A SPECTRAL THEORY OF GROKKING: WEIGHT DECAY INDUCES FEATURE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In grokking an early fit to the training data separates from a much later improvement in generalization. During this delay, training can move from a fixed neural tangent kernel (NTK) regime to one in which task-relevant kernel eigendirections continue to evolve. We provide a quantitative theory for how this transition from lazy to rich learning can produce delayed generalization. For homogeneous networks trained with squared loss and L_2 weight decay, we show that a finite residual remains after memorization, with larger residual fractions in target components associated with smaller NTK eigenvalues. These residuals feed back into the dynamics of the NTK itself, and projecting the resulting dynamics onto task-relevant spectral directions yields a reduced system in which residual-driven kernel growth competes with weight decay. This system predicts that the grokking timescale is controlled by the product of learning rate and weight decay, that feature learning slows logarithmically near a critical decay above which task-aligned NTK structure can no longer support generalization, and that stronger decay can prevent fitting altogether. We test these predictions in modular addition. In a homogeneous MLP, task-aligned Fourier structure continues to emerge in the NTK after training accuracy has saturated, and an 84×90-grid of trained networks across varying learning rate and weight decay recovers the predicted phase geometry and inverse-product scaling of the generalization time with learning rate and weight decay. A one-block Transformer shows similar macroscopic phase structure in a 42×45-grid, as well as the same transition-time scaling despite violating exact homogeneity. Together, these results provide a mechanistic derivation connecting post-fit feature learning to both the onset of generalization and its phase structure in the learning rate and weight decay plane.

1 INTRODUCTION

Analyzing the dynamics of generalization in deep neural networks is challenging, because optimization processes that lead to memorizing or overfitting, representation learning, and changes in validation set predictions usually occur together. However, grokking separates these timescales, as a network can reach high training accuracy, while remaining inaccurate on held-out examples for thousands of additional updates and only then generalize (Power et al., 2022). This separation lets us ask what happens after the training labels are already fitted and predicted correctly. *What error remains after the training labels are already fitted, how does it reshape the learned features, and what sets the delay before those changes improve held-out predictions?*

The neural tangent kernel (NTK) provides a framework for distinguishing approximately fixed-feature dynamics from feature learning. Let $\mathbf{f}$ denote the network outputs, $J = \nabla_{\boldsymbol{\theta}}\mathbf{f}$ the parameter Jacobian, and $K = JJ^\top$ the empirical NTK, which quantifies how network outputs adapt to changes to the parameters. For gradient flow on a differentiable data loss $\mathcal{L}_{\text{data}}$, the chain rule gives

$$\dot{\mathbf{f}}_{\text{data}} = -K\nabla_{\mathbf{f}}\mathcal{L}_{\text{data}}\,, \tag{1}$$

which is architecture-independent. When K remains constant, training is thus described by a fixed kernel. On the other hand, changes in K reflect changes in the tangent features of the network (Jacot et al., 2018). The neural tangent hierarchy extends this description by expressing the evolution of K in terms of higher-order tangent tensors (Huang & Yau, 2020). The residual error, the deviation

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.

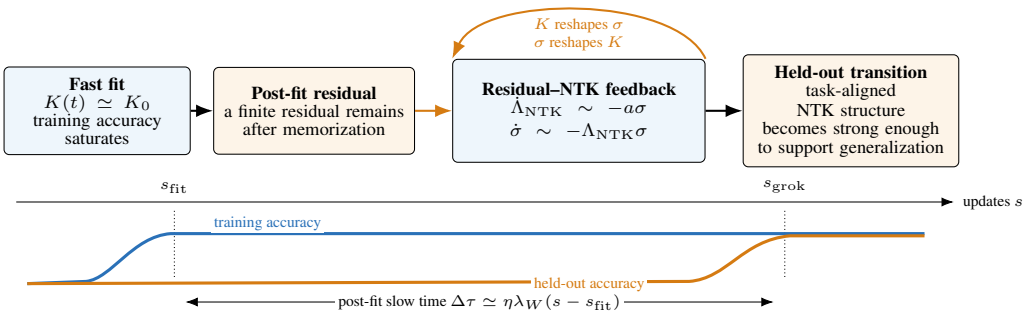


Figure 1: A fast, near-fixed-kernel stage fits the training data but leaves a finite residual under coupled weight decay. In the reduced dynamics that follow, the residual and task-aligned NTK strength evolve together. That is, the kernel reshapes the residual, and the residual reshapes the kernel, until task-aligned structure is strong enough to move held-out predictions across their decision margins. The lower panel separates the training fit from the later generalization transition and shows the post-fit slow-time increment $\Delta\tau \simeq \eta\lambda_W(s - s_{\mathrm{fit}})$.

between the network prediction and the true output, then enters the dynamics of K itself. Hence, the neural tangent hierarchy framework links the residual that is left after saturation of the training accuracy to subsequent changes in the NTK structure that are relevant to the task at hand.

Here, we derive a post-fit mechanism for delayed generalization in homogeneous networks trained with squared loss and coupled L_2 weight decay. Weight decay leaves residual error after fitting, and this drives continued task-aligned evolution of the empirical NTK. For an approximately fixed NTK during the initial fit, we prove that the fraction of each target component remaining as residual error decreases with the corresponding NTK eigenvalue. The neural tangent hierarchy couples these residuals to the evolution of the NTK, allowing task-aligned tangent structure to continue developing after memorization. Modular addition provides a natural Fourier basis for projecting these dynamics onto a reduced residual-NTK system in which residual-driven growth competes with weight decay. Its adiabatic limit predicts a slow post-fit timescale controlled by the product $\eta\lambda_W$ of learning rate and weight decay, together with finite-training boundaries in the corresponding (η, λ_W)-phase space. Empirically, we observe the corresponding post-fit NTK organization and optimizer-space structure in a homogeneous MLP, with the same scaling of generalization time with learning rate and weight decay also appearing in a Transformer. Figure 1 summarizes the proposed mechanism.

Related works. Delayed generalization can arise without evolving neural features. Linear estimators, Gaussian processes, and logistic models exhibit grokking under suitable conditions (Levi et al., 2024; Miller et al., 2024; Beck et al., 2025). Most directly, Xu et al. (2026) prove end-to-end grokking in overparameterized ridge regression trained by gradient descent with weight decay and derive quantitative hyperparameter dependence of the grokking time, while Kim (2026) derives an exactly solvable late-time weight-decay relaxation recovering the $(\eta\lambda_W)^{-1}$ scale in linear models. These results show that $(\eta\lambda_W)^{-1}$-type timing alone does not distinguish feature learning from fixed-feature dynamics.

Mechanistic work on neural-network grokking instead points to gradual representation change during the apparent plateau. In modular arithmetic, this includes the emergence of Fourier-structured circuits (Nanda et al., 2023), structured features and competition between memorizing and generalizing solutions (Liu et al., 2022; 2023a; Varma et al., 2024; Merrill et al., 2023; Ding et al., 2024), and transitions from an early kernel-like regime to later feature learning (Kumar et al., 2024; Lyu et al., 2024; Mohamadi et al., 2024; Rubin et al., 2024; Tian, 2026). A related line of empirical work tracks how the tangent features of the network reorganize over this transition. In particular, leading empirical-NTK eigenfunctions become increasingly task-relevant as delayed generalization emerges (Sanguino Bautiste et al., 2024), while substantial empirical-NTK movement can precede the representational changes that more closely track generalization (Zheng et al., 2024), and empirical-NTK eigenspaces in modular-arithmetic MLPs and Transformers align increasingly with Fourier features used by the learned solution (Lin, 2025). These works show that learned representations and the empirical NTK can continue to evolve during grokking, but leave open what drives this post-memorization

feature learning and how the evolution of task-aligned features is linked to generalization. As pointed out by Xu et al. (2026), a rigorous theoretical analysis that connects grokking to the transition from the lazy to the rich regime of training neural networks is missing.

In this work, we provide, to the best of our knowledge, the first quantitative theory linking this transition from lazy to rich training to delayed generalization. Appendix A develops the connections to other works in more detail. Specifically, we make the following contributions:

Contributions. We study delayed generalization in homogeneous networks trained with squared loss and coupled L_2 weight decay and derive a mechanism connecting post-fit residual error to continued task-aligned feature learning:

- We prove that, after the initial fit, coupled weight decay leaves a finite residual in the eigendirections of an approximately fixed NTK, with the largest relative residuals in directions that are weakly represented by the current features (Theorem 1).
- We show that these residuals enter the subsequent evolution of the empirical NTK. Projecting the neural tangent hierarchy onto a task-aligned Fourier direction yields a reduced residual–NTK system in which residual-driven growth competes with weight decay (Theorem 2).
- We derive a decay-controlled slow timescale for post-fit feature learning, predicting $(\eta\lambda_W)^{-1}$ scaling in the grokking regime and high-decay cutoffs where feature growth or training fit fails (Section 2.3).
- We test these predictions in homogeneous MLPs on modular addition, observing continued Fourier organization of the NTK and the predicted weak-decay timing and optimizer-space structure. A non-homogeneous one-block Transformer shows similar macroscopic behavior (Section 3).

2 Post-fit residuals drive tangent feature learning

We derive the post-fit residual under coupled weight decay, show how it drives subsequent NTK evolution, and reduce the dynamics to a task-aligned spectral mode. We then connect this mode growth to held-out generalization and the resulting optimizer-space grokking boundaries.

2.1 Weak target modes retain larger residual fractions

We perform our analysis in the setting of modular arithmetic, where the task is to learn the mapping $(a, b) \mapsto c = (a + b) \bmod p$, for a given prime p. Inputs and targets are one-hot encoded such that the network has p output coordinates. Let $\mathbb{Z}_p$ denote the additive group of integers modulo p. Its characters are $\chi_k(z) = \exp(2\pi ikz/p)$, $k \in \mathbb{Z}_p$, such that for output class c the target can be written as

$$y_c(a, b) = \frac{1}{p} \sum_{k=0}^{p-1} e^{-2\pi ikc/p} \chi_k(a)\chi_k(b). \tag{2}$$

Because the target contains only products $\chi_k(a)\chi_k(b)$ with the same frequency k, this specifies the relevant, task-aligned Fourier components that a successful model must learn to generalize. Note that the fixed-kernel residual dynamics derived below are not specific to modular addition, but hold for arbitrary datasets and targets under the homogeneous squared-loss setting. Modularity enters only when we use the task symmetry to identify an explicit Fourier basis for the subsequent spectral reduction.

Let n be the number of training examples and stack all p-dimensional predictions and targets into $\mathbf{f}, \mathbf{y} \in \mathbb{R}^{np}$. With residual $\boldsymbol{\sigma} = \mathbf{f} - \mathbf{y} \in \mathbb{R}^{np}$, the training objective can be written as

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{\mathcal{N}}{2} \|\boldsymbol{\sigma}\|^2 + \frac{\lambda_W}{2} \|\boldsymbol{\theta}\|^2 , \tag{3}$$

where $\mathcal{N}$ is an arbitrary normalization factor[1], and λ_W is the coefficient of the L_2 penalty. Throughout, lowercase λ_W denotes optimizer weight decay, whereas capital Λ denotes an NTK spectral strength or eigenvalue. We define $\lambda_W^{\text{norm}} = \lambda_W/\mathcal{N}$ as the decay coefficient in normalized gradient-flow time.

[1] As shown below, the slow update time governing grokking dynamics is invariant to the specific normalization.

One gradient-descent update with learning rate η advances this time by $\mathcal{N}\eta$ to first order. For P trainable parameters, $J = \nabla_{\boldsymbol{\theta}}\mathbf{f} \in \mathbb{R}^{np\times P}$ is the parameter Jacobian and $K = JJ^\top \in \mathbb{R}^{np\times np}$ is the empirical NTK. Gradient flow then leads to

$$\dot{\boldsymbol{\theta}} = -J^\top \boldsymbol{\sigma} - \lambda_W^{\text{norm}}\boldsymbol{\theta}, \qquad \dot{\boldsymbol{\sigma}} = -K\boldsymbol{\sigma} - \lambda_W^{\text{norm}} J\boldsymbol{\theta}. \tag{4}$$

Furthermore, if $\mathbf{f}_{\boldsymbol{\theta}}$ is D-homogeneous under uniform parameter rescaling, Euler's identity gives $J\boldsymbol{\theta} = D\mathbf{f}$. Bias-free feed-forward ReLU networks satisfy this identity, whereas biases, normalization, mixed-degree residual paths, and standard attention generally break it. Substituting this identity into Eq. (4) closes the dynamics in residual space.

Theorem 1 (Persistent ridge residual under homogeneous dynamics)**.** *Let $\mathbf{f}_{\boldsymbol{\theta}}$ be D-homogeneous and let $\lambda_W^{\text{norm}} > 0$. Gradient flow on Eq. (3) obeys*

$$\dot{\boldsymbol{\sigma}} = -(K + D\lambda_W^{\text{norm}} I)\boldsymbol{\sigma} - D\lambda_W^{\text{norm}}\mathbf{y}. \tag{5}$$

If $K(t) \equiv K$, then

$$\boldsymbol{\sigma}(t) \longrightarrow \boldsymbol{\sigma}_* = -D\lambda_W^{\text{norm}}(K + D\lambda_W^{\text{norm}} I)^{-1}\mathbf{y}. \tag{6}$$

Every finite-eigenvalue direction with a nonzero target projection therefore retains a nonzero residual.

