FROM ANTI-GENERALIZATION TO GROKKING: TRAIN–TEST REPRESENTATION KERNEL DYNAMICS IN MODULAR ADDITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Grokking is a phenomenon in which a model generalizes long after fitting the training data. It is commonly attributed to delayed feature learning, yet a quantitative explanation linking evolving hidden representations to generalization remains incomplete. We develop a quantitative framework based on evolving train–train and train–test representation kernels, which directly connect hidden representations to training and test logits. An approximately label-aligned residual persists after fitting and sustains slow feature learning under weight decay. Within this framework, we show that early random-feature structure actively drives anti-generalization, pushing test accuracy from chance to nearly zero through spurious train–test correlations, whereas subsequent feature learning reorganizes the cross-kernel toward task-aligned structure and drives generalization. The sharp transition is controlled by finite-width fluctuations associated with the effective number of feature-aligned neurons. At late-time equilibrium, the learned feature-kernel geometry further predicts the cross-entropy loss through a mean-field scaling law in weight decay and problem size. Together, these results provide a quantitative representation-level account of grokking from early anti-generalization to late generalization.

1 INTRODUCTION

In conventional overfitting, continued training after fitting the training data tends to worsen test performance. Grokking instead exhibits the opposite late-time behavior: a model can fit or memorize the training set while generalizing poorly, yet only much later abruptly improve on unseen data (Power et al., 2022). Originally discovered in modular arithmetic, grokking has since been reproduced across different architectures and learning settings (Liu et al., 2023). A central question is therefore not only why a model can eventually generalize, but what continues to change internally after the training loss has already become very small.

Various mechanisms have been proposed to explain delayed generalization, including regularization, implicit bias, and competition between memorizing and generalizing solutions (Liu et al., 2022; 2023; Varma et al., 2023; Lyu et al., 2024). Recent studies connect grokking to transitions from kernel or lazy-training regimes to rich feature learning (Kumar et al., 2024; Mohamadi et al., 2024; Rubin et al., 2024; Lyu et al., 2024), while recent work has further separated fast fitting from slower representation learning (Chou et al., 2026; Tan et al., 2026). Another line of work characterizes the features eventually learned by the model. Nanda et al. (2023) showed that Transformers trained on modular addition develop Fourier representations, and Tian (2025) characterized the emergence of group-representation features in two-layer quadratic networks on finite-group tasks. He et al. (2026) further studied Fourier feature formation, frequency competition, and phase coupling in two-layer networks, with experiments using ReLU and cross-entropy loss. Recent work has also described post-fit grokking dynamics through the evolution of task-aligned neural tangent kernel modes under weight decay (Pracher et al., 2026). These works clarify when feature learning can occur and what features may emerge, but a quantitative description connecting the evolving hidden representation to train–test behavior throughout grokking remains elusive.

In this paper, we propose a quantitative framework based on the evolving train–train and train–test representation kernels, which act as dynamical representations of feature learning. We study modular

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.

addition in a two-layer ReLU network trained with cross-entropy (CE) loss and weight decay. We focus on this practical setting, for which analytical theories of grokking remain substantially less developed than for quadratic networks with MSE. We find that the late-stage residual field remains approximately aligned with the centered labels, allowing feature learning to persist even after the training loss has nearly converged. This provides a unified description of early- and late-stage feature learning and quantitatively characterizes their distinct learning timescales. We further find that the degree-one homogeneity of ReLU sustains substantial late-stage feature learning, in contrast to the quadratic activation considered in previous analytical models. This reduced dynamics naturally introduces the train–train and train–test Gram matrices as representation kernels. After the output-layer transient, these kernels directly determine the training and test logits. In this representation-kernel picture, grokking corresponds to a reorganization of the train–test kernel structure from spurious correlations toward task-aligned structure.

Within this framework, we show that (1) early training actively drives anti-generalization: spurious correlations dominate the train–test kernel, and test accuracy falls from random guessing to nearly zero; (2) the sharp accuracy rise during grokking begins near a characteristic train–test alignment threshold, while finite width controls the sharpness of this transition; and (3) at late-time equilibrium, the structure of the learned feature kernel quantitatively determines the CE loss through a scaling law in weight decay and problem size. Together, these results provide a representation-level description of grokking from early anti-generalization, through slow feature learning and the finite-width transition, to late-time generalization.

2 Gradient Descent Dynamics

2.1 Problem settings

We study modular addition $a + b \bmod p$ as a classification task in a two-layer ReLU network. Figure 1(a)(b) shows the corresponding training and test accuracy and loss dynamics. The inputs are two-hot codes and labels are one-hot codes. The output of the model is

$$Z = V\sigma(WX),\ P = \mathrm{softmax}(Z) \tag{1}$$

where W, V are the first- and second-layer weights, σ is the activation function and softmax is column-wise. Let X and Y be the inputs and labels and treat columns as input/output codes. The network width is N, and the number of samples is M. Let $x_{i\mu}, y_{i\mu}, p_{i\mu}$ be the matrix entries of X, Y, P respectively, $\mu = 0 \dots M-1$. For the full dataset, $M = p^2$, and for a training fraction α, $M = \lfloor \alpha p^2 \rfloor$. The L^2 weight decay parameter is $\lambda > 0$ and the learning rate is η. The CE loss is $\mathcal{L} = -\frac{1}{M}\sum_\mu \sum_h y_{h\mu} \log p_{h\mu}$. We further define the zero-mean label matrix $\tilde{Y} = Y - \frac{1}{p}$ and activations $F = \sigma(WX)$.

Define the residual field $S = Y - P$; the backpropagated gradients (without weight decay) on F and V are

$$\begin{aligned} G_F &= -\frac{\partial \mathcal{L}}{\partial F} = \frac{1}{M} V^\top (Y - P) = \frac{1}{M} V^\top S \\ G_V &= -\frac{\partial \mathcal{L}}{\partial V} = \frac{1}{M} (Y - P) F^\top = \frac{1}{M} S F^\top \end{aligned} \tag{2}$$

For analytical convenience, we describe gradient descent in its continuous-time form, with the learning rate absorbed into t, so that $\dot{V} = G_V - \lambda V$. We initialize the first-layer weights from a Gaussian distribution and set the second-layer weights to zero for the analytical treatment. Thus $Z(0) = 0$, $P(0) = \frac{1}{p}$, and $S(0) = \tilde{Y}$. A sufficiently small random initialization of V approaches this behavior as its scale tends to zero. To first order in t, with F evaluated at initialization, $V(t) = G_V t + O(t^2) = t \cdot \frac{1}{M}\tilde{Y}F^\top + O(t^2)$, $G_F = t \cdot \frac{1}{M^2} F\tilde{Y}^\top \tilde{Y} + O(t^2)$.

2.2 Label-Aligned Residual-Field Ansatz

The nonlinear dependence of the residual field $S = Y - P$ on the logits makes the CE dynamics difficult to track analytically. At late times, when the training loss has nearly converged, we estimate the residual field using the following ansatz. We focus on the overparameterized regime $p \ll N$ and

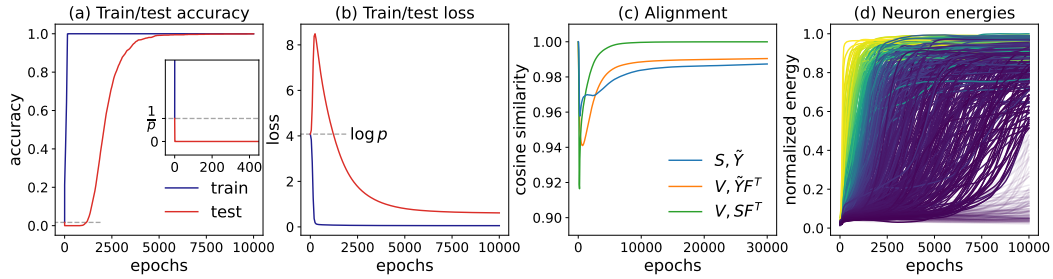

Figure 1: (a) Accuracy and (b) loss of the training and test sets for $p = 59, \alpha = 0.5, N = 512$. The test accuracy drops to 0 at the first epoch. (c) Cosine similarities of $\cos(S, \tilde{Y})$, $\cos(V, \tilde{Y}F^\top)$ and $\cos(V, SF^\top)$. Define $\cos(A, B) := \langle A, B\rangle_F / (\|A\|_F \|B\|_F)$. $\cos(S, \tilde{Y})$ checks the residual-field ansatz while $\cos(V, \tilde{Y}F^\top)$ and $\cos(V, SF^\top)$ check the V-locking relation. (d) Feature learning of each neuron represented in energy defined in Section 4. Plotted energy is normalized to $[0, 1]$.

on late-stage states with sufficiently small training loss on every sample. The softmax probabilities are normalized and peak at the correct label. For sample μ, let the correct label be h; its loss is $\mathcal{L}_\mu = -\log p_{h\mu}$. If the loss is sufficiently small on this sample, $p_{h\mu} \approx 1$ and $\mathcal{L}_\mu \approx 1 - p_{h\mu} \ll 1$. First, assume approximately uniform wrong-class probabilities. Normalization then gives $p_{i\mu} \approx \frac{1}{p-1}(1 - p_{h\mu})$ for $i \neq h$. Therefore the residual field satisfies $s_{i\mu} = \delta_{ih} - p_{i\mu} \approx \epsilon_\mu \tilde{y}_{i\mu}$, where $\epsilon_\mu = \frac{p}{p-1}\mathcal{L}_\mu$. Like $\tilde{Y}$, $S = Y - P$ has zero mean across classes in each column. Second, assume that late-stage losses are sufficiently concentrated across samples, so that $\epsilon_\mu \approx \epsilon$. With $\epsilon = \frac{p}{p-1}\mathcal{L} \ll 1$, these two approximations give the label-aligned residual-field ansatz,

$$S \simeq \epsilon \tilde{Y}. \tag{3}$$

Late-stage feature-learning. For t sufficiently large, V converges, so that $\dot{V} = G_V - \lambda V \simeq 0$. Then $\frac{1}{M} SF^\top - \lambda V \simeq 0$, $V(t) \simeq \frac{1}{\lambda M} SF^\top \simeq \frac{\epsilon}{\lambda M} \tilde{Y} F^\top$, therefore

$$G_F^{\text{late}} = \frac{1}{M} V^\top S \simeq \frac{\epsilon^2}{\lambda M^2} F \tilde{Y}^\top \tilde{Y} \tag{4}$$

Although the training loss is already small, the data-gradient contribution driving feature alignment remains nonzero. Before full parameter equilibrium is reached, this residual drive can support continued slow feature learning. At full equilibrium, the data gradient on W balances weight decay, giving $\dot{W} = 0$. We numerically verify the residual-field ansatz and the V-locking relation in Figure 1(c) by cosine similarity. We also note that this relation holds under CE loss, whereas fitting the output layer with fixed features under MSE yields a linear or ridge regression solution (Tian, 2025; Pracher et al., 2026).

Early-stage feature-learning. For $t \to 0$, we have already obtained

$$G_F(t) = t \cdot \frac{1}{M^2} F \tilde{Y}^\top \tilde{Y} + O(t^2) \tag{5}$$

The scalar prefactor of the task-aligned feature-learning drive changes from t/M^2 at early times to $\epsilon^2/(\lambda M^2)$ at late times. Their ratio is $\kappa = \frac{\lambda t}{\epsilon^2}$. Since F itself evolves, this is a ratio of gradient prefactors, rather than an unconditional ratio of gradient norms or actual feature-learning speeds. Further define $\dot{\kappa} = \frac{\kappa}{t} = \frac{\lambda}{\epsilon^2}$. A test run on $p = 59, \alpha = 0.5, N = 512$ shows that $\dot{\kappa}_{\text{real}} \approx 0.07$ which agrees with theoretically computed $\dot{\kappa}_{\text{th}} \approx 0.07$. $\dot{\kappa}_{\text{real}}$ is measured by the directional changing rate, see Appendix B.2. We illustrate in Figure 1(d) that early-stage learning curves have steeper slope than the late-stage. Moreover, the late-stage feature learning is almost absent in quadratic activation, see Appendix B.2.1.

3 KERNEL REPRESENTATION

One important question in grokking is why the model performs well on the training set but totally fails on the test set. We relate this difference to the sample-kernel structure under the label-aligned residual-field ansatz. In Section 2, we showed that at late-time quasi-equilibrium the output layer V is approximately locked to $SF^\top$. Motivated by the alignment evolution in Figure 1(c), we approximate V as a fast variable slaved to the slowly evolving W in the V-locking relation, and extend the quasi-equilibrium equation $G_V - \lambda V \approx 0$ and uniformity ansatz $S \propto \tilde{Y}$ to the starting time of loss convergence. Therefore, $V \propto \tilde{Y}F^\top$ during the low-loss training period. Moreover, as shown in Section 2, the leading-order $t \approx 0$ dynamics gives $V \propto t \cdot \tilde{Y}F^\top$.

As a consequence, the logit output is $Z = VF \propto \tilde{Y}F^\top F$. We then define the sample kernel as $K = F^\top F$. We use subscripts t and v to denote training and test (validation) sets. When the model is trained on the training set, $V \propto \tilde{Y}_t F_t^\top$, and the logit output is $Z_t \propto \tilde{Y}_t F_t^\top F_t$. Define the training kernel $K_{tt} = F_t^\top F_t$. The model logit output on the test set is $Z_v \propto \tilde{Y}_t F_t^\top F_v$. Define the train-test cross-kernel $K_{tv} = F_t^\top F_v$. Correct classification is ensured when the logits approximately align with the labels up to a class-independent shift. After centering across classes, this gives

$$\begin{aligned} &\text{Training set fitted} && \tilde{Y}_t K_{tt} \propto \tilde{Y}_t \\ &\text{Test set fitted} && \tilde{Y}_t K_{tv} \propto \tilde{Y}_v \end{aligned} \tag{6}$$

The training condition preserves the training-label subspace, whereas the cross-kernel condition transfers the task-aligned structure from training to unseen samples. For a test sample ν, define the class score $R_h(\nu) = \sum_{\mu:[\mu]=h} K^{tv}_{\mu\nu}$, where $[\mu]$ denotes the label of training sample μ. Under output locking, $Z_{h\nu} \approx \gamma(t)[R_h(\nu) - \frac{1}{p}\sum_g R_g(\nu)]$, where $\gamma(t) > 0$ is the locking coefficient. Thus, within this approximation, a test sample with label h_ν is classified correctly when $R_{h_\nu}(\nu) > \max_{g \neq h_\nu} R_g(\nu)$.

To better illustrate kernel representation, we set a specific selection rule for train and test sets which we use in the following context. For each label h, we randomly select two disjoint groups of samples with that label. Each group has size q. The maximal selection is therefore $q = \lfloor p/2 \rfloor$. Also, $M = pq$ and $\alpha = q/p$. Let X_t, Y_t and X_v, Y_v denote train and test datasets. We also require the sample index to be label-first, i.e., $\mu = (ha)$ corresponds to the data pair $(a, h-a; h)$. With the same label-first ordering in both sets, $Y_t = Y_v$. Such selection is balanced on each label, different from a totally random selection. Comparison between these two is discussed in Appendix B.1. Then $Y_t^\top Y_t = I_p \otimes \mathbf{1}_q \mathbf{1}_q^\top$, where I_p is the $p \times p$ identity and $\mathbf{1}_p \mathbf{1}_p^\top$ denotes the $p \times p$ all-ones matrix. Define the label kernels $B_{tt} \equiv \tilde{Y}_t^\top \tilde{Y}_t = I_p \otimes \mathbf{1}_q \mathbf{1}_q^\top - \frac{1}{p}\mathbf{1}_{pq}\mathbf{1}_{pq}^\top$ and $B_{tv} \equiv \tilde{Y}_t^\top \tilde{Y}_v = B_{tt}$. The rows of $\tilde{Y}_t$ satisfy $\tilde{Y}_t(Y_t^\top Y_t) = q\tilde{Y}_t$. Kernel components proportional to $Y_t^\top Y_t$ and $Y_t^\top Y_v$ support the training and test fitting conditions, respectively. We refer to the $q \times q$ diagonal blocks in the kernel as feature blocks.

3.1 RANDOM-INITIALIZATION KERNEL

In Section 2, we showed that the leading-order dynamics at $t \to 0$ gives $V(t) = t \cdot \frac{1}{M}\tilde{Y}_t F_{t0}^\top + O(t^2)$, where $F_{t0} = F_t(0)$. The leading-order training logits are therefore $Z_t(t) = t \cdot \frac{1}{M}\tilde{Y}_t F_{t0}^\top F_{t0} + O(t^2)$. Thus they are governed by the frozen random-feature Gram matrix $K_{tt}(0) = F_{t0}^\top F_{t0}$.

We then compute the expectations of K_{tt} and K_{tv} for Gaussian initialization of W. We set independently initialized weights $w_{jk} \sim \mathcal{N}(0, \frac{1}{2})$, such that $\mathbb{E}[\|\boldsymbol{w}_j\|_2^2] = p$, with j indexing neurons. Let $[\mu] = h$ denote the label of sample μ. The normalized full-dataset kernel has the following large-width limit (Appendix A.3.4),

$$\frac{1}{N}K_{\mu\nu} \xrightarrow[N\to\infty]{} \begin{cases} \kappa^g_{\text{diag}}, & \mu = \nu, \\ \kappa^g_{\text{feat}}, & [\mu] = [\nu] = h,\ \mu \neq \nu, \\ \kappa^g_{\text{coll}}, & [\mu] \neq [\nu] \text{ with input collision}, \\ \kappa^g_{\text{bg}}, & \text{otherwise}. \end{cases} \tag{7}$$

The superscript "g" denotes Gaussian. The subscripts "diag", "feat", "coll" and "bg" refer to diagonal, feature, input collision and background respectively. The input collision refers to cases $\mu = (a, b; h)$, $\nu = (c, d; g)$, $a = c$ or $b = d$, i.e., the off-diagonal entries of input correlation matrix $C = X^\top X$. For Gaussian initialization, $\kappa^g_{\text{feat}} \equiv \kappa^g_{\text{bg}}$. For ReLU, $\kappa^g_{\text{diag}} = \frac{1}{2}, \kappa^g_{\text{feat}} = \kappa^g_{\text{bg}} = \frac{1}{2\pi}, \kappa^g_{\text{coll}} = \frac{\sqrt{3}}{4\pi} + \frac{1}{6} \approx 0.304$. Quadratic results are given in Appendix A.3.3. For both activation functions, $\kappa^g_{\text{diag}} > \kappa^g_{\text{coll}} > \kappa^g_{\text{bg}} = \kappa^g_{\text{feat}}$. We show a numerical demonstration in Figure 2(a)–(c).

The training kernel K_{tt} is a submatrix of the full sample kernel K. Its expectation retains the strong diagonal, while input-collision correlations appear sparsely in the $q \times q$ off-diagonal blocks. The strong diagonal gives K_{tt} an identity-like structure, as shown in Figure 2(a). For $K_{tt} \sim I$, the criterion $\tilde{Y}_t K_{tt} \propto \tilde{Y}_t$ in Equation 6 is satisfied, helping the model fit the training data.

The cross-kernel K_{tv} is also a submatrix of K, but it contains no diagonal entries of K because the two sets are disjoint. Its feature blocks have the same expectation as the background, while input-collision correlations appear sparsely in wrong-label blocks. As shown in Figure 2(b,c), the feature blocks are therefore weaker than the off-diagonal blocks. For a test sample ν, the expected correct-label score is $\mathbb{E}[R_{h_\nu}(\nu)] = Nq\kappa^g_{\text{bg}}$. For a wrong label g, $\mathbb{E}[R_g(\nu)] = Nq\kappa^g_{\text{bg}} + Nc_g(\nu)(\kappa^g_{\text{coll}} - \kappa^g_{\text{bg}})$, where $c_g(\nu)$ counts training samples of label g sharing an input with ν. Thus the correct-label score is minimal in the expected kernel and is strictly below a wrong-label score whenever an input collision is present. This provides a mechanism for performance below the random-guessing accuracy $1/p$.

The expected random-feature kernel therefore predicts anti-generalization at early times: test samples with input collisions are misclassified once the leading-order logits become nonzero. Numerical experiments show the corresponding trend in loss, with training loss decreasing while test loss increases.

