The Riemannian Geometry Associated to Gradient Flows of Linear Convolutional Networks

El Mehdi Achour
UM6P College of Computing
Rabat, Morocco

Kathlén Kohn
KTH Royal Inst. of Technology
& Digital Futures, Stockholm, Sweden

Holger Rauhut
LMU Dept. of Math & MCML
Munich, Germany

Abstract

We study geometric properties of the gradient flow for learning deep linear convolutional networks. For linear fully connected networks, it has been shown recently that the corresponding gradient flow on parameter space can be written as a Riemannian gradient flow on function space (i.e., on the product of weight matrices) if the initialization satisfies a so-called balancedness condition. We establish that the gradient flow on parameter space for learning linear convolutional networks can be written as a Riemannian gradient flow on function space regardless of the initialization. This result holds for D-dimensional convolutions with $D \geq 2$, and for $D = 1$ it holds if all so-called strides of the convolutions are greater than one. The corresponding Riemannian metric depends on the initialization. Our results also apply to shallow ReLU networks.

1 INTRODUCTION

Convergence properties of (stochastic) gradient descent schemes for learning deep neural networks are challenging to understand mainly due to the nonconvexity of the corresponding loss functions. In recent years progress could be achieved in specific simplified settings Arora et al. (2018); Jacot et al. (2018); Bah et al. (2022); Kohn et al. (2022). One line of work considers deep linear networks Arora et al. (2018); Bah et al. (2022); Kohn et al. (2022, 2024); Nguegnang et al. (2024). While linear networks represent linear functions so that their expressivity is limited, convergence properties of training algorithms are still highly non-trivial and thus interesting to investigate. Such investigations should be seen as a first step before passing to networks with nonlinear activation, as linear networks are common building blocks of nonlinear architectures (e.g., of ReLU networks; see Sec. 5). For fully connected linear networks and the square loss it was shown in Bah et al. (2022) that (essentially) the corresponding gradient flow converges to a global minimizer for almost all initializations; this result was extended to gradient descent under a suitable upper bound on the step sizes Nguegnang et al. (2024). Additionally, Bah et al. (2022) revealed that under so-called balanced initialization, the flow of the network, i.e., the product of the weight matrices, is independent of its parametrization and follows a Riemannian gradient flow with respect to a suitable Riemannian metric. Expressed differently, the so-called neural tangent kernel (determining the Riemannian metric) is independent of the parameterization. This is a weaker property of the neural tangent kernel (NTK) than the well-known one that in the limit of infinite width the NTK becomes constant Jacot et al. (2018) (for random initialization) – of course, it is then in particular also independent of the parameterization. But in contrast, the mentioned result of Arora et al. (2018); Bah et al. (2022) holds for any finite-width network.

In this paper, we study similar geometric and convergence properties for linear **convolutional** neural networks, i.e., we pass from the fully connected case to certain structured neural networks. More precisely, we study the NTK's sole dependence on the network as a function (instead of its parameters). Each layer in such a network is a convolution on D-dimensional

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

signals that is given by its filter w_l, which is a D-dimensional tensor, and its stride $s_l \in \mathbb{Z}_{>0}^D$. Given a D-dimensional input tensor, the convolution computes the inner product of various parts of the input tensor with the filter w_l, by moving the filter through the input tensor with step size $s_{i,d}$ in the d-th direction. For a formal definition, see Sections 2 and 3. For a linear convolutional network with H layers, the following algebraic scalars stay constant throughout gradient flow (Kohn et al., 2022, Prop. 5.13):

$$\delta_l := \|w_{l+1}\|_F^2 - \|w_l\|_F^2, \text{ for } l = 1, \dots, H-1. \quad (1)$$

If all $\delta_1, \dots, \delta_{H-1}$ are zero, we say that the filters $(w_1, \dots, w_H)$ are balanced in analogy to the fully connected case Arora et al. (2018); Bah et al. (2022). One of our main findings is that in most cases the corresponding NTK and thereby the associated Riemannian metric is independent of the parameterization of the convolutional network, regardless whether the initialization is balanced or not. The NTK only depends on the δ_l (at initialization).

