A MECHANISTIC THEORY OF FEATURE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a theory of feature learning in overparameterized neural networks built on the dynamics of the "active parameter subspace": the low-dimensional subspace of parameter space spanned by the network Jacobian. We show that this subspace is determined by the instantaneous gradient of the network with respect to its parameters, that it evolves throughout training, and that the curvature of the loss acts as the infinitesimal generator of its evolution. Thus, feature learning becomes a trajectory on the Grassmannian of parameter subspaces, driven by the Hessian of the loss, which in turn depends on weight decay. This perspective contrasts with the neural tangent kernel (NTK) approximation, in which learning reduces to coefficient updates within a fixed basis, and with phenomenological kinematics accounts that describe feature evolution without an underlying generator. Numerical experiments on fully connected networks trained with full-batch gradient descent support the picture: weight decay sustains the subspace trajectory after training loss has plateaued, enabling the network to discover a representation that also generalizes, while without weight decay the active subspace freezes and generalization fails. This framework may explain why delayed generalization occurs, and why weight decay is strongly associated with grokking.

1 INTRODUCTION

Large efforts have been devoted in recent years to attain a mechanistic understanding of deep learning (Simon et al., 2026). The pioneering work of Jacot et al. (2018) introduced the concept of neural tangent kernels (NTKs), which allow to understand training of neural networks as a "lazy" kernel regression process, by which the network residual (the mismatch between predictions and targets) is reduced constantly over training time. The original NTK was conceptualized in the limit of large network width (Arora et al., 2019), and this was enough to attain important insights (Lee et al., 2019), e.g, initial learning to be strongly biased by the NTK spectrum at initialization (Bordelon et al., 2020; Baratin et al., 2021). The concept has been generalized to other architectures as GNNs, CNNs and Transformers (see Du et al. (2019), Arora et al. (2019) and Hron et al. (2020), respectively). Currently, researchers focus on identifying "adaptive tangent" structures, an "empirical" NTK, that evolve during training (Huang & Yau, 2020; Fort et al., 2020),

Despite these advances, a unified picture of the learning dynamics remains elusive (Simon et al., 2026). While evolving kernels describe changes in prediction accuracy during training, they often obscure how parameters evolve to capture data structure. The eventual alignment between network and task may be simply due to a long enough search guided by gradient descent and enough adjustable degrees of freedom in the network. Alternatively, we formulate training dynamics as motion on a Grassmannian (Helgason, 2001) in which a point represents an attainable subspace in parameter space. The loss Hessian — often called the "sharpness" of the landscape (Hochreiter & Schmidhuber, 1997; Keskar et al., 2017) — drives this subspace exploration. This gives a mechanistic causal account of feature learning beyond the lazy regime.

Fundamentally, we propose to shift the emphasis of the analysis from *what the residual is doing* to *what the parameters of the network are doing* during training. In doing so, we develop a theory in which training and test loss, memorization and generalization (feature learning), can be seen as different phases in the evolution of the parameters of the network, driven by well-defined causes. We set out to spell out the governing equations of gradient-descent training in a simplified setting that is sufficient to reveal all the fundamental forces at play. For a visual summary, see Figure 1.

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

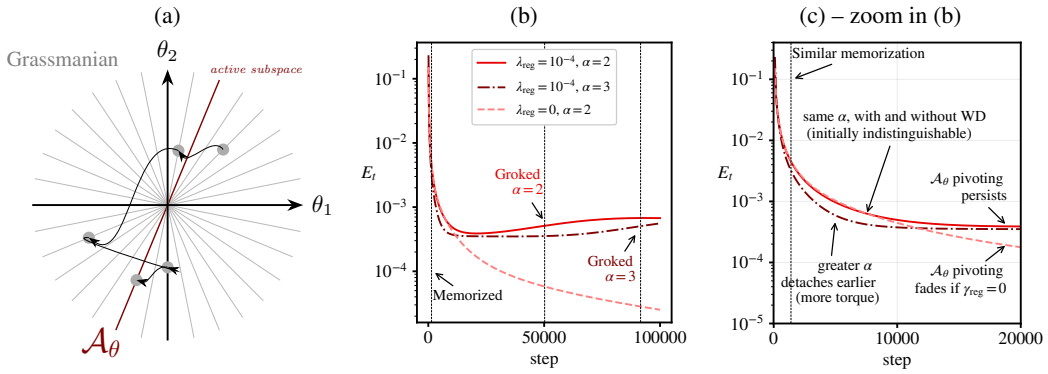


Figure 1: Visual summary of the paper's contributions. (a) Training viewed as pivoting between subspaces on the Grassmannian, driven by the loss Hessian. The current subspace at step t is the active subspace $\mathcal{A}_\theta$. (b) The diagnostic E_t measures the active-subspace rotation velocity for three canonical cases: (1) $\alpha = 2$ with weight decay ($\gamma_{\text{reg}} = 10^{-4}$), (2) $\alpha = 2$ without weight decay, (3) $\alpha = 3$ with weight decay. During memorization, all three rotate similarly; afterward, weight decay keeps cases (1) and (3) pivoting until generalization, while case (2) freezes and fails to generalize. (c) Zoom of the early phase: larger initial weights in case (3) change trajectory immediately after memorization, and case (2) detaches once the residual-driven Hessian contribution vanishes.

This perspective offers a novel explanation for phenomena such as the delayed generalization phenomenon known as grokking (Liu et al., 2023): delayed generalization can occur as long as the force driving the movement in the Grassmannian does not fade, its main component being tied to the value of the predictions residuals (which can decrease too fast after memorization) but weight decay introduces a second component that remains fueling the successive testing of attainable subspaces until an adequate one is found. We implement this framework within a controlled setting (teacher-student MLPs, MSE loss, full-batch gradient descent with and without constant weight decay) to isolate the main effects, laying a foundation for broader application.

The paper is structured as follows: Section 2 reviews the literature in depth to put our contribution in context; Section 3 presents the mathematical backbone and introduces the fundamental concepts in order to, in Section 4, show explicitly how the search for an adequate admissible set of parameters happens. We test the predictions of our framework against simple numerical training examples mentioned above in Section 5, while Section 6 proposes final remarks and future work directions.

2 Related Work

Neural Tangent Kernels and Lazy Training. The foundational work of Jacot et al. (2018) established the Neural Tangent Kernel (NTK) framework, demonstrating that in the infinite-width limit, neural networks evolve as linear models in function space. Subsequent work extended the NTK to finite widths, treating it as an evolving object (Huang & Yau, 2020; Fort et al., 2020) that adapts to data (Jiang et al., 2025). However, while evolving kernels capture residual-space dynamics, they may tend to obscure the logic behind the evolution of the network parameters. Our work complements this by focusing directly on the active parameter subspace geometry, rather than on the induced kernel evolution.

Feature Learning and Mean-Field Limits. To capture feature learning, alternative infinite-width limits were developed where weights evolve significantly (Mei et al., 2018). The Maximal Update Parameterization (Yang & Hu, 2021) enables a nontrivial feature-learning regime at large width while making many optimal hyperparameters stable across widths.. Recent efforts along "the mechanics of learning" (Simon et al., 2026) advocate for first-principles descriptions of these dynamics, and we have seen meaningful contributions such as self-consistent dynamical mean-field theory (Bordelon & Pehlevan, 2022), effective Gram matrices (Yang & Chaudhari, 2025) and central flows (Cohen et al., 2025). These approaches characterize feature evolution either by averaging over many

parameters or over continuous time. In contrast, we study an explicit discrete flow: finite-size networks, finite learning rates, and a clear infinitesimal generator at each step.

Geometry of Optimization. In deep learning, geometric perspectives have already revealed that gradient descent is intrinsically constrained to a subspace (Gur-Ari et al., 2018), that there exist conservation laws in neural networks (Kunin et al., 2021; Marcotte et al., 2023) and characterized the sharpness of the loss landscape (Cohen et al., 2021). The "edge of stability" phenomenon (Cohen et al., 2021) highlights the interplay between learning rate and curvature, which we extend by showing how curvature actively propagates parameter evolution. Recently, Marcotte et al. (2025; 2026) have shown, for architectures including Transformers, that gradient flow reduces to a manifold flow governed by conservation laws, and they can account for weight decay as a symmetry-breaking factor. Our work differs in two ways. First, we pose training as a discrete-time flow generated by finite-size matrices updated at each learning-rate step, rather than as continuous time or an infinite-width limit. Second, we argue that the important dynamics happens in the space of admissible parameter subspaces that must be explored. On top of that, we always work at the level of fundamental primitives (network predictions, residuals and parameters).

Grokking and delayed generalization. For the grokking phenomenon (Power et al., 2022; Bertolotti & Cazzola, 2026), where generalization occurs long after training loss has minimized, has sparked various theoretical explanations, and different mechanisms have been proposed: phase transitions in circuit formation (Nanda et al., 2023), Bayesian posterior evolution (Rubin et al., 2024), and implicit regularization towards minimal-norm solutions (Wang et al., 2024). These works, and many others that followed, have been instrumental in understanding when, where, and (in some particular cases) how grokking occurs. Our theory focuses instead on how grokking proceeds dynamically: we posit that subspace evolution settles the network into an adequate set of parameters that not only memorize the training targets but also successfully predict unseen validation data.

3 Governing equations

We assume a neural network (NN) described by a function $\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})$, where $\boldsymbol{x} \in \mathbb{R}^m$ is the vector collecting all the inputs and $\boldsymbol{\theta} \in \mathbb{R}^p$ is the vector collecting all the parameters of the network (p being the total number of parameters in the NN). The training set contains not only inputs $\boldsymbol{x}$ but also $\boldsymbol{y} \in \mathbb{R}^n$ scalar datapoints the network must predict (n being the total number of scalar targets). As we assume that the inputs and outputs in the training sets are given, we will think of the NN as $\boldsymbol{f} : \mathbb{R}^p \to \mathbb{R}^n$, i.e., a nonlinear map from parameters to full-batch predictions (either training or test), and we will simply write $\boldsymbol{f}(\boldsymbol{\theta}) \in \mathbb{R}^n$. The residual is therefore $\boldsymbol{r}(\boldsymbol{\theta}) = \boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{y}$. For convenience, we call $\boldsymbol{v} = \dot{\boldsymbol{\theta}}$ "instantaneous parameter velocity".