For ReLU networks, Eq. (5) holds on each differentiable interval and therefore piecewise across activation-boundary crossings. The fixed-kernel limit in Eq. (6) additionally requires $K(t)$ itself to remain constant. We prove the theorem in Appendix B.1. Now, we diagonalize K, writing $K\mathbf{u}_j = \Lambda_j \mathbf{u}_j$, where $\mathbf{u}_j$ is an eigenvector of K with eigenvalue Λ_j, and denote the corresponding target and residual coefficients by $y_j = \mathbf{u}_j^\top \mathbf{y}$ and $\sigma_j = \mathbf{u}_j^\top \boldsymbol{\sigma}$. The fixed-kernel limit in Eq. (6) then gives the equilibrium residual coefficient in the jth NTK eigendirection,

$$\sigma_{j,*} = -\frac{D\lambda_W^{\text{norm}}}{\Lambda_j + D\lambda_W^{\text{norm}}}\, y_j. \tag{7}$$

This shows that for fixed $|y_j|$, the fraction of the jth target component that remains as residual error is larger for smaller Λ_j. These weaker modes correspond to directions that the current features represent poorly. However, in a finite network the NTK can continue to evolve after training accuracy has saturated. Therefore, we now study whether the larger relative residuals in weak task-aligned modes can drive continued task-aligned NTK evolution.

2.2 Residuals drive task-aligned tangent growth

For a one-hidden-layer, $D = 2$, bias-free ReLU MLP, combining the neural tangent hierarchy (Huang & Yau, 2020) with the two-homogeneous weight-decay terms gives

$$\dot{\boldsymbol{\sigma}} = -(K + 2\lambda_W^{\text{norm}} I)\boldsymbol{\sigma} - 2\lambda_W^{\text{norm}}\mathbf{y}, \qquad \dot{K} = -K^{(2)}\boldsymbol{\sigma} - 2\lambda_W^{\text{norm}} K. \tag{8}$$

Here $K^{(2)}$ is a third-order tangent tensor, and $K^{(2)}\boldsymbol{\sigma}$ denotes contraction over its residual index, $(K^{(2)}\boldsymbol{\sigma})_{ij} := \sum_k K^{(2)}_{ijk}\sigma_k$. This contraction couples the residual and kernel dynamics, and again employing two-homogeneity the parameter decay term becomes $-2\lambda_W^{\text{norm}} K$. Although this term shrinks the NTK, the residual-dependent part $-K^{(2)}\boldsymbol{\sigma}$ can simultaneously increase NTK strength along task-aligned directions.

To proceed, choose a normalized real Fourier direction $\mathbf{u}$ with nonzero target projection and orient it so that $y = \mathbf{u}^\top \mathbf{y} \geq 0$. We define the task-aligned NTK strength Λ and projected residual σ by

$$\Lambda = \mathbf{u}^\top K \mathbf{u}, \qquad \sigma = \mathbf{u}^\top \boldsymbol{\sigma}. \tag{9}$$

Λ measures how strongly the network can change its output along the selected task direction under parameter updates.

The Fourier structure of modular addition makes $\mathbf{u}$ a natural task direction. This motivates a local one-mode approximation in which $\mathbf{u}$ remains approximately an eigendirection of the NTK, $K\mathbf{u} \simeq \Lambda\mathbf{u}$, with weak mixing into other directions, and its dynamics are driven mainly by the residual component along $\mathbf{u}$. Appendix C.1 states these conditions explicitly. Section 3 later shows the corresponding Fourier organization empirically. Projecting the hierarchy dynamics onto $\mathbf{u}$ then gives

$$\dot{\sigma} \simeq -(\Lambda + 2\lambda_W^{\text{norm}})\sigma - 2\lambda_W^{\text{norm}} y, \qquad \dot{\Lambda} \simeq -a\sigma - 2\lambda_W^{\text{norm}}\Lambda. \tag{10}$$

Here, $a(\boldsymbol{\theta})$ is the local coupling between the projected residual and the residual-driven change in Λ, as defined in Appendix C.1. We consider $a(\boldsymbol{\theta}) > 0$, qualitatively consistent with the observed post-fit emergence of task-aligned NTK structure. It implies that correcting the output along $\mathbf{u}$ tends to increase the NTK strength in the same direction. This sign is a dynamical condition rather than a consequence of homogeneity or symmetry. Equation (10) then describes a feedback. For $\sigma < 0$, larger Λ accelerates residual relaxation, while the remaining residual drives Λ upward.

We further assume that $a(\boldsymbol{\theta})$ is locally constant over the post-fit interval considered. Appendix C.1 shows that once the residual is close to its instantaneous ridge value, weight decay does not introduce an additional fast timescale for a, supporting this approximation in the slow regime considered below. After the fast ridge relaxation, $\sigma < 0$ for $y > 0$, so $-a\sigma > 0$ and the residual increases the NTK strength along the selected task direction. Under the locally constant-a approximation, the fixed points of Eq. (10) and their stability can be characterized exactly.

Theorem 2 (Spectral selection in the reduced tangent-mode system)**.** *Under the locally constant-a approximation, for $\lambda_W^{\mathrm{norm}} > 0$, $a > 0$, and $y \geq 0$, Eq. (10) has one equilibrium with $\Lambda \geq 0$:*

$$\Lambda_* = -\lambda_W^{\mathrm{norm}} + \sqrt{\lambda_W^{\mathrm{norm}\,2} + ay}, \qquad \sigma_* = -\frac{2\lambda_W^{\mathrm{norm}}}{a}\Lambda_*. \tag{11}$$

The equilibrium is locally asymptotically stable. It satisfies $\Lambda_* = 0$ for $y = 0$ and $\Lambda_* > 0$ for $y > 0$.

We give the fixed-point and stability calculations in Appendix C.2. Within the reduced system, unsupported directions decay, while positive residual-to-NTK coupling sustains NTK strength for target-aligned directions. Under the adiabatic approximation that residual relaxation is faster than mode motion, we set $\dot{\sigma} \simeq 0$ at the current Λ and obtain

$$\sigma_{\mathrm{ad}}(\Lambda) = -\frac{2\lambda_W^{\mathrm{norm}} y}{\Lambda + 2\lambda_W^{\mathrm{norm}}}, \qquad \dot{\Lambda} = 2\lambda_W^{\mathrm{norm}}\left(\frac{ay}{\Lambda + 2\lambda_W^{\mathrm{norm}}} - \Lambda\right). \tag{12}$$

This shows that mode evolution occurs on a normalized-time scale $1/\lambda_W^{\mathrm{norm}}$. Since $t \simeq \mathcal{N}\eta s$ after s updates, we define the normalization-invariant slow time

$$\tau := \lambda_W^{\mathrm{norm}} t \simeq \eta\lambda_W s\,. \tag{13}$$

Here, s is the optimizer update index and $\eta\lambda_W$ is the per-update decay scale. If s_{fit} denotes the end of the initial fitting stage, the subsequent slow-time increment is $\Delta\tau \simeq \eta\lambda_W(s - s_{\mathrm{fit}})$. Thus the reduced dynamics predict the leading scaling of post-fit transition times with $(\eta\lambda_W)^{-1}$.

2.3 Grokking boundaries in the (η, λ_W) plane

The same adiabatic calculation shows how the task-aligned NTK strength Λ controls how much of the corresponding target component is expressed in the network output. Since $\sigma = \mathbf{u}^\top(\mathbf{f} - \mathbf{y})$, the projected output coefficient $f = \mathbf{u}^\top\mathbf{f}$ satisfies

$$f = y + \sigma_{\mathrm{ad}}(\Lambda) = g(\Lambda)y, \quad g(\Lambda) = \frac{\Lambda}{\Lambda + 2\lambda_W^{\mathrm{norm}}}. \tag{14}$$

Hence, increasing Λ along a task-aligned mode increases the corresponding component of the training outputs continuously. This training coordinate is related to generalization through the evolution of held-out margins, the difference between the correct-class logit and the largest competing logit, along the same trajectory. Because modular addition is represented by several matched Fourier components as shown in Eq. (2), these margins can change smoothly as the corresponding components grow, while the predicted label changes only when a margin crosses zero. If many held-out examples cross that threshold at similar mode strengths, smooth spectral growth can therefore produce a sharp rise in validation accuracy. Appendix D.1 formalizes this connection using the empirical distribution of the mode strengths at which individual held-out examples change class.

The scalar dynamics in Eq. (10) track a single task-aligned Fourier mode and do not resolve the individual held-out margins that produce the sharp accuracy transition. We therefore now replace the collection of margin crossings by a threshold Λ_{g} on the task-aligned NTK strength Λ. Reaching Λ_{g} represents reaching a chosen held-out-accuracy threshold. We denote the NTK strength at the beginning of the slow stage by Λ_0, let T denote the number of optimizer updates available from that point, and recall the stable equilibrium Λ_* from Eq. (11). Since Eq. (12) gives $\dot{\Lambda} > 0$ for $\Lambda < \Lambda_*$,

a trajectory with $\Lambda_0 < \Lambda_g < \Lambda_*$ crosses the threshold Λ_g in finite time. Separating variables in Eq. (12) and integrating then gives the boundary in the (η, λ_W) plane between trajectories that reach Λ_g within T updates and those that remain below it.

Corollary 1 (Finite-training grokking boundary)**.** *Under the one-mode, adiabatic, and gradient-flow approximations,*

$$\eta_*(\lambda_W) = \frac{1}{2\lambda_W T} \int_{\Lambda_0}^{\Lambda_g} \frac{\Lambda + 2\lambda_W^{\mathrm{norm}}}{ay - \Lambda^2 - 2\lambda_W^{\mathrm{norm}}\Lambda}\, d\Lambda\,. \tag{15}$$

At small normalized decay and away from the fixed-point boundary, the leading dependence is $\eta_* \propto 1/\lambda_W$.

A derivation and closed form solution of this integral can be found in Appendices D.2– D.3. Equation (15) predicts that increasing Λ_0 removes a positive portion of the crossing-time integral and thereby shortens the delay before grokking occurs. We elaborate on this prediction and provide an empirical test in Appendices F.1– F.2. This finite-training grokking boundary is obtained under the assumption that the threshold is reachable in the first place, $\Lambda_g < \Lambda_*$. Whether or not this is possible follows from the limiting case $\Lambda_g = \Lambda_*$, which gives

$$\lambda_W^{\mathrm{mode}} = \mathcal{N}\,\frac{ay - \Lambda_g^2}{2\Lambda_g}, \tag{16}$$

provided $ay > \Lambda_g^2$. For $\lambda_W < \lambda_W^{\mathrm{mode}}$, the equilibrium lies above Λ_g so the mode can reach Λ_g in finite time, with Corollary 1 determining whether this occurs within T updates. For $\lambda_W \geq \lambda_W^{\mathrm{mode}}$ the threshold cannot be crossed in finite time. As λ_W approaches this cutoff from below, the net growth rate of the task-aligned mode at $\Lambda = \Lambda_g$ approaches zero, causing $\eta_*(\lambda_W)$ to diverge logarithmically as $\lambda_W \to \lambda_W^{\mathrm{mode}}$ (Appendix D.3). This shows that as λ_W approaches $\lambda_W^{\mathrm{mode}}$ from below, reaching Λ_g and thereby generalizing within a fixed update budget requires an increasingly large learning rate. In the (η, λ_W) plane, this produces an upward turn of the finite-time boundary near the cutoff at $\lambda_W^{\mathrm{mode}}$.

A further constraint arises from fitting the training data. This boundary depends on the relative gains of multiple output modes and is therefore not fixed exactly by the one-mode dynamics. To retain a scalar description, we summarize the onset of fitting failure by the gain of an effective training mode. Requiring $g(\Lambda_{\mathrm{fit}}) \geq g_{\min}$ gives

$$\lambda_W \leq \mathcal{N}\,\frac{\Lambda_{\mathrm{fit}}(1 - g_{\min})}{2g_{\min}}, \tag{17}$$

This provides an approximately vertical effective decay scale beyond which the network no longer reaches the chosen training-accuracy threshold.

Finally, sufficiently large learning rates encounter the discrete-time stability edge of the approximately fixed-kernel dynamics. As derived in Appendix D.4, linear stability requires

$$\eta < \frac{2}{\mathcal{N}\Lambda_{\max}(K) + 2\lambda_W}\,, \tag{18}$$

where $\Lambda_{\max}(K)$ is the largest eigenvalue of the frozen NTK. This is the frozen-kernel analogue of the usual edge-of-stability condition (Cohen et al., 2021). Because the NTK subsequently evolves, it provides only a local estimate of the large-η boundary of the full training dynamics. Appendices D.3– D.4 give the derivations and validity conditions for the fitting and stability bounds.