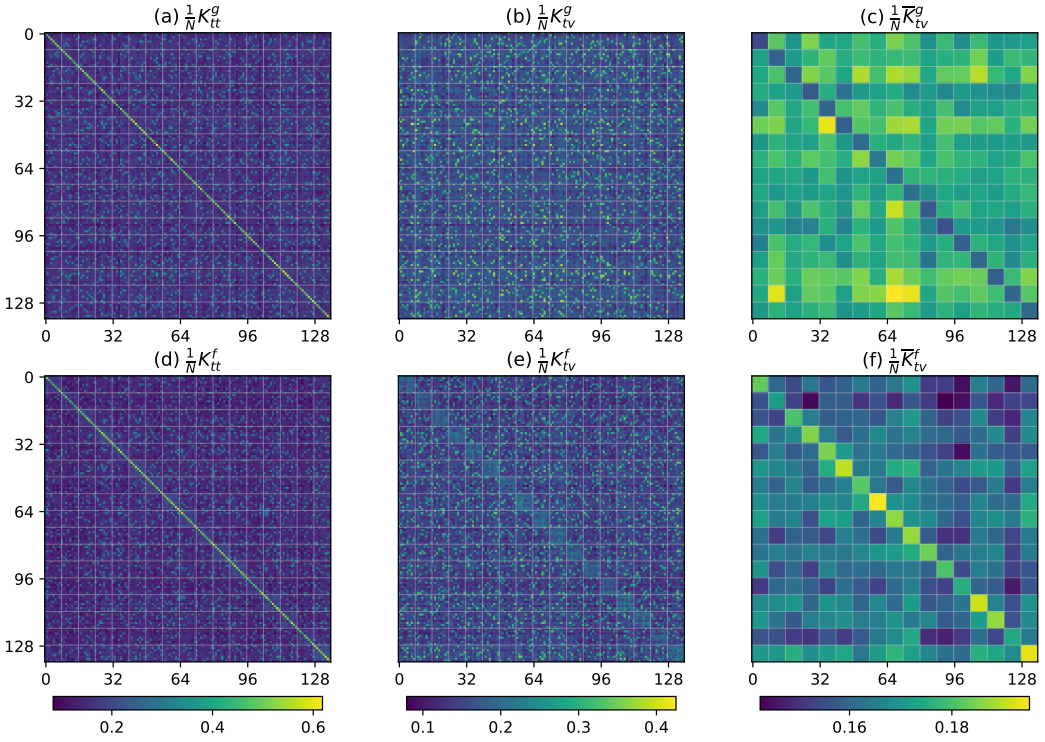

Figure 2: Training and cross-kernel structure for Gaussian random features (top row) and Fourier features (bottom row). (a) and (d) exhibit strong diagonals of K_{tt}. (b) and (e) exhibit sparsely distributed collisions in off-diagonal blocks. (c) and (f) show block-average (denoted by $\bar{K}_{tv}$). Feature blocks are on average weaker than off-diagonal blocks in Gaussian initialization (c) but stronger in Fourier bases (f). Use $p = 17, q = 8, N = 512$ for illustration.

3.2 KERNEL OF LEARNED FOURIER FEATURES

For modular addition, Nanda et al. (2023), Gromov (2023), and Tian (2025) have shown from numerical and theoretical perspectives that the first layer W eventually develops Fourier features. He et al. (2026) also observed Fourier-like features in two-layer ReLU networks trained with CE loss. Recent work (Swaroop, 2026) has reported convergence of W to square-wave bases in numerical experiments.

Here we assume that W follows a Fourier feature distribution with random phases for the two input channels, i.e., $\boldsymbol{w}_j = [\boldsymbol{w}_{j1}; \boldsymbol{w}_{j2}]$, $[\boldsymbol{w}_{j1,2}]_k = \cos(\omega_j k + \phi_{j1,2})$ where $\omega_j = \frac{2\pi n_j}{p}$, $\phi_{j1,2} \sim U(0, 2\pi)$(i.i.d.), n_j is a random integer picked from $1 \dots p-1$. Although previous studies (Tian, 2025, Theorem 2) suggested that the phases of $\boldsymbol{w}_j$ are constrained, we found no explicit constraint in numerical experiments in ReLU-CE networks (Appendix B.4). Such distribution gives $\|\boldsymbol{w}_j\|_2^2 = p$ for odd p. At large width, $K_{\mu\nu}/N$ approaches its ensemble mean. For large odd prime p, we approximate this mean by the following five classes, retaining input collisions and using a decorrelation approximation for the ReLU background (Appendix A.3.7),

$$\frac{1}{N}\mathbb{E}[K_{\mu\nu}] \simeq \begin{cases} \kappa^f_{\text{diag}}, & \mu = \nu, \\ \kappa^f_{\text{feat}}, & [\mu] = [\nu] = h,\ \mu \neq \nu, \\ \kappa^f_{\text{mod}}, & [\mu] \neq [\nu] \text{ with modular collision}, \\ \kappa^f_{\text{coll}}, & [\mu] \neq [\nu] \text{ with input collision}, \\ \kappa^f_{\text{bg}}, & \text{otherwise}. \end{cases} \tag{8}$$

The superscript "f" denotes Fourier. The modular collision refers to cases $\mu = (a, b; h)$, $\nu = (c, d; g)$, and $a - b \equiv c - d \mod p$. We visualize the feature and collision structures in Appendix A.3.2. For Fourier bases, $\kappa^f_{\text{feat}} \equiv \kappa^f_{\text{mod}}$. For ReLU, $\kappa^f_{\text{diag}} = \frac{1}{2}, \kappa^f_{\text{feat}} = \kappa^f_{\text{mod}} = \frac{2}{\pi^2} \approx 0.203, \kappa^f_{\text{coll}} = \frac{1}{6} + \frac{5}{4\pi^2} \approx 0.293, \kappa^f_{\text{bg}} = \frac{16}{\pi^4} \approx 0.164$. These coefficients specify the large-p approximation and decorrelated assumption above, explained in Appendix A.3.7. We show a numerical demonstration in Figure 2(d)–(f).

As shown by the training kernel K_{tt} in Figure 2(d), the Fourier kernel retains a strong diagonal with value κ^f_{diag}, and its feature blocks also exceed the background. Off-diagonal label blocks contain only $O(q)$ modular- and input-collision entries in each block, whereas all q^2 entries in a feature block are elevated. For each test sample ν, the correct-label block contributes q feature entries to $R_{h_\nu}(\nu)$, whereas each wrong-label block has at most one modular-collision entry and two input-collision entries for odd p. Neglecting the sparse collision corrections if $q \gg 1$, the mean sample kernel is approximately $\mathbb{E}[K] \simeq aY^\top Y + bI + c\mathbf{1}_{p^2}\mathbf{1}_{p^2}^\top$, with $a, b, c > 0$.

The cross-kernel K_{tv} contains no diagonal entries of K, but its feature blocks exceed the background. Under the same approximation, $\mathbb{E}[K_{tv}] \simeq aY_t^\top Y_v + c\mathbf{1}\mathbf{1}_{pq}^\top$, as shown in Figure 2(e,f). This $Y_t^\top Y_v$-like structure supports generalization according to Equation 6 when the correct-label scores exceed fluctuations and the collision corrections. This also explains the immediate generalization if the model is initialized using Fourier bases, see Appendix B.5.

4 MEAN FIELD DYNAMICS AND LOSS SCALING LAW

Following Equation 2, we compute the data-gradient dynamics of each w_{jk} in W, initially omitting weight decay, $\dot{W} = (G_F \circ F')X^\top$ where $\circ$ is the Hadamard product and $F' = \sigma'(WX)$. To characterize the mean-field training dynamics of individual neurons, we define an energy function for each neuron. Let $\boldsymbol{f}_j$ be the j-th row of F, the energy function $\mathcal{E}_j = \mathcal{E}(\boldsymbol{w}_j; S)$ is,

$$\mathcal{E}_j = \frac{1}{2}\|\boldsymbol{f}_j S^\top\|_2^2. \tag{9}$$

Under the output-layer locking approximation, $G_F \simeq (\lambda M^2)^{-1} F S^\top S$. Holding S fixed, the reduced data-gradient dynamics ascends $\mathcal{E}_j$ as $\partial\mathcal{E}_j/\partial w_{jk} = \lambda M^2 \dot{w}_{jk}$. These mean field dynamics are invariant under a width scaling $N \to cN$, as the rescaling $W, F, V \to c^{-1/2}(W, F, V)$ leaves

Z unchanged, while the data gradients scale identically to the weight-decay terms. Replacing S by the initial residual field $\tilde{Y}$ defines $\mathcal{E}_{0,j} = \mathcal{E}(\boldsymbol{w}_j; \tilde{Y}) = \frac{1}{2}\|\boldsymbol{f}_j \tilde{Y}^\top\|_2^2$, the label-based energy function defined by Tian (2025). Under the label-aligned residual-field ansatz, $\mathcal{E}_j \simeq \epsilon^2 \mathcal{E}_{0,j}$ during the low-loss training period.

Label Alignment. The label-based energy is exactly related to kernel-label alignment. Note that $\mathcal{E}_{0,j} = \frac{1}{2}\,\mathrm{Tr}\left[\boldsymbol{f}_j^\top \boldsymbol{f}_j \tilde{Y}_t^\top \tilde{Y}_t\right]$ on the training set. Define the total label-based energy as $\mathcal{E}_t = \sum_j \mathcal{E}_{0,j}$ and the label alignment as

$$A_{tt} = \langle K_{tt}, B_{tt} \rangle_F, \quad B_{tt} = \tilde{Y}_t^\top \tilde{Y}_t, \tag{10}$$

we have $A_{tt} = 2\mathcal{E}_t$. A_{tt} measures the unnormalized alignment of the kernel with the zero-mean label Gram matrix B_{tt}.

Radial Equilibrium. We derive the self-consistent norm-balance equation for each neuron in the reduced dynamics. At late-time norm equilibrium, $\dot{\boldsymbol{w}}_j \cdot \boldsymbol{w}_j = 0$, or equivalently $\frac{d}{dt}\|\boldsymbol{w}_j\|_2^2 = 0$. Using the gradient relation of $\mathcal{E}_j$ and the ReLU property $\boldsymbol{w}_j \cdot \partial \mathcal{E}_j / \partial \boldsymbol{w}_j = 2\mathcal{E}_j$ (see Appendix A.2) we obtain the radial equilibrium equation,

$$2\mathcal{E}_j - \lambda^2 M^2 \|\boldsymbol{w}_j\|_2^2 = 0. \tag{11}$$

By ReLU homogeneity, $\mathcal{E}(c\boldsymbol{w}_j; S) = c^2 \mathcal{E}(\boldsymbol{w}_j; S)$. Thus radial balance constrains the energy per unit squared norm, rather than the norm itself.

4.1 Loss Scaling Law

We derive a mean-field equilibrium loss equation using the learned feature kernel, as expected after generalization. For odd p, the feature ensemble in Section 3 uses $\|\boldsymbol{w}_j\|_2^2 = p$ as a reference normalization. By ReLU homogeneity, rescaling a reference neuron to norm $\|\boldsymbol{w}_j\|_2$ multiplies its kernel contribution by $\|\boldsymbol{w}_j\|_2^2/p$. We then compute the label alignment for the feature kernel as $A_{tt} = 2\sum_j \mathcal{E}_{0,j}$, without assuming equal norms or a common energy extremum. Also from the definition, $A_{tt} = \langle K_{tt}, B_{tt} \rangle_F$, $B_{tt} = \tilde{Y}_t^\top \tilde{Y}_t = I_p \otimes (\mathbf{1}_q \mathbf{1}_q^\top) - \frac{1}{p}\mathbf{1}_{pq}\mathbf{1}_{pq}^\top$. Note that B_{tt} consists of feature blocks minus a uniform $1/p$ matrix. Assuming the norm-weighted feature statistics follow the ensemble in Section 3, for $p, q \gg 1$ we neglect the diagonal and sparse off-diagonal corrections in the mean-field kernel, giving $\left\langle K_{tt}, I_p \otimes (\mathbf{1}_q \mathbf{1}_q^\top) \right\rangle_F \simeq \kappa_{\mathrm{feat}}^f q^2 \|W\|_F^2$, $\left\langle K_{tt}, \frac{1}{p}\mathbf{1}_{pq}\mathbf{1}_{pq}^\top \right\rangle_F \simeq \kappa_{\mathrm{bg}}^f q^2 \|W\|_F^2$, therefore,

$$A_{tt} = \langle K_{tt}, B_{tt} \rangle_F \simeq (\kappa_{\mathrm{feat}}^f - \kappa_{\mathrm{bg}}^f) q^2 \|W\|_F^2. \tag{12}$$

Using the residual-field ansatz $\mathcal{E}_j \simeq \epsilon^2 \mathcal{E}_{0,j}$ and summing radial equilibrium over neurons, using $A_{tt} = 2\sum_j \mathcal{E}_{0,j}$ and $M = pq$, gives

$$\epsilon^2 A_{tt} \simeq \lambda^2 p^2 q^2 \|W\|_F^2, \tag{13}$$

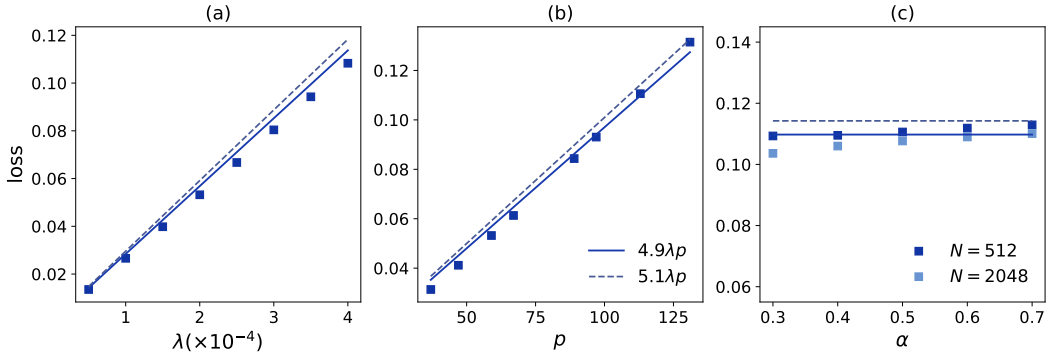

Figure 3: Late-time loss scaling with (a) weight decay λ, (b) problem size p, and (c) training fraction α. Solid and dashed lines show the predictions $4.9\lambda p$ and $5.1\lambda p$, respectively.

$$\implies \epsilon \simeq \frac{\lambda p}{\sqrt{\Delta\kappa^f}}, \quad \Delta\kappa^f = \kappa^f_{\text{feat}} - \kappa^f_{\text{bg}}. \tag{14}$$

For the loss, $\mathcal{L} = \frac{p-1}{p}\epsilon \simeq \lambda(p-1)/\sqrt{\Delta\kappa^f}$. For $p \gg 1$, $\mathcal{L} \approx \epsilon$. Substitute the computation result in Section 3, we obtain $\epsilon \approx 5.1\lambda p$. The square-wave feature kernel gives $\epsilon \approx 4.9\lambda p$ (Appendix A.3.8), thus yielding a slightly lower predicted equilibrium loss than the Fourier kernel, consistent with the numerical preference reported in Swaroop (2026).

Within these mean-field assumptions, the predicted final loss for grokked CE modular-addition models depends on weight decay λ and problem size p, with no explicit dependence on width or training fraction as shown in Figure 3(a)-(c). Finite width and random selection of training data can change the realized alignment and cause the loss to deviate from this mean-field prediction.

5 Finite-Width Dynamics of the Grokking Transition

A hallmark of grokking is the rapid rise of test accuracy long after the model fits the training set. Nanda et al. (2023) proposed a continuous feature learning progress towards generalization for modular addition. Based on the kernel methods in Section 3 and the mean-field dynamics in Section 4, we describe grokking as a continuous evolution of K_{tt} and K_{tv} from Gaussian initialization to the feature kernel. We then characterize the transition at finite width through its onset alignment, its extent in alignment space, and its duration. The numerical results are interpreted using kernel margins and effective neuron participation.

5.1 Alignment threshold for the onset of grokking

We define the transition onset time t_0 as when the test accuracy returns to $1/p$. For a target accuracy $\rho > 1/p$, let t_ρ denote the corresponding first crossing. The actual model accuracy ρ is distinguished from the kernel accuracy ρ_k determined by Equation 6. We compare these accuracies in Appendix B.3.1.

Define normalized cross-kernel alignment $a_{tv} = \langle K_{tv}, B_{tv}\rangle_F / \|W\|_F^2$ which is invariant under a overall scaling of $W \to cW$. We use the balanced data selection setting of Section 3. At fixed p, q, our experiments indicate that $a_{tv}(t_0)$ is positive, increases with width, and approaches an approximately width-independent positive value at large N. Since classification depends on the largest competing class score rather than their mean, $a_{tv} = 0$ does not imply chance accuracy; a positive onset alignment can therefore be required. A detailed discussion of the transition onset is in Appendix B.3.

5.2 Transition sharpness and effective width

Accuracy-alignment gap. For a target accuracy ρ, define $\Delta_\rho = a_{tv}(t_\rho) - a_{tv}(t_0)$. We compare the onset $\rho = 1/p$ with target accuracies $\rho = 0.1, 0.2, \ldots, 0.9$. The gap is zero at the onset by definition. Across the tested widths, Δ_ρ decreases with N, indicating a sharper transition, as shown in Figure 4(a). Fits against the nominal width N deviate from an exponent of $-1/2$. Unequal neuron amplitudes and unequal contributions to feature alignment motivate a more specific measure of participation.

Feature effective width. Neurons contribute unequally to the kernel. We therefore define the feature effective width using the training-set energy $\mathcal{E}_{0,j} = \mathcal{E}_0(\boldsymbol{w}_j)$ from Section 4,

$$N^F_{\text{eff}} = \frac{1}{\sum_j (p^f_j)^2}, \quad p^f_j = \frac{\mathcal{E}_{0,j}}{\sum_k \mathcal{E}_{0,k}} = \frac{\|\boldsymbol{f}^t_j \tilde{Y}_t^\top\|_2^2}{\sum_k \|\boldsymbol{f}^t_k \tilde{Y}_t^\top\|_2^2}, \quad \sum_j p^f_j = 1. \tag{15}$$

Since $A_{tt} = 2\sum_j \mathcal{E}_{0,j}$, N^F_{eff} measures the effective number of neurons contributing to training-label alignment. If finite-width fluctuations from different neuron contributions combine approximately independently, their relative magnitude scales as $(N^F_{\text{eff}})^{-1/2}$. If the class-score distributions change mainly by shifting during the transition, this suggests an approximate $\Delta_\rho \sim (N^F_{\text{eff}})^{-1/2}$. Fits across multiple settings and data splits are consistent with this estimate over the tested range, where fitted exponent fluctuates around -0.5, see Figure 4(c) and Appendix B.3.

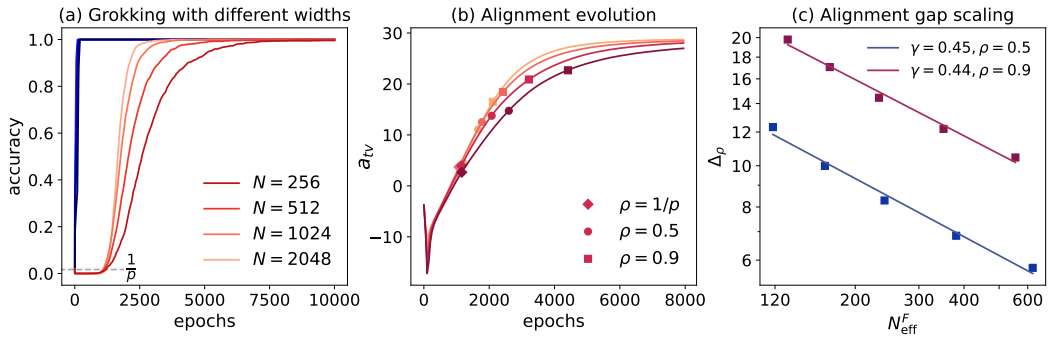

Figure 4: (a) Accuracy evolution and (b) train-test alignment of different widths. We mark the points of accuracy $1/p$, 0.5 and 0.9 on the alignment curves. (c) Alignment gaps fitted with feature effective width at accuracy 0.5 and 0.9. Use $p = 59, q = 29$ and run for seeds $0 \ldots 9$. N_{eff}^F is taken at t_ρ.