3.1 Without weight decay

The Jacobian. We assume full-batch gradient descent with MSE loss, $\mathcal{L} = 1/2\|\boldsymbol{r}\|^2 = (\boldsymbol{r}^\top \boldsymbol{r})/2$, which can be written as:

$$\dot{\boldsymbol{\theta}} = -\nabla_\theta \mathcal{L} = -\boldsymbol{J}^\top \boldsymbol{r}\,, \tag{1a}$$

where $\boldsymbol{J} = \nabla_\theta \boldsymbol{f} \in \mathbb{R}^{n\times p}$ is the Jacobian of the NN map at time t, thus, applying the chain rule

$$\dot{\boldsymbol{r}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} = -\boldsymbol{J}\boldsymbol{J}^\top \boldsymbol{r} = -\boldsymbol{K}\boldsymbol{r}\,, \tag{1b}$$

where $\boldsymbol{K}$ is the so-called neural tangent kernel (NTK) at time t [1]. For $n < p$, we can naturally assume that $\mathrm{rank}(\boldsymbol{K}) = n$, i.e., it is a full-rank matrix, which moreover is positive semidefinite since it is built from $\boldsymbol{J}\boldsymbol{J}^\top$, its eigenvalues ($\lambda_i$ for $i = 1,\dots,n$) are the n singular values of $\boldsymbol{J}$ squared, and thus they are greater of equal to zero) . In this paper, NTK will be considered a function of training time.

[1] We are using NTK somewhat loosely: in its original meaning, the NTK corresponded to the value at $t = 0$ (initial conditions) and it was assumed a wide network, what leads to assuming that $\boldsymbol{K}$ does not change.

The Hessians Let us introduce the Hessian of $\boldsymbol{f}$ and that of the MSE loss. The Hessian of the NN map is a third-order tensor,

$$\mathsf{H} = \mathsf{H}_{ijk} = \frac{\partial f_i}{\partial \theta_j \partial \theta_k} = \frac{\partial r_i}{\partial \theta_j \partial \theta_k} = \nabla_\theta \boldsymbol{J}\,, \tag{2a}$$

while the Hessian of the loss is a $p \times p$ matrix,

$$\boldsymbol{H} = H_{jk} = \frac{\partial^2}{\partial \theta_j \partial \theta_k}\left(\frac{1}{2}\sum_{l=1}^{n} r_l r_l\right) = \underbrace{\boldsymbol{J}^\top \boldsymbol{J}}_{\boldsymbol{G}} + \underbrace{\boldsymbol{r}^\top \mathsf{H}}_{\boldsymbol{H}_{\text{net}}} = \boldsymbol{G} + \boldsymbol{H}_{\text{net}}\,, \tag{2b}$$

note that the Hessian of the loss depends not only on the gradient of the residual (first addend, $\boldsymbol{G}$) but also on the magnitude and curvature of the network (second addend). See that the Hessian of the loss appears as the forcing term if we write an equation for the parameter velocity: taking another time derivative of eq. (1a)

$$\dot{\boldsymbol{v}} = \frac{d}{dt}\left(-\nabla_\theta \mathcal{L}\right) = -\nabla^2_\theta \mathcal{L}\, \dot{\boldsymbol{\theta}} = -\boldsymbol{H}\boldsymbol{v}\,. \tag{3}$$

This equation tell us that curvature of the loss is the infinitesimal generator of the parameter rate of change.

Jacobian and Hessians do not share the same spectrum. The Jacobian is the matrix that connects the space where the residuals are measured ($\mathbb{R}^n$) and the space from where we draw parameters ($\mathbb{R}^p$). Since $p > n$ (over-parametrized setting), it follows that $\boldsymbol{J}$ is a rectangular matrix, for which we have a SVD decomposition:

$$\boldsymbol{J} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \in \mathbb{R}^{n\times p} \rightarrow \boldsymbol{J}^\top = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top \in \mathbb{R}^{p\times n}\,, \tag{4}$$

where

- $\boldsymbol{U} \in SO(n) \rightarrow \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_n$, and the columns of $\boldsymbol{U}$ are also the eigenvectors of $\boldsymbol{K}$, which we will refer to as "learning modes" or "learning directions" (Jacot et al., 2018),
- $\boldsymbol{\Sigma} = \text{diag}(\sigma_1\,, \ldots,\, \sigma_n)$ where the singular values of $\boldsymbol{J}$ (σ_i) squared are the eigenvalues of $\boldsymbol{K}$.
- $\boldsymbol{V} \in \mathbb{R}^{p\times n}$ such that $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}_n$.

Let us also highlight that the orthogonal projector that takes any point in $\mathbb{R}^p$ back to the span of the columns of $\boldsymbol{V}$, i.e., back to $\text{Im}(\boldsymbol{J}^\top) = \text{Im}(\boldsymbol{V})$, is $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^\top \in \mathbb{R}^{p\times p}$, such that $\boldsymbol{P}^2 = \boldsymbol{P}$, $\boldsymbol{P}^\top = \boldsymbol{P}$ and $\text{rank}\ \boldsymbol{P} = n$. The columns of the orthogonal complement of $\boldsymbol{V}$, termed $\boldsymbol{V}^\perp$, span the rest of possible direction in parameter space, i.e., $\text{span}([\boldsymbol{V},\, \boldsymbol{V}^\perp]) = \mathbb{R}^p$.

The SVD decomposition therefore can help to understand the spectrum of the following objects:

$$\boldsymbol{K} = \boldsymbol{J}\boldsymbol{J}^\top = \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^T\,, \tag{5a}$$

$$\boldsymbol{G} = \boldsymbol{J}^\top\boldsymbol{J} = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T\,, \tag{5b}$$

thus $\boldsymbol{G}$ and $\boldsymbol{K}$ share the same eigenvalues ($\boldsymbol{\Sigma}^2$), unlike the loss Hessian (Liao & Mahoney, 2021):

$$\boldsymbol{H} = \underbrace{\begin{bmatrix}\boldsymbol{\Sigma}^2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}}_{\boldsymbol{G}} + \underbrace{\begin{bmatrix}\boldsymbol{\mathcal{H}}_1 & \boldsymbol{\mathcal{H}}_3^\top\\ \boldsymbol{\mathcal{H}}_3 & \boldsymbol{\mathcal{H}}_2\end{bmatrix}}_{\boldsymbol{H}_{\text{net}}}\,. \tag{5c}$$

where we have chosen to express the entries of $\boldsymbol{H}$ in the basis of $\mathbb{R}^p$ formed by $\boldsymbol{V}$ and its orthogonal complement $\boldsymbol{V}^\perp$. Let us look carefully at the structure of the component associated to $\boldsymbol{H}_{\text{net}}$ in the loss Hessian, in relation to eq. (3):

- $\boldsymbol{\mathcal{H}}_1 = \boldsymbol{V}^\top\boldsymbol{H}_{\text{net}}\boldsymbol{V} \in \mathbb{R}^{n\times n}$ contains spectral elements connected to the residual space (NTK).
- $\boldsymbol{\mathcal{H}}_2 = (\boldsymbol{V}^\perp)^\top\boldsymbol{H}_{\text{net}}\boldsymbol{V}^\perp \in \mathbb{R}^{(p-n)\times(p-n)}$ represents components in parameter space decoupled from the residual space.
- $\boldsymbol{\mathcal{H}}_3 = (\boldsymbol{V}^\perp)^\top\boldsymbol{H}_{\text{net}}\boldsymbol{V} \in \mathbb{R}^{(p-n)\times n}$ can be regarded as a coupling term that connects a subspace of parameter space of the same size as residual space to the other portion of dimension $p - n$.

See that unlike $\boldsymbol{G}$, (which is rank-deficient and, at most, has rank n), $\boldsymbol{H}_{\text{net}}$ can be full rank (rank p). Moreover, $\boldsymbol{H}_{\text{net}} = \boldsymbol{r}^\top \mathbf{H}$ also includes weighting with the current residual values (can be positive or negative), thus, the eigenvalues of $\boldsymbol{H}_{\text{net}}$ are not constrained to be positive since the matrix does not have to be positive definite. This means that the curvature of the network induces extra flexibility in the parameter rate evolution, and thus on the parameters themselves. Evidently, despite the eigenvalues of $\boldsymbol{K}$ appearing in eq. (5c), the eigenvalues and the eigenvectors of $\boldsymbol{H}$ can be very different.

3.2 With weight decay

The Jacobian does not change. Adding weight decay, $\gamma_{\text{reg}}\|\boldsymbol{\theta}\|^2/2$ ($\gamma_{\text{reg}} > 0$), to the loss entails a second term driving the gradient descent:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{v} = -\nabla_\theta \mathcal{L} = -\boldsymbol{J}^\top \boldsymbol{r} - \gamma_{\text{reg}} \boldsymbol{\theta} \,. \tag{6}$$

Since adding weight decay does not affect the architecture of the network, everything we said in Section 3.1 as to $\boldsymbol{J}$, $\boldsymbol{K}$ and $\boldsymbol{G}$ still holds.

The loss of the Hessian does change. Conversely, the Hessian of the loss gains an extra term

$$\boldsymbol{H} = \boldsymbol{G} + \boldsymbol{H}_{\text{net}} + \gamma_{\text{reg}} \boldsymbol{I}_p \,, \tag{7}$$

that will perturb the spectrum of the generator eq. (3), but does not modify the non-diagonal blocks in eq. (5c) discussed at the end of Section 3.1 .

3.3 Propagators

Henceforth, to clearly convey that the the variables evolve over training steps, we add the subindex t in order to number iterations explicitly. The instantaneous solution of eq. (1b) at any time is a time-ordered path integral (Yang & Chaudhari, 2025)

$$\boldsymbol{r}_t = \mathcal{T} \exp\left(- \int_0^t \boldsymbol{K}(s) ds \right) \boldsymbol{r}_0 = \exp\left(-\boldsymbol{K} t\right) \boldsymbol{r}_0 + \text{higher-order terms} \tag{8a}$$

which can be simplified if we ignore higher-order terms (Magnus expansion)

$$\boldsymbol{r}_t \approx \exp\left(-\boldsymbol{K} t\right) \boldsymbol{r}_0 \,, \tag{8b}$$

i.e., we assume that the propagator remains effectively constant over a timestep $\Delta t = \eta$ short enough. Forming the chain of propagators (a causal chain, in which each new term is unknown until the prior one has been obtained), and assuming constant learning rate, we can write the residual after $T+1$ steps as

$$\boldsymbol{r}_{T+1} = \left[\prod_{t=0}^{T} \exp(-\eta \boldsymbol{K}_t) \right] \boldsymbol{r}_0 = \Big[\exp(-\eta \boldsymbol{K}_T) \exp(-\eta \boldsymbol{K}_{T-1}) \dots \exp(-\eta \boldsymbol{K}_1) \exp(-\eta \boldsymbol{K}_0) \Big] \boldsymbol{r}_0 \,. \tag{9}$$

Forming the other chain of causal propagators, and assuming constant learning rate short enough so the Hessian does not change either, we can solve eq. (3) as in eq. (8b) and approximate the solution after $T+1$ steps to express the rate of change of parameters as causal chain of propagators similar to eq. (9)

$$\boldsymbol{v}_{T+1} = \left[\prod_{t=0}^{T} \exp(-\eta \boldsymbol{H}_t) \right] \boldsymbol{v}_0 = \Big[\exp(-\eta \boldsymbol{H}_T) \exp(-\eta \boldsymbol{H}_{T-1}) \dots \exp(-\eta \boldsymbol{H}_1) \exp(-\eta \boldsymbol{H}_0) \Big] \boldsymbol{v}_0 \,. \tag{10}$$

We could obtain the parameters $\boldsymbol{\theta}$ by integrating over all the trajectory. Notice how, unlike in eq. (9), adding weight decay does affect this chain explicitly because the Hessian of the loss gains a constant diagonal term, eq. (7).