To summarize, these constraints predict the following organization of the (η, λ_W) plane. When the training data are fitted and Λ_g is reachable but the finite-training condition in Corollary 1 is not satisfied, the network remains in the memorization regime over the available training interval. Grokking is possible when the training fit succeeds (Eq. (17)), the task-aligned NTK strength can reach Λ_g (Eq. (16)), the threshold is crossed within T updates (Eq. (15)), and gradient descent remains stable (Eq. (18)). Increasing weight decay can eventually prevent either task-mode growth or training fit, while sufficiently large learning rates produce a separate instability boundary.

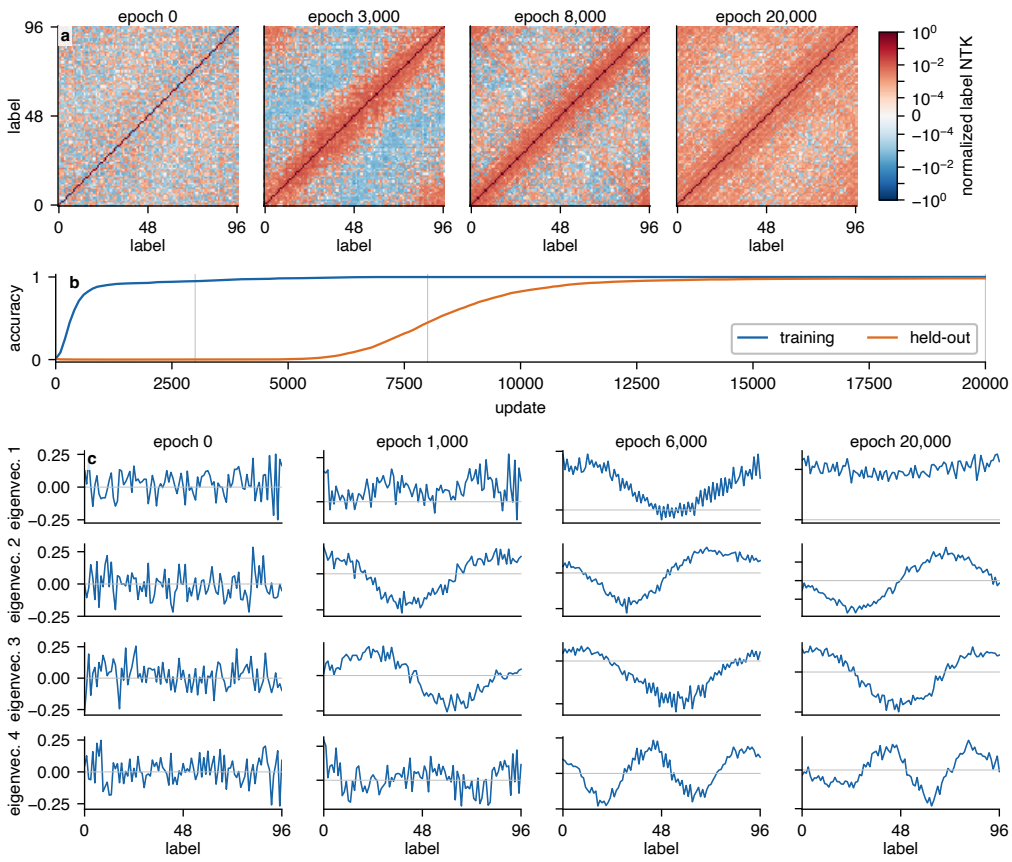


Figure 2: Post-fit change in the normalized label-space NTK for a bias-free MLP on addition modulo 97. **(a)** We project the sample NTK onto the true-label, average by output label, and normalize the resulting 97×97 label kernel. **(b)** Training and held-out accuracy for the same run. **(c)** Four leading eigenvectors of the same normalized label kernel, showing Fourier-like standing waves that sharpen after training accuracy is already high.

3 Experiments

We now test these predictions and measure post-fit NTK reorganization as well as the (η, λ_W)-plane boundaries, and whether similar transition-time behavior appears in Transformers. Appendix G provides experimental details[2].

3.1 Task-aligned NTK structure develops after fitting

We trained a bias-free one-hidden-layer ReLU MLP on addition modulo 97, using half of the samples for training and the remainder for evaluation. We computed the empirical NTK, projected it onto the true-label output of each sample, and averaged the resulting entries by output label. This isolates tangent feature organization with respect to the task labels rather than individual examples. To further isolate structure from changes in overall NTK scale, we normalized the resulting 97×97 kernel. Figure 2a shows that periodic structure in this kernel strengthens between updates $3{,}000$ and $20{,}000$, while Figure 2b shows that training accuracy has already saturated during much of this evolution. In Figure 2c, the leading eigenvectors dynamics show the emergence of standing-wave profiles in label space, consistent with the Fourier task structure in Eq. (2). Their continued organization after fitting is also qualitatively consistent with the post-fit growth of task-aligned NTK modes described by Eq. (10).

[2]The supplementary anonymous code package contains all code used for the reported results.

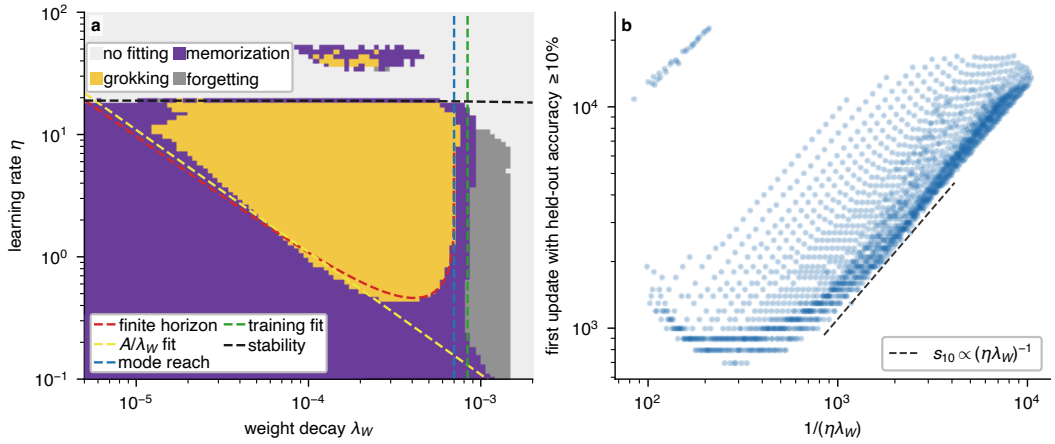


Figure 3: Optimizer-plane structure for the homogeneous MLP on addition modulo 23. **(a)** Outcomes after 24,000 updates on an 84×90 learning-rate/decay grid. The overlays show the calibrated finite-time boundary, a separately fitted inverse-decay relation, the mode-reachability cutoff, the training-fit cutoff, and the calibrated frozen-kernel stability form. **(b)** First held-out 10% crossing among runs that eventually meet the 90% grokking criterion, plotted against $(\eta\lambda_W)^{-1}$. The dashed segment indicates the predicted scaling $s_{10} \propto (\eta\lambda_W)^{-1}$ and is not fitted to the data.

3.2 Grokking boundaries in the MLP optimizer plane

Next, we trained an (84×90)-grid of MLPs for varying (η, λ_W) on addition modulo 23 for a fixed total of 24,000 updates. Figure 3 shows the resulting (η,λ_W)-plane, categorized into memorizing, grokking, forgetting, and non-fitting runs, as described in Appendix G.2. At small $\eta\lambda_W$, the task-aligned NTK strength Λ does not have enough time to reach the generalization threshold Λ_g within the fixed training budget, producing the broad memorization region in Figure 3a. The red curve fits the finite-training grokking boundary from Eq. (15). Note that since the fit-parameters are inferred from the boundary, they are not direct measurements of $\Lambda(t)$. Hence, this comparison tests the predicted optimizer dependence of the crossing time rather than the absolute amount of task-aligned NTK growth. At weak decay, it follows the transition from memorization to grokking and closely tracks the yellow $\eta \propto \lambda_W^{-1}$ fit. As λ_W increases, the finite-time curve turns upward toward the blue mode-reachability cutoff from Eq. (16), beyond which the task-aligned NTK strength cannot reach the required threshold. The green line shows the training-fit cutoff from Eq. (17), while the black line corresponds to the frozen-kernel stability condition in Eq. (18), fitted to the observed large-learning-rate boundary. The full fitting procedures for all curves are given in Appendix G.2.1.

Figure 3b tests the predicted slow-time scaling. For every run that reaches 90% held-out accuracy, we plot the first update s_{10} at which that accuracy reaches 10% against $(\eta\lambda_W)^{-1}$. Equation (13) predicts $s_{10} \propto (\eta\lambda_W)^{-1}$ if the transition is governed by the slow time $\tau \simeq \eta\lambda_W s$. Indeed, the observed crossing times follow this scaling across the grokking region.

3.3 Grokking boundaries and transition-time scaling in a Transformer

Biases, LayerNorm, residual paths, and attention all break the global homogeneity used in Theorems 1 and 2. However, differentiating $K = JJ^\top$ under gradient flow still yields an exact residual-dependent contribution to $\dot{K}$ for any differentiable model (Appendix E). We therefore tested whether the homogeneous predictions also hold in a non-homogeneous case, and trained a (42×45)-grid of Transformers. In line with the MLP phase diagram, Figure 4a shows a weak-decay memorization region, an intermediate grokking band, and failure at stronger decay or large learning rate. The fitted curve of the form $\eta = A/\lambda_W$ describes the grokking-memorization boundary well, in line with the homogeneous theory. Additionally, within the grokking networks, Figure 4b exhibits the same linear scaling of the first held-out 80% crossing s_{80} with $(\eta\lambda_W)^{-1}$. We use 80% here because early 10% accuracy crossings are often reversible in practice. This suggests that the optimizer-level phase and transition-time scaling may extend beyond exact homogeneity.

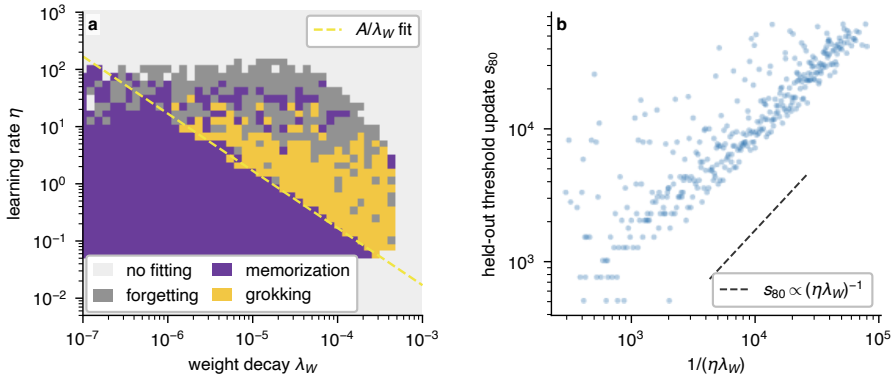


Figure 4: Phase and transition-time measurements for a one-block Transformer on the same modulo-23 split. **(a)** Outcomes on a 42×45 optimizer grid with an inverse-decay fit to the main memorization–grokking boundary. **(b)** First held-out 80% crossing among runs that ultimately grok at the 90% criterion, plotted against $(\eta\lambda_W)^{-1}$. The dashed segment indicates the scaling $s_{80} \propto (\eta\lambda_W)^{-1}$.

4 Limitations and Outlook

Our theory holds for homogeneous networks trained with squared loss, coupled weight decay, and full-batch gradient flow. Extending the reduced dynamics to more general objectives and architectures is a natural next step. The Transformer experiments already show that similar macroscopic optimizer-space structure can persist without exact homogeneity, but the corresponding microscopic dynamics remain open. Further, the scalar reduction follows a single decoupled task direction. Although the natural Fourier basis of modular addition motivates this, interactions among task-relevant components are neglected, and more general datasets need not provide an equally natural spectral basis. Future work should therefore extend the dynamics to interacting task directions and identify suitable coordinates beyond cyclic modular arithmetic. In this direction, Marchetti et al. (2026) show that for sequential group-composition tasks, networks learn irreducible representations one at a time, suggesting a route from the present one-mode reduction toward multiple representation-theoretic components. Finally, modular addition and grokking provide explicit task structure and separate fitting from generalization, but parts of our mechanism are more general. The fixed-kernel residual result does not rely on modular arithmetic, and the residual-dependent contribution to NTK evolution persists for differentiable models. This raises the question of whether our framework also applies to broader representation-learning problems. The J-space framework of Gurnee et al. (2026) provides a possible connection, using an activation-to-output Jacobian to identify representations with strong downstream influence. The NTK similarly uses the parameter Jacobian to track task-aligned directions during training. Establishing whether these directions emerge together would connect the optimization dynamics studied here to the broader formation of internal representations used for generalization.