5.3 Transition Duration

Define $r_{\text{align}} = da_{tv}/dt$ and $\Delta t_\rho = t_\rho - t_0$. For monotonic $a_{tv}(t)$, $\Delta t_\rho \simeq \Delta_\rho / \bar{r}_{\text{align}}$, where $\bar{r}_{\text{align}}$ is the average alignment growth rate over the transition. At sufficiently large N, $\bar{r}_{\text{align}}$ is approximately width-independent, so $\Delta t_\rho \propto \Delta_\rho$, linking the sharper transition to the narrowing alignment interval. At smaller N, both Δ_ρ and $\bar{r}_{\text{align}}$ vary with width, so both contribute to the transition duration as shown in Figure 4(b). Still, a closed-form prediction for r_{align} remains open.

6 Discussion and Future Directions

Our work introduced a representation-kernel framework that captures the temporal evolution of feature learning throughout grokking in two-layer ReLU networks trained with cross-entropy loss and weight decay. Across varying problem sizes, network widths, and regularizations, our kernel representation directly connects evolving hidden representations to training and test logits, quantitatively explaining both early anti-generalization and delayed generalization. We further demonstrated that the label-aligned residual field sustains a persistent task-aligned gradient drive after the training loss has nearly converged, while the degree-one homogeneity of ReLU allows this drive to remain competitive with weight decay. At late-time equilibrium, the learned feature-kernel geometry accurately predicts the cross-entropy loss with a mean-field scaling law in weight decay and problem size. Our framework clarifies two complementary aspects of the grokking trajectory: (i) resolving how early spurious input-collision correlations dominate the cross-kernel and actively suppress test accuracy below chance level, and (ii) explaining how late-stage kernel reorganization drives a sharp accuracy transition associated with finite-width fluctuations. Crucially, as detailed in the Appendix, this kernel perspective extends to random data splits, multilayer networks, other modular operations, and a synthetic 3D frame-chirality classification task specifically designed to induce anti-generalization via correlated train-test splits, suggesting spurious cross-kernel correlations as a broader mechanism for anti-generalization.

There are several compelling opportunities for future work. Building upon the empirically and structurally general framework demonstrated across our arithmetic tasks, a natural next step is to extend this representation-kernel formulation to non-Abelian group operations and composite combinatorial tasks, systematically mapping how underlying symmetries dictate spurious collision patterns. In addition, while our mean-field analysis explains macroscopic equilibrium scaling, our experiments reveal an intriguing and unresolved phenomenon in the underlying dynamics: the temporal growth of the feature-neuron fraction appears to be width-independent for large t (Figure 12). Analytically modeling the initial-condition and gradient dynamics that govern this unequal neuron progression remains an important open theoretical question. Finally, our residual-field and output-locking relations are mathematically controlled primarily in the asymptotic early and late quasi-equilibrium regimes, leaving the closed-form prediction of the intermediate alignment growth rate r_{align} open. Formulating a fully unified, non-equilibrium dynamical theory across this non-stationary crossover trajectory represents an exciting and essential challenge for future representation-learning research.

AI USE STATEMENT

Generative AI tools were used for discussion and refinement of theoretical arguments, manuscript editing and consistency checking, literature discovery, and limited coding support including debugging. The main theoretical framework, core analytical results, and the majority of the derivations and code were developed by the authors. The central ideas and results presented in the main text were initially formulated by the authors, with AI tools subsequently used to refine arguments, improve derivations, and assist with presentation. AI tools provided additional assistance with several technical derivations in the Appendix. In particular, they helped refine the authors' calculation of the Fourier representation kernels and assisted in completing the derivation of the square-wave representation kernel. These derivations were subsequently reviewed and reproduced by the authors and independently checked against numerical calculations. AI tools generated the code for the frame-chirality task presented in the Appendix and provided an initial scaffold for the training code. All other code used in this work was written by the authors, with AI assistance limited to debugging. The AI-generated frame-chirality code was inspected, tested, and executed by the authors. All final mathematical claims, numerical results, code used to produce the reported results, and citations were reviewed by the authors, who take full responsibility for the content of the paper.

ETHICS STATEMENT

This work is theoretical and uses only synthetic tasks and computational experiments. It does not involve human subjects, personal data, or sensitive datasets, and we do not identify any direct ethical risks arising from the methods or results presented.

REPRODUCIBILITY STATEMENT

The assumptions and derivations underlying the theoretical results are stated in the main text and detailed in the Appendix. Experimental settings, including model architectures, optimization parameters, dataset construction, and random seeds, are provided alongside the corresponding experiments. Code for reproducing the numerical results is released as supplementary material.

REFERENCES

Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, volume 22, pp. 342–350, 2009. URL https://proceedings.neurips.cc/paper/2009/hash/5751ec3e9a4feab575962e78e006250d-Abstract.html.

Chi-Ning Chou, Oscar Uzdelewicz, Neng-Chun Chiu, Yao-Yuan Yang, and SueYeon Chung. Two speeds of learning: A representation-readout decomposition of grokking and double descent. *arXiv preprint arXiv:2605.27078*, 2026. URL https://arxiv.org/abs/2605.27078.

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

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. URL https://arxiv.org/abs/2602.16849.

Tanishq Kumar, Blake Bordelon, Samuel Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/63ed15a46a143ff57484b38cd6b85d91-Abstract-Conference.html.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1711.00165.

Ziming Liu, Ouail Kitouni, Niklas S. Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. In *Advances in Neural Information Processing Systems*, volume 35,

2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/dfc310e81992d2e4cedc09ac47eff13e-Abstract-Conference.html.

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/11799.

Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon Du, Jason Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/909c8fef63e1cede406ce9e6794f99a2-Abstract-Conference.html.

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. PMLR, 2024. URL https://proceedings.mlr.press/v235/mohamadi24a.html.

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. URL https://iclr.cc/virtual/2023/oral/12572.

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. URL https://arxiv.org/abs/2201.02177.

Lenz Pracher, Pascal de Jong, Oskar Lieshaus, Alan Jeffares, and Steffen Rulands. A spectral theory of grokking: Weight decay induces feature learning. *arXiv preprint arXiv:2609.26679*, 2026. URL https://arxiv.org/abs/2609.26679.

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. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/682f87a8c306098ec8be29019bd76aa4-Abstract-Conference.html.

Anand Swaroop. Latent algorithmic structure precedes grokking: A mechanistic study of ReLU MLPs on modular arithmetic. *arXiv preprint arXiv:2603.23784*, 2026. URL https://arxiv.org/abs/2603.23784v1.

Hu Tan, Kuo Gai, and Shihua Zhang. Deciphering two training clocks in grokking via deep linear network theory with conditional ReLU reduction. *arXiv preprint arXiv:2606.05863*, 2026. URL https://arxiv.org/abs/2606.05863.

Yuandong Tian. Provable scaling laws of feature emergence from learning dynamics of grokking. *arXiv preprint arXiv:2509.21519*, 2025. URL https://arxiv.org/abs/2509.21519v5.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023. URL https://arxiv.org/abs/2309.02390.

A Additional Derivations

A.1 Uniform Residual for the Complete Dataset

For prime p and the complete dataset $\{X, Y\}$, we show that a deterministic, symmetry-preserving infinite-width limit satisfies $\bar{S}(t) = \epsilon(t)\tilde{Y}$ at every finite time t. We assume that the initialization law is invariant under the permutations below and that the chosen infinite-width scaling admits such a limit. The symmetry argument establishes the form of the limiting residual, rather than convergence to the limit itself.

For triplets $(a, b; c)$, let transformation $g = (u, r, s)$, $u \in \mathbb{Z}_p^*$ and $r, s \in \mathbb{Z}_p$ such that $(a, b; c) \mapsto (ua + r, ub + s; uc + r + s)$.

The triplet set has two orbits. For $(a, b; h)$, $h \equiv a + b \mod p$, $ua + ub + r + s \equiv u(a + b) + r + s \equiv uh + r + s \mod p$. For two incorrect triplets $(a, b; c)$ and $(a', b'; c')$, let $d = c - a - b \neq 0$ and $d' = c' - a' - b' \neq 0$ in $\mathbb{Z}_p$. Set $u = d'/d$, $r = a' - ua$, and $s = b' - ub$. Then $g(a, b; c) = (a', b'; c')$. Correct triplets are also connected by translations with $u = 1$. Therefore the two orbits are classified by $c = a + b \mod p$ and $c \neq a + b \mod p$.

Let A_g be the block-diagonal permutation on the $2p$ input coordinates that maps $\boldsymbol{x}^{(a,b)}$ to $\boldsymbol{x}^{(ua+r,ub+s)}$, and let R_g permute output classes by $c \mapsto uc + r + s$. For neuron weight $(\boldsymbol{w}_j, \boldsymbol{v}_j)$, let transformation $T_g : (\boldsymbol{w}_j, \boldsymbol{v}_j) \mapsto (A_g\boldsymbol{w}_j, R_g\boldsymbol{v}_j)$. A_g satisfies $(A_g\boldsymbol{w}_j) \cdot (A_g\boldsymbol{x}) = \boldsymbol{w}_j \cdot \boldsymbol{x}$, so $\boldsymbol{z}_{T_g\theta}(A_g\boldsymbol{x}) = R_g\boldsymbol{z}_\theta(\boldsymbol{x})$. Therefore, the loss is invariant under the transformation $\mathcal{L}(T_g\theta) = \mathcal{L}(\theta)$. The gradient and gradient descent dynamics are equivariant under these orthogonal parameter permutations; isotropic weight decay preserves this property.

The invariant initialization law and equivariant dynamics preserve symmetry in distribution. Under the assumed deterministic infinite-width limit, the limiting output and residual inherit this symmetry. So for residual field, $\lim_{N\to\infty} S_N(t) = \bar{S}(t)$ for all finite t. The two orbits imply one common correct-class residual $s_{\mathrm{c}}(t)$ and one common incorrect-class residual $s_{\mathrm{w}}(t)$ across all samples. Since each residual column sums to zero, $s_{\mathrm{c}} + (p - 1)s_{\mathrm{w}} = 0$. Setting $\epsilon(t) = p\, s_{\mathrm{c}}(t)/(p - 1) = -p\, s_{\mathrm{w}}(t)$ gives $\bar{S}(t) = \epsilon(t)\tilde{Y}$.

A.2 Mean-field dynamics

We expand the reduced dynamics used in Section 4. All sample sums in this subsection run over the training set of size M; thus $F = F_t$, $S = S_t$, and $\tilde{Y} = \tilde{Y}_t$ when the train/test distinction is needed. Derivatives of the energy below hold S fixed.

A.2.1 Data-gradient dynamics and neuron energy

Starting from the backpropagated gradient in Equation 2, the chain rule gives, initially omitting weight decay,

$$\dot{w}_{jk} = -\frac{\partial \mathcal{L}}{\partial w_{jk}} = -\sum_\mu \frac{\partial \mathcal{L}}{\partial f_j^\mu}\frac{\partial f_j^\mu}{\partial w_{jk}} = \sum_\mu (g_F)_j^\mu \sigma'(\boldsymbol{w}_j \cdot \boldsymbol{x}^\mu) x_k^\mu. \tag{16}$$

In matrix form, $\dot{W} = (G_F \circ F')X^\top$, where $\circ$ is the Hadamard product and $F' = \sigma'(WX)$. Under the output-layer locking approximation,

$$\dot{V} = \frac{1}{M}SF^\top - \lambda V \simeq 0, \qquad V \simeq \frac{1}{\lambda M}SF^\top, \qquad G_F = \frac{1}{M}V^\top S \simeq \frac{1}{\lambda M^2}FS^\top S. \tag{17}$$

We use this expression for G_F in the reduced dynamics. For the j-th row $\boldsymbol{f}_j$ of F, define

$$\mathcal{E}_j = \mathcal{E}(\boldsymbol{w}_j; S) = \frac{1}{2}\|\boldsymbol{f}_j S^\top\|_2^2 = \frac{1}{2}\sum_i \left[\sum_\mu s_i^\mu \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x}^\mu)\right]^2. \tag{18}$$

Differentiating the squared sum at fixed S gives

$$\begin{aligned}\frac{\partial \mathcal{E}_j}{\partial w_{jk}} &= \sum_i \left[\sum_\nu s_i^\nu \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x}^\nu)\right]\left[\sum_\mu s_i^\mu \sigma'(\boldsymbol{w}_j \cdot \boldsymbol{x}^\mu) x_k^\mu\right] \\ &= \sum_\mu (FS^\top S)_j^\mu \sigma'(\boldsymbol{w}_j \cdot \boldsymbol{x}^\mu) x_k^\mu.\end{aligned} \tag{19}$$

Comparing with Equation 16, the reduced data-gradient dynamics is $\dot{\boldsymbol{w}}_j = (\lambda M^2)^{-1} \nabla_{\boldsymbol{w}_j} \mathcal{E}_j$. Thus, for fixed S and without weight decay, $d\mathcal{E}_j/dt = (\lambda M^2)^{-1} \|\nabla_{\boldsymbol{w}_j} \mathcal{E}_j\|_2^2 \geq 0$.

A.2.2 Weight decay and the residual-field ansatz

Restoring weight decay gives

$$\dot{\boldsymbol{w}}_j = \frac{1}{\lambda M^2} \nabla_{\boldsymbol{w}_j} \mathcal{E}_j - \lambda \boldsymbol{w}_j = -\nabla_{\boldsymbol{w}_j} \Phi_j, \qquad \Phi_j = \frac{\lambda}{2} \|\boldsymbol{w}_j\|_2^2 - \frac{1}{\lambda M^2} \mathcal{E}(\boldsymbol{w}_j; S). \tag{20}$$

For fixed S, $d\Phi_j/dt = -\|\nabla_{\boldsymbol{w}_j} \Phi_j\|_2^2$. In the network, however, $S = Y - P$ evolves with the parameters. Self-consistency enters through this residual field (Rubin et al., 2024), and the total derivative contains an additional term,

$$\frac{d\Phi_j}{dt} = -\|\nabla_{\boldsymbol{w}_j} \Phi_j\|_2^2 - \frac{1}{\lambda M^2} \left\langle \frac{\partial \mathcal{E}_j}{\partial S}, \dot{S} \right\rangle_F. \tag{21}$$

Consequently, the fixed-field potential need not decrease along the full evolving dynamics.

Replacing S by the initial residual $\tilde{Y}$ defines the label-based energy used in Tian (2025),

$$\mathcal{E}_{0,j} = \mathcal{E}(\boldsymbol{w}_j; \tilde{Y}) = \frac{1}{2} \|\boldsymbol{f}_j \tilde{Y}^\top\|_2^2. \tag{22}$$

The energy is quadratic in S, so the ansatz $S \simeq \epsilon \tilde{Y}$ implies $\mathcal{E}_j \simeq \epsilon^2 \mathcal{E}_{0,j}$. The factor ϵ^2 therefore comes directly from the residual-field amplitude.

A.2.3 Label alignment

Writing the squared norm as a trace yields

$$\mathcal{E}_{0,j} = \frac{1}{2} \operatorname{Tr}\left[\boldsymbol{f}_j^\top \boldsymbol{f}_j \tilde{Y}_t^\top \tilde{Y}_t\right] = \frac{1}{2} \left\langle \boldsymbol{f}_j^\top \boldsymbol{f}_j, B_{tt} \right\rangle_F, \qquad B_{tt} = \tilde{Y}_t^\top \tilde{Y}_t. \tag{23}$$

Since $K_{tt} = F_t^\top F_t = \sum_j \boldsymbol{f}_j^\top \boldsymbol{f}_j$, summing over neurons gives the exact identity

$$A_{tt} = \langle K_{tt}, B_{tt} \rangle_F = 2 \sum_j \mathcal{E}_{0,j} = 2\mathcal{E}_t. \tag{24}$$

For ReLU and $c > 0$, scaling $W \to cW$ gives $F_t \to cF_t$ and $K_{tt} \to c^2 K_{tt}$. Hence $a_{tt} = A_{tt}/\|W\|_F^2$ is invariant under this overall rescaling and measures label alignment per unit squared weight norm.

A.2.4 Radial equilibrium

The radial direction is the direction parallel to $\boldsymbol{w}_j$, along which its norm changes while its orientation stays fixed. Taking the inner product of Equation 20 with $\boldsymbol{w}_j$ gives

$$\frac{1}{2} \frac{d}{dt} \|\boldsymbol{w}_j\|_2^2 = \frac{1}{\lambda M^2} \boldsymbol{w}_j \cdot \nabla_{\boldsymbol{w}_j} \mathcal{E}_j - \lambda \|\boldsymbol{w}_j\|_2^2. \tag{25}$$

For ReLU, $\sigma'(z)z = \sigma(z)$, and therefore

$$\begin{aligned}\boldsymbol{w}_j \cdot \nabla_{\boldsymbol{w}_j} \mathcal{E}_j &= \sum_{\mu,\nu,i} s_i^\mu s_i^\nu \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x}^\nu) \sigma'(\boldsymbol{w}_j \cdot \boldsymbol{x}^\mu)(\boldsymbol{w}_j \cdot \boldsymbol{x}^\mu) \\ &= \sum_{\mu,\nu,i} s_i^\mu s_i^\nu \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x}^\nu) \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x}^\mu) = 2\mathcal{E}_j.\end{aligned} \tag{26}$$

Equivalently, this is Euler's identity for $\mathcal{E}(c\boldsymbol{w}_j; S) = c^2\mathcal{E}(\boldsymbol{w}_j; S)$. At norm equilibrium, Equation 25 becomes

$$2\mathcal{E}_j - \lambda^2 M^2 \|\boldsymbol{w}_j\|_2^2 = 0. \tag{27}$$

For the complete dataset, $M = p^2$, giving $2\mathcal{E}_j - \lambda^2 p^4 \|\boldsymbol{w}_j\|_2^2 = 0$. For a nonzero neuron, radial balance fixes $\mathcal{E}_j/\|\boldsymbol{w}_j\|_2^2 = \lambda^2 M^2/2$. At fixed S, both terms scale quadratically with the neuron norm, so this relation alone determines neither the norm nor a stationary feature direction. Directional stationarity additionally requires the component of $\nabla_{\boldsymbol{w}_j}\mathcal{E}_j$ tangent to the sphere of fixed norm to vanish.

Finally, inserting $\mathcal{E}_j \simeq \epsilon^2\mathcal{E}_{0,j}$ and summing Equation 27 over neurons yields

$$\epsilon^2 A_{tt} \simeq \lambda^2 M^2 \|W\|_F^2, \qquad \epsilon^2 a_{tt} \simeq \lambda^2 M^2. \tag{28}$$

For balanced sampling, $M = pq$. The feature-kernel estimate $A_{tt} \simeq \Delta\kappa^f q^2 \|W\|_F^2$ then gives $\epsilon \simeq \lambda p/\sqrt{\Delta\kappa^f}$, recovering the loss-scaling argument in Section 4.

A.3 Computation of the Sample Kernel

A.3.1 Notation and normalization

For $\mu = (a, b; h)$ and $\nu = (c, d; g)$, let $h = a + b \bmod p$ and $g = c + d \bmod p$. The two input channels occupy different coordinates: $\boldsymbol{x}^\mu = [\boldsymbol{e}_a; \boldsymbol{e}_b]$, $\boldsymbol{w}_j = [\boldsymbol{w}_{j1}; \boldsymbol{w}_{j2}]$, and $z_j^\mu = [\boldsymbol{w}_{j1}]_a + [\boldsymbol{w}_{j2}]_b$. Thus $f_j^\mu = \sigma(z_j^\mu)$ and

$$K = F^\top F, \qquad K_{\mu\nu} = \sum_{j=1}^{N} f_j^\mu f_j^\nu, \qquad \overline{k}_{\mu\nu} = \mathbb{E}[f_j^\mu f_j^\nu]. \tag{29}$$

For independent, identically distributed neurons with the required finite moments, $\mathbb{E}[K_{\mu\nu}] = N\overline{k}_{\mu\nu}$ and $K_{\mu\nu}/N \to \overline{k}_{\mu\nu}$ almost surely as $N \to \infty$. The training and cross-kernels $K_{tt} = F_t^\top F_t$ and $K_{tv} = F_t^\top F_v$ are restrictions of this full-dataset kernel. Under the balanced sampling of Section 3, $M_t = M_v = pq$.