3.4 INTRODUCING THE ACTIVE PARAMETER SUBSPACE

Recall that we assume an over-parameterized setting $p > n$. This fact is built into the gradient flows eq. (1a) and eq. (6), which gives the parameter velocity at every instant. Because the p-dimensional parameter velocity is spanned by only n residuals (an observation well-documented by Gur-Ari et al. (2018)), the parameters cannot explore all of $\mathbb{R}^p$; they are confined at each time to a short-lived n-dimensional subspace defined by the instantaneous Jacobian $\boldsymbol{J}$. We call this n-dimensional subspace the "active" subspace within the parameter space, while the remaining $p - n$ "bulk" coordinates remain "frozen". We therefore surmise that feature learning amounts to finding an adequate active plane, and that the search is governed by the infinitesimal generators in eq. (10), i.e., the Hessian in eq. (5c). The next question is: *how does the active parameter subspace $\mathcal{A}_\theta = \mathrm{Im}(\boldsymbol{V})$ evolve during training?* We answer the question in next section.

4 INFINITESIMAL EVOLUTION OF THE ACTIVE PARAMETER SUBSPACE

Recap. Parameters evolve in $\mathbb{R}^p$ but gradient flow confines them instantaneously to the n-dimensional active subspace $\mathcal{A}_\theta = \mathrm{Im}(\boldsymbol{V}_t)$. This section characterizes how $\boldsymbol{V}_t$ changes from step to step and expresses that perturbation in terms of the Jacobian, the Hessian, and the learning rate. The first step is to estimate the instantaneous rate of change of the Jacobian.

Instantaneous rates of change. Assuming that the parameters of the network at any given time are known, i.e., the vector $\boldsymbol{\theta}_t$ to be available (this happens at initialization when we pick $\boldsymbol{\theta}_0$, but it also happens at the end of each training step), then one can evaluate the residual $\boldsymbol{r}_t = \boldsymbol{f}(\boldsymbol{\theta}_t) - \boldsymbol{y}$, the gradient of the network is the Jacobian $\boldsymbol{J}_t = \nabla_\theta \boldsymbol{f}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$ and the network curvature is given by $\mathbf{H}_t = \nabla^2_\theta \boldsymbol{f}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.

Likewise, $\boldsymbol{K}_t = \boldsymbol{J}_t\boldsymbol{J}_t^\top$, $\boldsymbol{G}_t = \boldsymbol{J}_t^\top\boldsymbol{J}_t$, $\nabla_\theta\mathcal{L}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} = -\boldsymbol{J}_t^\top\boldsymbol{r}_t$, $\nabla_\theta\mathcal{L}|^2_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} = \boldsymbol{H}_t = \boldsymbol{G}_t + \boldsymbol{r}_t^\top\mathbf{H}_t + \gamma_{\mathrm{reg}}\boldsymbol{I}_p$. Now, to estimate the rate of change of those variables based on the current data, Begin with $\boldsymbol{v}_t = \dot{\boldsymbol{\theta}}_t$, and combine with $\dot{\boldsymbol{J}}_t = \nabla^2_\theta\mathcal{L}\dot{\boldsymbol{\theta}}_t$:

$$\dot{\boldsymbol{J}}_t = \dot{\boldsymbol{\theta}}_t^\top\mathbf{H}_t = \boldsymbol{v}_t^\top\mathbf{H}_t = -\left(\boldsymbol{J}_t^\top\boldsymbol{r}_t + \gamma_{\mathrm{reg}}\boldsymbol{\theta}_t\right)^\top\mathbf{H}_t\,, \tag{11}$$

from here it follows, using the chain rule again, $\dot{\boldsymbol{K}}_t = \dot{\boldsymbol{J}}_t\boldsymbol{J}_t^\top + \boldsymbol{J}_t\dot{\boldsymbol{J}}_t^\top$, $\dot{\boldsymbol{G}}_t = \dot{\boldsymbol{J}}_t^\top\boldsymbol{J}_t + \boldsymbol{J}_t^\top\dot{\boldsymbol{J}}_t$ and so on and so forth. Ignoring terms $\mathcal{O}(\eta^2)$, the variations of these variables across steps can be approximated using these rates, e.g., $\delta\boldsymbol{J}_t \approx \eta\dot{\boldsymbol{J}}_t$, and then one can substitute eq. (11).

4.1 CLOSED-FORM EXPRESSION OF INFINITESIMAL ACTIVE SUBSPACE PERTURBATION

We can visualize the impending motion ("pivoting") of the current active subspace $\mathcal{A}_\theta = \mathrm{Im}(\boldsymbol{V}_t)$ at the current step by looking at the instantaneous rate of change of the matrix $\boldsymbol{V}_t$.

The explicit form of the perturbation. Recall $\boldsymbol{V}$ together with its orthogonal complement, $\boldsymbol{V}_t^\perp$ span the rest of possible directions in $\mathbb{R}^p$. Then, an infinitesimal change in the matrix $\boldsymbol{V}_t$ can be written without lack of generality as

$$\delta\boldsymbol{V}_t = \delta\boldsymbol{V}_t^{\parallel} + \delta\boldsymbol{V}_t^{\perp} = \boldsymbol{V}_t\boldsymbol{A}_t + \boldsymbol{V}_t^\perp\boldsymbol{B}_t \in \mathbb{R}^{p\times n}\,, \tag{12}$$

such that the matrices $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ contain the projections of the perturbation over the basis vectors, $\boldsymbol{V}_t^\top\delta\boldsymbol{V}_t = \boldsymbol{A}_t \in \mathbb{R}^{n\times n}$ and $(\boldsymbol{V}_t^\perp)^\top\delta\boldsymbol{V}_t = \boldsymbol{B}_t$. Their exact form is shown next. These are classic results in the theory of matrix perturbations (Stewart & Sun, 1990), but we also include their step-by-step derivation in Appendix E.

$$\begin{aligned}\boldsymbol{A}_t &= \boldsymbol{\Lambda}_t \circ \left(\boldsymbol{\Sigma}_t\boldsymbol{U}_t^\top\delta\boldsymbol{J}_t\boldsymbol{V}_t + \boldsymbol{V}_t^\top\delta\boldsymbol{J}_t^\top\boldsymbol{U}_t\boldsymbol{\Sigma}_t\right) = \\ &= A_{ij} = \frac{1}{\sigma_j^2 - \sigma_i^2}\left[\sum_{k=1}^{n}\sum_{\mu=1}^{p}\left(\sigma_i U_{ik}\delta J_{k\mu}V_{\mu j} + V_{\mu i}\delta J_{k\mu}U_{jk}\sigma_j\right)\right]\end{aligned} \tag{13a}$$

for $i, j = 1, \dots, n$ and $i \neq j$ ($A_{ii} = 0$), where $\circ$ refers to Hadamard product and the spectral gap matrix $\mathbf{\Lambda} = \Lambda_{ij} = (\sigma_j^2 - \sigma_i^2)^{-1}$,

$$\boldsymbol{B}_t = (\boldsymbol{V}_t^{\perp})^{\top} (\delta \boldsymbol{J}_t)^{\top} \boldsymbol{U}_t \boldsymbol{\Sigma}_t^{-1} = B_{\mu j} = \sum_{k=1}^{n} \sum_{\beta=1}^{p} V_{\mu\beta}^{\perp} \delta J_{k\beta} U_{jk} \sigma_k^{-1} \tag{13b}$$

for $j = 1, \dots, n$ and $\mu = 1, \dots, p - n$. The last piece to interpret these expressions is to recall eq. (11): $\delta \boldsymbol{J}_t \approx \eta \dot{\boldsymbol{J}} = -\eta \left(\boldsymbol{J}_t^{\top} \boldsymbol{r}_t + \gamma_{\text{reg}} \boldsymbol{\theta}_t \right)^{\top} \mathbf{H}_t$. Hence, both perturbations are linear in the learning rate and in the Jacobian velocity. The latter depends instantaneously on the residual $\boldsymbol{r}_t$, the Jacobian $\boldsymbol{J}_t$, the network curvature $\mathbf{H}_t$, and — with weight decay — on $\gamma_{\text{reg}} \boldsymbol{\theta}$. Thus $\delta \boldsymbol{V}^{\parallel}$ (proportional to $\boldsymbol{A}_t$) describes infinitesimal rotations within the active subspace, while $\delta \boldsymbol{V}^{\perp}$ (proportional to $\boldsymbol{B}_t$) describes leaking of $\mathcal{A}_\theta$ into the orthogonal complement, a true mutation of $\mathcal{A}_\theta$.

The in-subspace rotation. Particularly, eq. (13a) reveals that $\boldsymbol{A} = -\boldsymbol{A}^{\top}$, i.e., it resembles the infinitesimal generator of a rotation (Hall, 2015), and also – through $\boldsymbol{U}_t^{\top}(\cdot)\boldsymbol{V}_t$ and $\boldsymbol{V}_t^{\top}(\cdot)\boldsymbol{U}_t$ — that the perturbation inside $\mathcal{A}_\theta$ is sensitive to the alignment with both the current basis of $\mathcal{A}_\theta$ and the learning modes (columns of $\boldsymbol{U}_t$) . Moreover, the eigenvalue gaps $(\sigma_j^2 - \sigma_i^2)$ signal that degenerate singular values lead to large rotation and strong reorganization of the basis of $\mathcal{A}_\theta$.