5 Conclusions

We developed a spectral theory of grokking in which coupled weight decay leaves a finite post-fit residual, with target components at smaller NTK eigenvalues retaining a larger residual fraction. We used the neural tangent hierarchy to show that this residual drives task-aligned NTK growth. The resulting slow dynamics determine whether task-aligned NTK strength can grow sufficiently far and sufficiently quickly within the training budget, while training fit and discrete-time stability impose additional constraints. Empirically, a homogeneous MLP on modular addition shows continued Fourier organization of the NTK after training accuracy saturates, an MLP hyperparameter sweep recovers the predicted (η, λ_W) phase geometry and $(\eta\lambda_W)^{-1}$ transition-time scaling, and a one-block Transformer reproduces the same macroscopic phase structure and transition-time scaling despite lacking exact homogeneity. Together, these results suggest that residual error left after fitting can actively shape how and when neural networks acquire generalizing representations, connecting feature learning, generalization time, and optimizer-space structure within a single mechanistic framework.

AI USE STATEMENT

AI was used to assist with text editing, coding, and with literature research. AI was not involved in research ideation, the development of theoretical results, research methodology, and experimental design. All AI-assisted outputs were reviewed, revised where necessary, and verified by the authors.

REPRODUCIBILITY STATEMENT

The assumptions and approximations underlying the homogeneous and reduced-mode theory are stated in the main text and Appendices C.1–D.4, which also provide complete proofs of the exact theoretical claims and derivations of the reduced predictions. Appendix E.1 gives the corresponding differentiable-model identities. Appendix F.2 and Appendices G.1–G.4 specify the procedural data construction and splits, architectures, random seeds, losses, optimizers, training budgets, evaluation conventions, kernel normalizations, phase definitions, and boundary-fitting procedures used in the experiments. The accompanying anonymous supplementary code package contains the centralized configurations, archived data used for the reported figures, environment specification, and training, analysis, and figure-generation entry points needed to reproduce the reported results.

REFERENCES

Aristide Baratin, Thomas George, César Laurent, R. Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2269–2277, 2021.

Alon Beck, Noam Itzhak Levi, and Yohai Bar-Sinai. Grokking at the edge of linear separability. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3307–3334, 2025.

Francesco Bertolotti and Walter Cazzola. A survey on grokking. *ACM Computing Surveys*, 58(13): 327:1–327:25, 2026. doi: 10.1145/3814603.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jh-rTtvkGeM.

Xiaoman Delores Ding, Zifan Carl Guo, Eric J. Michaud, Ziming Liu, and Max Tegmark. Survival of the fittest representation: A case study with modular addition. *arXiv preprint arXiv:2405.17420*, 2024.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, T. Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026.

Jianliang He, Leda Wang, Siyu Chen, and Zhuoran Yang. On the mechanism and dynamics of modular addition: Fourier features, lottery ticket, and grokking. *arXiv preprint arXiv:2602.16849*, 2026.

Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Alan Jeffares, Alicia Curth, and Mihaela van der Schaar. Deep learning through a telescoping lens: A simple model provides empirical insights on grokking, gradient boosting and beyond. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Tikeng Notsawo Pascal Junior, Guillaume Dumas, and Guillaume Rabusseau. Grokking beyond the euclidean norm of model parameters. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 28552–28618, 2025.

Taeyoung Kim. Grokking on the weight-decay clock: A rate hierarchy from softly broken symmetries. *arXiv preprint arXiv:2607.23967*, 2026.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *International Conference on Learning Representations*, 2024.

Noam Itzhak Levi, Alon Beck, and Yohai Bar-Sinai. Grokking in linear estimators: A solvable model that groks without understanding. In *International Conference on Learning Representations*, 2024.

Jennifer Lin. Feature identification via the empirical ntk. *arXiv preprint arXiv:2510.00468*, 2025.

Ziming Liu, Ouail Kitouni, Niklas 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*, 2022.

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations*, 2023a.

Ziming Liu, Ziqian Zhong, and Max Tegmark. Grokking as compression: A nonlinear complexity perspective. *arXiv preprint arXiv:2310.05918*, 2023b.

Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon Shaolei Du, Jason D. Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. In *International Conference on Learning Representations*, 2024.

Neil Rohit Mallinar, Daniel Beaglehole, Libin Zhu, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. Emergence in non-neural models: Grokking modular arithmetic via average gradient outer product. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 42834–42856, 2025.

Giovanni Luca Marchetti, Daniel Kunin, Adele Myers, Francisco Acosta, and Nina Miolane. Sequential group composition: A window into the mechanics of deep learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oFD9pc53pF.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *ICLR Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28699–28722, 2023.

Jack William Miller, Charles O'Neill, and Thang D. Bui. Grokking beyond neural networks: An empirical exploration with model complexity. *Transactions on Machine Learning Research*, 2024.

Mohamad Amin Mohamadi, Zhiyuan Li, Lei Wu, and Danica J. Sutherland. Why do you grok? a theoretical analysis on grokking modular addition. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35934–35967, 2024.

Prudhviraj Naidu, Zixian Wang, Leon Bergen, and Ramamohan Paturi. Quiet feature learning in algorithmic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37756–37764, 2026.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*, 2023.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Lucas Prieto, Melih Barsbey, Pedro A. M. Mediano, and Tolga Birdal. Grokking at the edge of numerical stability. In *International Conference on Learning Representations*, 2025.

Noa Rubin, Inbar Seroussi, and Zohar Ringel. Grokking as a first order phase transition in two layer networks. In *International Conference on Learning Representations*, 2024.

Javier Sanguino Bautiste, Gregor Bachmann, Bobby He, Lorenzo Noci, and Thomas Hofmann. Feature learning dynamics under grokking in a sparse parity task. *ICML 2024 Workshop on High-dimensional Learning Dynamics: The Emergence of Structure and Reasoning*, 2024.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua M. Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. In *Has it Trained Yet? Workshop at NeurIPS*, 2022.

Yuandong Tian. Provable scaling laws of feature emergence from learning dynamics of grokking. In *International Conference on Learning Representations*, 2026.

Vikrant Varma, Rohin Shah, Zachary Kenton, Janos Kramar, and Ramana Kumar. Explaining grokking through circuit efficiency. In *International Conference on Learning Representations*, 2024.

Mingyue Xu, Gal Vardi, and Itay Safran. To grok grokking: Provable grokking in ridge regression. *arXiv preprint arXiv:2601.19791*, 2026.

Xingyu Zheng, Kyle Daruwalla, Ari S. Benjamin, and David Klindt. Delays in generalization match delayed changes in representational geometry. In *Proceedings of UniReps: the Second Edition of the Workshop on Unifying Representations in Neural Models*, volume 285 of *Proceedings of Machine Learning Research*, pp. 324–334, 2024.

A Related mechanisms for delayed generalization

Grokking has been explained through circuit formation, implicit bias, tangent-feature motion, fixed spectra, and optimizer state. Surveys emphasize that several mechanisms can coexist across losses, architectures, and training regimes (Bertolotti & Cazzola, 2026). We position our account by asking which dynamical object changes during the plateau and which observations distinguish that change from a slow fixed-feature relaxation.

A.1 Mechanistic progress and representation competition

Modular arithmetic provides an interpretable Fourier basis for circuit analysis. Nanda et al. (Nanda et al., 2023) reverse-engineer a trigonometric algorithm in a grokked Transformer and introduce progress measures that reveal gradual circuit formation during the apparent plateau. Varma et al. (Varma et al., 2024) explain delayed generalization through competition between memorizing and generalizing circuits, while Merrill et al. (Merrill et al., 2023) study competition between sparse and dense subnetworks. Ding et al. (Ding et al., 2024) track competing circular Fourier representations with low-dimensional dynamics. He et al. (He et al., 2026) analyze how frequency competition, phase alignment, and weight decay shape Fourier-feature formation in two-layer networks on modular addition. Across a broader set of algorithmic Transformer tasks, Naidu et al. (Naidu et al., 2026) find causally relevant features developing during long plateaus before the output loss improves. Other representation-level studies connect grokking to changes in norms, learned features, and compression (Liu et al., 2022; 2023a;b). We use the same task-defined Fourier basis but study the residual that drives tangent-feature motion within it.

A.2 Kernel-to-feature transitions and empirical NTK dynamics

Several theories describe grokking as a transition from an early linearized regime to later feature learning. Kumar et al. (Kumar et al., 2024) use a controllable laziness parameter to induce or remove grokking in polynomial regression, MNIST, and modular addition. Lyu et al. (Lyu et al., 2024) prove a separation between early kernel-like and late implicit biases in homogeneous networks with weight decay. Mohamadi et al. (Mohamadi et al., 2024) show that an early permutation-equivariant kernel regime can be sample inefficient for modular addition, whereas bounded-norm feature-learning solutions generalize from fewer examples. Rubin et al. (Rubin et al., 2024) analyze an adaptive kernel and relate grokking in two-layer networks to a first-order phase transition. Tian (Tian, 2026) derives a sequence of lazy, independent-feature, and interacting-feature stages with scaling laws for feature emergence and generalization.

Empirical NTKs distinguish departure from the lazy regime from the task content acquired by the kernel. On sparse parity, Sanguino Bautiste et al. (Sanguino Bautiste et al., 2024) observe leading NTK eigenfunctions moving from non-predictive directions toward predictive features as generalization emerges. On image classification, Zheng et al. (Zheng et al., 2024) find substantial empirical-NTK movement before delayed test improvement and closer synchronization between representational geometry and generalization. For modular arithmetic, Lin (Lin, 2025) shows that leading empirical-NTK eigenspaces align with Fourier feature families in trained MLPs and Transformers and that the alignment changes through grokking. These results establish feature motion but do not identify its source. We derive a residual-dependent equation for task-aligned kernel components. Equation (8)

inserts the post-fit residual into $\dot{K}$, and the Fourier projection yields the reduced feedback, slow time, and finite-training boundary. This mechanism is also compatible with dynamic alignment between finite-network tangent features and task directions (Baratin et al., 2021).

A.3 Static-feature grokking and the weight-decay transition-time scaling

Fixed-feature models show that delayed generalization can occur without representation change. Levi et al. (Levi et al., 2024) analyze grokking in linear estimators, Miller et al. (Miller et al., 2024) document related non-neural behavior, and Beck et al. (Beck et al., 2025) study long delays near linear separability. Xu et al. (Xu et al., 2026) prove end-to-end grokking for overparameterized ridge regression trained with gradient descent and weight decay. Their bounds separate early training fit from later population improvement and express the delay in terms of learning rate, decay, sample size, feature dimension, and initialization. Kim (Kim, 2026) derives a complementary late-time relaxation in linear models with weight decay and heavy-ball optimization; the zero-momentum limit again yields $1/(\eta\lambda)$ scaling.

The transition-time scaling in Eq. (13) is therefore shared by static-feature and feature-learning mechanisms. Our finite-network account adds the empirical NTK as a state variable and predicts task-aligned motion after training accuracy saturates. The normalized kernel measurement, the residual-dependent hierarchy term, and the directional intervention separate this claim from a fixed-feature delay.

Feature learning can also arise outside ordinary neural-network gradient descent. Mallinar et al. (Mallinar et al., 2025) demonstrate grokking in recursive feature machines driven by an average gradient outer product; block-circulant task features emerge after training loss has vanished. Jeffares et al. (Jeffares et al., 2024) analyze grokking through a sequence of local linear approximations, providing another time-resolved view of representation change.

A.4 Jacobian representations and emergent capabilities

Jacobian geometry can isolate directions with direct behavioral leverage. Gurnee et al. (Gurnee et al., 2026) introduce the Jacobian lens and identify a small "J-space" of verbalizable representations in language models; interventions show that these directions can mediate downstream reasoning and reporting. Their Jacobian transports downstream activations to outputs, whereas our NTK uses the parameter Jacobian, but both approaches focus on directions that couple strongly to behavior. We study how such task-relevant parameter-Jacobian directions form during training.

The quantization model of neural scaling (Michaud et al., 2023) proposes that knowledge and skills may be acquired in discrete modules even when aggregate loss follows smooth power laws. Grokking offers a small setting in which abrupt behavior can be resolved in time. Equation (58) gives one route from a smooth internal coordinate to a sharp output transition, so behavioral discontinuity need not imply discontinuous internal learning. At the same time, task-aligned modes provide a concrete object for testing more modular accounts of capability acquisition.

A.5 Regularization and optimization mechanisms

Regularization mechanisms extend beyond Euclidean weight decay. Junior et al. (Junior et al., 2025) show that sparsity, low-rank penalties, implicit regularization, and depth can induce grokking-like transitions. Adaptive optimizers can produce late-training instabilities (Thilak et al., 2022), while cross-entropy training can involve numerical and logit-scaling effects (Prieto et al., 2025). The relevant slow variable changes with the objective and optimizer. We focus on homogeneous squared-loss training because coupled L_2 decay leaves the ridge residual in Theorem 1 and directly contracts the two-homogeneous empirical NTK in Eq. (8).