1.1 Gradient flow for fully-connected and convolutional linear networks

For a fixed neural network architecture, we denote by $\mu : \Theta \to \mathcal{M}$ the network parametrization map that assigns to parameters $\theta \in \Theta$ the end-to-end function $\mu_\theta := \mu(\theta)$. The image of μ is the neuromanifold $\mathcal{M}$ of the network architecture.

We consider both fully connected and convolutional linear neural networks:

- **Fully connected linear neural networks** take the form

  $$\mu_\theta : \mathbb{R}^{d_0} \to \mathbb{R}^{d_H}, \mu_\theta(X) = W_H \cdots W_1 X,$$

  where $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and $\theta = (W_1, \dots, W_H)$, i.e, $\mu_\theta = W_H \cdots W_1$. Setting $r = \min\{d_0, \dots, d_H\}$, the neuromanifold $\mathcal{M}$ is the manifold of $d_H \times d_0$ matrices of rank at most r.

- **A convolutional linear neural network** is a composition of linear convolutions. On one-dimensional signals, the convolution in the l-th layer is given by a filter $w_l \in \mathbb{R}^{k_l}$ and a stride $s_l \in \mathbb{Z}_{>0}$, $l = 1, \dots, H$. It is a linear map $\alpha_{w_l, s_l} : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$ with output dimension d_l and input dimension $d_{l-1} = s_l(d_l - 1) + k_l$ that sends a vector $X \in \mathbb{R}^{d_{l-1}}$ to the vector with i-th entry for $i = 0, 1, \dots, d_l - 1$:

  $$(\alpha_{w_l, s_l}(X))_i = \sum_{j=0}^{k_l - 1} w_{l,j} X_{i s_l + j}. \quad (2)$$

  The network composes H such convolutions, i.e., $f = \alpha_{w_H, s_H} \circ \cdots \circ \alpha_{w_1, s_1}$, which is again a convolution of stride $s_1 \cdots s_H$ and filter size $k := k_1 + \sum_{l=2}^{H} (k_l - 1) \prod_{i=1}^{l-1} s_i$. Hence, the network parametrization map

  $$\mu : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_H} \to \mathbb{R}^k$$

  assigns to a filter tuple $(w_1, \dots, w_H)$ the filter of the end-to-end convolution.

  Convolutions and linear convolutional neural networks on D-dimensional signals are defined analogously; see (10) below.

Given training data $\mathcal{D} = \{(X_i, Y_i) \in \mathbb{R}^{d_0} \times \mathbb{R}^{d_H} \mid i = 1, 2, \dots, N\}$, one aims at learning a hypothesis function $\alpha : \mathbb{R}^{d_0} \to \mathbb{R}^{d_H}$ (e.g. a neural network) such that $\alpha(X_i) \approx Y_i$ for $i = 1, \dots, N$. Given a (differentiable) loss function $\ell : \mathbb{R}^{d_H} \times \mathbb{R}^{d_H} \to \mathbb{R}$, the empirical risk of α is defined as

$$\ell_{\mathcal{D}}(\alpha) = \sum_{i=1}^{N} \ell(\alpha(X_i), Y_i).$$

Below we will exclusively work with the square loss

$$\ell(\alpha(X_i), Y_i) = \|\alpha(X_i) - Y_i\|_2^2.$$

The empirical square loss in parameter space is then

$$\mathcal{L}_{\mathcal{D}}(\theta) := \ell_{\mathcal{D}} \circ \mu(\theta) = \sum_{i=1}^{N} \|\mu_\theta(X_i) - Y_i\|_2^2.$$

Learning via empirical risk minimization requires minimization of $\mathcal{L}_{\mathcal{D}}$. In practice, one often uses (variants of) gradient descent or stochastic gradient descent. For a mathematical analysis it is convenient to view gradient descent as an Euler discretization of gradient flow, which we will consider here exclusively. Introducing a time dependence $\theta(t), t \in [0, \infty)$ and given an initialization θ_0, the gradient flow is the solution to the ODE

$$\frac{d}{dt}\theta(t) = -\nabla \mathcal{L}_{\mathcal{D}}(\theta(t)), \quad \theta(0) = \theta_0. \quad (3)$$