The subspace mixing. Conversely, looking at eq. (13b), we acknowledge that magnitude of the components of $\boldsymbol{B}_t$ depend – through $(\boldsymbol{V}_t^{\perp})^{\top}(\cdot)\boldsymbol{U}_t$ – on the alignment between the current learning modes and the basis directions out of the current active plane; the more aligned these are (assuming that $\delta J_{k\beta} \neq 0$), the more the current j-th basis vector tends to leave $\mathcal{A}_\theta$ in favor of the μ-th basis element of $\boldsymbol{V}^{\perp}$ Remarkably, not only alignment in this way matters, the global contribution depends on how "fast" (large σ_k) or "slow" (small σ_k) the current learning modes (are through the factor σ_k^{-1}); this suggest that slow modes control the evolution of $\mathcal{A}_\theta$.

Grassmannian velocity and generalization. Based on the previous paragraph, we can thus identify $\dot{\boldsymbol{B}}_t = \boldsymbol{B}_t/\eta$ to be the instantaneous velocity of the Grassmannian, since it quantifies the tendency to move from one active subspace to another, i.e., from a given point in $\mathrm{Gr}(n, p)$ at iteration t to the next (see Appendix D for details). Hence, eqs. (11) and (13b) reveal that weight decay is critical because it sustains $\dot{\boldsymbol{B}}_t$ even after memorization, when the residual becomes negligible. Without it, the Grassmannian velocity stalls and the active subspace stops pivoting; with it, the term $\gamma_{\text{reg}} \boldsymbol{\theta}_t$ this keeps $\dot{\boldsymbol{B}}_t$ nonzero, allowing the search for an adequate active plane to continue.
The final question is therefore: *why does $\mathcal{A}_\theta$ drift towards generalization?* We argue that the answer lies in how the population covariance is built into the expected value of the loss of the Hessian after interpolation (once $\boldsymbol{r} \approx 0$). Due to limited space, we develop the argument in Appendix A. In a nutshell, $\mathbb{E}[\boldsymbol{H}_t] \approx \mathbf{\Sigma}_{\text{pop}} = \mathbb{E}[\boldsymbol{J}_t^{\top} \boldsymbol{J}_t]$ (do not confuse the population covariance $\mathbf{\Sigma}_{\text{pop}}$ with $\mathbf{\Sigma}$, the matrix of singular values). This is the mechanism that orients both the parameter updates, via eq. (10), and the Grassmannian velocity, eq. (13b), towards a generalizing active plane. The role of weight decay is to prevent the drift along that direction from stalling. The active plane aligns with the principal directions of the data.

5 Numerical experiments

Diagnostic variables. We will use the following variables to peek into the learning process and understand how the active plane is behaving:

$$C_t = \frac{\|[\boldsymbol{K}_t, \boldsymbol{K}_{t+\Delta}]\|_F}{\|\boldsymbol{K}_t\|_F \, \|\boldsymbol{K}_{t+\Delta}\|_F}, \quad D_t = \frac{\|\boldsymbol{K}_{t+\Delta} - \boldsymbol{K}_t\|_F}{\|\boldsymbol{K}_t\|_F}, \quad E_t = \frac{\|\boldsymbol{P}_{t+\Delta} - \boldsymbol{P}_t\|_F}{\sqrt{2n}}, \tag{14}$$

where $\|\cdot\|_F$ indicates taking the Frobenius norm and Δ is the number of steps between consecutive checkpoints. C_t measures temporal noncommutativity between consecutive checkpoints, D_t measures local tangent-operator magnitude changes (it focuses on global magnitude while C_t would tell us if the eigenframe is changing), and E_t measures drift of the active subspace using consecutive projectors $\boldsymbol{P}_t = \boldsymbol{V}_t \boldsymbol{V}_t^{\top}$ (recall that each projector defines one point in the Grassmannian). One can show (see Appendix C) that $E_t \sim \eta \|\dot{\boldsymbol{B}}_t\|_F$ to first order, so it directly measures the magnitude of the Grassmannian velocity. This last parameter is the most interesting to us (we expect it to be

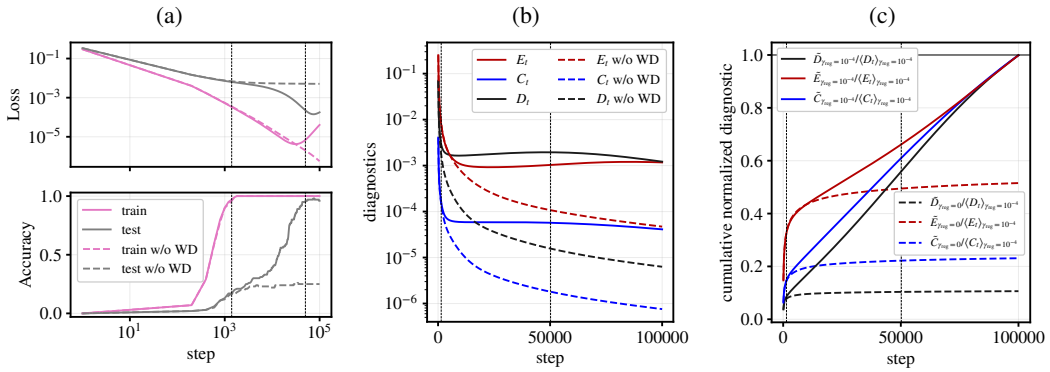


Figure 2: Comparison $\alpha = 2$, with and without weight decay ($\gamma_{\text{reg}} = 10^{-4}$ or $= 0$). Left vertical line indicates end of memorization and beginning of feature learning phase; right one indicates end of feature learning phase (95% test accuracy). (a) Loss and accuracy evolution – notice LU shape (Liu et al., 2023). (b) Diagnostics – see eq. (14) – between consecutive checkpoints. (c) Cumulative diagnostics normalized using results from $\gamma_{\text{reg}} \neq 0$.

meaningfully during the feature learning phase), the other two can be thought as control parameters that reveal if the dynamics in residual space decouple and their intrinsic complexity.

The model. We study the teacher-student regression setup inspired by Liu et al. (2023). A fixed teacher network $\boldsymbol{f}$ is a three-layer MLP with architecture 5–100–100–5 and $\tanh$ activations. The network has $p = 11,205$ parameters. We train for 10^5 steps and checkpoint the network state every $\Delta = 100$. Inputs are drawn i.i.d. from $\mathcal{N}(0, \boldsymbol{I}_5)$, and the student has the same architecture but its initial parameters are rescaled: $\|\boldsymbol{\theta}_0\|_{\text{student}} = \alpha\|\boldsymbol{\theta}_0\|_{\text{teacher}}$, with $\alpha = 2$. The greater the α, the more far-away the student is initialized away from the teacher. Training uses 100 examples and is evaluated on a held-out test set of 100 examples. Since each training datum is a five-entry vector, we have $n = 500$. The original paper employs AdamW with learning rate 3×10^{-4} and weight decay 0.05 for 10^5 steps, and reports delayed generalization. We have made two salient modifications:

- To be consistent with the rest of the paper, we replace AdamW by full-batch gradient descent with a manually controllable L2 penalty $\gamma_{\text{reg}}\|\boldsymbol{\theta}\|^2/2$ applied to all parameters.
- We set $\eta = 0.1$ and $\gamma_{\text{reg}} = 10^{-4}$ (to reproduce grokking).

Experiment I: weight decay sustains E_t and enables generalization. The results for runs with ($\gamma_{\text{reg}} = 10^{-4}$) and without weight decay ($\gamma_{\text{reg}} = 0$) are reported in Figure 2. Figure 2a confirms that the model does require weight decay to generalize despite the training loss (residual) being minimized irrespective of it. Figure 2b is revealing: after and early and fast memorization phase in which the three parameters are meaningful $\gamma_{\text{reg}} = 0$ entails that the tangent kernel magnitude, its eigenbasis and the active subspace rotation, they all freeze rapidly (i.e, C_t, D_t, $E_t \to 0$). Conversely, despite not recovering the magnitude seen during memorization, the three diagnostic parameters settle around relatively constant values when $\gamma_{\text{reg}} = 10^{-4}$, which we interpret as their feature-learning averages; thus, the active subspace pivoting –the discrete trajectory in the Grassmannian, recall eqs. (11), (12) and (13b) – continues until the test loss decreases and test accuracy hits $\approx 100\%$. Finally, Figure 3c reports cumulative diagnostics, defined as the running sum of each diagnostic up to step t, normalized by its average value for the case with weight decay. For example, $\tilde{E}_t = \sum_{s \leq t} E_t$, and then it is normalized as $\tilde{E}_t / \langle E_t \rangle_{\gamma=10^{-4}} / N_{\text{checkpoints}}$. Naturally, all the normalized cumulative diagnostics reach unity at the end of the process for $\gamma_{\text{reg}} = 10^{-4}$, whereas we clearly see that those corresponding to $\gamma_{\text{reg}} = 0$ froze at a value much smaller than one shortly after memorization, particularly the tangent kernel (C_t and D_t). Remarkably, E_t is the last one to converge to its plateau (consistent with the residual keeping the subspace drift for some time).

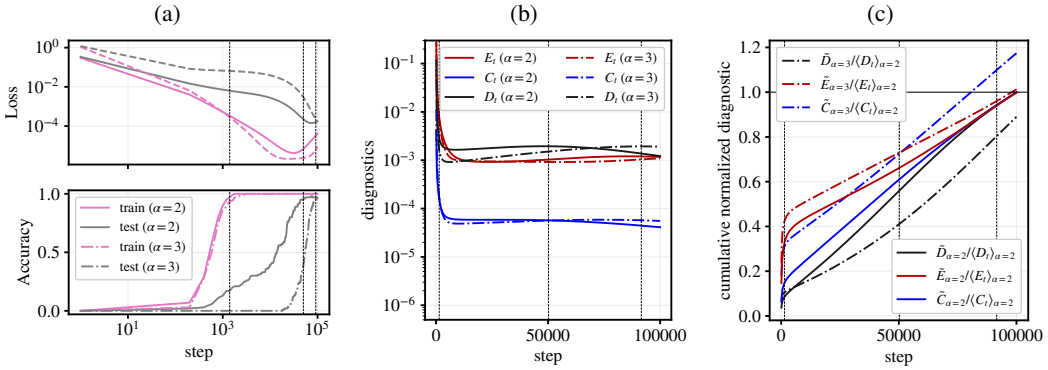


Figure 3: Comparison $\gamma_{\text{reg}} = 10^{-4}$ plus either $\alpha = 2$ or $\alpha = 3$. Leftmost vertical line indicates end of memorization and beginning of feature learning phase; middle one indicates end of feature learning phase (95% test accuracy) for $\alpha = 2$; rightmost one, same but for $\alpha = 3$. (a) Loss and accuracy evolution – notice LU shape (Liu et al., 2023). (b) Diagnostics – see eq. (14) – between consecutive checkpoints. (c) Cumulative diagnostics normalized using results of $\alpha = 2$.