A.6 Relation to fixed-feature and circuit-level accounts

Our account links the residual left by an early approximately fixed kernel to later tangent-feature motion in a task-defined Fourier direction. The resulting feature strength determines whether held-out margins are reachable and how much optimizer time is required to reach them. Fixed-feature analyses explain how weight decay can create a long transition-time without changing the representation;

circuit analyses describe the algorithm implemented after generalization. Residual-dependent neural-tangent-hierarchy dynamics connect the two by specifying how post-fit supervision can reshape the tangent geometry from which a generalizing circuit is built.

B EXACT HOMOGENEOUS FUNCTION-SPACE DYNAMICS

For reference, the notation used throughout the derivations is summarized below.

Symbol	Meaning
$\boldsymbol{\sigma} = \mathbf{f} - \mathbf{y}$	stacked training residual
$J = \nabla_{\boldsymbol{\theta}} \mathbf{f}$	training-set parameter Jacobian
$K = JJ^\top$	empirical neural tangent kernel
D	degree of parameter homogeneity
$\mathcal{N}$	global normalization multiplying $\|\boldsymbol{\sigma}\|^2/2$ in the implemented loss
λ_W	optimizer coupled L_2 weight-decay coefficient
$\lambda_W^{\mathrm{norm}} = \lambda_W/\mathcal{N}$	normalized decay used in continuous time
η	optimizer gradient-descent learning rate
t	normalized gradient-flow time, $t = \mathcal{N} t_{\mathrm{opt}}$
s	discrete optimizer update index
Λ	task-direction NTK strength $\mathbf{u}^\top K \mathbf{u}$
a	projected residual-to-NTK coupling coefficient
y	target coefficient in the selected task direction
Λ_{g}	reduced strength representing a held-out criterion
T	finite optimizer update budget

Table 1: Notation used in the main derivations.

B.1 PROOF OF THEOREM 1

We absorb the global data-loss prefactor into the normalized time introduced in Section 2.1. The data-gradient term then has unit coefficient, and optimizer weight decay appears through $\lambda_W^{\mathrm{norm}}$,

$$\dot{\boldsymbol{\theta}} = -J^\top \boldsymbol{\sigma} - \lambda_W^{\mathrm{norm}} \boldsymbol{\theta}, \tag{19}$$

$$\dot{\boldsymbol{\sigma}} = J\dot{\boldsymbol{\theta}} = -JJ^\top \boldsymbol{\sigma} - \lambda_W^{\mathrm{norm}} J\boldsymbol{\theta} = -K\boldsymbol{\sigma} - \lambda_W^{\mathrm{norm}} J\boldsymbol{\theta}. \tag{20}$$

For a D-homogeneous network, uniform rescaling of all trainable weights rescales the output by degree D. Differentiating this identity at unit scale gives the Euler relation that closes the parameter-space decay term in function space.

$$\mathbf{f}_{c\boldsymbol{\theta}} = c^D \mathbf{f}_{\boldsymbol{\theta}}, \tag{21}$$

$$\frac{d}{dc}\mathbf{f}_{c\boldsymbol{\theta}}\bigg|_{c=1} = J\boldsymbol{\theta} = D\mathbf{f} = D(\boldsymbol{\sigma} + \mathbf{y}). \tag{22}$$

Substituting this relation removes the remaining explicit dependence on $\boldsymbol{\theta}$.

$$\dot{\boldsymbol{\sigma}} = -(K + D\lambda_W^{\mathrm{norm}} I)\boldsymbol{\sigma} - D\lambda_W^{\mathrm{norm}} \mathbf{y}, \tag{23}$$

which is Eq. (5).

For fixed K, the residual equation is affine and linear. The matrix $M := K + D\lambda_W^{\mathrm{norm}} I$ is positive definite for $\lambda_W^{\mathrm{norm}} > 0$, even when K has null directions, so every residual component relaxes exponentially to a unique fixed point. We set

$$M := K + D\lambda_W^{\mathrm{norm}} I \succ 0, \qquad \boldsymbol{\sigma}_* := -D\lambda_W^{\mathrm{norm}} M^{-1}\mathbf{y},$$

and the solution can be written as

$$\dot{\boldsymbol{\sigma}} = -M(\boldsymbol{\sigma} - \boldsymbol{\sigma}_*), \tag{24}$$

$$\boldsymbol{\sigma}(t) - \boldsymbol{\sigma}_* = e^{-Mt}\big(\boldsymbol{\sigma}(0) - \boldsymbol{\sigma}_*\big). \tag{25}$$

We diagonalize K to expose the ridge form. Each eigenmode follows an independent scalar relaxation, and the ratio of its equilibrium residual to its target coefficient is determined by Λ_j relative to $D\lambda_W^{\mathrm{norm}}$. For $K\mathbf{u}_j = \Lambda_j \mathbf{u}_j$,

$$\dot{\sigma}_j = -(\Lambda_j + D\lambda_W^{\mathrm{norm}})\sigma_j - D\lambda_W^{\mathrm{norm}} y_j, \tag{26}$$

$$\sigma_{j,*} = -\frac{D\lambda_W^{\mathrm{norm}}}{\Lambda_j + D\lambda_W^{\mathrm{norm}}}\, y_j. \tag{27}$$

For ReLU networks, these identities hold on every interval on which the activation pattern is fixed. At activation-boundary crossings the network remains continuous and the gradient-flow equation holds almost everywhere, so the same function-space relation extends piecewise across the trajectory.

This proves Theorem 1. □

C One-mode neural-tangent-hierarchy closure

C.1 Assumptions behind the modal closure

For the normalized Fourier direction $\mathbf{u}$ used in the main text, define

$$q_{\boldsymbol{\theta}} := \mathbf{u}^\top \mathbf{f}_{\boldsymbol{\theta}}, \qquad \Lambda := \mathbf{u}^\top K \mathbf{u} = \|\nabla_{\boldsymbol{\theta}} q_{\boldsymbol{\theta}}\|^2, \qquad \sigma := \mathbf{u}^\top \boldsymbol{\sigma}. \tag{28}$$

The reduction follows one task-bearing direction rather than approximating the full kernel. Over the transition interval, we assume that the kernel acts approximately diagonally on this direction and that the residual driving it is dominated by the component along $\mathbf{u}$:

$$\boldsymbol{\sigma} \approx \sigma \mathbf{u}, \qquad K\mathbf{u} \approx \Lambda \mathbf{u}. \tag{29}$$

These conditions suppress leading-order mixing with the remaining modes.

With

$$K^{(2)}_{ijk} = \langle \nabla_{\boldsymbol{\theta}} K_{ij}, \nabla_{\boldsymbol{\theta}} f_k \rangle, \tag{30}$$

the projected hierarchy term becomes

$$\mathbf{u}^\top (K^{(2)}\boldsymbol{\sigma})\mathbf{u} \approx \sigma \sum_{ijk} u_i u_j u_k K^{(2)}_{ijk}, \tag{31}$$

$$a(\boldsymbol{\theta}) := \sum_{ijk} u_i u_j u_k K^{(2)}_{ijk} = \langle \nabla_{\boldsymbol{\theta}} \Lambda, \nabla_{\boldsymbol{\theta}} q_{\boldsymbol{\theta}} \rangle. \tag{32}$$

Thus $a(\boldsymbol{\theta})$ is the local proportionality between the residual coefficient σ and the residual-driven change in the NTK strength Λ. When $a > 0$ and $\sigma < 0$ for a positive target coefficient, this term increases Λ. The sign of a is a dynamical property; homogeneity does not determine it. The projected kernel equation is therefore

$$\dot{\Lambda} = -a(\boldsymbol{\theta})\sigma - 2\lambda_W^{\mathrm{norm}}\Lambda. \tag{33}$$

Whenever $\sigma \neq 0$, the same equation can be inverted along a measured trajectory:

$$a_{\mathrm{eff}}(t) = -\frac{\dot{\Lambda}(t) + 2\lambda_W^{\mathrm{norm}}\Lambda(t)}{\sigma(t)}. \tag{34}$$

The constant-a approximation is local and does not require a to remain fixed throughout training. In the two-homogeneous MLP, both Λ and a are degree two under uniform parameter rescaling,

$$\Lambda(c\boldsymbol{\theta}) = c^2\Lambda(\boldsymbol{\theta}), \qquad a(c\boldsymbol{\theta}) = c^2 a(\boldsymbol{\theta}). \tag{35}$$

Under the one-mode parameter approximation,

$$\dot{\boldsymbol{\theta}} \approx -\sigma \nabla_{\boldsymbol{\theta}} q_{\boldsymbol{\theta}} - \lambda_W^{\mathrm{norm}} \boldsymbol{\theta}, \tag{36}$$

$$\dot{a} = -b(\boldsymbol{\theta})\sigma - 2\lambda_W^{\mathrm{norm}} a, \qquad b(\boldsymbol{\theta}) := \langle \nabla_{\boldsymbol{\theta}} a, \nabla_{\boldsymbol{\theta}} q_{\boldsymbol{\theta}} \rangle. \tag{37}$$

After the residual relaxes near its instantaneous ridge value, substitution into Eq. (37) gives

$$\dot{a} \simeq 2\lambda_W^{\mathrm{norm}} \left(\frac{b(\boldsymbol{\theta}) y}{\Lambda + 2\lambda_W^{\mathrm{norm}}} - a \right), \tag{38}$$

and

$$\frac{d}{dt}\left(\frac{a}{\Lambda}\right) = \frac{\sigma}{\Lambda^2}\left(a^2 - b\Lambda\right). \tag{39}$$

Equations (37)–(38) show that weight decay does not introduce an additional fast timescale for a after residual relaxation. Pure radial decay also leaves a/Λ unchanged, so any remaining drift of this ratio is residual driven. We therefore use a locally constant coupling subject to four explicit conditions:

1. the projected residual and kernel satisfy Eq. (29);
2. $a(t)$ remains positive over the interval of interest;
3. $a(t)$ changes slowly relative to residual relaxation; and
4. coupling to other task modes is weak enough to be absorbed into the local coefficient a and the effective threshold Λ_{g}.

The cyclic symmetry of modular addition and the difference-structured split motivate approximate Fourier decoupling. The remaining conditions are assumptions of the scalar reduction.

C.2 Proof of Theorem 2

With a fixed locally, we combine the two first-order equations into a second-order equation for Λ. This form separates damping from the force that selects the equilibrium. Equation (10) gives

$$\ddot{\Lambda} = -a\dot{\sigma} - 2\lambda_W^{\mathrm{norm}}\dot{\Lambda} \tag{40}$$

$$= a(\Lambda + 2\lambda_W^{\mathrm{norm}})\sigma + 2a\lambda_W^{\mathrm{norm}} y - 2\lambda_W^{\mathrm{norm}}\dot{\Lambda}, \tag{41}$$

$$\sigma = -\frac{\dot{\Lambda} + 2\lambda_W^{\mathrm{norm}}\Lambda}{a}, \tag{42}$$

hence

$$\ddot{\Lambda} + (\Lambda + 4\lambda_W^{\mathrm{norm}})\dot{\Lambda} + 2\lambda_W^{\mathrm{norm}}\Lambda(\Lambda + 2\lambda_W^{\mathrm{norm}}) - 2a\lambda_W^{\mathrm{norm}} y = 0. \tag{43}$$

Equation (43) describes one-dimensional damped motion with state-dependent coefficient $\Lambda + 4\lambda_W^{\mathrm{norm}}$. We collect the remaining terms into the cubic potential

$$V(\Lambda) = \frac{2\lambda_W^{\mathrm{norm}}}{3}\Lambda^3 + 2\lambda_W^{\mathrm{norm}\,2}\Lambda^2 - 2a\lambda_W^{\mathrm{norm}} y\Lambda, \tag{44}$$

$$\mathcal{H} = \frac{1}{2}\dot{\Lambda}^2 + V(\Lambda) - V(\Lambda_*). \tag{45}$$

On the branch $\Lambda \geq 0$, positive decay makes the potential strictly convex. The shifted energy $\mathcal{H}$ combines displacement from its unique minimum with the kinetic term, and

$$V'(\Lambda) = 2\lambda_W^{\mathrm{norm}}\Lambda(\Lambda + 2\lambda_W^{\mathrm{norm}}) - 2a\lambda_W^{\mathrm{norm}} y, \tag{46}$$

$$V''(\Lambda) = 4\lambda_W^{\mathrm{norm}}(\Lambda + \lambda_W^{\mathrm{norm}}) > 0, \tag{47}$$

$$\dot{\mathcal{H}} = -(\Lambda + 4\lambda_W^{\mathrm{norm}})\dot{\Lambda}^2 \leq 0. \tag{48}$$

The stationary points of the first-order system coincide with the extrema of this potential. We solve the equilibrium conditions to obtain

$$\sigma_* = -\frac{2\lambda_W^{\mathrm{norm}}}{a}\Lambda_*, \tag{49}$$

$$\Lambda_*(\Lambda_* + 2\lambda_W^{\mathrm{norm}}) = ay, \tag{50}$$

$$\Lambda_\pm = -\lambda_W^{\mathrm{norm}} \pm \sqrt{\lambda_W^{\mathrm{norm}\,2} + ay}. \tag{51}$$