A.3.2 KERNEL STRUCTURE VISUALIZATION

We visualize the feature blocks, input collisions, and modular collisions using $B = Y^\top Y$, $C = X^\top X$, and $C_{\text{mod}} = D^\top D$, respectively, where

$$D_{r\mu} = \mathbf{1}[r = d_\mu], \qquad d_\mu \equiv a_\mu - b_\mu \pmod{p}. \tag{30}$$

Here D is the one-hot matrix of modular differences, and B denotes the uncentered label Gram matrix for this visualization. Input and modular collisions correspond to the nonzero off-diagonal entries of C and C_{mod}. The top row of Figure 5 shows these three structures. The bottom row compares them with the Fourier kernel K^f and Gaussian kernel K^g, and also shows K^f with its feature blocks removed to expose the collision structure.

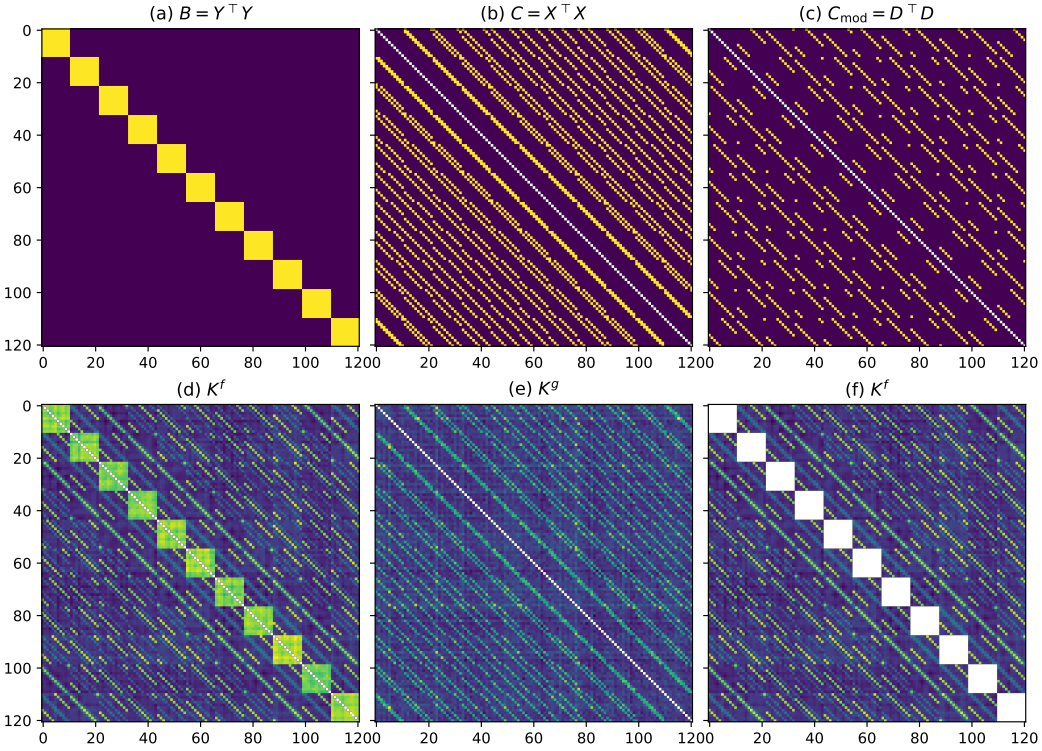

Figure 5: Kernel structure visualization for $p = 11$ and $N = 2048$. $\sigma(x) = x^2$. Top row: (a) feature blocks $B = Y^\top Y$, (b) input collisions $C = X^\top X$, and (c) modular collisions $C_{\text{mod}} = D^\top D$. Diagonals are removed in (b) and (c). Bottom row: (d) Fourier kernel K^f, (e) Gaussian kernel K^g, and (f) K^f with feature blocks removed. Diagonals are removed in (d) and (e). The enhanced kernel entries reflect the feature and collision structures in the top row, as described in Section 3.

A.3.3 GAUSSIAN INITIALIZATION: QUADRATIC ACTIVATION

Assume independent $w_{jk} \sim \mathcal{N}(0, \sigma_w^2)$. Section 3 uses $\sigma_w^2 = 1/2$, so $\mathbb{E}\|\boldsymbol{w}_j\|_2^2 = p$. Set $X = z_j^\mu$, $Y = z_j^\nu$, and $s^2 = 2\sigma_w^2$. Their correlation is

$$\operatorname{Var} X = \operatorname{Var} Y = s^2, \qquad \operatorname{Cov}(X, Y) = \sigma_w^2(\delta_{ac} + \delta_{bd}), \qquad \rho = \frac{\delta_{ac} + \delta_{bd}}{2}. \tag{31}$$

Thus $\rho = 1$ on the diagonal, $\rho = 1/2$ for one input collision, and $\rho = 0$ otherwise. For $\sigma(x) = x^2$, the Gaussian fourth-moment identity gives

$$\overline{k}_{\mu\nu} = \mathbb{E}[X^2 Y^2] = \mathbb{E}[X^2]\mathbb{E}[Y^2] + 2\mathbb{E}[XY]^2 = s^4(1 + 2\rho^2). \tag{32}$$

Consequently,

$$\kappa^g_{\text{diag}} = 12\sigma_w^4, \qquad \kappa^g_{\text{feat}} = \kappa^g_{\text{bg}} = 4\sigma_w^4, \qquad \kappa^g_{\text{coll}} = 6\sigma_w^4. \tag{33}$$

With $\sigma_w^2 = 1/2$, these become the values in Section 3:

$$\boxed{\kappa^g_{\text{diag}} = 3, \qquad \kappa^g_{\text{feat}} = \kappa^g_{\text{bg}} = 1, \qquad \kappa^g_{\text{coll}} = \frac{3}{2}.} \tag{34}$$

Scope of the block-score estimate. For a test sample ν and a wrong label g, let $m_g(\nu)$ be the number of input-collision samples in that training-label block. Then

$$\mathbb{E}[R_{h_\nu}(\nu)] = Nq\kappa^g_{\text{bg}}, \qquad \mathbb{E}[R_g(\nu)] = N\left[q\kappa^g_{\text{bg}} + m_g(\nu)(\kappa^g_{\text{coll}} - \kappa^g_{\text{bg}})\right]. \tag{35}$$

For a training sample, the correct block instead has expectation $N[\kappa^g_{\text{diag}} + (q-1)\kappa^g_{\text{bg}}]$.

A.3.4 Gaussian initialization: ReLU activation

Write $x_+ = \max(x, 0)$. For $X \sim \mathcal{N}(0, s^2)$,

$$\mathbb{E}[X_+] = \frac{s}{\sqrt{2\pi}}, \qquad \mathbb{E}[X_+^2] = \frac{s^2}{2}. \tag{36}$$

For a correlated pair, write $\rho = \cos\theta$, with $\theta \in [0, \pi]$, and represent $X = sR\cos\varphi$, $Y = sR\cos(\varphi - \theta)$ using an isotropic standard two-dimensional Gaussian. Here φ is uniform and independent of R, with $\mathbb{E}R^2 = 2$. The overlap of the two positive half-circles gives

$$\mathbb{E}[X_+Y_+] = \frac{s^2}{\pi}\int_{\theta-\pi/2}^{\pi/2} \cos\varphi\cos(\varphi-\theta)\,d\varphi = \frac{s^2}{2\pi}\left[\sin\theta + (\pi-\theta)\cos\theta\right]. \tag{37}$$

Equivalently,

$$\overline{k}_{\mu\nu} = \frac{s^2}{2\pi}\left[\sqrt{1-\rho^2} + (\pi - \arccos\rho)\rho\right]. \tag{38}$$

Evaluating at $\rho = 1, 0, 1/2$ yields

$$\kappa^g_{\text{diag}} = \sigma_w^2, \qquad \kappa^g_{\text{feat}} = \kappa^g_{\text{bg}} = \frac{\sigma_w^2}{\pi}, \qquad \kappa^g_{\text{coll}} = \sigma_w^2\left(\frac{\sqrt{3}}{2\pi} + \frac{1}{3}\right). \tag{39}$$

In the normalization of Section 3,

$$\boxed{\kappa^g_{\text{diag}} = \frac{1}{2}, \qquad \kappa^g_{\text{feat}} = \kappa^g_{\text{bg}} = \frac{1}{2\pi}, \qquad \kappa^g_{\text{coll}} = \frac{\sqrt{3}}{4\pi} + \frac{1}{6}.} \tag{40}$$

A.3.5 Gaussian initialization: Hermite expansion

Use the probabilists' Hermite polynomials H_n, defined by $e^{tz-t^2/2} = \sum_{n=0}^{\infty} H_n(z)t^n/n!$. For jointly standard Gaussian U, V with correlation ρ, the generating functions give

$$\mathbb{E}[e^{tU-t^2/2}e^{uV-u^2/2}] = e^{\rho tu}, \qquad \mathbb{E}[H_m(U)H_n(V)] = \delta_{mn}n!\rho^n. \tag{41}$$

For an activation with $\mathbb{E}[\sigma(sU)^2] < \infty$, expand the scaled function, not the unscaled activation:

$$g(u) = \sigma(su) = \sum_{n=0}^{\infty} c_n H_n(u), \qquad c_n = \frac{1}{n!}\mathbb{E}[\sigma(sU)H_n(U)]. \tag{42}$$

Then

$$\overline{k}_{\mu\nu} = \mathbb{E}[g(U)g(V)] = \sum_{n=0}^{\infty} n! c_n^2 \rho^n. \tag{43}$$

The Gaussian kernel values are therefore

$$\kappa^g_{\mathrm{diag}} = \sum_{n=0}^{\infty} n! c_n^2, \qquad \kappa^g_{\mathrm{feat}} = \kappa^g_{\mathrm{bg}} = c_0^2, \qquad \kappa^g_{\mathrm{coll}} = \sum_{n=0}^{\infty} n! c_n^2\, 2^{-n}. \tag{44}$$

In particular,

$$\kappa^g_{\mathrm{diag}} - \kappa^g_{\mathrm{bg}} = \sum_{n=1}^{\infty} n! c_n^2, \qquad \kappa^g_{\mathrm{coll}} - \kappa^g_{\mathrm{bg}} = \sum_{n=1}^{\infty} n! c_n^2\, 2^{-n}. \tag{45}$$

For a nonconstant activation in Gaussian L^2, both gaps are strictly positive, and $\kappa^g_{\mathrm{diag}} > \kappa^g_{\mathrm{coll}} > \kappa^g_{\mathrm{bg}}$. The equality of feature and background values follows from independence, independently of the particular activation.

For $\sigma(x) = x^2$, $g(u) = s^2[H_2(u) + H_0(u)]$, so the series gives $s^4(1 + 2\rho^2)$. For ReLU, summing the series recovers the expression in Section A.3.4.

The coefficients depend on $s = \sqrt{2}\sigma_w$; coefficients of $\sigma(u)$ cannot be reused at arbitrary weight variance without rescaling. The factors $n!$ correspond to the probabilists' convention and would change for an orthonormal Hermite basis.

A.3.6 Fourier features: quadratic activation

Assume odd p and the ensemble in Section 3:

$$[\boldsymbol{w}_{j1}]_k = \cos(\omega_j k + \phi_{j1}), \qquad [\boldsymbol{w}_{j2}]_k = \cos(\omega_j k + \phi_{j2}), \qquad \omega_j = \frac{2\pi n_j}{p}, \tag{46}$$

where n_j is uniform on $\{1, \dots, p-1\}$ and the two phases are independent and uniform on $[0, 2\pi)$. The two channel norms sum to p. Define $\Delta_a = a - c$, $\Delta_b = b - d$, and, conditional on a frequency ω,

$$u = \omega c + \phi_1, \qquad v = \omega d + \phi_2, \qquad A = \omega\Delta_a, \qquad B = \omega\Delta_b. \tag{47}$$

Then $z^\nu = \cos u + \cos v$ and $z^\mu = \cos(u + A) + \cos(v + B)$. The elementary phase averages are

$$\mathbb{E}[\cos^2 u] = \frac{1}{2}, \qquad \mathbb{E}[\cos^4 u] = \frac{3}{8}, \qquad \mathbb{E}[\cos(u + A)\cos u] = \frac{1}{2}\cos A, \tag{48}$$

$$\mathbb{E}[\cos^2(u + A)\cos^2 u] = \frac{1}{4} + \frac{1}{8}\cos(2A). \tag{49}$$

Expanding the product of squared preactivations and averaging the independent phases gives the exact conditional kernel

$$\mathbb{E}_{\phi_1,\phi_2}[(z^\mu)^2(z^\nu)^2 \mid \omega] = 1 + \frac{1}{8}[\cos(2A) + \cos(2B)] + \cos A \cos B. \tag{50}$$

Define the exact nonzero-frequency average

$$D_p(r) = \frac{1}{p-1}\sum_{n=1}^{p-1}\cos\left(\frac{2\pi nr}{p}\right) = \begin{cases} 1, & r \equiv 0 \pmod p, \\ -\frac{1}{p-1}, & r \not\equiv 0 \pmod p. \end{cases} \tag{51}$$

Thus

$$\overline{k}^f_{\mu\nu} = 1 + \frac{1}{8}[D_p(2\Delta_a) + D_p(2\Delta_b)] + \frac{1}{2}[D_p(\Delta_a + \Delta_b) + D_p(\Delta_a - \Delta_b)]. \tag{52}$$

For odd p, the diagonal, same-label condition, and modular-collision condition correspond to $(\Delta_a, \Delta_b) = (0, 0)$, $\Delta_a + \Delta_b = 0$, and $\Delta_a - \Delta_b = 0$, respectively. Distinct same-label and modular-collision samples have no input collision. The exact values are

$$\overline{k}^f_{\mu\nu} = \begin{cases} \frac{9}{4}, & \mu = \nu, \\ \frac{3}{2} - \frac{3}{4(p-1)}, & \mu \neq \nu,\ h = g \text{ or } \Delta_a = \Delta_b, \\ \frac{9}{8} - \frac{9}{8(p-1)}, & \text{exactly one of } \Delta_a, \Delta_b \text{ is zero}, \\ 1 - \frac{5}{4(p-1)}, & \text{otherwise}. \end{cases} \tag{53}$$

Taking large p gives the Fourier coefficients used in Section 3, including the input-collision class:

$$\boxed{\kappa^f_{\text{diag}} = \frac{9}{4}, \qquad \kappa^f_{\text{feat}} = \kappa^f_{\text{mod}} \simeq \frac{3}{2}, \qquad \kappa^f_{\text{coll}} \simeq \frac{9}{8}, \qquad \kappa^f_{\text{bg}} \simeq 1.} \tag{54}$$

The finite-p expressions retain the effect of excluding the zero frequency. The input-collision class has limiting value $9/8$ and is retained in the five-class kernel of Section 3. Only the subsequent block-average approximation omits sparse collision entries. For quadratic activation, the displayed finite-p formula is exact even for odd composite p.

Figure 6 compares the Gaussian and Fourier kernels for quadratic activation, illustrating the emergence of label-block structure in the Fourier ensemble.

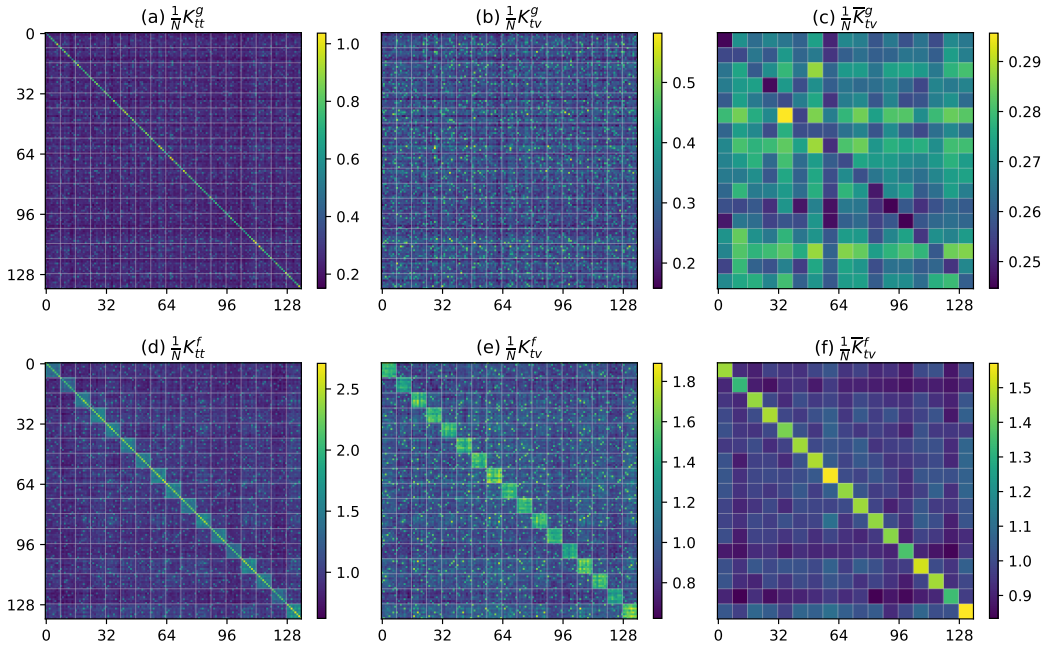

Figure 6: Sample kernels for quadratic activation with Gaussian weights (top row) and sinusoidal Fourier features (bottom row). Columns show K_{tt}, K_{tv}, and $\overline{K}_{tv}$.

A.3.7 Fourier features: ReLU activation

Exact phase average. Using the same u, v, A, B, define

$$C(t) = \frac{1}{2\pi}\int_0^{2\pi} |\cos x|\,|\cos(x+t)|\,dx. \tag{55}$$

Let $\delta(t) = \min_{m\in\mathbb{Z}} |t - m\pi| \in [0, \pi/2]$. Direct integration gives

$$C(t) = \frac{\sin\delta(t) + [\pi/2 - \delta(t)]\cos\delta(t)}{\pi}, \qquad C(0) = \frac{1}{2}, \qquad \frac{1}{\pi}\int_0^{\pi} C(t)\,dt = \frac{4}{\pi^2}. \tag{56}$$

Write $S = (u+v)/2$ and $D = (u-v)/2$. The sum-to-product identity gives $z^\nu = 2\cos S\cos D$ and $z^\mu = 2\cos[S + (A+B)/2]\cos[D + (A-B)/2]$. Since $x_+ = (x + |x|)/2$ and the joint law is invariant under $(z^\mu, z^\nu) \mapsto (-z^\mu, -z^\nu)$, the mixed terms vanish:

$$\mathbb{E}_\phi[z^\mu_+ z^\nu_+ \mid \omega] = \frac{1}{4}\mathbb{E}_\phi[z^\mu z^\nu \mid \omega] + \frac{1}{4}\mathbb{E}_\phi[|z^\mu|\,|z^\nu| \mid \omega]. \tag{57}$$

The factors involving absolute cosines are π-periodic, so the phase integral factorizes in S, D on their periodic domain. Therefore

$$\mathbb{E}_\phi[z^\mu_+ z^\nu_+ \mid \omega] = \frac{1}{8}(\cos A + \cos B) + C\left(\frac{A+B}{2}\right) C\left(\frac{A-B}{2}\right). \tag{58}$$

The exact ensemble kernel is consequently

$$\overline{k}^f_{\mu\nu} = \frac{1}{8}[D_p(\Delta_a) + D_p(\Delta_b)] + \frac{1}{p-1}\sum_{n=1}^{p-1} C\left(\frac{\pi n(\Delta_a + \Delta_b)}{p}\right) C\left(\frac{\pi n(\Delta_a - \Delta_b)}{p}\right). \tag{59}$$

This finite sum completes the phase-average calculation and remains valid at finite odd p, including input collisions and composite moduli. It also avoids assuming independence of two shifts generated by the same frequency.