Experiment II: greater α selects a different E_t trajectory. Now the two cases (both with weight decay $\gamma_{\text{reg}} = 10^{-4}$) do generalize within the training window, but the greater α takes many more steps, see Figure 3a. Figure 3b showcases how the diagnostics may vary in precise magnitude, while their trends are very similar qualitative. Figure 3c displays the cumulative diagnostics, normalized this time with the average values at $\alpha = 2$ (e.g., $\langle E_t \rangle_{\alpha=2}$). Interestingly, the training trajectory for $\alpha = 3$ is more strongly non-commutative (normalized cumulative C_t is greater at the end), while the total variation of the tangent kernel (normalized cumulative D_t) is actually smaller; these two facts taken together suggest that the tangent kernel may not change that much in magnitude but its eigenbasis is evolving more meaningfully for the greater α. Remarkably, the cumulative E_t is very similar at $\alpha = 3$ and $\alpha = 2$; this we interpret asthe trajectory in the Grassmannian $\alpha = 3$ taking the same route despite starting at a different point. We provide extra plots (Appendix B), to illustrate how E_t converges with the number of columns of $\boldsymbol{V}$, Figures 4 and 5. and to address the grokking mechanism proposed in Liu et al. (2023) ("LU mechanism"), arguing that it is a concurrent phenomenon (Appendix A), not the ultimate cause, see Figure 6.

6 Final Remarks

We have presented a geometric perspective on feature learning in over-parameterized neural networks, centered on the evolution of the active parameter subspace $\mathcal{A}_\theta = \text{Im}(\boldsymbol{V})$ during training. By deriving the perturbation of the Jacobian eigenbasis, we showed that the loss Hessian acts as the infinitesimal generator of parameter evolution, and that weight decay contributes a persistent term to this generator that remains active even after the training residual has vanished. Numerical experiments on a teacher-student regression task confirm that, without weight decay, the active subspace freezes after memorization and generalization does not occur; with weight decay, the subspace keeps evolving in the Grassmannian until a configuration compatible with the test data is found (generalization). Increasing the student initialization scale α slows down feature learning, the active plane appears to enter the same trajectory towards generalization later on.

Several directions remain open. First, the present experiments are restricted to MSE loss, full-batch, and small MLPs; extending the diagnostic framework to cross-entropy loss, mini-batching, adaptive optimizers, and larger architectures (including transformers) is needed to assess the generality of the mechanism. Second, a more quantitative link between the Grassmannian velocity $\dot{\boldsymbol{B}}_t$ and the population risk would strengthen the theory by providing a better quantitative understanding of how the drift is biased towards generalization. Third, the perturbative analysis assumes small learning-rate steps; connecting this fact to the edge-of-stability regime (Cohen et al., 2021) within the same geometric language is an important next step. Finally, the evolving dynamics in the Grassmannian as $p \to n$ may provide an interpretation for double descent (Belkin et al., 2019).

AI USE STATEMENT

In this work, we used generative AI tools to assist with the following required-disclosure activities: implementing, refactoring, and debugging code (Kimi3 Code); checking mathematical arguments and editing/improving the readability of text (Kimi2.6 and Qwen3.5), as well as researching the state-of-the-art literature (GPT-5.6). We did not use generative AI tools to formulate research hypotheses or to develop the theoretical framework, nor to draft the scientific narrative of the paper; other required-disclosure activities were not applicable to this work. All AI-assisted code was reviewed, executed, and tested by the authors and all literature suggestions were manually verified against original sources. The authors take full responsibility for the final content of this work, including all text, code and figures produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have taken the following steps to ensure the reproducibility of this work. All experimental setup, model architectures, optimization hyperparameters, and evaluation metrics are described in detail in Section 5 . The full training pipeline, including data generation, model initialization, full-batch gradient-descent loop, checkpoint saving, and the post-processing diagnostics used to produce figures, is provided as supplementary material in the Jupyter notebooks `running_saving_template_run.ipynb` and `create_paper_images_run.ipynb`. These notebooks are self-contained and can be run sequentially to reproduce the numerical experiments and all figures that report results. For the theoretical claims, all assumptions are stated formally in Section 3 and ancillary mathematical details are given in Appendices C, D and E.

ACKNOWLEDGMENTS

To be filled after double-blind review.

REFERENCES

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1904.11955.

Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2269–2277. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/baratin21a.html.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. doi: 10.1073/pnas.1903070116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1903070116.

Francesco Bertolotti and Walter Cazzola. A survey on grokking. *ACM Computing Surveys*, 58(13), June 2026. ISSN 0360-0300. doi: 10.1145/3814603. URL https://doi.org/10.1145/3814603.

Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32240–32256, 2022.

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

Jeremy M. Cohen, Simran Kaur, Yuanzhi Li, J. Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2103.00065.

Jeremy M. Cohen, Alex Damian, Ameet Talwalkar, J. Zico Kolter, and Jason D. Lee. Understanding optimization in deep learning with central flows. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.24206.

Simon S. Du, Kangcheng Hou, Ruslan Salakhutdinov, Barnabás Póczos, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M. Roy, and Surya Ganguli. Deep learning versus kernel learning: An empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5850–5861, 2020.

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

Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, volume 222 of *Graduate Texts in Mathematics*. Springer International Publishing, Cham, 2 edition, 2015. ISBN 978-3-319-13466-6. doi: 10.1007/978-3-319-13467-3. URL https://doi.org/10.1007/978-3-319-13467-3.

Sigurdur Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*, volume 34 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2001. ISBN 0-8218-2848-7.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, January 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

Jiří Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: NNGP and NTK for deep attention networks, 2020. URL https://arxiv.org/abs/2006.10540. Presented at the 37th International Conference on Machine Learning.

Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4542–4551. PMLR, 2020. URL https://proceedings.mlr.press/v119/huang20l.html.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pp. 8571–8580, 2018.

Kaiqi Jiang, Jeremy Cohen, and Yuanzhi Li. Understanding the evolution of the neural tangent kernel at the edge of stability. In *Advances in Neural Information Processing Systems*, volume 38, pp. 31996–32036, 2025. URL https://arxiv.org/abs/2507.12837.

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

Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel L. K. Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2012.04728.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8572–8583, 2019.

Zhenyu Liao and Michael W Mahoney. Hessian eigenspectra of more realistic nonlinear models. *Advances in Neural Information Processing Systems*, 34:20104–20117, 2021.

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data, 2023. URL https://arxiv.org/abs/2210.01117.

Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyré. Abide by the law and follow the flow: Conservation laws for gradient flows. In *Advances in Neural Information Processing Systems*, volume 36, pp. 63210–63221, 2023.

Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyré. Transformative or conservative? conservation laws for ResNets and Transformers. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2506.06194.

Sibylle Marcotte, Gabriel Peyré, and Rémi Gribonval. Intrinsic training dynamics of deep neural networks. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/184b2454410a88c280d29aa95fbac6e2-Abstract-Conference.html.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. doi: 10.1073/pnas.1806579115. URL https://doi.org/10.1073/pnas.1806579115.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, 2023. URL https://arxiv.org/abs/2301.05217.

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

Noa Rubin, Inbar Seroussi, and Zohar Ringel. Grokking as a first-order phase transition in two-layer networks, 2024. URL https://arxiv.org/abs/2310.03789.

Jamie Simon, Daniel Kunin, Alexander Atanasov, Enric Boix-Adserà, Blake Bordelon, Jeremy Cohen, Nikhil Ghosh, Florentin Guth, Arthur Jacot, Mason Kamb, Dhruva Karkada, Eric J. Michaud, Berkan Ottlik, and Joseph Turnbull. There will be a scientific theory of deep learning. *arXiv preprint arXiv:2604.21691*, 2026. URL https://arxiv.org/abs/2604.21691.

Gilbert W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, Boston, 1990. ISBN 978-0-12-670230-9.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15071.

Greg Yang and Edward J. Hu. Tensor programs IV: Feature learning in infinite-width neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11727–11737. PMLR, 2021. URL https://proceedings.mlr.press/v139/yang21c.html.

Rubing Yang and Pratik Chaudhari. An effective gram matrix characterizes generalization in deep networks, 2025. URL https://arxiv.org/abs/2504.16450.

A POPULATION ALIGNMENT ARGUMENT

The first part of the argument is acknowledging the relation between the loss of the Hessian, eq. (7), and the empirical population covariance (let us consider weight decay later):

$$\boldsymbol{\Sigma}_{\text{pop}} = \mathbb{E}[(\nabla_\theta \boldsymbol{f})^\top \nabla_\theta \boldsymbol{f}] = \mathbb{E}[\boldsymbol{J}^\top \boldsymbol{J}] \to \mathbb{E}[\boldsymbol{H}] = \mathbb{E}[\boldsymbol{J}^\top \boldsymbol{J}] + \mathbb{E}[\boldsymbol{r}^\top \mathbf{H}]\,, \tag{15}$$

(do not confuse the empirical population covariance $\boldsymbol{\Sigma}_{\text{pop}}$ with $\boldsymbol{\Sigma}$, the matrix of singular values of $\boldsymbol{J}$) thus, after memorization (once $\boldsymbol{r} \approx 0$) we get $\mathbb{E}[\boldsymbol{H}] \approx \boldsymbol{\Sigma}_{\text{pop}}$, i.e., the eigenvectors of the the Hessian tend to align with the principal components of the population covariance. Recall the propagator keeps the eigenbasis of the infinitesimal generator through exponentiation, this means that the chain of parameter-space propagators, eq. (10), becomes

$$\mathbb{E}[\boldsymbol{v}_t] \approx \left[\prod \exp(-\eta \boldsymbol{\Sigma}_{\text{pop}})\right] \boldsymbol{v}_0\,, \tag{16}$$

ignoring the initial memorization steps. This means that the parameter updates are aligned with the population covariance, i.e., the parameters are updated by the underlying data structure. Finally, recall the expression of the Grassmannian velocity, estimated dividing eq. (13b) by η,

$$\dot{\boldsymbol{B}}_t = (\boldsymbol{V}_t^\perp)^\top (\boldsymbol{v}_t^\top \mathbf{H}_t)^\top \boldsymbol{U}_t \Sigma^{-1} \to \mathbb{E}[\dot{\boldsymbol{B}}_t] \propto (\boldsymbol{V}_t^\perp)^\top \left[\prod \exp(-\eta \boldsymbol{\Sigma}_{\text{pop}})\right]\,, \tag{17}$$

which indicates that that if a column of $\boldsymbol{V}_t^\perp$ is orthogonal to the principal directions of $\boldsymbol{\Sigma}_{\text{pop}}$, that velocity component is zero (that direction does not tend to enter the active subspace), whereas if they overlap, we have non-negative velocity component along that direction (that direction tends to enter). That is the mechanism by which the Grassmannian drifts toward directions that capture the population signal implicit in the dataset, i.e., that is how an eigenbasis of the active plane that not just interpolates but generalizes is found.