The main interest of this paper is to study geometric properties of this gradient flow as well as convergence properties, in particular, whether the flow converges to a minimum of $\mathcal{L}_{\mathcal{D}}$. The **NTK** introduced in Jacot et al. (2018) is a useful tool to this end. Denoting by J_θ the Jacobi matrix of μ at θ, the evolution in function space, i.e., the evolution of $\mu(\theta(t))$ can be written as

$$\frac{d}{dt}\mu(\theta(t)) = -J_{\theta(t)} J_{\theta(t)}^\top \nabla \ell_{\mathcal{D}}(\mu(\theta(t))),$$

where $\nabla \ell_{\mathcal{D}}(\alpha)$ denotes the (functional) gradient of $\ell_{\mathcal{D}}$ with respect to α. The above equation motivates to introduce the NTK as

$$K_\theta := J_\theta J_\theta^\top.$$

In case of fully connected linear networks where $\mu(\theta) = W_N \cdots W_1$, the gradient flow equation reads

$$\frac{d}{dt} W_j(t) = -\nabla_{W_j} \mathcal{L}_{\mathcal{D}}(W_1(t), \dots, W_N(t)), \quad (4)$$

for $j = 1, \dots, H$. It is shown in Arora et al. (2018) that for all $i = 1, \dots, H-1$,

$$\Delta_i := \Delta_i(t) = W_{i+1}^\top(t) W_{i+1}(t) - W_i(t) W_i^\top(t) \quad (5)$$

is constant in t, i.e., $\Delta_i(t) = \Delta_i(0)$ for all $t \geq 0$ and all $i = 1, \dots, H-1$. We refer to Marcotte et al. (2024) for more information on such invariants. In function space, the corresponding evolution is

$$\mu(\theta(t)) = W(t) = W_H(t) \cdots W_1(t).$$

For balanced initialization, i.e., $\Delta_i(0) = 0$ for all $i = 1, \dots, H-1$, the evolution of $W(t)$ only depends on $W(t)$ itself but not on the individual factors $W_l(t)$. More precisely, it is shown in Arora et al. (2018) that in this case $W(t)$ satisfies the ODE

$$\begin{aligned} &\frac{d}{dt} W(t) = \\ &-\sum_{j=1}^{H} (W(t)W(t)^T)^{\frac{H-j}{H}} \nabla \mathcal{L}_{\mathcal{D}}(W(t)) (W(t)^T W(t))^{\frac{j-1}{H}} \\ &=: -\mathcal{A}_W(t)\big(\nabla \mathcal{L}_{\mathcal{D}}(W(t))\big). \end{aligned} \quad (6)$$

It is shown in Bah et al. (2022) that the restriction $\bar{\mathcal{A}}_W$ of $\mathcal{A}_W$ to the tangent space $T_W(\mathcal{M}_r)$ of the manifold $\mathcal{M}_r$ of $d_H \times d_0$ matrices of rank exactly r is invertible. The bilinear form

$$g_W(Y, Z) = \langle \bar{\mathcal{A}}_W^{-1} Y, Z \rangle, \quad Y, Z \in T_W(\mathcal{M}_r)$$

generates a Riemannian metric of class C^1 on $\mathcal{M}_r$ (Bah et al., 2022, Propositions 10 and 11). Moreover, with the associated Riemannian gradient ∇^g we can write the evolution (6) of $W(t)$ (in case of balanced initialization, i.e., $\Delta_i(0) = 0$, $i = 1, \dots, H-1$) as Riemannian gradient flow (see also Bah et al. (2022)):

$$\frac{d}{dt} W(t) = -\nabla^g \mathcal{L}_{\mathcal{D}}(W(t)).$$

The NTK at $\theta = (W_1, \dots, W_H)$ (not necessarily balanced) is given by its action on $Z \in \mathbb{R}^{d_H \times d_H}$ via (see, e.g., (Bah et al., 2022, Lemma 2)

$$\begin{aligned} K_\theta(Z) = \textstyle\sum_{l=1}^{H} & W_H \cdots W_{l+1} W_{l+1}^\top \cdots W_H^\top Z \\ & W_1^\top \cdots W_{l-1}^\top W_{l-1} \cdots W_1. \end{aligned} \quad (7)$$