For $y = 0$, the only nonnegative root is zero. For $y > 0$, $\Lambda_+ > 0$ and $\Lambda_- < 0$, giving one positive target-bearing equilibrium. The Lyapunov calculation establishes dissipative motion toward the potential minimum; we use the Jacobian to verify the local asymptotic stability in Theorem 2. At the positive equilibrium,

$$A = \begin{pmatrix} -(\Lambda_* + 2\lambda_W^{\mathrm{norm}}) & -\sigma_* \\ -a & -2\lambda_W^{\mathrm{norm}} \end{pmatrix}, \tag{52}$$

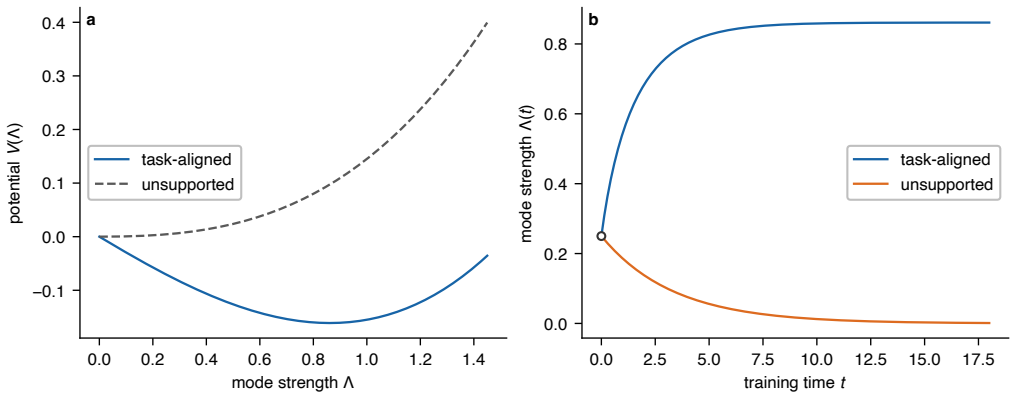


Figure 5: Reduced potential and adiabatic trajectories. **(a)** With illustrative values $\lambda_W^{\mathrm{norm}} = 0.15$ and $a = 1$, the task-aligned mode ($y = 1$) has its potential minimum at positive Λ, whereas the unsupported mode ($y = 0$) is minimized at $\Lambda = 0$. **(b)** Starting from the same initial strength, the two adiabatic trajectories move toward their respective minima. These dimensionless values illustrate the reduced dynamics and are not fitted to an empirical run.

with

$$\operatorname{tr} A = -(\Lambda_* + 4\lambda_W^{\mathrm{norm}}) < 0, \tag{53}$$

$$\det A = 2\lambda_W^{\mathrm{norm}}(\Lambda_* + 2\lambda_W^{\mathrm{norm}}) - a\sigma_* \tag{54}$$

$$= 4\lambda_W^{\mathrm{norm}}(\Lambda_* + \lambda_W^{\mathrm{norm}}) > 0. \tag{55}$$

The negative trace and positive determinant place both eigenvalues in the open left half-plane, so the positive equilibrium is locally asymptotically stable on the nonnegative branch. □

D Adiabatic reduction and finite-training boundary

D.1 From smooth mode growth to an accuracy transition

The reduced dynamics produce a smooth trajectory $\Lambda(t)$, whereas held-out accuracy changes only when individual predictions change class. For held-out example μ with correct class c_μ, define its correct-class margin along the training trajectory by

$$m_\mu(t) = f_{c_\mu}(\mathbf{x}_\mu; t) - \max_{c \neq c_\mu} f_c(\mathbf{x}_\mu; t). \tag{56}$$

When $\Lambda(t)$ is monotone over the transition, the trajectory can be reparameterized by Λ, so that $\widetilde{m}_\mu(\Lambda) := m_\mu(t(\Lambda))$. If each relevant $\widetilde{m}_\mu(\Lambda)$ increases monotonically through zero over its classification transition, then each example has a mode strength at which its predicted class changes. Denote this threshold by

$$\widetilde{m}_\mu(\Lambda_{\mu,\mathrm{g}}) = 0. \tag{57}$$

In this case, held-out accuracy at a given Λ is the empirical cumulative distribution of the example-specific thresholds,

$$\mathrm{Acc}(\Lambda) = \frac{1}{M}\sum_{\mu=1}^{M} \mathbf{1}\{\Lambda > \Lambda_{\mu,\mathrm{g}}\} = \widehat{F}_{\mathrm{g}}(\Lambda). \tag{58}$$

A narrow distribution of $\Lambda_{\mu,\mathrm{g}}$ therefore produces a sharp accuracy rise even when $\Lambda(t)$ changes smoothly. No discontinuity in the optimization dynamics is required. As an illustration, a logistic threshold distribution with location μ_g and scale b_g gives

$$\mathrm{Acc}(\Lambda) \approx \frac{1}{1 + \exp[-(\Lambda - \mu_g)/b_g]}. \tag{59}$$

This logistic form is only an example of the readout, the one-mode dynamics do not depend on it.

D.2 Adiabatic slow equation

The residual relaxes at rate $\Lambda + 2\lambda_W^{\mathrm{norm}}$, while the NTK mode moves on the slower decay-controlled scale once the residual is close to its ridge value. Under this separation of timescales, we set $\dot{\sigma} \simeq 0$ at the current Λ and define the resulting adiabatic residual $\sigma_{\mathrm{ad}}(\Lambda)$. Equation (10) gives

$$0 = -(\Lambda + 2\lambda_W^{\mathrm{norm}})\sigma - 2\lambda_W^{\mathrm{norm}} y, \tag{60}$$

$$\sigma_{\mathrm{ad}}(\Lambda) = -\frac{2\lambda_W^{\mathrm{norm}} y}{\Lambda + 2\lambda_W^{\mathrm{norm}}}, \tag{61}$$

$$\dot{\Lambda} = -a\sigma_{\mathrm{ad}}(\Lambda) - 2\lambda_W^{\mathrm{norm}}\Lambda \tag{62}$$

$$= 2\lambda_W^{\mathrm{norm}} \frac{ay - \Lambda^2 - 2\lambda_W^{\mathrm{norm}}\Lambda}{\Lambda + 2\lambda_W^{\mathrm{norm}}}. \tag{63}$$

The numerator in Eq. (63) determines the sign of the mode growth. It is positive below the stable fixed point and vanishes at that fixed point, so the upward motion slows as Λ approaches equilibrium. Separating variables gives

$$dt = \frac{1}{2\lambda_W^{\mathrm{norm}}} \frac{\Lambda + 2\lambda_W^{\mathrm{norm}}}{ay - \Lambda^2 - 2\lambda_W^{\mathrm{norm}}\Lambda}\, d\Lambda. \tag{64}$$

Integrating from Λ_0 to Λ_{g} gives the normalized gradient-flow time required to reach the held-out threshold. After T optimizer updates measured from the beginning of the slow stage,

$$t \simeq \mathcal{N}\eta T, \tag{65}$$

$$\mathcal{N}\lambda_W^{\mathrm{norm}} = \lambda_W. \tag{66}$$

Equating the available normalized time to the crossing time gives Eq. (15). The conversion from updates contributes $\mathcal{N}$, while $\mathcal{N}\lambda_W^{\mathrm{norm}} = \lambda_W$, leaving optimizer decay in the prefactor and normalized decay inside the integrand.

D.3 Closed form, critical slowing, and the fitting cutoff

We factor the denominator of the crossing-time integrand at the two fixed points of the slow equation. Define

$$q := ay, \qquad \Lambda_\pm := -\lambda_W^{\mathrm{norm}} \pm \sqrt{\lambda_W^{\mathrm{norm}\,2} + q}. \tag{67}$$

Then

$$q - \Lambda^2 - 2\lambda_W^{\mathrm{norm}}\Lambda = -(\Lambda - \Lambda_+)(\Lambda - \Lambda_-), \tag{68}$$

$$C_- := \frac{2\lambda_W^{\mathrm{norm}} + \Lambda_+}{\Lambda_+ - \Lambda_-}, \tag{69}$$

$$C_+ := \frac{2\lambda_W^{\mathrm{norm}} + \Lambda_-}{\Lambda_- - \Lambda_+}. \tag{70}$$

The integrand becomes

$$\frac{\Lambda + 2\lambda_W^{\mathrm{norm}}}{q - \Lambda^2 - 2\lambda_W^{\mathrm{norm}}\Lambda} = -\frac{C_-}{\Lambda - \Lambda_+} - \frac{C_+}{\Lambda - \Lambda_-}. \tag{71}$$

Integrating between Λ_0 and Λ_{g} gives

$$\eta_*(\lambda_W) = \frac{1}{2\lambda_W T}\left[C_- \log\left|\frac{\Lambda_0 - \Lambda_+}{\Lambda_{\mathrm{g}} - \Lambda_+}\right| + C_+ \log\left|\frac{\Lambda_0 - \Lambda_-}{\Lambda_{\mathrm{g}} - \Lambda_-}\right|\right]. \tag{72}$$

As the requested threshold approaches the stable fixed point Λ_+, the pole at Λ_+ reaches the upper integration limit while the Λ_- contribution remains finite. Hence, for $\Lambda_{\mathrm{g}} \uparrow \Lambda_+$,

$$\eta_*(\lambda_W) \sim \frac{C_-}{2\lambda_W T} \log\frac{1}{\Lambda_+ - \Lambda_{\mathrm{g}}}, \tag{73}$$

$$\Lambda_+ - \Lambda_{\mathrm{g}} = \Theta\big(\lambda_W^{\mathrm{mode}} - \lambda_W\big)\ . \tag{74}$$

The logarithm arises because the growth rate of Λ vanishes at the reachability boundary. Away from this cutoff, the logarithm remains finite and the explicit prefactor gives the leading $1/(\lambda_W T)$ dependence.

The training-fit cutoff in Eq. (17) follows from the same adiabatic gain. For a representative training mode, the expressed fraction of its target component is

$$g(\Lambda_{\rm fit}) = \frac{\Lambda_{\rm fit}}{\Lambda_{\rm fit} + 2\lambda_W^{\rm norm}}. \tag{75}$$

Requiring $g(\Lambda_{\rm fit}) \geq g_{\rm min}$ gives

$$\frac{\Lambda_{\rm fit}}{\Lambda_{\rm fit} + 2\lambda_W^{\rm norm}} \geq g_{\rm min}, \tag{76}$$

$$\lambda_W^{\rm norm} \leq \frac{\Lambda_{\rm fit}(1 - g_{\rm min})}{2g_{\rm min}}, \tag{77}$$

$$\lambda_W \leq \mathcal{N}\frac{\Lambda_{\rm fit}(1 - g_{\rm min})}{2g_{\rm min}}, \tag{78}$$

which is Eq. (17). The approximately vertical boundary used in the phase diagram further assumes that the representative training-mode strength $\Lambda_{\rm fit}$ varies weakly with η over the range of interest.

D.4 Discrete-time stability of the frozen-kernel dynamics

Our continuous-time reduction does not describe the large-step edge of the optimizer plane. We hold the kernel fixed to isolate the Euler stability condition for the fast residual dynamics and set

$$h := \mathcal{N}\eta.$$

For fixed K,

$$\boldsymbol{\sigma}_{s+1} = \boldsymbol{\sigma}_s - h\left[(K + 2\lambda_W^{\rm norm} I)\boldsymbol{\sigma}_s + 2\lambda_W^{\rm norm}\mathbf{y}\right], \tag{79}$$

$$\boldsymbol{\sigma}_* = -2\lambda_W^{\rm norm}(K + 2\lambda_W^{\rm norm} I)^{-1}\mathbf{y}, \tag{80}$$

$$\boldsymbol{\delta}_s := \boldsymbol{\sigma}_s - \boldsymbol{\sigma}_*, \tag{81}$$

$$\boldsymbol{\delta}_{s+1} = \left[I - h(K + 2\lambda_W^{\rm norm} I)\right]\boldsymbol{\delta}_s. \tag{82}$$

After we subtract the fixed point, each kernel eigendirection evolves independently. For $K\mathbf{u}_j = \Lambda_j \mathbf{u}_j$,

$$\delta_{j,s+1} = \left[1 - h(\Lambda_j + 2\lambda_W^{\rm norm})\right]\delta_{j,s}. \tag{83}$$

Therefore

$$|1 - h(\Lambda_j + 2\lambda_W^{\rm norm})| < 1, \tag{84}$$

$$\mathcal{N}\eta < \frac{2}{\Lambda_{\max}(K) + 2\lambda_W^{\rm norm}}, \tag{85}$$

$$\eta < \frac{2}{\mathcal{N}\Lambda_{\max}(K) + 2\lambda_W}, \tag{86}$$

which recovers Eq. (18). The most restrictive eigenvalue is $\Lambda_{\max}(K)$. Feature learning changes this spectrum, so we use the bound only as a local stability condition for the frozen-kernel approximation, not as a global guarantee for nonlinear training.