The prior argument can explain alignment between active plane and the task, but it has not explained why the process can stall and how weight decay can sustain the drift for longer, enabling eventual generalization. To see how this last piece fits, recall that the instantaneous parameter velocity not only can be expressed in terms of prior steps using the propagators, it also has a revealing form in terms of other instantaneous variables, eq. (7), $\boldsymbol{v}_t = \boldsymbol{J}_t^\top \boldsymbol{r}_t - \gamma_{\text{reg}} \boldsymbol{\theta}_t$. Without weight decay, shortly after interpolation $\boldsymbol{v}_t \approx 0$, the network freezes; with weight decay, it becomes purely "radial" $\boldsymbol{v}_t \approx -\gamma_{\text{reg}} \boldsymbol{\theta}_t$, non-zero and aligned with the current parameter: the velocity will keep pointing in the direction that the weights were pointing when the residual vanished. They key is that by that moment, the Grassmannian velocity is already aligned with the population covariance, i.e., the training must continue in the same direction, and weight direction provides the needed impulse. Using an analogy: after memorization the wheel is steering in the right direction, we just need to keep enough "fuel" to keep moving along the already-fixed direction.

This interpretation is somewhat consistent with the one proposed by Liu et al. (2023): weight decay sustains the parameter evolution radially, but we claim that grokking does not happen because there is a special "goldilocks" parameter norm that ensures generalization; rather, there is an active plane that is able to generalize the model response, weight decay makes it possible to attain it.

We believe out interpretation is strongly backed by the results in Section 5: Figures 2 and 3 show that the subspace drift stalls without weight decay ($\dot{\boldsymbol{B}} \approx 0$, $\boldsymbol{v} \approx 0$, and $E_t \to 0$), whereas weight decay ensures that all these parameter settle to non-negative, relatively stable value. The latter is consistent with $|\boldsymbol{v}| \approx \gamma_{\text{reg}} |\boldsymbol{\theta}|$ and $E_t \to \text{constant}$.

B EXTRA DIAGNOSTICS TEACHER-STUDENT EXAMPLE

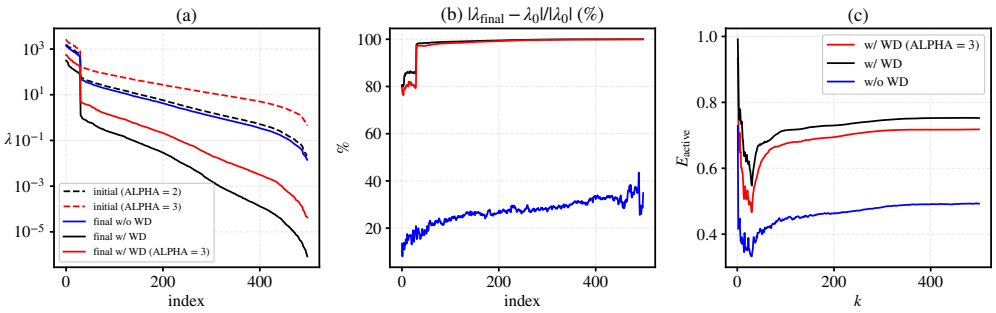


Figure 4: Spectrum evolution and influence over Grassmannian velocity. (a) Spectra for the three canonical cases (ordered by eigenvalue magnitude) studied in Section 5 at both initilization and at the end of training. Both cases weight decay evolve (decrease substantually), without weight decay they remain close to initialization. (b) Same as (a) but in terms percentual change. With weight decay, higher modes decrease so substantially over training that change $\approx 100\%$ (tiny final magnitude compared with initial one). Conversely, change is much more balanced across the spectrum without weight decay (clear $\sim 30\%$ on average). (c) Evolution of $E_{\text{active}} = \|\boldsymbol{P}_{\text{final}} - \boldsymbol{P}_0\|/\sqrt{2k}$ as a function of number of columns in $\boldsymbol{V}$ retained. Keeping only first modes can lead to large errors.

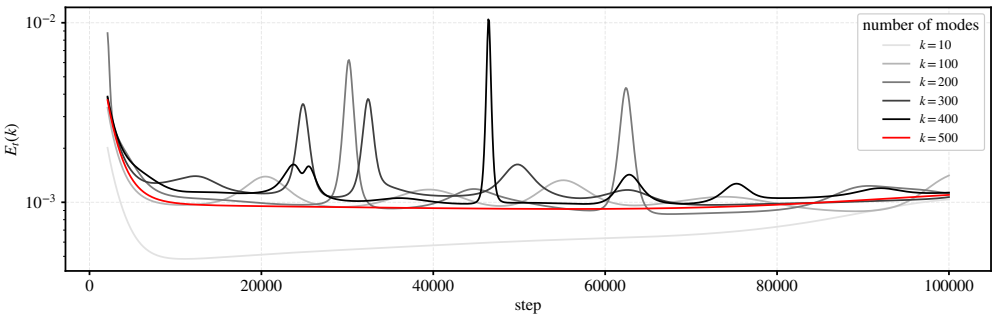


Figure 5: For $\gamma_{\text{reg}} = 10^{-4}$ and $\alpha = 3$: convergence of E_t as more columns of $\boldsymbol{V}$ are added (results reported in the paper $k = 500$). The overall trend requires at least first 100, but the fact that there can be intense oscillations points to how sensitive this parameter can be to slow learning modes.

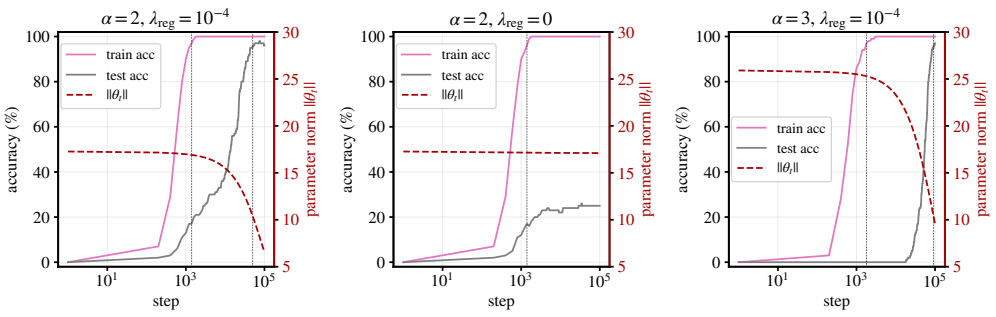


Figure 6: Evolution of the norm of the parameters for the three canonical cases. Without weight decay, the process stalls but it still reaches a 20% accuracy, proving that the process can run for some time even in the absence of weight decay (powered by the component of the Hessian proportional to the residual in $\boldsymbol{H}_{\text{net}}$). Thus, weight decay helps generalization, but it is not the ultimate cause; a model may therefore generalize out of the "goldilocks zone" (Liu et al., 2023).

C CONNECTING E_t TO THE ACTIVE SUBSPACE PERTURBATION

Recall the active-subspace projector $\boldsymbol{P}_t = \boldsymbol{V}_t\boldsymbol{V}_t^\top$ and the first-order eigenbasis perturbation, eq. (12),

$$\delta\boldsymbol{V}_t = \boldsymbol{V}_t\boldsymbol{A}_t + \boldsymbol{V}_t^\perp\boldsymbol{B}_t\,, \qquad \boldsymbol{A}_t = -\boldsymbol{A}_t^\top\,.$$

To first order,

$$\begin{aligned}\boldsymbol{P}_{t+\Delta} &= \boldsymbol{V}_{t+\Delta}\boldsymbol{V}_{t+\Delta}^\top \\ &\approx \big[\boldsymbol{V}_t(\boldsymbol{I}+\boldsymbol{A}_t) + \boldsymbol{V}_t^\perp\boldsymbol{B}_t\big]\big[(\boldsymbol{I}+\boldsymbol{A}_t)^\top\boldsymbol{V}_t^\top + \boldsymbol{B}_t^\top(\boldsymbol{V}_t^\perp)^\top\big] \\ &\approx \boldsymbol{V}_t\boldsymbol{V}_t^\top + \boldsymbol{V}_t\boldsymbol{B}_t^\top(\boldsymbol{V}_t^\perp)^\top + \boldsymbol{V}_t^\perp\boldsymbol{B}_t\boldsymbol{V}_t^\top,\end{aligned}$$

where we used $(\boldsymbol{I}+\boldsymbol{A}_t)(\boldsymbol{I}+\boldsymbol{A}_t)^\top \approx \boldsymbol{I}$ and dropped all $\mathcal{O}(\|\boldsymbol{A}_t\|^2)$, $\mathcal{O}(\|\boldsymbol{B}_t\|^2)$, and $\mathcal{O}(\boldsymbol{A}_t\boldsymbol{B}_t)$ terms. Hence, in the basis $[\boldsymbol{V}_t, \boldsymbol{V}_t^\perp]$.

$$\delta\boldsymbol{P}_t = \boldsymbol{P}_{t+\Delta} - \boldsymbol{P}_t \approx \begin{bmatrix}\boldsymbol{0} & \boldsymbol{B}_t^\top \\ \boldsymbol{B}_t & \boldsymbol{0}\end{bmatrix}$$

The Frobenius norm is therefore

$$\begin{aligned}\|\delta\boldsymbol{P}_t\|_F^2 &= \operatorname{trace}\big((\delta\boldsymbol{P}_t)^\top\delta\boldsymbol{P}_t\big) \\ &\approx \operatorname{trace}\big(\boldsymbol{V}_t\boldsymbol{B}_t^\top\boldsymbol{B}_t\boldsymbol{V}_t^\top\big) + \operatorname{trace}\big(\boldsymbol{V}_t^\perp\boldsymbol{B}_t\boldsymbol{B}_t^\top(\boldsymbol{V}_t^\perp)^\top\big) = 2\,\|\boldsymbol{B}_t\|_F^2\,,\end{aligned}$$

where we exploited that $\boldsymbol{V}$ and $\boldsymbol{V}^\perp$ contain vectors that form an orthogonal basis, thus the trace is indifferent to them, and the two final addends are equal by virtue of the cyclicity of the trace. So the diagnostic E_t introduced in eq. (14) satisfies

$$E_t = \frac{\|\boldsymbol{P}_{t+\Delta} - \boldsymbol{P}_t\|_F}{\sqrt{2n}} \approx \frac{\|\boldsymbol{B}_t\|_F}{\sqrt{n}} \approx \eta\Delta\frac{\|\dot{\boldsymbol{B}}_t\|_F}{\sqrt{n}}\,.$$

Thus, to first order, E_t measures only the out-of-subspace mixing block $\boldsymbol{B}_t$; the in-subspace rotation $\boldsymbol{A}_t$ drops out because it preserves the projector. One last comment: we chose to normalize E_t by $\sqrt{2n}$ because in our example we can efficiently compute all the vectors in $\boldsymbol{V}$ (there are only $n = 500$); we foresee that for larger problems only the first $k < n$ modes would be necessary, thus $E_{k,t}$ would be normalized by $\sqrt{2k}$.