Diagonal and feature/modular-collision values. On the diagonal, symmetry and $\mathbb{E}[(\cos u + \cos v)^2] = 1$ give

$$\kappa^f_{\mathrm{diag}} = \mathbb{E}[(z^\mu_+)^2] = \frac{1}{2}. \tag{60}$$

For distinct samples with $\Delta_b = -\Delta_a$ or $\Delta_b = \Delta_a$, put $r = \Delta_a \not\equiv 0$. Both cases have the same exact phase/frequency average:

$$\overline{k}^f_{\mathrm{feat/mod}}(r) = -\frac{1}{4(p-1)} + \frac{1}{2(p-1)} \sum_{n=1}^{p-1} C\left(\frac{2\pi n r}{p}\right). \tag{61}$$

When p is prime, every nonzero r permutes the frequency residues. In the large-p limit, the sum approaches the mean of C, giving

$$\kappa^f_{\mathrm{feat}} = \kappa^f_{\mathrm{mod}} \simeq \frac{1}{2}\frac{4}{\pi^2} = \frac{2}{\pi^2}. \tag{62}$$

For composite p, this continuum approximation requires the order $p/\gcd(r,p)$ to be large. Oddness alone does not ensure it.

Background approximation and its limitation. If the two shifts $(A+B)/2$ and $(A-B)/2$ may be averaged as independent uniform angles modulo π, the product term becomes

$$\mathbb{E}\left[C\left(\frac{A+B}{2}\right) C\left(\frac{A-B}{2}\right)\right] \simeq \left(\frac{4}{\pi^2}\right)^2 = \frac{16}{\pi^4}. \tag{63}$$

The linear cosine term has mean $-1/[4(p-1)]$ for noncolliding samples and vanishes at large p. This gives the background approximation used in Section 3:

$$\boxed{\kappa^f_{\mathrm{diag}} = \frac{1}{2}, \qquad \kappa^f_{\mathrm{feat}} = \kappa^f_{\mathrm{mod}} \simeq \frac{2}{\pi^2}, \qquad \kappa^f_{\mathrm{coll}} \simeq \frac{1}{6} + \frac{5}{4\pi^2}, \qquad \kappa^f_{\mathrm{bg}} \simeq \frac{16}{\pi^4}.} \tag{64}$$

Background decorrelation. A uniform frequency is a single random variable. The two shifts need not be independent for a fixed sample pair, even as $p \to \infty$. Thus $16/\pi^4$ is a decorrelation/background approximation used in Section 3, not the exact large-width limit of every background entry. For example, $(\Delta_a, \Delta_b) = (1, 3)$ has neither an input collision nor a label/modular collision for sufficiently large odd primes, but its limiting product average is

$$\frac{1}{2\pi}\int_0^{2\pi} C(2\omega)C(\omega)\, d\omega \approx 0.1644183265, \qquad \frac{16}{\pi^4} \approx 0.1642557161. \tag{65}$$

The exact finite sum above should be used when individual entries or finite-modulus effects matter. Approximate averaging over many offsets may still justify the simpler background value, but this is a further assumption.

Input-collision class. For exactly one input collision, take $\Delta_a = 0$ and $\Delta_b = r \neq 0$. The exact formula reduces to

$$\overline{k}^f_{\mathrm{coll}}(r) = \frac{1}{8}\left(1 - \frac{1}{p-1}\right) + \frac{1}{p-1}\sum_{n=1}^{p-1} C\left(\frac{\pi n r}{p}\right)^2. \tag{66}$$

For prime $p \to \infty$ (or sufficiently large residue order),

$$\frac{1}{\pi}\int_0^{\pi} C(t)^2\,dt = \frac{2}{\pi^3}\int_0^{\pi/2}\Big[\sin t + \Big(\frac{\pi}{2}-t\Big)\cos t\Big]^2\,dt = \frac{1}{24}+\frac{5}{4\pi^2}. \tag{67}$$

Hence

$$\boxed{\kappa^f_{\mathrm{coll}} \simeq \frac{1}{6}+\frac{5}{4\pi^2} \approx 0.2933181462.} \tag{68}$$

This is the input-collision coefficient used in Section 3. It is substantially larger than the generic background and even the feature value, so neglecting these entries in the subsequent kernel reduction is a sparse-block approximation, not an entrywise approximation.

A.3.8 Square-wave features: ReLU activation

This section uses the same sample indices, frequency distribution, and normalization as above. The superscript "sq" distinguishes square-wave coefficients from the sinusoidal coefficients κ^f. The subscripts diag, feat, mod, coll, and bg retain their meanings in Section 3.

Feature normalization and activation. Define $Q(\theta) = \mathrm{sgn}(\cos\theta)$; its values at zero crossings do not affect phase averages. We use Q instead of q to avoid confusion with the number of samples per label. With the same independent uniform phases and nonzero frequencies as in Section A.3.6, set

$$[\boldsymbol{w}_{j1}]_a = \frac{1}{\sqrt{2}}Q(\omega_j a+\phi_{j1}), \qquad [\boldsymbol{w}_{j2}]_b = \frac{1}{\sqrt{2}}Q(\omega_j b+\phi_{j2}), \qquad \|\boldsymbol{w}_j\|_2^2 = p. \tag{69}$$

The normalization matches the sinusoidal ensemble. The preactivation takes values in $\{-\sqrt{2},0,\sqrt{2}\}$, so ReLU selects the event that both square waves are positive:

$$z_j^{\mu} = \frac{Q(\omega_j a+\phi_{j1})+Q(\omega_j b+\phi_{j2})}{\sqrt{2}}, \qquad f_j^{\mu} = \sqrt{2}\,\mathbf{1}\{Q(\omega_j a+\phi_{j1}) = Q(\omega_j b+\phi_{j2}) = 1\}. \tag{70}$$

Positive-half-cycle overlap and exact kernel. For a uniform angle θ, the overlap probability of two positive half-cycles is

$$G(t) = \Pr[Q(\theta)=Q(\theta+t)=1] = \frac{\pi - r(t)}{2\pi}, \qquad r(t) = \min_{m\in\mathbb{Z}}|t-2\pi m| \in [0,\pi]. \tag{71}$$

The overlap length is $\pi - r(t)$. Thus G is even and 2π-periodic, with $G(0) = 1/2$. Retain $\Delta_a = a-c$, $\Delta_b = b-d$, $A = \omega\Delta_a$, and $B = \omega\Delta_b$ from Section A.3.6. Independence of the two input phases gives

$$\mathbb{E}_{\phi_1,\phi_2}[f_j^{\mu} f_j^{\nu} \mid \omega] = 2G(A)G(B). \tag{72}$$

The exact phase/frequency-averaged kernel is therefore

$$\overline{k}^{\mathrm{sq}}_{\mu\nu} = \frac{2}{p-1}\sum_{n=1}^{p-1} G\left(\frac{2\pi n\Delta_a}{p}\right) G\left(\frac{2\pi n\Delta_b}{p}\right), \qquad \mathbb{E}[K_{\mu\nu}] = N\overline{k}^{\mathrm{sq}}_{\mu\nu}. \tag{73}$$

This formula applies to individual sample pairs at finite p, without decorrelating the frequency-induced shifts.

Diagonal, feature, and modular-collision classes. The diagonal value is exact:

$$\kappa^{\mathrm{sq}}_{\mathrm{diag}} = 2G(0)^2 = \frac{1}{2}. \tag{74}$$

For distinct same-label samples, $\Delta_b = -\Delta_a$; for a modular collision, $\Delta_b = \Delta_a$. Since G is even, both give an average of $2G(A)^2$. For a uniform angle U,

$$\mathbb{E}[G(U)] = \frac{1}{\pi}\int_0^\pi \left(\frac{1}{2} - \frac{r}{2\pi}\right) dr = \frac{1}{4}, \qquad \mathbb{E}[G(U)^2] = \frac{1}{\pi}\int_0^\pi \left(\frac{1}{2} - \frac{r}{2\pi}\right)^2 dr = \frac{1}{12}. \tag{75}$$

Consequently,

$$\kappa^{\mathrm{sq}}_{\mathrm{feat}} = \kappa^{\mathrm{sq}}_{\mathrm{mod}} \simeq 2\mathbb{E}[G(U)^2] = \frac{1}{6}. \tag{76}$$

The continuum-frequency approximation requires large prime p, or sufficiently large residue order for composite moduli.

Input-collision and background classes. For exactly one input collision, say $\Delta_a = 0$ and $\Delta_b \neq 0$, the conditional kernel is $2G(0)G(B) = G(B)$. Hence

$$\kappa^{\mathrm{sq}}_{\mathrm{coll}} \simeq \mathbb{E}[G(U)] = \frac{1}{4}. \tag{77}$$

For the background, treating the two shifts as independent uniform angles U, V gives

$$\kappa^{\mathrm{sq}}_{\mathrm{bg}} \simeq 2\mathbb{E}[G(U)]\mathbb{E}[G(V)] = \frac{1}{8}. \tag{78}$$

The five-class approximation is therefore

$$\boxed{\kappa^{\mathrm{sq}}_{\mathrm{diag}} = \frac{1}{2}, \qquad \kappa^{\mathrm{sq}}_{\mathrm{feat}} = \kappa^{\mathrm{sq}}_{\mathrm{mod}} \simeq \frac{1}{6}, \qquad \kappa^{\mathrm{sq}}_{\mathrm{coll}} \simeq \frac{1}{4}, \qquad \kappa^{\mathrm{sq}}_{\mathrm{bg}} \simeq \frac{1}{8}.} \tag{79}$$

The ordering is $\kappa^{\mathrm{sq}}_{\mathrm{diag}} > \kappa^{\mathrm{sq}}_{\mathrm{coll}} > \kappa^{\mathrm{sq}}_{\mathrm{feat}} = \kappa^{\mathrm{sq}}_{\mathrm{mod}} > \kappa^{\mathrm{sq}}_{\mathrm{bg}}$. Input collisions are retained as their own class; omitting them in a subsequent block-average calculation is a sparsity approximation.

Background averaging. For a fixed sample pair, A, B arise from the same random frequency and need not be independent. Thus $1/8$ is a decorrelated or offset-averaged background, not an exact value for every background entry. The exact finite sum retains these correlations.

Figure 7 illustrates the corresponding label-block structure for square-wave features with ReLU activation.

Finite-p coefficients. For odd prime p, multiplication by a nonzero offset permutes the nonzero residues. Let $g_n = G(2\pi n/p)$ and $m = (p-1)/2$. Direct summation gives

$$\begin{aligned} S_1 &= \sum_{n=1}^{p-1} g_n = 2\sum_{n=1}^{m}\left(\frac{1}{2} - \frac{n}{p}\right) = \frac{(p-1)^2}{4p}, \\ S_2 &= \sum_{n=1}^{p-1} g_n^2 = 2\sum_{n=1}^{m}\left(\frac{1}{2} - \frac{n}{p}\right)^2 = \frac{(p-1)(p-2)}{12p}. \end{aligned} \tag{80}$$

Therefore,

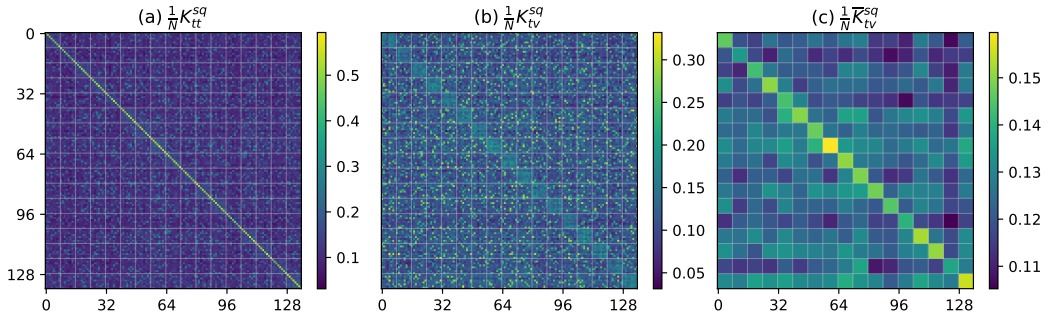

Figure 7: Sample kernels for square-wave features with ReLU activation: (a) K_{tt}, (b) K_{tv}, and (c) $\overline{K}_{tv}$. The superscript s in the panels denotes the square-wave ensemble, written sq in the text.

$$\kappa_{\text{diag}}^{\text{sq}}(p) = \frac{1}{2}, \qquad \kappa_{\text{feat}}^{\text{sq}}(p) = \kappa_{\text{mod}}^{\text{sq}}(p) = \frac{2S_2}{p-1} = \frac{p-2}{6p}, \qquad \kappa_{\text{coll}}^{\text{sq}}(p) = \frac{S_1}{p-1} = \frac{p-1}{4p}. \tag{81}$$

To define a finite-p background coefficient, average uniformly over $\mathcal{B}_p = \{(r,s) : r,s \neq 0,\ s \neq r,\ s \neq -r\}$ in $\mathbb{Z}_p^2$, excluding input, feature, and modular collisions. For prime $p > 3$, $|\mathcal{B}_p| = (p-1)(p-3)$, and multiplication by a nonzero frequency permutes this set. Since $g_{p-n} = g_n$,

$$\left\langle \overline{k}_{\mu\nu}^{\text{sq}} \right\rangle_{\mathcal{B}_p} = \frac{2(S_1^2 - 2S_2)}{(p-1)(p-3)} = \frac{3p^2 - 8p + 1}{24p^2}. \tag{82}$$

This is an average over background offsets, not the value for every background pair. For $p = 3$, this averaging set is empty; for composite moduli, use the exact frequency sum instead of these prime-modulus formulas.

At $p = 59$, the diagonal, feature/modular-collision, input-collision, and offset-averaged background values are approximately 0.5, 0.16102, 0.24576, and 0.11936, respectively.

Comparison with sinusoidal features and loss scaling. Under the same large-p, background-decorrelation, and sparse-collision approximations, the feature-background gaps are

$$\Delta\kappa^{\text{sq}} = \kappa_{\text{feat}}^{\text{sq}} - \kappa_{\text{bg}}^{\text{sq}} = \frac{1}{24}, \qquad \Delta\kappa^{f} = \frac{2}{\pi^2} - \frac{16}{\pi^4} \approx 0.03839, \qquad \frac{\Delta\kappa^{\text{sq}}}{\Delta\kappa^{f}} \approx 1.085. \tag{83}$$

Inserting this gap into the mean-field relation of Section 4 gives

$$\epsilon \simeq \frac{\lambda p}{\sqrt{\Delta\kappa^{\text{sq}}}} = 2\sqrt{6}\,\lambda p \approx 4.899\,\lambda p, \qquad \mathcal{L} \simeq 2\sqrt{6}\,\lambda(p-1). \tag{84}$$

The square-wave ensemble thus predicts a slightly smaller equilibrium loss coefficient than the sinusoidal ensemble. This comparison requires the mean-field assumptions of Section 4; similar block contrasts alone do not imply identical samplewise predictions or guarantee generalization when collision terms and fluctuations are retained.

Numerical verification of the Fourier and square-wave coefficients at finite p and N is provided in Section A.3.9 below, Tables 1 and 2. In particular, the measured square-wave feature–background gap remains close to $1/24$, supporting the loss coefficient above.

A.3.9 Numerical Verification of Feature Kernels

We numerically estimate the ensemble-averaged entries of the normalized kernel K/N for the Fourier and square-wave feature ensembles with ReLU activation. We use $(p, N) =$

$(59, 2048), (113, 4096), (151, 8192)$ and seed 42. Tables 1 and 2 compare the results with the large-p coefficients derived above.

Table 1: Fourier feature-kernel coefficients with ReLU activation. The theory column gives the large-p predictions, using the background-decorrelation approximation. Numerical values use seed 42.

Coefficient	Theory	$p = 59$ $N = 2048$	$p = 113$ $N = 4096$	$p = 151$ $N = 8192$
κ^f_{diag}	0.500	0.500	0.500	0.500
κ^f_{feat}	0.203	0.198	0.200	0.201
κ^f_{coll}	0.293	0.290	0.291	0.292
κ^f_{mod}	0.203	0.198	0.200	0.201
κ^f_{bg}	0.164	0.158	0.161	0.162

Table 2: Square-wave feature-kernel coefficients with ReLU activation. The theory column gives the large-p predictions, with an offset-averaged background. Numerical values use seed 42.

Coefficient	Theory	$p = 59$ $N = 2048$	$p = 113$ $N = 4096$	$p = 151$ $N = 8192$
$\kappa^{\mathrm{sq}}_{\mathrm{diag}}$	0.500	0.500	0.500	0.500
$\kappa^{\mathrm{sq}}_{\mathrm{feat}}$	0.167	0.161	0.164	0.164
$\kappa^{\mathrm{sq}}_{\mathrm{coll}}$	0.250	0.246	0.248	0.248
$\kappa^{\mathrm{sq}}_{\mathrm{mod}}$	0.167	0.161	0.164	0.164
$\kappa^{\mathrm{sq}}_{\mathrm{bg}}$	0.125	0.119	0.122	0.123

For both ensembles, the tabulated coefficients approach the large-p predictions as p and N increase. The deviations are consistent with the finite-p effects and background-averaging approximations discussed in Sections A.3.7 and A.3.8. The quantity controlling the square-wave loss prediction is the gap $\Delta\kappa^{\mathrm{sq}} = \kappa^{\mathrm{sq}}_{\mathrm{feat}} - \kappa^{\mathrm{sq}}_{\mathrm{bg}}$. From the tabulated values, this gap is 0.042, 0.042, and 0.041 at $p = 59, 113, 151$, respectively, close to $1/24 \approx 0.0417$. These checks support the feature-kernel coefficients used in Sections 3 and 4, and indicate that the resulting $4.9\lambda p$ prediction remains a good approximation at the finite problem sizes considered here.

A.4 Anti-generalization in Multilayer Networks

Anti-generalization also occurs in multilayer networks trained on modular addition. The Gaussian kernel calculation in Section A.3 shows that diagonal and input-collision entries are elevated above the background, with their locations determined by the input Gram matrix $C = X^\top X$. Following the kernel construction of Cho & Saul (2009) and the infinite-width recursion of Lee et al. (2018), we show how this ordering propagates through random hidden layers.

A.4.1 Gaussian random layers and input geometry

For this subsection, normalize the inputs to unit norm: $X = [\boldsymbol{x}^1, \dots, \boldsymbol{x}^M]$, $\|\boldsymbol{x}^\mu\| = 1$. For the two-hot modular-addition inputs, this amounts to dividing each input by $\sqrt{2}$. A Gaussian random layer has

$$F_{j\mu} = \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}^\mu), \qquad \boldsymbol{w}_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I). \tag{85}$$

To retain the main-text convention $K = F^\top F$, write the normalized infinite-width kernel as

$$\overline{K}_{\mu\nu} = \lim_{N\to\infty} \frac{1}{N} K_{\mu\nu} = \mathbb{E}_{\boldsymbol{w}}\left[\sigma(\boldsymbol{w}^\top \boldsymbol{x}^\mu)\sigma(\boldsymbol{w}^\top \boldsymbol{x}^\nu)\right]. \tag{86}$$

The preactivations $z_\mu = \boldsymbol{w}^\top \boldsymbol{x}^\mu$ and $z_\nu = \boldsymbol{w}^\top \boldsymbol{x}^\nu$ are jointly standard Gaussian, with correlation $\rho_{\mu\nu} = (\boldsymbol{x}^\mu)^\top \boldsymbol{x}^\nu = C_{\mu\nu}$. Hence an activation-dependent scalar function κ determines the kernel:

$$\overline{K}_{\mu\nu} = \kappa(\rho_{\mu\nu}), \qquad \overline{K} = \kappa^\circ(C), \tag{87}$$

where κ° acts entrywise. Thus the random layer transforms the input Gram geometry entrywise.