If $\theta = (W_1, \dots, W_H)$ is balanced, i.e., $\Delta_i = 0$, $i = 1, \dots, H-1$, and $W = W_H \cdots W_1$ then

$$K_\theta(Z) = \mathcal{A}_W(Z);$$

in particular, the NTK is independent of the parameterization of $W = \mu(\theta)$, i.e., independent of the individual factors W_l in the product $W = W_H \cdots W_1$. Moreover, the Riemannian metric g_W on $\mathcal{M}_k$ is given by the quadratic form induced by K_θ^{-1} (where the inverse is taken for the restriction on the tangent space $T_W(\mathcal{M}_k)$).

It is shown in Bah et al. (2022) that for almost all initializations, the gradient flow (4) converges to a global minimum of $\ell_{\mathcal{D}}(\alpha)$ restricted to the manifold $\mathcal{M}_k$ of rank k-matrices where $k \leq r := \min\{d_l, l = 0, \dots, H\}$. For $k = r$, this corresponds to the global minimum of $\mathcal{L}_{\mathcal{D}}(\theta)$, but it is currently open whether $k = r$ holds for almost all initializations.

For linear convolutional networks, where $\theta = (w_1, \dots, w_H)$ and $\mu(\theta)$ is the corresponding end-to-end convolutional filter (taking into account the strides $s_1, \dots, s_H$), we will study the parameter-independence of NTKs; see Section 1.2. Convergence of the corresponding gradient flow for learning linear convolutional neural networks on one-dimensional signals, i.e., $D = 1$, is studied in Diederen et al. (2026). It is shown that for certain loss functions, including the square loss, the gradient flow (3) converges to a critical point of $\mathcal{L}_{\mathcal{D}}$.

1.2 Main contributions

We prove the following:

- For linear convolutional networks on D-dimensional signals where $D \geq 2$: For any fixed $\delta = (\delta_1, \dots, \delta_{H-1}) \in \mathbb{R}^{H-1}$ and any end-to-end function v in the smooth locus of the neuromanifold $\mathcal{M}$, there are finitely many parameters $\theta = (w_1, \dots, w_H)$ in $\mu^{-1}(v)$ that satisfy (1); they all have the same NTK K_θ. Hence, the NTK only depends on the end-to-end function v and the invariants δ.
  - For every fixed δ, the NTKs are a Riemannian metric on the smooth locus of $\mathcal{M}$.

  - For fixed δ, the NTK is generally parameter dependent on the singular locus of $\mathcal{M}$.

- For linear convolutional networks on one-dimensional signals where the strides $s_1, \ldots, s_{H-1}$ are larger than 1: All results from the previous bullet points still hold.
  - It is known that for enough and sufficiently generic training data, every critical point θ of the squared-error loss is either the zero function $\mu(\theta) = 0$, or θ is a regular point of μ and its image $\mu(\theta)$ is in the smooth locus of $\mathcal{M}$ (Kohn et al., 2024, Theorem 2.12). We show that gradient flow with generic initialization does not converge to the zero function. Note that the zero function corresponds to a saddle point of $\mathcal{L}_{\mathcal{D}}$ (unless all Y_i are 0). As noted above, it has been shown in Diederen et al. (2026) that (for convolutional networks on one-dimensional signals) gradient flow always converges to a critical point. This means that altogether we can say that we have a metric on the smooth part of $\mathcal{M}$ with convergence to a point in that smooth part, and that this point is a critical point of the loss in both parameter and function space.

- For linear convolutional networks on one-dimensional signals where some strides are one: For fixed δ, the NTK is generally parameter dependent.