D LIE-THEORETICAL FRAMEWORK

The Jacobian $\boldsymbol{J}$ is the object that influences the group-theoretical structure of both the generators of the dynamics in either residual and parameter space. It controls the former via $\boldsymbol{K} = \boldsymbol{J}\boldsymbol{J}^\top$, and defines the active plane of the latter via $\mathcal{A}_\theta = \mathrm{Im}\boldsymbol{J}^\top$.

Also, for different reasons, the dynamics in either space carry the features of symmetric spaces. All groups we consider are always over the field of real numbers, $\mathbb{R}$.

D.1 STRUCTURE IN PARAMETER SPACE

At any training step the SVD of the Jacobian, $\boldsymbol{J} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, gives an orthonormal basis $\boldsymbol{V} \in \mathbb{R}^{p\times n}$ for $\mathcal{A}_\theta$. The set of all such rank-n subspaces is the Grassmannian $\mathrm{Gr}(n,p) \simeq O(p)/(O(n)\times O(p-n))$ (Helgason, 2001), where the modulo operation reflects that a different choice of basis inside $\mathcal{A}_\theta$ or inside its orthogonal complement does not change $\mathrm{Gr}(n,p)$.

To see the geometry, consider the Cartan involution

$$\vartheta(\boldsymbol{X}) = \boldsymbol{Q}\boldsymbol{X}\boldsymbol{Q}, \qquad \boldsymbol{Q} = \begin{bmatrix}\boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_{p-n}\end{bmatrix}, \tag{18}$$

which induces the Cartan decomposition of the Lie algebra ($\mathfrak{k}$ associated to -1, $\mathfrak{m}$ to +1)

$$\mathsf{so}(p) = \underbrace{(\mathsf{so}(n)\oplus\mathsf{so}(p-n))}_{\mathfrak{k}}\oplus\mathfrak{m}\,. \tag{19}$$

This splitting can be interpreted as follows: $\mathfrak{k}$ generates rotations that stay inside the active subspace or inside its complement (a local reparametrization), while $\mathfrak{m}$ generates rotations that mix the two. Only $\mathfrak{m}$ moves the point on the Grassmannian. If we write an infinitesimal perturbation of $\boldsymbol{V}$ as in eq. (12), then $\boldsymbol{A} \in \mathfrak{k}$ is in-space reparametrization and $\boldsymbol{B} \in \mathfrak{m}$ is the physical Grassmannian velocity, i.e., the genuine change from one point in $\mathrm{Gr}(n,p)$ to the next. The diagnostic E_t used in the experiments is precisely the norm of $\boldsymbol{B}$.

The commutation relations that follow from eq. (19) sharpen the picture by telling how the successive active subspace pivots are composed:

$$[\mathfrak{k},\mathfrak{k}] \subset \mathfrak{k}\,, \qquad [\mathfrak{k},\mathfrak{m}] \subset \mathfrak{m}\,, \qquad [\mathfrak{m},\mathfrak{m}] \subset \mathfrak{k}\,. \tag{20}$$

where the first relation means that two in-space mixings compose into another in-space mixing. The third means that two successive subspace rotations can produce an in-space mixing. The second says that an in-space mixing applied to a subspace-rotation direction yields another subspace-rotation direction.

Weight decay enters the parameter-space dynamics through the Hessian $\boldsymbol{H} = \boldsymbol{G} + \boldsymbol{H}_{\mathrm{net}} + \gamma_{\mathrm{reg}}\boldsymbol{I}_p$. The term $\gamma_{\mathrm{reg}}\boldsymbol{I}_p$ belongs to the center (the part the algebra that commutes with everything else): it scales every direction in $\mathbb{R}^p$ uniformly and therefore does not alter the Grassmannian decomposition. What it does change is the magnitude of the generator.

In summary, $\mathrm{Gr}(n,p)$ is a symmetric space whose tangent directions are exactly the off-diagonal blocks $\mathfrak{m}$. We posit that feature learning/generalization is motion along those tangent directions, driven by the Hessian, and weight decay controls whether that motion continues after the training residual has vanished.

D.2 Structure in residual space

The neural tangent kernel $\boldsymbol{K} = \boldsymbol{J}\boldsymbol{J}^\top$ is a symmetric positive semidefinite matrix in $\mathbb{R}^{n\times n}$, so $\exp(-t\boldsymbol{K})$ belong in $\mathrm{SPD}(n)$, the set of positive definite matrices, which is again a symmetric space. Indeed, the Lie algebra of the general linear group decomposes as $\mathsf{gl}(n,\mathbb{R}) = \mathbb{R}\,\boldsymbol{I}_n \oplus \mathsf{sl}(n,\mathbb{R})$, and the special linear algebra further decomposes under the involution $\vartheta(\boldsymbol{X}) = -\boldsymbol{X}^\top$ as $\mathsf{sl}(n,\mathbb{R}) = \mathsf{so}(n) \oplus \mathrm{Sym}_0(n)$. At the group level this gives $\mathrm{SPD}(n) \simeq \mathbb{R}^+ \times SL(n,\mathbb{R})/SO(n)$: a positive scale factor times a shape deformation of unit determinant.

In practice, this manifests as follows. let us decompose $\boldsymbol{K}$ into volumetric and deviatoric parts: $\boldsymbol{K} = \mathrm{trace}(\boldsymbol{K})/n\,\boldsymbol{I} + (\boldsymbol{K} - \mathrm{trace}(\boldsymbol{K})/n\,\boldsymbol{I}) = c/n\,\boldsymbol{I} + \bar{\boldsymbol{K}}$; note that $\mathrm{trace}(\bar{\boldsymbol{K}}) = 0$ and therefore

$$\exp\left(-\eta\boldsymbol{K}_0\right) = e^{-\eta c^{(0)}} \exp\left(-\eta\bar{\boldsymbol{K}}_0\right) = e^{-\eta c^{(0)}} \left(\boldsymbol{U}_0\,\mathrm{diag}\left(e^{-\eta\mu_1^{(0)}} \dots e^{-\eta\mu_n^{(0)}}\right)\boldsymbol{U}_0^\top\right)\,, \tag{21}$$

where the superindices in parenthesis are meant to identify eigenvalues across iterations. So each propagator contains a component of pure compression (decreasing all residuals by the same factor $e^{-\eta c}$) plus anisotropic fluctuations along different residual components (μ_i). In other words, $\lambda_i^{(t)} = c^{(t)} + \mu_i^{(t)} \geq 0$, but this does not prevent some μ_i from being negative: since $\mathrm{trace}(\bar{\boldsymbol{K}}) = \sum_{i=1}^n \mu_i = 0$, there must be both positive and negative μ_i eventually canceling each other.

The diagonal matrix can therefore be interpreted as determining which modes in the NTK spectrum adapt faster. Notice also that the matrices $\boldsymbol{U}$ are legit rotation matrices in $\mathbb{R}^n$ ($\boldsymbol{U} \in SO(n)$, hence isometries): they redistribute the total amplitude in the modes directly to the residuals without distorting the overall magnitude.

Forming the chain of propagator and assuming constant learning rate, we can write the residual after $T+1$ steps as in eq. (9):

$$\begin{aligned}
\boldsymbol{r}_{T+1} &= \left[\prod_{t=0}^{T} \exp(-\eta\boldsymbol{K}_t)\right]\boldsymbol{r}_0 = \left[\exp(-\eta\boldsymbol{K}_T)\exp(-\eta\boldsymbol{K}_{T-1})\dots\exp(-\eta\boldsymbol{K}_0)\right]\boldsymbol{r}_0 \\
&= e^{-\eta c^{(T+1)}}\left[\boldsymbol{U}_T\exp(-\eta\boldsymbol{\Lambda}_T)\underbrace{\boldsymbol{U}_T^\top\boldsymbol{U}_{T-1}}_{\boldsymbol{R}_T}\exp(-\eta\boldsymbol{\Lambda}_{T-1})\boldsymbol{U}_{T-1}^\top\dots\boldsymbol{U}_0\exp(-\eta\boldsymbol{\Lambda}_0)\boldsymbol{U}_0^\top\right]\boldsymbol{r}_0\,,
\end{aligned} \tag{22}$$

where $c^{(T+1)} = \sum_{t=0}^{T} c^{(t)}$ is the accumulated compression of residuals after T steps, and we have highlighted one of the rotation matrices $\boldsymbol{R}_t = \boldsymbol{U}_t^\top \boldsymbol{U}_{t-1} \in SO(n)$, which transports the eigenvector basis at $t-1$ to that of the next propagator at iteration t. The fact that these changes of basis are isometries means that they do not modify the spectral invariant content.

The payoff of this decomposition is twofold. First, the maximal torus factors $\exp(-\eta \boldsymbol{\Lambda}_t)$ contract or expand fixed learning modes, while the rotations $\boldsymbol{R}_t$ mix slow and fast modes at each step. So even if a single $\boldsymbol{K}_t$ has a clean spectral ordering, the chain redistributes that structure across modes. This is how noncommutativity enters during training: the frozen NTK picture (Jacot et al., 2018) assumes $\boldsymbol{U}_t^\top \boldsymbol{U}_{t-1} = \boldsymbol{I}_n$, so that all terms commute.

Second, because each $\boldsymbol{K}_t$ is positive semidefinite, the accumulated center factor $e^{-\eta c^{(T+1)}}$ is a contraction. Even though individual shape deformations can have negative eigenvalues μ_i, the full eigenvalues $\lambda_i^{(t)} = c^{(t)} + \mu_i^{(t)}$ remain nonnegative, and the overall product stays positive definite. Thus the residual cannot blow up in the exact exponential flow; any observed increase must therefore be an artifact of the first-order approximation $\boldsymbol{\theta}_{t+1} \approx \boldsymbol{\theta}_t - \eta \nabla_\theta \mathcal{L}$ used in discrete gradient descent.