A.4.2 Hermite expansion

For $\mathbb{E}_{z\sim\mathcal{N}(0,1)}[\sigma(z)^2] < \infty$, expand the activation in probabilists' Hermite polynomials:

$$\sigma(z) = \sum_{n=0}^{\infty} a_n H_n(z), \qquad \mathbb{E}[H_n(z)H_m(z)] = n!\,\delta_{nm}. \tag{88}$$

For jointly standard Gaussian z, z' with correlation ρ, $\mathbb{E}[H_n(z)H_m(z')] = n!\rho^n\delta_{nm}$, giving

$$\kappa(\rho) = \sum_{n=0}^{\infty} n! a_n^2 \rho^n, \qquad \overline{K} = c_0 \mathbf{1}_M \mathbf{1}_M^\top + \sum_{n=1}^{\infty} c_n C^{\circ n}, \qquad c_n = n! a_n^2 \geq 0. \tag{89}$$

For nonnegative correlations, κ is nondecreasing, so entrywise ordering is preserved. For arbitrary negative correlations this need not hold for a general activation. If $\sigma(z) \simeq a + bz$ over the relevant preactivation range, then $\overline{K} \simeq a^2 \mathbf{1}_M \mathbf{1}_M^\top + b^2 C$. Centering with $H = I_M - \mathbf{1}_M \mathbf{1}_M^\top / M$ removes the constant term and gives $H\overline{K}H \simeq b^2 HCH$.

A.4.3 ReLU and multilayer recursion

For ReLU and unit-variance preactivations, the kernel and its derivative are

$$\kappa_{\mathrm{ReLU}}(\rho) = \frac{\sqrt{1-\rho^2} + (\pi - \arccos\rho)\rho}{2\pi}, \qquad \kappa'_{\mathrm{ReLU}}(\rho) = \frac{\pi - \arccos\rho}{2\pi} > 0, \quad -1 < \rho < 1. \tag{90}$$

Thus $\rho_1 > \rho_2$ implies $\kappa_{\mathrm{ReLU}}(\rho_1) > \kappa_{\mathrm{ReLU}}(\rho_2)$: ReLU preserves the ordering even for negative input correlations.

For a deep, bias-free fully connected network, take independent Gaussian weights of variance $1/N_\ell$ between successive hidden layers. Let $\overline{K}^{(0)} = C$, and let $\overline{K}^{(\ell)}$ be the infinite-width normalized activation Gram matrix at layer ℓ. The covariance recursion is

$$\overline{K}_{\mu\nu}^{(\ell+1)} = \mathbb{E}[\sigma(u)\sigma(v)], \qquad \begin{pmatrix} u \\ v \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \overline{K}_{\mu\mu}^{(\ell)} & \overline{K}_{\mu\nu}^{(\ell)} \\ \overline{K}_{\mu\nu}^{(\ell)} & \overline{K}_{\nu\nu}^{(\ell)} \end{pmatrix}\right). \tag{91}$$

Equal input norms give a common diagonal variance $q_\ell = \overline{K}^{(\ell)}_{\mu\mu}$. For ReLU, homogeneity makes the variance and correlation updates explicit:

$$\overline{K}^{(\ell+1)}_{\mu\nu} = q_\ell \kappa_{\mathrm{ReLU}}(c^{(\ell)}_{\mu\nu}), \qquad q_{\ell+1} = \frac{q_\ell}{2}, \qquad c^{(\ell+1)}_{\mu\nu} = 2\kappa_{\mathrm{ReLU}}(c^{(\ell)}_{\mu\nu}), \tag{92}$$

where $c^{(\ell)}_{\mu\nu} = \overline{K}^{(\ell)}_{\mu\nu}/q_\ell$. Using weight variance $2/N_\ell$ instead keeps q_ℓ constant and gives the same correlation update.

Since this correlation map is strictly increasing, the input-Gram ordering persists through any fixed finite number of infinite-width random ReLU layers, although the contrast can decrease with depth. In particular, the initial modular-addition correlations 1, $1/2$, and 0 remain three ordered classes corresponding to diagonal, input-collision, and background entries. Their locations are therefore preserved in both K_{tt} and K_{tv}. In the early t-linear regime, this structure remains the leading-order description while hidden features stay close to initialization; the random-layer recursion alone does not describe subsequent feature learning.

Figure 8 illustrates the kernel structure at finite width. Figure 9 shows the corresponding early anti-generalization and later grokking behavior in three- and four-layer networks.

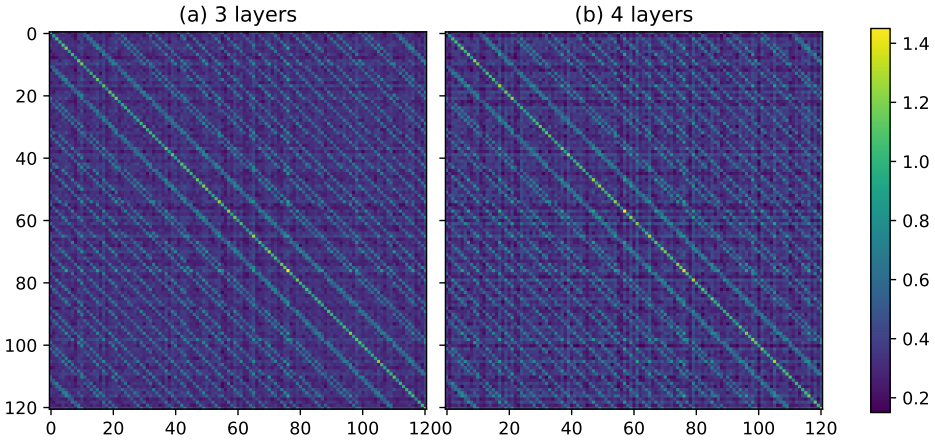

Figure 8: Normalized sample kernels $K = \frac{1}{N} F^\top F$ for three-layer (left) and four-layer (right) fully connected networks with $p = 11$ and $N = 512$. Here F denotes the final hidden-layer activations.

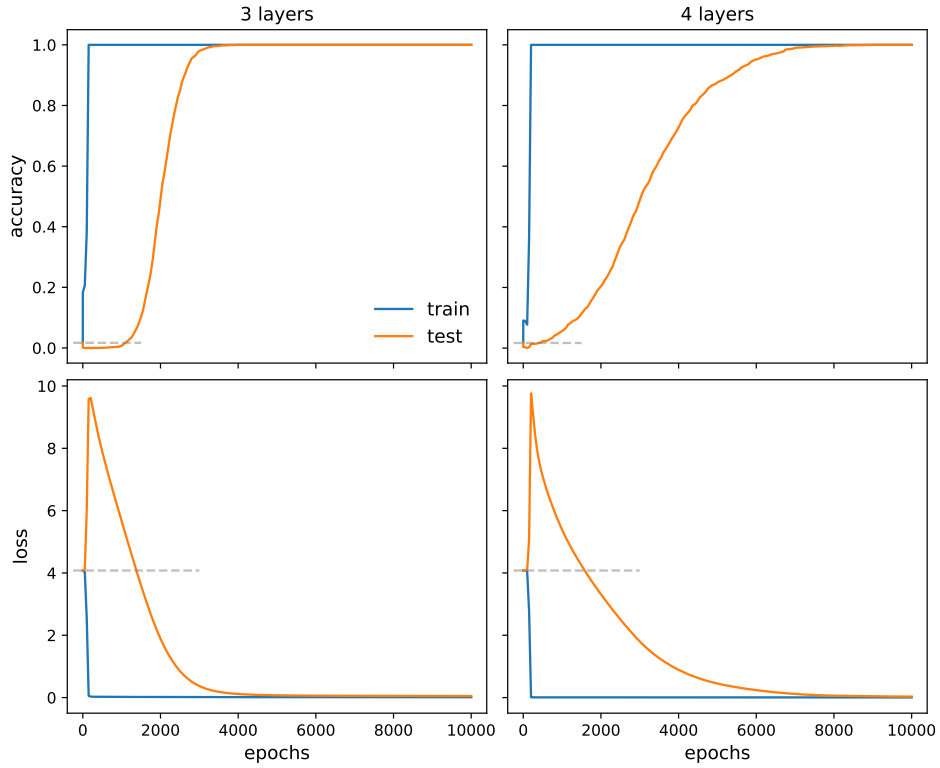

Figure 9: Training and test accuracy (top) and loss (bottom) for three-layer (left) and four-layer (right) networks. Gray dashed lines indicate chance accuracy $1/p$ and uniform-prediction loss $\log p$, respectively.

B Additional Numerical Results

B.1 Balanced and Random Data Selection

In the main text, we use a balanced data split with the same number of samples per label in each set. A more common setting is a random data split, in which samples are selected randomly from the complete dataset and label counts are generally unequal. We compare the accuracy and loss of two selection rules in Figure 10, and observe similar behavior.

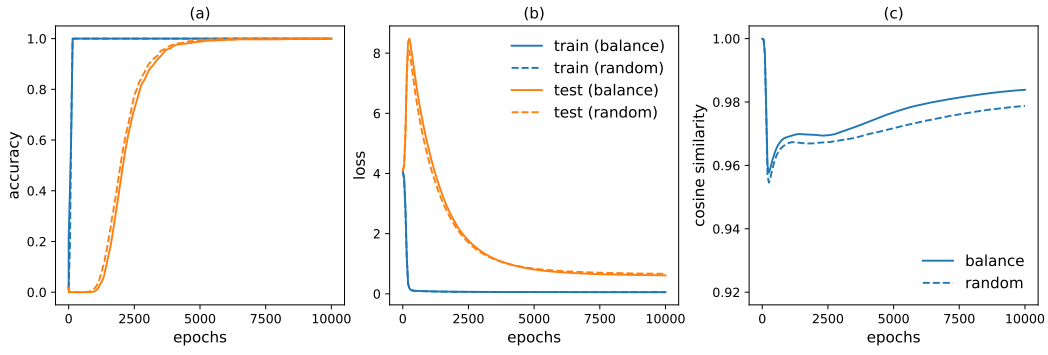

Figure 10: (a) Accuracy, (b) loss, and (c) cosine similarity between S and $\tilde{Y}$ for balanced and random data selection. We use $p = 59$ and $N = 512$, with $q = 29$ for balanced selection and $\alpha = q/p \approx 0.49$ for random selection.

In these experiments, random selection preserves the overall grokking dynamics but affects residual-field fluctuations. Let a group consist of samples sharing the same label. Figure 11(a) shows the equilibrium loss of each sample as a function of its group size. Sample loss is correlated with group size, with larger loss in smaller groups. For random selection, the total variance of loss l is divided into two parts, the inter-group part and intra-group part. Let p_h be the fraction of analyzed samples in group h, l_h its mean loss, and $\bar{l} = \sum_h p_h l_h$ the overall mean loss. The total variance is

$$\mathrm{Var}(l) = \sum_h p_h \,\mathrm{Var}(l|h) + \sum_h p_h (l_h - \bar{l})^2 \tag{93}$$

After removing the inter-group variance, the variance (or standard deviation) of loss is approximately the same as that under the balanced selection, as shown in Figure 11(b).

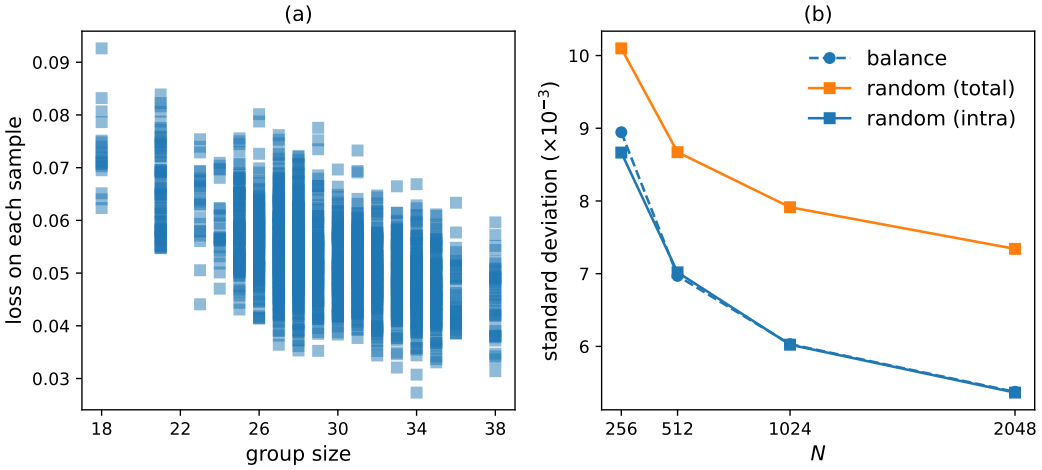

Figure 11: (a) Equilibrium sample loss as a function of label-group size. (b) Standard deviation of sample losses at epoch 30000 for widths $N = 256, 512, 1024, 2048$, using balanced and random selection. Both total and intra-group standard deviations are shown.

B.2 Early- and Late-Stage Learning

Early-stage learning is less common in wider networks. Figure 12 shows the growth of the fraction of feature neurons at different widths.

Here we consider the feature changing rate of each neuron, defined using the angular changing rate,

$$v_j = \frac{\angle[\boldsymbol{w}_j(t), \boldsymbol{w}_j(t+\Delta t)]}{\Delta t}. \tag{94}$$

This is a direct proxy for directional feature learning. We plot the changing rates for each neuron in the early-stage. The early-stage plot shows a clear t-linear increase during the first 45 epochs, as shown in Figure 13(a). For late-stage, we only consider and plot neurons with $\tilde{\mathcal{E}}_{0j} \leq 0.3$ at $t_{\mathrm{mid}} = 2000$ epochs, where $\tilde{\mathcal{E}}_{0j} = \mathcal{E}_0(\boldsymbol{w}_j)/\mathcal{E}_0^*$ is the normalized energy in $[0, 1]$. That is to say we only consider those neurons which are learning features during the late-stage. We test for $p = 59, N = 512, \lambda = 2 \times 10^{-4}$, learning rate $\eta = 2$ and 4 batches per epoch. Therefore $t = 8n_{\mathrm{epoch}}$.

For early-stage, we fit the v_j w.r.t time and extract the slope, $\bar{v}$, in the first 45 epochs. The result is $\bar{v} = 2.40 \times 10^{-6}$. In the t-linear regime, the mean early-stage is therefore $\bar{v}_{\mathrm{early}} \approx \bar{v}t$. For late-stage, we compute the average of maximum changing rates of qualified neurons. The late-stage angular changing rate is averaged over a 50-epoch interval ($\Delta t = 400$ in rescaled time). The results are,

$$\bar{v}_{\mathrm{early}} \approx 2.40 \times 10^{-6} \times t, \quad \bar{v}_{\mathrm{late}} = 3.54 \times 10^{-5}. \tag{95}$$

Then $\dot{\kappa}_{\mathrm{real}}$ in the main text is $\dot{\kappa}_{\mathrm{real}} = (\bar{v}_{\mathrm{early}}/\bar{v}_{\mathrm{late}})/t = 0.0680$. Take $\epsilon \approx \mathcal{L}(t_{\mathrm{final}}) = 0.054$, the theoretical ratio is $\dot{\kappa}_{\mathrm{th}} = \lambda/\epsilon^2 = 0.0686$, which is close to the numerical value. The fitting is shown in Figure 13

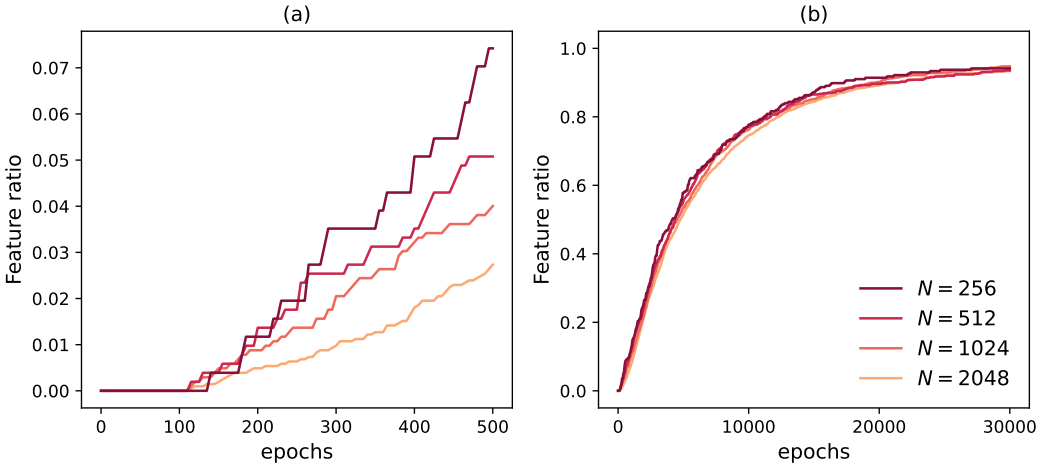

Figure 12: Growth of the feature-neuron fraction (a) at early times and (b) throughout training. The fraction is the number of neurons reaching a normalized energy of 0.5 divided by the network width N.

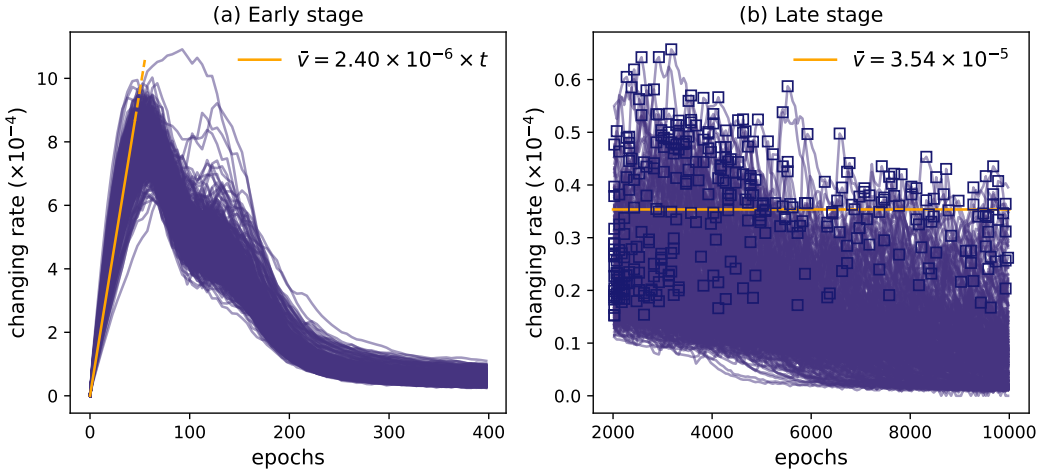

Figure 13: Neuron weight-direction change rates during (a) early-stage and (b) late-stage learning. Panel (a) shows the linear fit over the first 45 epochs, and panel (b) shows the average of the per-neuron maximum change rates. Each hollow square marks the maximum for one neuron.

We also consider the learning rate. Here the feature-learning rate is defined using $\tilde{\mathcal{E}}_{0j}$ which represents the feature alignment of a single neuron. The learning rate of a neuron is defined as

$$r_j = \frac{e_2 - e_1}{t(\tilde{\mathcal{E}}_{0j} = e_2) - t(\tilde{\mathcal{E}}_{0j} = e_1)} \tag{96}$$

Take $e_1 = 0.4$ and $e_2 = 0.6$, the learning rate is approximately the maximum ascending rate of $\tilde{\mathcal{E}}_{0j}$. To compare the early- and late- stage learning rate, we define early-stage learning rate as the average learning rate of all neurons reaching $\tilde{\mathcal{E}}_{0j} \geq 0.9$ at reference training time $t_{\text{ref}} = 400$ epochs. The late-stage learning rate is defined as the average learning rate of all neurons with $\tilde{\mathcal{E}}_{0j} \leq 0.3$ at t_{mid} and $\tilde{\mathcal{E}}_{0j} \geq 0.6$ at t_{final}, $t_{\text{final}} = 10000$ epochs. We use the same experimental settings. The late-stage learning is computed using a standard seed 42. For early-stage learning, we obtain results from seeds $0 \ldots 9$ together with 42 because the qualified neurons are few. The average early- and late-stage learning rates extracted from the numerical experiments are,

$$\bar{r}_{\text{early}} = 1.06 \times 10^{-3}, \quad \bar{r}_{\text{late}} = 3.87 \times 10^{-5} \tag{97}$$

Then the learning rate ratio in the main text is computed as $\kappa_{\text{real}} = \bar{r}_{\text{early}}/\bar{r}_{\text{late}} = 27.3$. The average time of early-stage transition (when $\tilde{\mathcal{E}}_{0j}$ reaches 0.5) is 148.8 epochs, i.e. $\bar{t}_{\text{early}} = 1190.4$. Take $\epsilon \approx \mathcal{L}(t_{\text{final}}) = 0.054$, the theoretical ratio is $\kappa_{\text{th}} = \lambda t/\epsilon^2 = 81.6$. The qualified neuron learning curves are shown in Figure 14.