- For fully-connected linear networks, the NTK is generally parameter dependent. While this may intuitively be clear given the available results, this has not been rigorously shown before. In this paper, we therefore provide formal proofs of the following facts.
  - For generic $\Delta = (\Delta_1, \ldots, \Delta_{H-1})$, the NTK is generally parameter dependent.
  - In the balanced case (i.e., all $\Delta_i = 0$), we explain the underlying geometry for why the NTK only depends on the function V at hand: The balanced parameters in $\mu^{-1}(V)$ are related by orthogonal matrices, which are isometries of the ambient Euclidean parameter space.

- Our results also apply to shallow ReLU networks; see Section 5.

We summarize the key difference between the studied architectures in Table 1.

Linear network architecture	**Is NTK parameter independent for fixed Δ resp. δ?**
fully-connected	yes, when balanced (i.e., $\Delta = 0$)
1-dim convolutions	yes, for all δ, when all strides > 1
high-dim convolutions	yes, for all δ and all strides

Table 1: Linear network architectures studied in this article. Without the assumptions after "when", the answer to the question becomes "generally no".

Remark 1. For networks with algebraic activation function, the network parametrization map $\mu : \Theta \to \mathcal{M}$ is algebraic, and so is the map that assigns to parameters $\theta \in \Theta$ the NTK K_θ. Therefore, the equality of NTKs $K_{\theta_1} = K_{\theta_2}$ is a polynomial condition in (θ_1, θ_2). On irreducible algebraic sets, polynomial conditions hold either everywhere or almost nowhere. Since the invariants (5) and (1) are polynomial, the set of parameter pairs (θ_1, θ_2) that satisfy the same invariants and give rise to the same function $\mu(\theta_i)$ is an algebraic set. Hence, to prove the bullet items above where "the NTK is generally parameter dependent", it is sufficient to exhibit one counterexample.

1.3 Relation to previous works

Implicit bias and optimization geometry in deep linear networks. Previous works observed that gradient-based methods in deep linear models exhibit non-trivial implicit regularization Saxe et al. (2014); Ji and Telgarsky (2019); Gunasekar et al. (2017, 2018). For fully-connected architectures with balanced initialization, one part of a geometric explanation for that phenomenon is that gradient flow on parameters induces in fact a Riemannian gradient flow on the function space Arora et al. (2019b); Bah et al. (2022). Our results demonstrate that linear CNNs exhibit the latter behavior even without balancedness (except in certain degenerate 1-dimensional cases). This suggests that implicit geometric regularization is more robust in structured convolutional architectures than in fully-connected networks that have been studied in e.g. Moroshko et al. (2020); Woodworth et al. (2020).

Finite-width NTK behavior. Most NTK analyses focus on infinite-width limits, where the kernel

becomes stationary Jacot et al. (2018); Lee et al. (2019); Arora et al. (2019a). Finite-width phenomena remain significantly less understood Huang and Yau (2020); Nguyen and Mondelli (2020). Our results contribute to this direction by showing that for finite-width CNNs, the NTK depends only on the end-to-end function and the invariants δ, not on internal parameterization. This contrasts sharply with fully connected networks, where parameter dependence persists generically.

Architectural inductive bias and identifiability in structured models. Convolutions encode translation-equivariance and local connectivity lec (2002); Cohen and Welling (2016); Mallat (2016), which are understood as sources of inductive bias Bietti and Mairal (2019). We show that this structural inductive bias constrains the function space so much that (except in degenerate 1-dimensional cases) generic end-to-end filters admit unique parametrizations (up to scaling). In particular, this leads to a well-defined Riemannian metric on the smooth part of the function space. This aligns with work showing that structural constraints can enforce uniqueness or restrict the representational geometry of deep models Haeffele and Vidal (2015); Sedghi et al. (2018), and more broadly with analyses of how architectural structure shapes reachable functions and training dynamics Geiger et al. (2020).