E Jacobian-mediated feature dynamics beyond homogeneity

For any differentiable model trained with squared loss and coupled L_2 decay, the chain rule separates the residual-dependent contribution to tangent-kernel evolution from the contribution of radial parameter decay. Homogeneity is only needed to convert the latter into fixed multiples of the output and NTK. Without homogeneity, the residual-dependent term remains exact, while the radial contractions become architecture dependent.

E.1 Exact differentiable-model identities

Proposition 1 (Exact Jacobian-mediated dynamics for differentiable models)**.** *Let*

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}) = \frac{1}{2}\left\|\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{y}\right\|^2 + \frac{\lambda_W^{\text{norm}}}{2}\left\|\boldsymbol{\theta}\right\|^2, \tag{87}$$

with $\boldsymbol{\sigma} = \mathbf{f} - \mathbf{y}$, $J = \nabla_{\boldsymbol{\theta}}\mathbf{f}$, *and* $K = JJ^\top$. *Then*

$$\dot{\boldsymbol{\sigma}} = -K\boldsymbol{\sigma} - \lambda_W^{\text{norm}} J\boldsymbol{\theta}, \tag{88}$$

$$\dot{K}_{ij} = -\sum_k K^{(2)}_{ijk}\sigma_k - \lambda_W^{\text{norm}} R_{ij}, \tag{89}$$

where

$$K^{(2)}_{ijk} = \langle \nabla_{\boldsymbol{\theta}} K_{ij}, \nabla_{\boldsymbol{\theta}} f_k \rangle, \qquad R_{ij} = \langle \nabla_{\boldsymbol{\theta}} K_{ij}, \boldsymbol{\theta} \rangle. \tag{90}$$

Along the same flow, the loss is monotone because

$$\frac{d}{dt}\widetilde{\mathcal{L}}(\boldsymbol{\theta}(t)) = -\left\|J^\top\boldsymbol{\sigma} + \lambda_W^{\text{norm}}\boldsymbol{\theta}\right\|^2 \le 0. \tag{91}$$

Proof. We differentiate the model output along parameter gradient flow to obtain the residual identity. Applying the same derivative to each entry of K produces one contraction with the data gradient and one with the radial weight-decay direction.

$$\dot{\boldsymbol{\theta}} = -J^\top\boldsymbol{\sigma} - \lambda_W^{\text{norm}}\boldsymbol{\theta}, \tag{92}$$

$$\dot{\boldsymbol{\sigma}} = J\dot{\boldsymbol{\theta}} = -K\boldsymbol{\sigma} - \lambda_W^{\text{norm}} J\boldsymbol{\theta}, \tag{93}$$

$$\dot{K}_{ij} = \left\langle \nabla_{\boldsymbol{\theta}} K_{ij}, \dot{\boldsymbol{\theta}} \right\rangle \tag{94}$$

$$= -\sum_k \langle \nabla_{\boldsymbol{\theta}} K_{ij}, \nabla_{\boldsymbol{\theta}} f_k \rangle \sigma_k - \lambda_W^{\text{norm}} \langle \nabla_{\boldsymbol{\theta}} K_{ij}, \boldsymbol{\theta} \rangle, \tag{95}$$

$$\frac{d}{dt}\widetilde{\mathcal{L}} = \left\langle \nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}, \dot{\boldsymbol{\theta}} \right\rangle = -\left\|\nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}\right\|^2. \tag{96}$$

□

Equation (89) isolates the extension needed beyond homogeneous networks. Whenever the $K^{(2)}$ contraction is nonzero, the residual that controls prediction error also changes the tangent kernel. The contraction may vanish at individual states; homogeneity is needed only to replace the remaining radial term by fixed coefficients.

E.2 Local projected dynamics

For a task direction $\mathbf{u}$, we suppose over a training interval that

$$K\mathbf{u} \approx \Lambda\mathbf{u}, \quad \boldsymbol{\sigma} \approx \sigma\mathbf{u}, \quad \mathbf{u}^\top J\boldsymbol{\theta} \approx \beta(\sigma + y), \quad \mathbf{u}^\top R\mathbf{u} \approx \rho\Lambda, \quad \mathbf{u}^\top(K^{(2)}\boldsymbol{\sigma})\mathbf{u} \approx a\sigma, \tag{97}$$

with slowly varying β, ρ, a. The coefficients β and ρ summarize radial parameter decay after projection onto the local task direction, while a retains its residual-to-kernel role. Projecting gives

$$\dot{\sigma} \approx -(\Lambda + \beta\lambda_W^{\text{norm}})\sigma - \beta\lambda_W^{\text{norm}} y, \qquad \dot{\Lambda} \approx -a\sigma - \rho\lambda_W^{\text{norm}}\Lambda. \tag{98}$$

For a two-homogeneous model, Euler identities set $\beta = \rho = 2$ and recover Eq. (10). In a non-homogeneous model, we must measure these local contractions; they need not equal the homogeneous coefficients. If they remain order one and vary slowly through the transition, both decay terms retain the factor λ_W^{norm}, leaving $1/\lambda_W^{\text{norm}}$ as the natural normalized-time scale and $(\eta\lambda_W)^{-1}$ as the corresponding update scale.

For the Transformer in Appendix G.3, we let $\mathbf{f}_{\boldsymbol{\theta}}$ denote the softmax probabilities entering the mean-squared-error loss. Proposition 1 applies on differentiable intervals, and hence almost everywhere along training, despite attention, LayerNorm, residual paths, biases, and the softmax output map.

F DIRECTIONAL NTK-ALIGNMENT INTERVENTION

F.1 EARLIER TASK ALIGNMENT SHORTENS THE PREDICTED DELAY

Starting closer to the held-out threshold removes a positive part of the crossing-time integral without invoking the global kinetic calibration. Equation (15) gives

$$s_{\mathrm{g}} = \frac{1}{2\eta\lambda_W} \int_{\Lambda_0}^{\Lambda_{\mathrm{g}}} \frac{\Lambda + 2\lambda_W^{\mathrm{norm}}}{ay - \Lambda^2 - 2\lambda_W^{\mathrm{norm}}\Lambda}\, d\Lambda, \tag{99}$$

and differentiation with respect to the initial strength gives

$$\frac{\partial s_{\mathrm{g}}}{\partial \Lambda_0} = -\frac{1}{2\eta\lambda_W} \frac{\Lambda_0 + 2\lambda_W^{\mathrm{norm}}}{ay - \Lambda_0^2 - 2\lambda_W^{\mathrm{norm}}\Lambda_0} < 0 \tag{100}$$

throughout the reachable branch, where the denominator is the positive spectral drive. The sign predicts that improving early alignment with a reachable task direction should advance the later held-out transition even if the auxiliary intervention is removed before generalization begins.

F.2 A TEMPORARY NTK INTERVENTION ADVANCES HELD-OUT GENERALIZATION

We change early task-indexed tangent geometry while keeping the later objective identical, which tests the sign in Eq. (100). We use addition modulo 97 on the 4,753 unordered input pairs. We train a bias-free $194 \to 128 \to 128 \to 128 \to 97$ ReLU teacher with He-normal initialization. Its $50/50$ split uses NumPy seed 0 and contains 2,376 training and 2,377 held-out examples. We run 30,000 full-batch SGD updates without momentum, using learning rate 180, coupled weight decay 10^{-5}, ordinary PyTorch MSE between logits and one-hot targets averaged over samples and outputs, and model seed 0.

For examples $\mathbf{x}_\mu, \mathbf{x}_\nu$ with labels c_μ, c_ν, we project the empirical NTK through the corresponding true-label logits,

$$K_{\mu\nu}^{\mathrm{TL}} = \left\langle \nabla_{\boldsymbol{\theta}} f_{c_\mu}(\mathbf{x}_\mu), \nabla_{\boldsymbol{\theta}} f_{c_\nu}(\mathbf{x}_\nu) \right\rangle. \tag{101}$$

The intervention uses a normalization order different from the diagnostic in Appendix G.1. We first cosine-normalize the sample kernel,

$$\widehat{K}_{\mu\nu}^{\mathrm{TL}} = \frac{K_{\mu\nu}^{\mathrm{TL}}}{\sqrt{K_{\mu\mu}^{\mathrm{TL}} K_{\nu\nu}^{\mathrm{TL}}}}, \tag{102}$$

and then average between label groups,

$$\bar{K}_{cc'} = \frac{1}{n_c n_{c'}} \sum_{\mu:c_\mu=c} \sum_{\nu:c_\nu=c'} \widehat{K}_{\mu\nu}^{\mathrm{TL}}. \tag{103}$$

We compute the fixed symmetric 97×97 target matrix from the teacher's training split and the differentiable student matrix from each student's training split.

We train 40 paired bias-free $194 \to 256 \to 97$ ReLU students with He-normal initialization. Pair $q \in \{0, \ldots, 39\}$ uses model seed q and an independently generated $50/50$ split with NumPy seed $10{,}000 + q$; every split has 2,376 training and 2,377 held-out examples, and both members of the pair share the same initialization and data. Every student uses 30,000 full-batch SGD updates without momentum, learning rate 35, coupled weight decay 3×10^{-6}, and ordinary PyTorch MSE between logits and one-hot targets averaged over samples and outputs. During updates 0 through 2,999, the intervention member minimizes

$$\mathcal{L}_{\mathrm{int}} = \mathcal{L}_{\mathrm{student}} + \lambda_{\mathrm{NTK}} \left\| \bar{K}_{\boldsymbol{\theta}} - \bar{K}_{\mathrm{teacher}} \right\|_F, \qquad \lambda_{\mathrm{NTK}} = 0.10. \tag{104}$$

We remove the auxiliary term at update 3,000; the baseline never receives it. We record training and held-out accuracy and MSE every 100 updates.

The paired mean speedup is 1.498, and the median paired speedup is 1.489. The intervention students also maintain lower held-out MSE after the penalty has been removed. Because the two members of each pair share their data and initialization and use the same objective after update 3,000, the shift is consistent with the predicted negative derivative in Eq. (100). Matching the full 97×97 label kernel changes several Fourier components and may alter other parts of the trajectory, so the experiment tests the direction of the initial-alignment effect rather than the scalar one-mode closure in isolation.

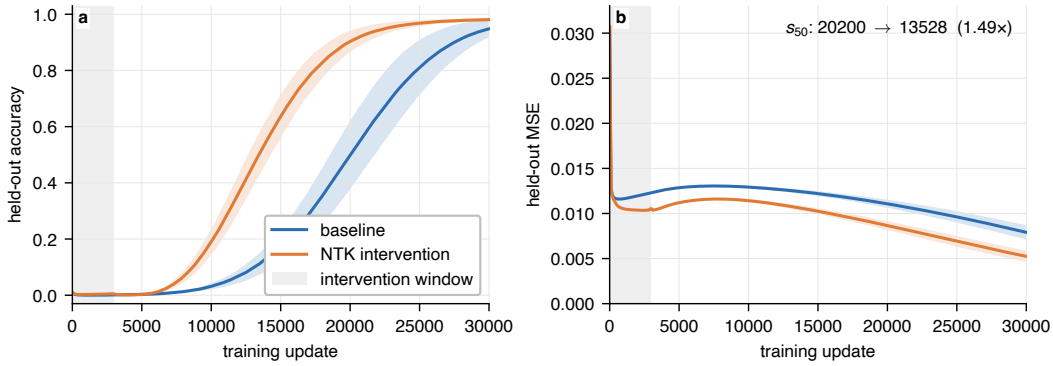


Figure 6: Temporary early NTK alignment advances held-out generalization. **(a)** Mean held-out accuracy for the paired baseline and intervention students. The shaded vertical region marks the first 3,000 updates, when the NTK penalty is active; bands show one sample standard deviation across 40 paired seeds. **(b)** Mean held-out MSE for the same runs. The mean first 50% held-out crossing moves from 20,200 updates for the baseline to 13,527.5 for the intervention, a ratio of 1.493.

G Experimental configurations and analysis conventions

The accompanying experiment package centralizes the configurations below in the model, training, and analysis code. We report optimizer weight decay in the convention passed directly to PyTorch SGD.

G.1 Spectral diagnostics on addition modulo 97

For Figure 2, we use the $97 \cdot 98/2 = 4{,}753$ unordered input pairs allowed by commutativity. A NumPy generator with seed 0 permutes the pairs; the first 2,376 are used for training and the remaining 2,377 are held out. Each input concatenates two 97-dimensional one-hot vectors, and the target is the 97-dimensional one-hot encoding of the modular sum.

The model is a bias-free $194 \to 256 \to 97$ one-hidden-layer ReLU MLP with He-normal initialization and model seed 0. We train for 20,000 full-batch SGD updates with learning rate 40, coupled weight decay 2.5×10^{-5}, and no momentum. The loss is ordinary PyTorch MSE averaged over samples and all 97 output coordinates, so $\mathcal{N} = 2/(2376 \cdot 97) = 1/115{,}236$ and $\lambda_W^{\mathrm{norm}} = 2.8809$. Accuracy is recorded every 100 updates.