Several effects can contribute to this quantitative difference. First, the residual field S is not equal to $\tilde{Y}$ at $\bar{t}_{\text{early}}$. The training loss is 1.27 at $\bar{t}_{\text{early}}$. Second, $\bar{t}_{\text{early}}$ lies outside the t-linear regime. As shown above, the linear increase lasts up to about 50 epochs. Third, the fluctuations of S at $\bar{t}_{\text{early}}$ affect the learning rate. Also, the estimation $\epsilon \approx \mathcal{L}(t_{\text{final}})$ is not accurate, as the training loss still remains ~ 0.07 at t_{mid}. These effects can reduce the early-stage feature-learning speed relative to the leading-order estimate.

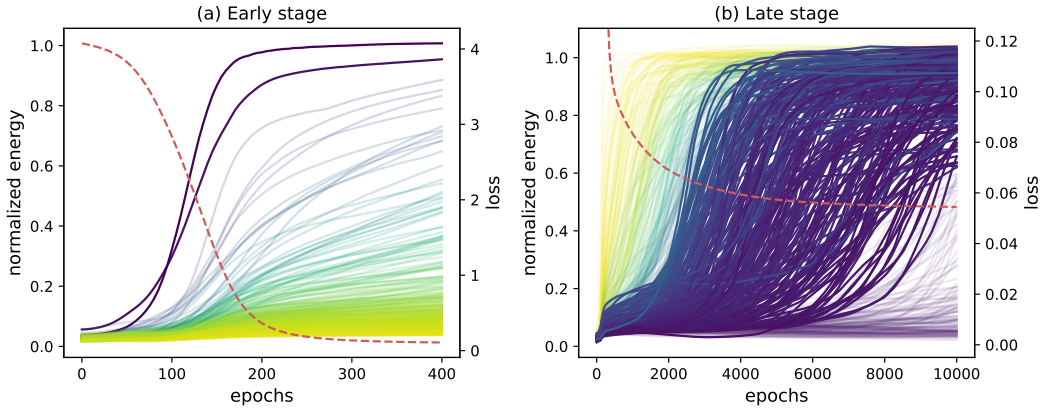

Figure 14: Normalized neuron-energy curves during (a) early-stage and (b) late-stage learning, used to extract feature-learning rates. Training loss is also shown. Opaque curves denote neurons satisfying the selection criteria.

B.2.1 Comparison of ReLU and Quadratic Activation in Late-Stage Learning

Under CE loss, our experiments show a clear contrast between ReLU and quadratic activation: many ReLU neurons develop task-aligned features after the training loss has nearly converged, whereas quadratic neurons show little additional feature acquisition during this late stage (Figure 15). This indicates that the persistence of late-stage feature learning depends on the activation function.

This contrast can be understood from the small-weight scaling under output-layer locking, $G_F \simeq (\lambda M^2)^{-1} F S^\top S$. Writing $\boldsymbol{w}_j = r\boldsymbol{u}_j$ with $\|\boldsymbol{u}_j\| = 1$, quadratic activation gives $f_j^\mu = O(r^2)$ and hence $(g_F)_j^\mu = O(r^2)$ at fixed residual field. The additional factor $\sigma'(\boldsymbol{w}_j \cdot \boldsymbol{x}^\mu) = O(r)$ makes the data-gradient contribution to $\dot{\boldsymbol{w}}_j$ cubic, $O(r^3)$, whereas L^2 weight decay contributes the linear term $-\lambda \boldsymbol{w}_j$. Thus sufficiently small quadratic neurons shrink rather than acquire features at late times. For ReLU, both terms are $O(r)$, so small weight magnitude alone does not suppress late-stage feature learning.

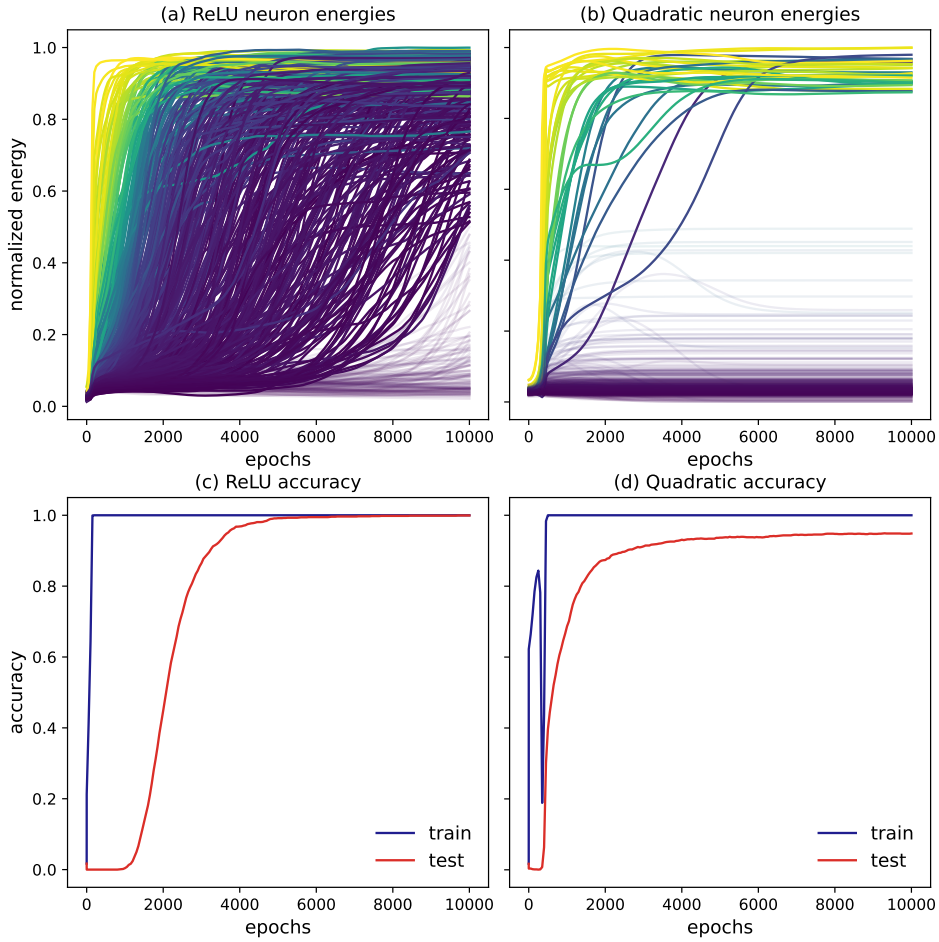

Figure 15: Feature learning of all neurons with (a) ReLU and (b) quadratic activation. Faint curves represent neurons whose normalized energy does not reach 0.5 within 10000 epochs. Panels (c) and (d) show training and test accuracy for ReLU and quadratic activation, respectively. Quadratic activation achieves a lower final test accuracy in this experiment.

B.3 Kernel Dynamics and Finite-Width Effects

B.3.1 Comparison of Model and Kernel Accuracies

We compare the model accuracy with the kernel accuracy obtained by taking the classwise argmax of $\tilde{Y}_t K_{tv}$, following the kernel representation in Section 3. As shown in Figure 16, the kernel accuracy generally reaches a given test-accuracy level later than the model. To examine the contribution of the residual-field approximation, we also compute accuracy from SK_{tv}, retaining the measured training residual S instead of replacing it by $\epsilon\tilde{Y}_t$. This gives closer agreement with the model, suggesting that deviations from the label-aligned residual-field ansatz contribute to the discrepancy. The difference between the model and kernel accuracies is also smaller at the larger width tested here.

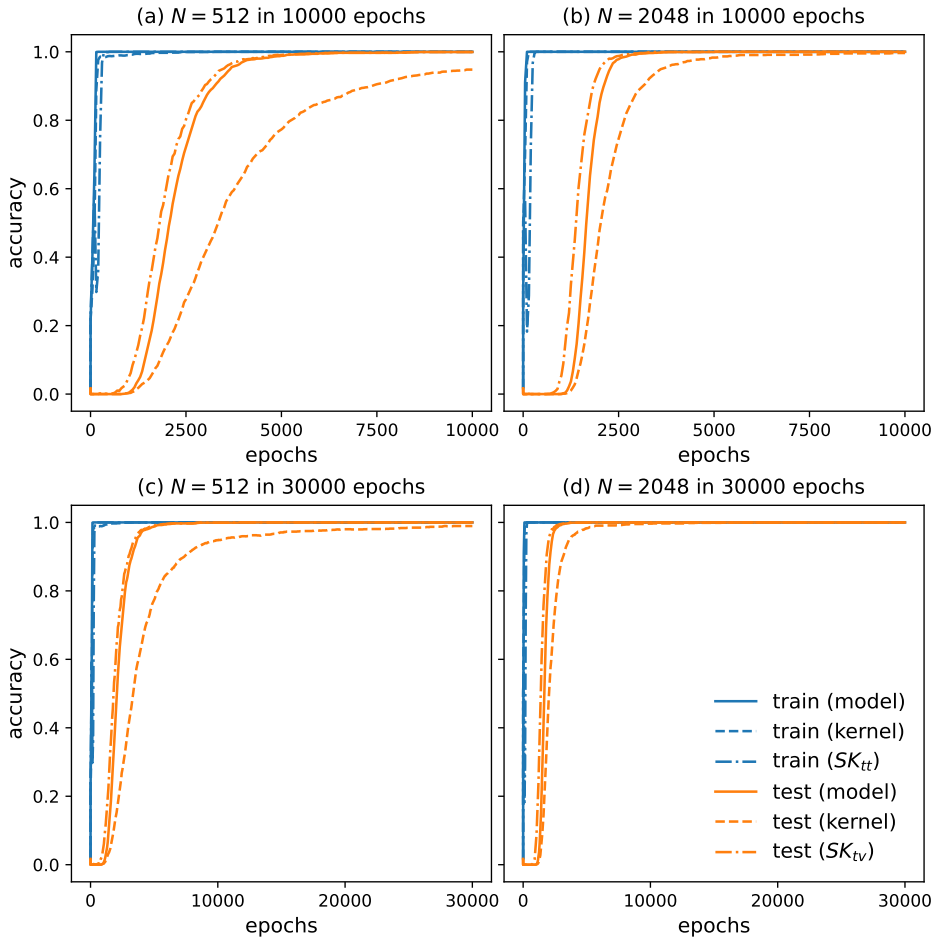

Figure 16: Comparison of model accuracies (solid), label-based kernel accuracies (dashed), and accuracies computed using the measured residual S (dash-dotted). Training and test predictions use $\tilde{Y}_t K_{tt}$ and $\tilde{Y}_t K_{tv}$ for the label-based kernel, and SK_{tt} and SK_{tv} for the residual-based scores. Panels (a), (c) use $N = 512$; panels (b), (d) use $N = 2048$. The top and bottom rows show the first 10000 and 30000 epochs, respectively.

B.3.2 Onset of the Grokking Transition

Numerical experiments show that the onset alignment $a_{tv}(t_0)$ is positive, increases with width, and approaches an approximately width-independent positive value at large N, as shown in Figure 17(b). The positive onset alignment can be understood as follows. Zero alignment means that the correct-class score balances the average wrong-class score after averaging over samples. It does not imply chance-level accuracy: a correct prediction requires correct-class score exceeds its largest competitor, as stated in Section 3. A positive average alignment can therefore be needed to overcome strong spurious competitors.

For a fixed test sample ν with label h, each wrong-label block column (defined as entries $K_{\mu\nu}$ with $[\mu] = g$ and $g \neq h$) contains at most two training samples sharing one of its inputs. Its column thus contains 0, 1, or 2 input collisions, whose kernel entries are enhanced relative to the background. Under independent balanced sampling for each label, the probability of a two-collision competitor is $P_2 = \frac{q(q-1)}{p(p-1)}$. The probability that at least one of the $p-1$ wrong labels has two collisions is $1-(1-P_2)^{p-1}$. For $q \simeq p/2$ and large p, these expressions reduce to $P_2 \simeq 1/4$ and $1-(3/4)^{p-1}$.

Let $C_{tv} = X_t^\top X_v$, the class score of C_{tv} corresponds to the collision count. Let $\tilde{K}_{tv} = \frac{p}{\|W\|_F^2} K_{tv}$ be the reduced kernel, normalized by the total squared weight norm relative to p. This removes the overall neuron-weight scale. We correspondingly define class score $\tilde{R}_h(\nu)$ on $\tilde{K}_{tv}$. Let $h = [\nu]$ denote the correct label and $g \neq h$ denote the wrong label in the following context. Let c denote the collision count and $\bar{\kappa}_{\text{coll}}, \bar{\kappa}^c_{\text{bg}}$ denote average kernel value of the collision entries and block-column backgrounds with c collisions. The class score is $\tilde{R}_g(\nu)|_c = c\bar{\kappa}_{\text{coll}} + (q-c)\bar{\kappa}^c_{\text{bg}} = c(\bar{\kappa}_{\text{coll}} - \bar{\kappa}^c_{\text{bg}}) + q\bar{\kappa}^c_{\text{bg}}$. Here we assume that the collision entry expectations are not affected by collision count, which is supported by the numerical results below. Near the chance-accuracy onset, the comparison with two-collision competitors motivates $\tilde{R}_h(\nu) \lesssim \mathbb{E}[\tilde{R}_g(\nu)|_{c=2}]$, as shown in Figure 17 (a). Assume that the conditional mean class score depends linearly on collision count. For $q \simeq p/2$, $\mathbb{E}[\tilde{R}_g(\nu)] \simeq \mathbb{E}[\tilde{R}_g(\nu)|_{c=1}]$. By Section 5, the alignment $a_{tv} \simeq \frac{q(p-1)}{p}\left\{\mathbb{E}[\tilde{R}_h(\nu)] - \mathbb{E}[\tilde{R}_g(\nu)]\right\}$. The characteristic alignment scale associated with two-collision competitors is

$$\begin{aligned} a_2 &= \frac{q(p-1)}{p}\left\{\mathbb{E}[\tilde{R}_g(\nu)|_{c=2}] - \mathbb{E}[\tilde{R}_g(\nu)]\right\} \\ &= \frac{q(p-1)}{p}[(\bar{\kappa}_{\text{coll}} - \bar{\kappa}^{c=2}_{\text{bg}}) + (q-1)(\bar{\kappa}^{c=2}_{\text{bg}} - \bar{\kappa}^{c=1}_{\text{bg}})]. \end{aligned} \tag{98}$$

If we assume that the background blocks follow exactly the Gaussian initialization, i.e., $\bar{\kappa}^c_{\text{bg}} = \kappa^g_{\text{bg}}$ and $\bar{\kappa}_{\text{coll}} = \kappa^g_{\text{coll}}$, we get $a_{2,\text{th}} = \frac{q(p-1)}{p}(\kappa^g_{\text{coll}} - \kappa^g_{\text{bg}})$.

We test for $p = 113, q = 56, N = 2048$. Let $D_{tv} = \tilde{K}_{tv} \circ C_{tv}$ be the contribution from collision entries. We compute the class sum of D_{tv}. The results for collision 0, 1 and 2 are 0, 0.34 and 0.68, meaning that the average contribution from one collision is 0.34, which is comparable to $\kappa^g_{\text{coll}} \approx 0.304$. This supports treating the mean collision-entry contribution as independent of collision count in this experiment. Numerical experiments also show a correlated enhancement of the background class score w.r.t. collision count. It gives $\bar{\kappa}^{c=1}_{\text{bg}} - \bar{\kappa}^{c=0}_{\text{bg}} = 3.0 \times 10^{-3}$ and $\bar{\kappa}^{c=2}_{\text{bg}} - \bar{\kappa}^{c=0}_{\text{bg}} = 6.6 \times 10^{-3}$. We estimate $a_{2,\text{num}} = 21.1$ from the numerical experiments and a simplified theoretical onset scale $a_{2,\text{th}} = 8.04$. The comparison between $a_{2,\text{num}}$ and real $a_{tv}(t_0)$ is shown in Figure 17 (b), where $a_{tv}(t_0) \lesssim a_{2,\text{num}}$ but exceeds $a_{2,\text{th}}$.

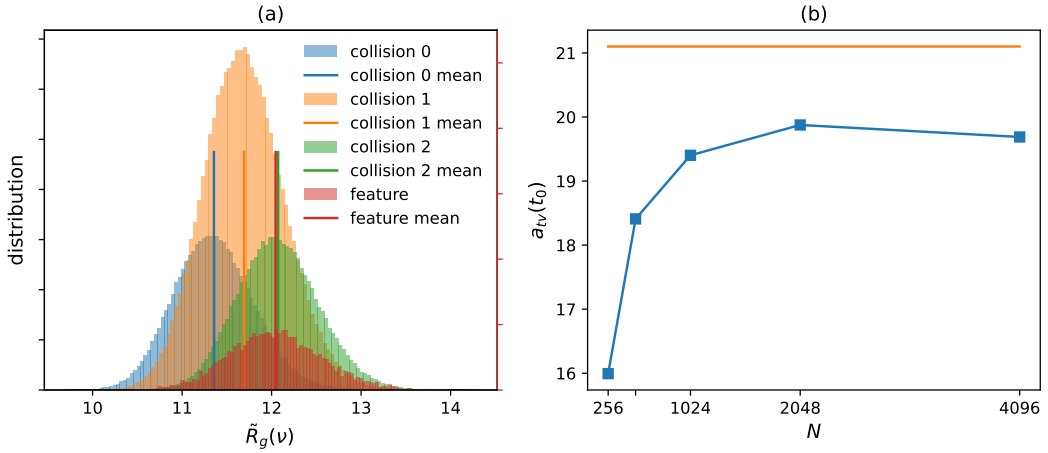

Figure 17: (a) Distributions and means of the reduced wrong-label class scores $\tilde{R}_g(\nu)$, grouped by collision count $c = 0, 1, 2$, at the onset of grokking for $p = 113$, $q = 56$, and $N = 2048$. The correct-label distribution $\tilde{R}_h(\nu)$ and its mean are also shown, using a separate vertical scale. (b) Onset alignment $a_{tv}(t_0)$ versus width N. The orange line marks $a_{2,\text{num}} = 21.1$.

B.3.3 Finite-Width Effects

Finite width causes fluctuations in the class scores. As shown in Figure 17 (a), the broadening of correct and wrong label class score causes a finite alignment gap Δ_ρ. We plot in Figure 18 for the

test run at accuracy 50%. Although in Figure 18 (a) the $\tilde{R}_h(\nu)$ has overlap with $\tilde{R}_g(\nu)$, we found strong positive correlation between these two which reduces fluctuations in their difference; overlap of the marginal distributions alone does not determine classification errors.

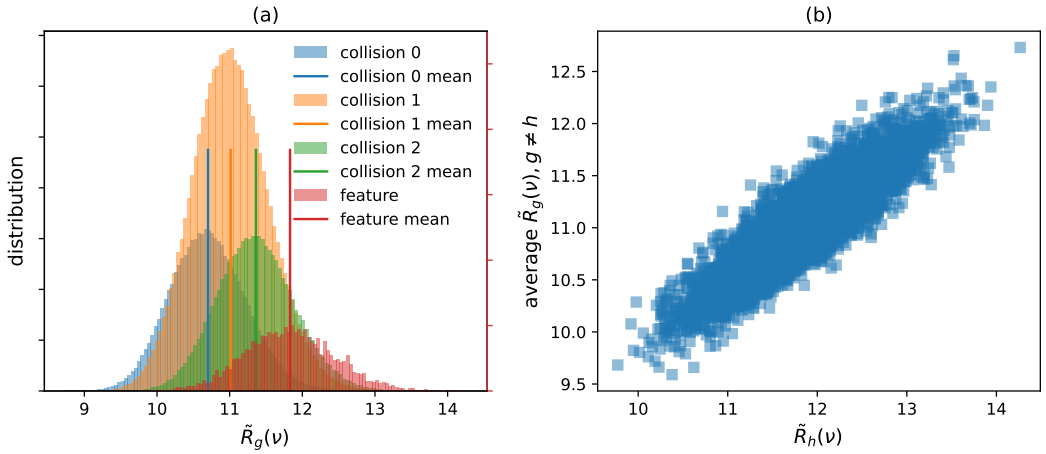

Figure 18: (a) Reduced class score distributions at test accuracy 0.5 for $p = 113$, $q = 56$, and $N = 2048$, grouped as in Figure 17(a). The smaller mean values compared to Figure 17(a) come from the shift of the kernel average $\mathbb{E}[\tilde{K}]$. (b) Samplewise correlation between the mean wrong-label score $\frac{1}{p-1}\sum_{g\neq h}\tilde{R}_g(\nu)$ and the correct-label score $\tilde{R}_h(\nu)$.