E Deriving the Grassmannian velocity

How does the matrix $\boldsymbol{V}_t \in \mathbb{R}^{p \times n}$, recall eq. (4), change during training? See that its orthogonal complement, $\boldsymbol{V}_t^\perp$, spans the rest of possible directions in $\mathbb{R}^{p \times p}$, we can say $\mathrm{span}(\boldsymbol{Q}) = \mathrm{span}([\boldsymbol{V}_t, \boldsymbol{V}_t^\perp]) = \mathbb{R}^p$.

Then, the rate of change of matrix $\boldsymbol{V}$ can be written as

$$\dot{\boldsymbol{V}}_t \in \mathbb{R}^{p \times n} \to \dot{\boldsymbol{V}}_t = \dot{\boldsymbol{V}}_t^{\parallel} + \dot{\boldsymbol{V}}_t^{\perp} = \boldsymbol{V}_t \boldsymbol{A}_t + \boldsymbol{V}_t^\top \boldsymbol{B}_t \,, \tag{23}$$

i.e, it can be uniquely decomposed in the base of $\mathbb{R}^p$ formed by $\boldsymbol{V}_t$ and $\boldsymbol{V}_t$ $\perp$. The matrix $\boldsymbol{A}_t = \boldsymbol{V}^\top \dot{\boldsymbol{V}} \in \mathbb{R}^{n \times n}$ represents rotation of the current subspace while $\boldsymbol{B}_t = (\boldsymbol{V}^\perp)^\top \dot{\boldsymbol{V}} \in \mathbb{R}^{(p-n) \times n}$ represents incipient mixing of active and null directions.

The right singular vectors $\boldsymbol{V}_t$ are the eigenvectors of the matrix $\boldsymbol{G}_t = \boldsymbol{J}_t^\top \boldsymbol{J}_t$, meaning:

$$\boldsymbol{G}_t \boldsymbol{V}_t = \boldsymbol{V}_t \boldsymbol{\Lambda}_t, \quad \text{where } \boldsymbol{\Lambda}_t = \boldsymbol{\Sigma}_t^2 \,.$$

Differentiating this equation with respect to time:

$$\dot{\boldsymbol{G}}_t \boldsymbol{V}_t + \boldsymbol{G}_t \dot{\boldsymbol{V}}_t = \dot{\boldsymbol{V}}_t \boldsymbol{\Lambda}_t + \boldsymbol{V}_t \dot{\boldsymbol{\Lambda}}_t.$$

Substitute the decomposition eq. (23):

$$\dot{\boldsymbol{G}}_t \boldsymbol{V}_t + \boldsymbol{G}_t (\boldsymbol{V}_t \boldsymbol{A}_t + \boldsymbol{V}_t^\perp \boldsymbol{B}_t) = (\boldsymbol{V}_t \boldsymbol{A}_t + \boldsymbol{V}_t^\perp \boldsymbol{B}_t) \boldsymbol{\Lambda}_t + \boldsymbol{V}_t \dot{\boldsymbol{\Lambda}}_t.$$

Recall that $\boldsymbol{G}_t \boldsymbol{V}_t = \boldsymbol{V}_t \boldsymbol{\Lambda}_t$ and $\boldsymbol{G}_t \boldsymbol{V}_t^\perp = 0$ (since $\boldsymbol{V}_t^\perp$ spans the null space of $\boldsymbol{J}_t$). The equation simplifies to:

$$\dot{\boldsymbol{G}}_t \boldsymbol{V}_t + \boldsymbol{V}_t \boldsymbol{\Lambda}_t \boldsymbol{A}_t = \boldsymbol{V}_t \boldsymbol{A}_t \boldsymbol{\Lambda}_t + \boldsymbol{V}_t^\perp \boldsymbol{B}_t \boldsymbol{\Lambda}_t + \boldsymbol{V}_t \dot{\boldsymbol{\Lambda}}_t \,. \tag{24}$$

To isolate $\boldsymbol{B}_t$, we project this equation onto the null space by left-multiplying by $(\boldsymbol{V}_t^\perp)^\top$:

$$(\boldsymbol{V}_t^\perp)^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t + \underbrace{(\boldsymbol{V}_t^\perp)^\top \boldsymbol{V}_t}_{0} \boldsymbol{\Lambda}_t \boldsymbol{A}_t = \underbrace{(\boldsymbol{V}_t^\perp)^\top \boldsymbol{V}_t}_{0} \boldsymbol{A}_t \boldsymbol{\Lambda}_t + \underbrace{(\boldsymbol{V}_t^\perp)^\top \boldsymbol{V}_t^\perp}_{\boldsymbol{I}} \boldsymbol{B}_t \boldsymbol{\Lambda}_t + \underbrace{(\boldsymbol{V}_t^\perp)^\top \boldsymbol{V}_t}_{0} \dot{\boldsymbol{\Lambda}}_t.$$

This yields the exact relation:

$$(\boldsymbol{V}_t^\perp)^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t = \boldsymbol{B}_t \boldsymbol{\Lambda}_t.$$

Assuming that $\boldsymbol{\Lambda}_t = \boldsymbol{\Sigma}_t^2$ is full rank and therefore invertible, solve for $\boldsymbol{B}_t$:

$$\begin{aligned}
\boldsymbol{B}_t &= (\boldsymbol{V}_t^\perp)^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t \boldsymbol{\Lambda}_t^{-1} = (\boldsymbol{V}_t^\perp)^\top \left(\dot{\boldsymbol{J}}_t^\top \boldsymbol{J}_t + \boldsymbol{J}_t^\top \dot{\boldsymbol{J}}_t \right) \boldsymbol{V}_t \boldsymbol{\Lambda}_t^{-1} \\
&= (\boldsymbol{V}_t^\perp)^\top \dot{\boldsymbol{J}}_t^\top \underbrace{\boldsymbol{J}_t \boldsymbol{V}_t}_{\boldsymbol{U}_t \boldsymbol{\Sigma}_t} \boldsymbol{\Lambda}_t^{-1} + \underbrace{(\boldsymbol{V}_t^\perp)^\top \boldsymbol{J}_t^\top}_{(\boldsymbol{J}_t \boldsymbol{V}_t^\perp)^\top = 0} \dot{\boldsymbol{J}}_t \boldsymbol{V}_t \boldsymbol{\Lambda}_t^{-1} \\
&= (\boldsymbol{V}_t^\perp)^\top \dot{\boldsymbol{J}}_t^\top \boldsymbol{U}_t \boldsymbol{\Sigma}_t^{-1} \,.
\end{aligned}$$

To find also the expression of $\boldsymbol{A}_t$, we project eq. (24) onto the active subspace by left-multiplying by $\boldsymbol{V}_t^\top$:

$$\boldsymbol{V}_t^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t + \underbrace{\boldsymbol{V}_t^\top \boldsymbol{V}_t}_{\boldsymbol{I}} \boldsymbol{\Lambda}_t \boldsymbol{A}_t = \underbrace{\boldsymbol{V}_t^\top \boldsymbol{V}_t}_{\boldsymbol{I}} \boldsymbol{A}_t \boldsymbol{\Lambda}_t + \underbrace{\boldsymbol{V}_t^\top \boldsymbol{V}_t^\perp}_{0} \boldsymbol{B}_t \boldsymbol{\Lambda}_t + \underbrace{\boldsymbol{V}_t^\top \boldsymbol{V}_t}_{\boldsymbol{I}} \dot{\boldsymbol{\Lambda}}_t \,,$$

which yields the relation

$$\boldsymbol{V}_t^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t = \boldsymbol{A}_t \boldsymbol{\Lambda}_t - \boldsymbol{\Lambda}_t \boldsymbol{A}_t + \dot{\boldsymbol{\Lambda}}_t \,. \tag{25}$$

Because $\boldsymbol{A}_t$ is necessarily skew-symmetric, its diagonal entries vanish. Consequently, taking the diagonal of eq. (25) gives the eigenvalue rates:

$$\dot{\boldsymbol{\Lambda}}_t = \mathrm{diag}\Big(\boldsymbol{V}_t^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t\Big) \,. \tag{26}$$

Putting it back in the prior equation, the off-diagonal entries give the in-subspace rotation (active-mode mixing without leakage to the frozen directions):

$$(\boldsymbol{A}_t)_{ij} = \frac{\Big(\boldsymbol{V}_t^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t\Big)_{ij}}{\lambda_j - \lambda_i}, \qquad i \neq j \,. \tag{27}$$

Remarkably, $\boldsymbol{A}_t$ is controlled by off-diagonal perturbations divided by eigenvalue differences (the "spectral gaps").

Substituting $\dot{\boldsymbol{G}}_t = \dot{\boldsymbol{J}}_t^\top \boldsymbol{J}_t + \boldsymbol{J}_t^\top \dot{\boldsymbol{J}}_t$ into $\boldsymbol{V}_t^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t$ gives

$$\begin{aligned} \boldsymbol{V}_t^\top \dot{\boldsymbol{G}}_t \boldsymbol{V}_t &= (\dot{\boldsymbol{J}}_t \boldsymbol{V}_t)^\top \boldsymbol{U}_t \boldsymbol{\Sigma}_t + \boldsymbol{\Sigma}_t \boldsymbol{U}_t^\top \dot{\boldsymbol{J}}_t \boldsymbol{V}_t \\ &= \boldsymbol{M}_t^\top \boldsymbol{\Sigma}_t + \boldsymbol{\Sigma}_t \boldsymbol{M}_t, \end{aligned} \tag{28}$$

where we introduced the "projected Jacobian perturbation"

$$\boldsymbol{M}_t = \boldsymbol{U}_t^\top \dot{\boldsymbol{J}}_t \boldsymbol{V}_t \in \mathbb{R}^{n \times n}. \tag{29}$$

Inserting eq. (28) into eq. (27) yields

$$(\boldsymbol{A}_t)_{ij} = \frac{\big(\boldsymbol{M}_t^\top \boldsymbol{\Sigma}_t + \boldsymbol{\Sigma}_t \boldsymbol{M}_t\big)_{ij}}{\sigma_j^2 - \sigma_i^2}, \qquad i \neq j. \tag{30}$$

Finally, using the network Hessian tensor contraction $\dot{\boldsymbol{J}}_t = \boldsymbol{v}_t^\top \mathbf{H}$, the projected perturbation becomes

$$\boldsymbol{M}_t = \boldsymbol{U}_t^\top (\boldsymbol{v}_t^\top \mathbf{H}) \boldsymbol{V}_t. \tag{31}$$