For training examples $\mathbf{x}_\mu, \mathbf{x}_\nu$ with labels c_μ, c_ν, the true-label-projected sample NTK is

$$K^{\mathrm{TL}}_{\mu\nu} = \left\langle \nabla_{\boldsymbol{\theta}} f_{c_\mu}(\mathbf{x}_\mu), \nabla_{\boldsymbol{\theta}} f_{c_\nu}(\mathbf{x}_\nu) \right\rangle . \tag{105}$$

We first average these entries between groups of examples with the same output labels, producing a 97×97 matrix $\widetilde{K}$. We then diagonal-normalize and symmetrize the label kernel,

$$\bar{K}_{cc'} = \frac{\widetilde{K}_{cc'}}{\sqrt{\widetilde{K}_{cc}\widetilde{K}_{c'c'}}}. \tag{106}$$

Panel (a) uses updates 0, 3,000, 8,000, and 20,000. Eigensystems are computed at updates 0, 100, 500, 1,000, 3,000, 6,000, 8,000, 10,000, and 20,000. Eigenvectors are matched backward through these checkpoints by maximum absolute overlap, with signs chosen continuously. Panel (c) displays four tracked leading eigenvectors at updates 0, 1,000, 6,000, and 20,000.

G.2 High-resolution MLP phase sweep on addition modulo 23

We use all $23^2 = 529$ ordered input pairs. Each input concatenates two 23-dimensional one-hot vectors, and the target is a 23-dimensional one-hot encoding of the modular sum. NumPy seed 2027 selects the training differences

$$\mathcal{D} = \{0, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14, 16, 17, 19, 20, 22\}, \tag{107}$$

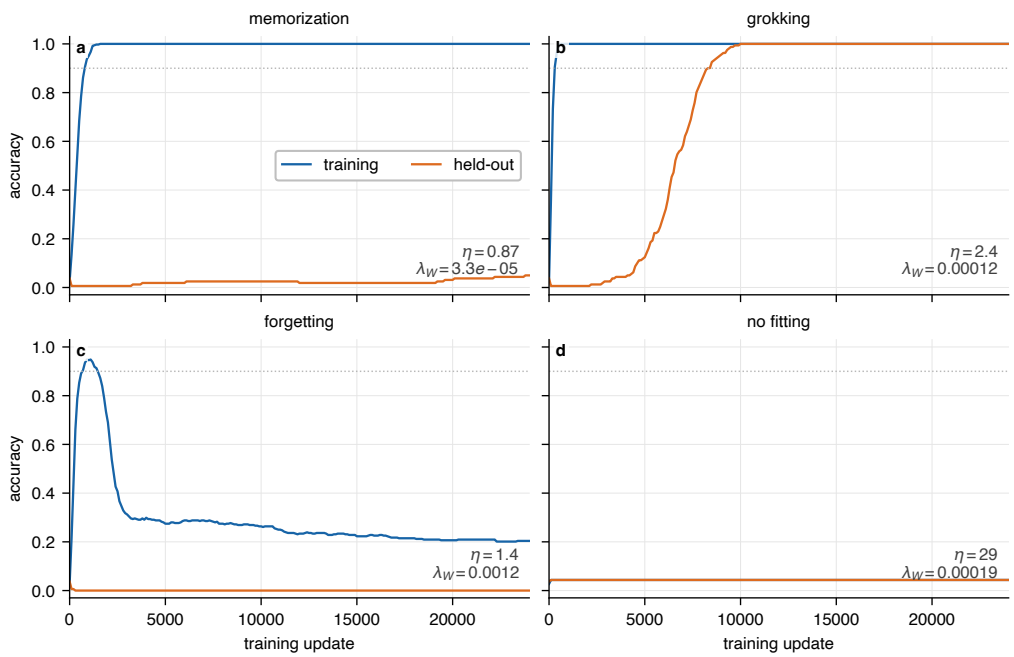


Figure 7: Representative trajectories for the four MLP phase outcomes. We show training and held-out accuracy for fixed cells from the modulo-23 sweep. The dotted horizontal line is the 90% threshold used to classify the phase map.

and we define

$$\mathcal{T}_{\mathcal{D}} = \{(a, b) : a - b \bmod 23 \in \mathcal{D}\}. \tag{108}$$

The split contains 368 training and 161 held-out pairs and is invariant under simultaneous shifts $(a, b) \mapsto (a + r, b + r)$.

At every grid point we initialize the same bias-free $46 \to 256 \to 23$ one-hidden-layer ReLU MLP with He-normal initialization and model seed 0. Training uses ordinary PyTorch MSE averaged over the 368×23 residual coordinates, full-batch SGD, coupled L_2 weight decay, and no momentum. Thus $\mathcal{N} = 2/(368 \cdot 23) = 1/4232$. Each run lasts 24,000 updates and is evaluated every 100 updates. The grid contains 84 logarithmically spaced learning rates from 0.1 to 100 and 90 logarithmically spaced weight decays from 5×10^{-6} to 2×10^{-3}, for 7,560 runs.

We classify runs using a 90% accuracy threshold. *Grokking* requires final training and held-out accuracy above 90%; *memorization* requires final training accuracy above 90% but held-out accuracy below it; *forgetting* means that training accuracy crossed 90% earlier but ends below it; all remaining runs are *no fitting*. For Figure 3b, we retain only runs that ultimately grok and record the first update at which held-out accuracy reaches 10%. This lower threshold marks the beginning of the final MLP generalization rise and does not affect the phase labels.

The representative runs in Figure 7 are fixed cells from the same dense sweep. Their (η, λ_W) values are $(0.87048, 3.2930 \times 10^{-5})$ for memorization, $(2.36316, 1.18329 \times 10^{-4})$ for grokking, $(1.43426, 1.24846 \times 10^{-3})$ for forgetting, and $(28.6967, 1.89561 \times 10^{-4})$ for no fitting. The grokking example first reaches 90% held-out accuracy at update 8,300. The forgetting example reaches peak training accuracy 0.948 and ends at 0.204.

G.2.1 Boundary extraction and calibration

For each usable weight-decay column, we define the memorization–grokking boundary as the geometric midpoint between the last memorizing learning rate and the first grokking learning rate. We exclude the detached high-step island. The finite-time boundary uses 53 midpoints over $2 \times 10^{-5} \le \lambda_W \le 7 \times 10^{-4}$ with equal weight in log learning rate.

To evaluate Eq. (15), define the scaled coordinates

$$\widehat{\Lambda} = \mathcal{N}\Lambda, \qquad \widehat{q} = \mathcal{N}^2 a y. \tag{109}$$

This removes $\mathcal{N}$ from the integrand while leaving the optimizer prefactor $1/(2\lambda_W T)$ unchanged. The calibrated finite-time curve is

$$\boxed{\eta_{\mathrm{fit}}(\lambda_W) = \frac{C_\eta}{2\lambda_W T} \int_{\widehat{\Lambda}_0}^{\widehat{\Lambda}_g} \frac{\widehat{\Lambda} + 2\lambda_W}{\widehat{q} - \widehat{\Lambda}^2 - 2\lambda_W \widehat{\Lambda}}\, d\widehat{\Lambda}} \tag{110}$$

with one global kinetic factor C_η. If omitted modes and slow variation in the projected coupling rescale the scalar velocity as $d\widehat{\Lambda}/dt \simeq \kappa F(\widehat{\Lambda})$, then crossing times are multiplied by $1/\kappa$ and $C_\eta = 1/\kappa$. In the empirical calibration we set $T = 24{,}000$, the nominal total update budget of the sweep, rather than using the run-dependent post-fit budget $24{,}000 - s_{\mathrm{fit}}$. Thus T is an effective approximation to the time available for the slow stage, with part of the resulting timing offset absorbed by the global factor C_η. Repeating the calibration with the post-fit budget produces only minor changes to the fitted boundary and does not alter the qualitative conclusions. The MSE normalization $\mathcal{N}$ is known exactly and is not part of this calibration. The quantities $\widehat{\Lambda}_0$, $\widehat{\Lambda}_g$, and $\widehat{q}$ are inferred jointly from the optimizer-plane boundary rather than measured from NTK trajectories. We therefore treat them as effective coordinates of the scalar boundary model: the fit tests the predicted dependence of the crossing time on learning rate and weight decay, but does not determine the absolute magnitude of post-fit NTK growth.

Panel (a) of Table 2 collects the values used for every overlay in Figure 3a. The mode-reachability cutoff is derived from the fitted reduced coordinates rather than fitted independently. The training-fit and stability overlays use the functional forms of Eqs. (17) and (18), with their locations calibrated separately to the observed boundaries. In particular, the stability calibration is summarized by an effective scale $\widehat{\Lambda}_{\mathrm{stab}}$ and is not obtained from a direct measurement of $\mathcal{N}\Lambda_{\max}(K)$.

Over the small-decay interval used for the inverse-decay comparison, the empirical boundary has log–log slope -1.1681, while the calibrated finite-time curve has slope -0.9043. These slopes are diagnostics of the boundary shape and are not additional fit parameters.

G.3 Transformer phase sweep on addition modulo 23

We use the same task, difference split, split seed, training size, and held-out size as in Appendix G.2, but present each input as a sequence of two 23-dimensional one-hot tokens. A learned affine encoder maps each token to $d_{\mathrm{model}} = 32$. The model contains one pre-norm Transformer block with four attention heads, a ReLU feed-forward width of 64, residual connections, learned embeddings for the two positions, a final LayerNorm, and an affine $64 \to 23$ decoder. We use model seed 0 at every grid point.

Training uses full-batch SGD with coupled L_2 weight decay and no momentum. The loss is PyTorch MSE between softmax probabilities and one-hot targets, averaged over samples and output coordinates, so $\mathcal{N} = 1/4232$. Each run lasts 60,000 updates and is evaluated every 250 updates. The phase file contains 42 uniformly logarithmically spaced learning rates from 5×10^{-3} to 10^{3} and 45 uniformly logarithmically spaced weight decays from 10^{-7} to 10^{-3}, for 1,890 independently trained networks. We use the same 90% phase definitions as for the MLP.

For every usable weight-decay column, we extract the geometric midpoint between the last memorizing and first grokking learning rate. We fit the inverse-decay form $\eta = A/\lambda_W$ over $10^{-5} \le \lambda_W \le 3 \times 10^{-4}$. The current phase grid yields the values in panel (b) of Table 2.

For the transition-time scaling panel, we retain runs that ultimately grok and record the first held-out 80% crossing. The 80% threshold is chosen to lie in the final generalization rise rather than near chance accuracy, $1/23 \approx 4.35\%$. On the current 1,890-run grid, the grokking cohort has median $\eta\lambda_W s_{80} = 0.920$.

G.4 Reduced-mode potential illustration

For Figure 5, we integrate Eq. (12) with dimensionless normalized decay $\lambda_W^{\mathrm{norm}} = 0.15$, coupling $a = 1$, initial strength $\Lambda(0) = 0.25$, fourth-order Runge–Kutta step 0.002, and final time 18. The

Curve	Relation	Calibration values	Fit range / quality
(a) MLP			
Finite-time boundary	Eq. (110)	$C_\eta = 24.840743$, $\widehat{\Lambda}_0 = 0.9281806$, $\widehat{\Lambda}_g = 0.9284920$, $\widehat{q} = 0.8633973$; $\kappa = C_\eta^{-1} = 0.0402564$	53 midpoints, $2 \times 10^{-5} \le \lambda_W \le 7 \times 10^{-4}$; log-MSE 0.026800; log-RMSE 0.163706
Inverse-decay asymptote	$\eta = A/\lambda_W$	$A = 1.0916395 \times 10^{-4}$	34 midpoints, $2 \times 10^{-5} \le \lambda_W \le 2 \times 10^{-4}$; log-RMSE 0.119902
Mode reachability	Eq. (16)	$\lambda_W^{\mathrm{mode}} = 7.0000 \times 10^{-4}$, derived from $\widehat{q}$ and $\widehat{\Lambda}_g$	no independent fit
Training fit	Eq. (17)	$\lambda_W^{\mathrm{fit}} = 8.3821151 \times 10^{-4}$	location minimizes final-training-accuracy misclassification
Frozen-kernel stability	Eq. (18)	effective $\widehat{\Lambda}_{\mathrm{stab}} = 0.1056011$; low-decay cutoff $\eta \simeq 18.9282$	location calibrated to observed instability boundary
(b) Transformer			
Inverse-decay boundary	$\eta = A/\lambda_W$	$A = 1.6875621 \times 10^{-5}$	16 boundary points, $10^{-5} \le \lambda_W \le 3 \times 10^{-4}$; log-RMSE 0.107016

Table 2: Boundary calibrations for the optimizer-plane experiments. Panel (a) lists the MLP phase-diagram overlays; panel (b) gives the inverse-decay fit for the Transformer memorization–grokking boundary. The theory specifies the functional forms of the MLP curves, while their locations are determined either by the reduced finite-time fit or by separate effective calibrations to the observed boundaries.

task-aligned trajectory uses $y = 1$, while the unsupported control uses $y = 0$. These values are chosen only to display the qualitative reduced dynamics and are not fitted to an empirical run.