To relate these score fluctuations to the alignment gap, we approximate the relevant class-score distributions as evolving primarily through a shift over the transition. In main text Section 5 we proposed $\Delta_\rho \propto \left(N_{\text{eff}}^F\right)^{-1/2}$ relation. Here we show the results for all three runs, seeded from 0 to 9.

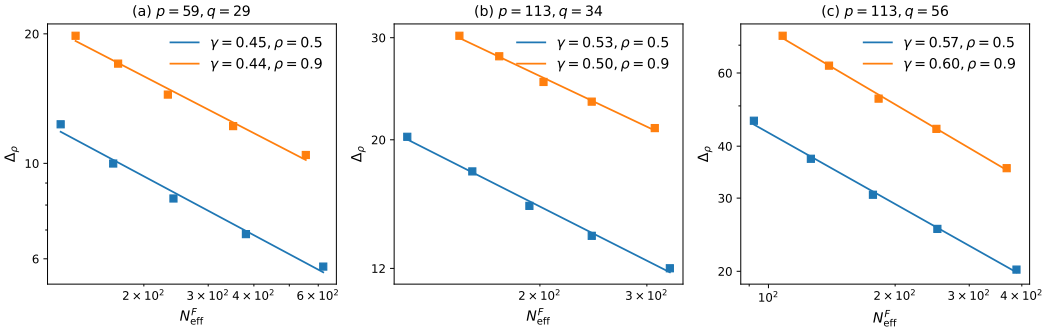

Figure 19: Alignment-gap scaling with feature effective width for (a) $p = 59, q = 29$, (b) $p = 113, q = 34$, and (c) $p = 113, q = 56$. Fits use target accuracies 0.5 and 0.9, widths $N = 256, 512, 1024, 2048, 4096$, and data-split seeds 0–9.

Norm-based effective width. The ReLU norm-balance argument in Section 4 allows nonuniform neuron norms while preserving the mean-field loss relation under the stated feature-statistics assumptions. Numerically, some neurons carry substantially more squared weight norm than others. Similar to the feature effective width defined in Section 5, we define a simpler norm-based effective width as

$$N_{\text{eff}} = \frac{\left(\sum_j \|\boldsymbol{w}_j\|_2^2\right)^2}{\sum_j \|\boldsymbol{w}_j\|_2^4} = \frac{\|W\|_F^4}{\sum_j \|\boldsymbol{w}_j\|_2^4}. \tag{99}$$

This is the participation ratio of squared neuron norms. Let $p_j^w = \frac{\|\boldsymbol{w}_j\|_2^2}{\sum_k \|\boldsymbol{w}_k\|_2^2}$, with $\sum_j p_j^w = 1$, then $N_{\text{eff}} = \left(\sum_j (p_j^w)^2\right)^{-1} \leq N$. It counts the equivalent number of equal-norm neurons. Unlike N_{eff}^F, N_{eff} accounts for only norm participation. At late norm equilibrium under the residual-field ansatz, Section 4 gives $\mathcal{E}_{0,j} \propto \|\boldsymbol{w}_j\|_2^2$, so $N_{\text{eff}}^F \simeq N_{\text{eff}}$. As an independent consistency check, let $\sigma_{\text{loss}}(t)$ be the standard deviation of training loss across samples and $\tilde{\sigma}_{\text{loss}}(t) = \sigma_{\text{loss}}(t)/\mathcal{L}(t)$. For the balanced datasets at late equilibrium, numerical results approximately follow $\tilde{\sigma}_{\text{loss}} \propto N_{\text{eff}}^{-1/2}$. We show the results for three settings in Figure 20.

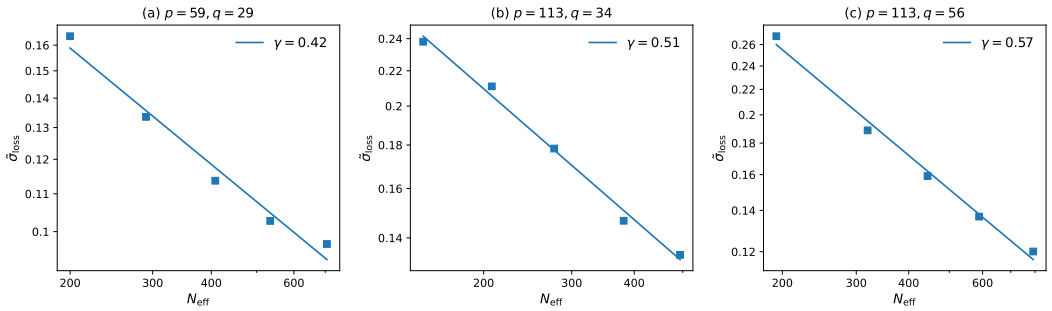

Figure 20: Fits of the relative loss fluctuations to $\tilde{\sigma}_{\text{loss}} \propto N_{\text{eff}}^{-\gamma}$ for (a) $p = 59, q = 29$, (b) $p = 113, q = 34$, and (c) $p = 113, q = 56$.

The effective width and feature effective width both increase with N, while their ratios to N evolve over training time. We show these relations in Figure 21(b). We also compare the alignment-gap scaling fitted using the feature effective width N_{eff}^F with that fitted using the bare width N in Figure 21(c).

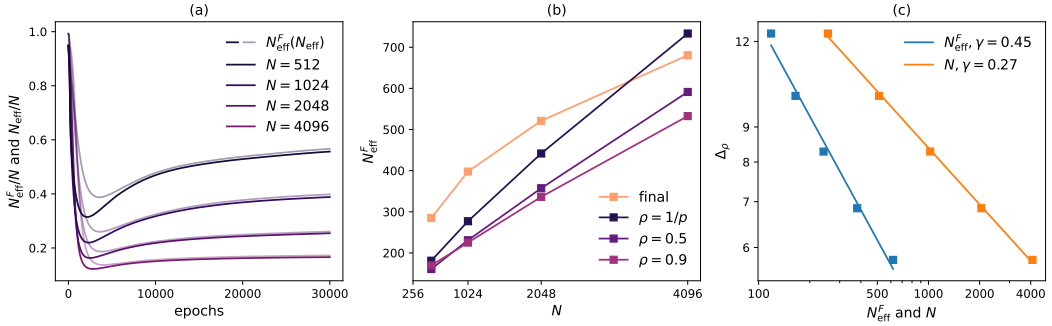

Figure 21: (a) Feature effective width ratio N_{eff}^F/N and effective width ratio N_{eff}/N versus training time. The two ratios converge at late times. (b) N_{eff}^F versus N at different accuracy levels. "Final" denotes the converged value, measured at epoch 30,000. (c) Comparison of the alignment-gap scaling fitted using the feature effective width N_{eff}^F and the bare width N at $\rho = 0.5$. All three panels use $p = 59, q = 29$.

B.4 Final Frequency and Phase Distribution

We test $p = 59$, $q = 29$, and $N = 2048$ with seeds 0–9. Figure 22 shows the frequency counts and phase-difference distribution pooled over all ten runs. The counts across frequencies are relatively uniform, as is the phase-difference distribution. Thus, the phase difference does not appear to impose a strong constraint on feature learning in these experiments.

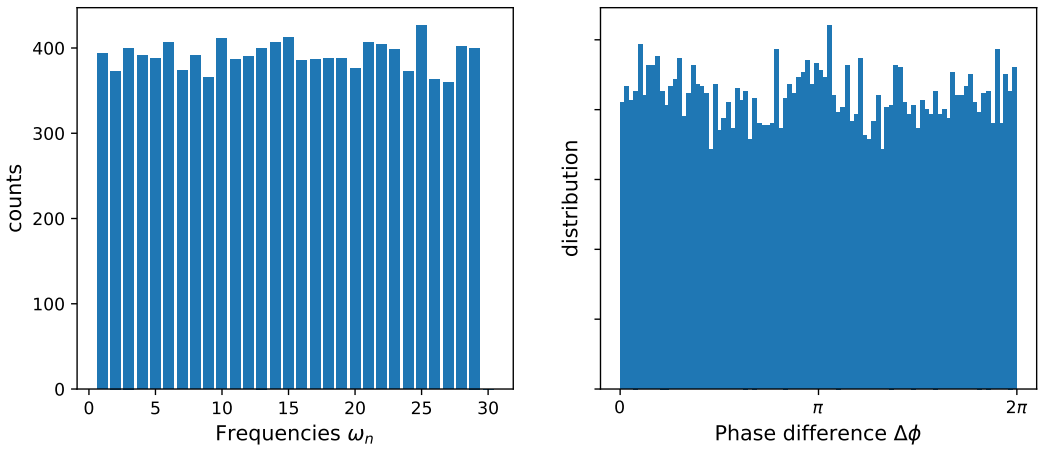

Figure 22: (a) Frequency counts of neurons pooled over ten runs. The frequencies are $\omega_n = 2\pi n/p$, with $n = 1, \ldots, (p-1)/2$. Neurons with frequency strength below 0.5 are excluded. (b) Distribution of the phase difference $\Delta\phi_j = \phi_{j,2} - \phi_{j,1}$ between the two input-channel components of $\boldsymbol{w}_j$.

B.5 Modular Addition with Fourier Initialization

We initialize the first layer with the random-phase Fourier bases defined in Section 3. The model generalizes immediately. The chance-level starting point is due to zero initialization of the second layer. The results are shown in Figure 23.

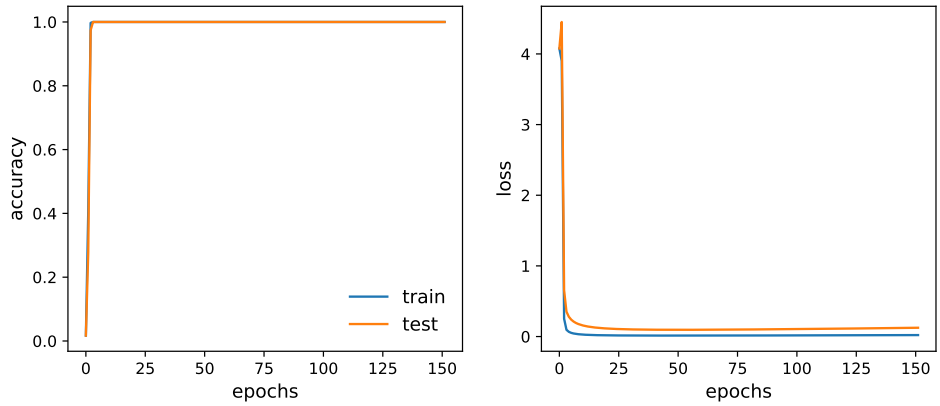

Figure 23: (a) Accuracy and (b) loss evolution of a model with Fourier initialization, using $p = 59$ and $N = 512$.

C Other Tasks

C.1 Other Modular Arithmetic Tasks

We test our framework on modular multiplication, $a \times b \bmod p$, and modular square, $(a^2+b^2) \bmod p$, where p is prime and $a, b \in \{1, \ldots, p-1\}$. We use $p = 59$, $q = 29$, and $N = 512$ for modular multiplication, and $p = 59$, $q = 20$, and $N = 512$ for modular square. Both tasks exhibit early anti-generalization followed by delayed generalization, although the early anti-generalization in modular square is weak. The normalized cross-kernel alignment a_{tv} initially dips for modular multiplication but not for modular square, suggesting that this dip may not be universal across modular tasks. We note that the initial evolution of a_{tv}, or equivalently the change of K_{tv} from its initialization, enters only at higher order in t. This is distinct from the early anti-generalization induced by the Gaussian

initialization kernel, which already appears at $O(t)$ in the logits. Figure 24 shows the accuracy, field alignments measured by cosine similarity, and dynamics of a_{tv} for both tasks.

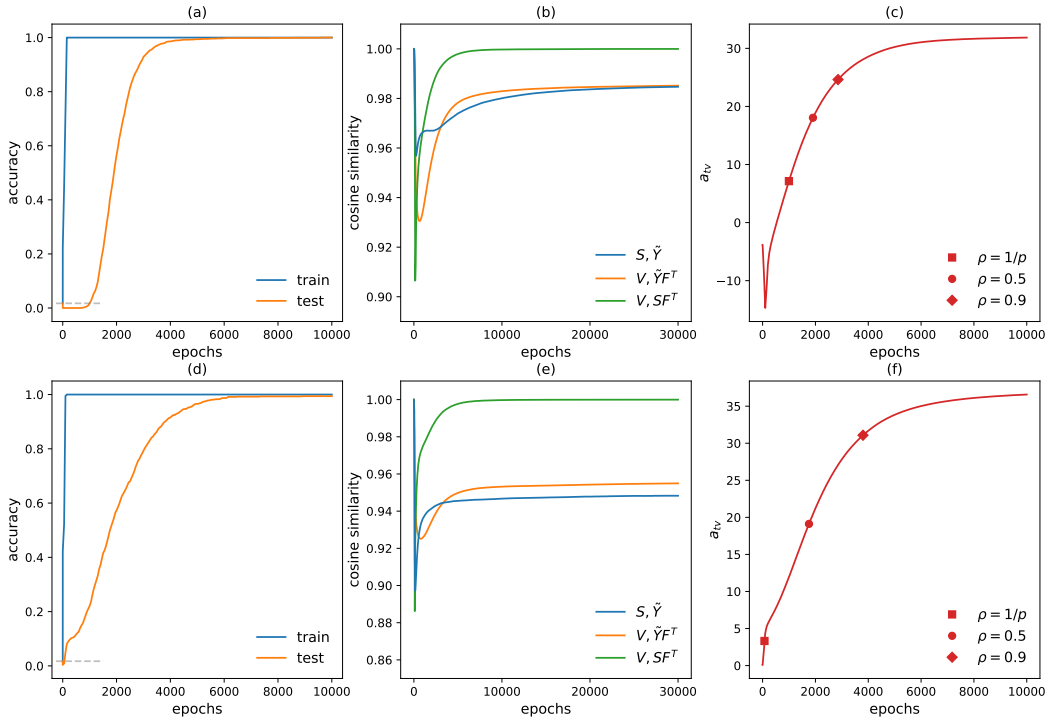

Figure 24: Numerical results for modular multiplication (top row) and modular square (bottom row). (a), (d) Training and test accuracy. (b), (e) Field alignments measured by cosine similarity. (c), (f) Dynamics of a_{tv}, with markers indicating test accuracies $\rho = 1/p$, 0.5, and 0.9. Modular square exhibits weak early anti-generalization but no initial dip in a_{tv}.

C.2 A Designed Anti-Generalization Task

We design a binary classification task that exhibits anti-generalization under a specific train–test splitting rule. The task is to classify the chirality of an ordered orthonormal frame consisting of three unit vectors.

In the first setting, the input contains three 3-dimensional vectors representing the frame axes and one additional 3-dimensional vector representing their shared origin. In total, the input has 12 dimensions.

In the second setting, the 3-dimensional frames are isometrically embedded in an n-dimensional space. A fixed matrix $A \in \mathbb{R}^{n\times 3}$, with $A^\top A = I_3$, is generated by QR decomposition of a Gaussian random matrix and shared by all samples. Independent Gaussian noise vectors restricted to the $(n-3)$-dimensional orthogonal complement of $A(\mathbb{R}^3)$ are added to the three embedded axes: $\boldsymbol{\xi}_k = (I_n - AA^\top)\boldsymbol{g}_k$, where $\boldsymbol{g}_k \sim \mathcal{N}(0, \sigma^2 I_n)$. The input has $3n$ dimensions, without an origin vector or subsequent normalization. Projection by $A^\top$ recovers the original frame exactly; A is not supplied to the network.

In both settings, frames are constructed in pairs: given two random orthogonal unit vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, we set $\boldsymbol{c} = \boldsymbol{a} \times \boldsymbol{b}$ and form $[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}]$ and $[\boldsymbol{a}, \boldsymbol{b}, -\boldsymbol{c}]$ with opposite chirality. The two members share the same origin in the first setting and the same three noise vectors in the second setting. We place one member of each pair in the training set and the other in the test set, choosing the assignment randomly while keeping both sets balanced between the two classes. This splitting rule places highly correlated, oppositely labelled pairs in the train–test block of $C = X^\top X$, providing a source of spurious correlations that can favor anti-generalization.

The numerical experiments exhibit pronounced early anti-generalization followed by delayed generalization. For both settings, we use a two-layer ReLU network with one hidden layer of width $N = 1024$, cross-entropy loss, weight decay $\lambda = 10^{-4}$, and SGD without momentum. For setting 1, each of the training and test sets contains 450 samples, $\eta = 0.02$, and the origin coordinates are independently drawn from $\mathcal{N}(0, 2^2)$. For setting 2, each set contains 1600 samples, $n = 7$, $\eta = 0.05$, and $\sigma = 1$ for the Gaussian noise defined above. The results are shown in Figure 25.

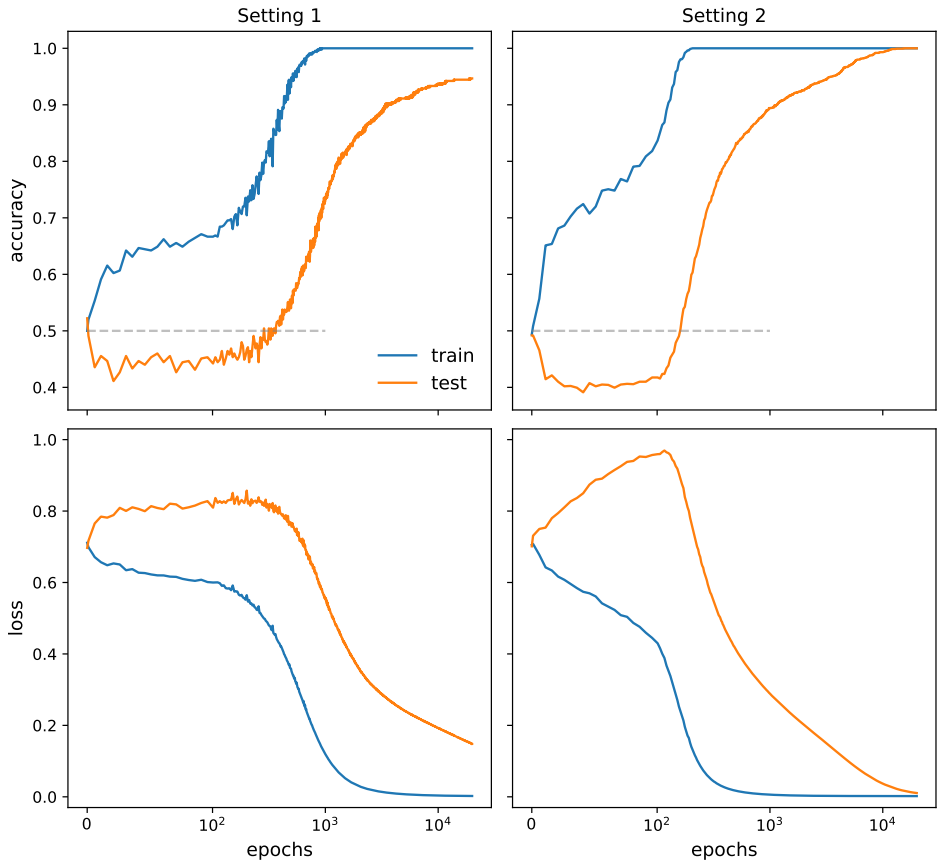

Figure 25: Numerical results for setting 1 (left column) and setting 2 (right column). Top row: training and test accuracy. Bottom row: training and test loss. The gray dashed line in the accuracy panels indicates the binary chance level of 0.5.