Algebra-geometric approaches to deep learning and their link to optimization. Recent work has applied algebraic geometry to analyze parameter symmetries and singularities of neuromanifolds Kileel et al. (2019); Kohn et al. (2024); Finkel et al. (2025); Usevich et al. (2025); Massarenti and Mella (2025); Arjevani et al. (2025); Shahverdi et al. (2025, 2026); Marchetti et al. (2025). Also, the critical points have been analyzed via the algebraic viewpoint Mehta et al. (2021); Bharadwaj and Hoşten (2023); Shahverdi (2025); Graziani (2026). Our paper extends this program by characterizing the parameter symmetries and smooth loci for convolutional neuromanifolds and by linking them to NTK-induced Riemannian metrics. The latter supports the idea that optimization dynamics can be understood in terms of intrinsic geometric data rather than solely statistical or approximation-theoretic properties Bronstein et al. (2021); Amari (1998).

Convergence and saddle avoidance under gradient flow. Our results complement recent analyses showing that gradient descent and flow avoid strict saddles under generic conditions Lee et al. (2016); Panageas and Piliouras (2016); Bah et al. (2022). In particular, for generic data and initialization, we prove that gradient flow converges to a critical point in the smooth locus rather than collapsing to the zero map. This relates to broader work on benign nonconvexity and landscape geometry, which shows that certain deep learning loss landscapes, despite being nonconvex, do not exhibit harmful local minima Du et al. (2019); Kawaguchi (2016); Choromanska et al. (2015).

2 CONVOLUTIONS ON ONE-DIMENSIONAL SIGNALS

In this section, we focus on linear networks where each layer is a convolution on one-dimensional signals, see (2). The composition of such convolutions is equivalent to multiplying sparse polynomials as follows. For positive integers t and k, we write $\mathbb{R}[x^t, y^t]_{k-1}$ for the vector space of polynomials that are homogeneous of degree $k-1$ in the variables (x^t, y^t). This vector space is isomorphic to $\mathbb{R}^k$ via

$$\pi_t : \mathbb{R}^k \to \mathbb{R}[x^t, y^t]_{k-1},$$
$$v \mapsto v_0 x^{t(k-1)} + v_1 x^{t(k-2)} y^t + \ldots + v_{k-1} y^{t(k-1)}.$$

The important insight is the following equality:

$$\pi_1(\mu(w_1, \ldots, w_H)) = \pi_{t_H}(w_H) \cdots \pi_{t_1}(w_1), \quad (8)$$
$$\text{where } t_l = s_1 \cdots s_{l-1}.$$

This allows us to view the network parametrization map μ as multiplying sparse polynomials. We also see directly from (8) that layers with $k_l = 1$ are irrelevant (i.e., such a layer can be omitted) and that the last stride s_H does not affect the map μ. Therefore, we assume from now on that $k_l > 1$ for all $l = 1, \ldots, H$ and that $s_H = 1$.

The neuromanifold $\mathcal{M} := \mathrm{im}(\mu)$ is a Euclidean closed, semi-algebraic subset of $\mathbb{R}^k$ of dimension $\sum_{l=1}^{H}(k_l - 1) + 1$ (Kohn et al., 2024, Theorem 2.4). If $H > 1$, it has singular points. This singular locus consists of 0 and the union of all neuromanifolds that are strictly contained in $\mathcal{M}$ and come from architectures with the same strides $s_1, \ldots, s_H$ (Kohn et al., 2024, Theorem 2.9). This stratification is analogous to the well-known fact that the neuromanifold of a linear fully-connected network consists of all matrices whose rank is upper bounded by some r, and that its singular locus (if r is not the maximal rank) is

precisely the matrices of rank at most $r-1$. We denote the smooth locus of $\mathcal{M}$ as $\mathrm{Reg}(\mathcal{M})$, which is the set of points in $\mathcal{M}$, which are not singular.

Due to the scaling ambiguity of the filters $\mu(\lambda_1 w_1, \ldots, \lambda_H w_H) = \lambda_1 \cdots \lambda_H \mu(w_1, \ldots, w_H)$ (for $\lambda_l \in \mathbb{R}$), we can projectify the map μ:

$$\begin{aligned}\mu_{\mathbb{P}} :&\mathbb{P}^{k_1-1} \times \ldots \times \mathbb{P}^{k_H-1} \to \mathbb{P}^{k-1},\\ &([w_1], \ldots, [w_H]) \mapsto [\mu(w_1, \ldots, w_H)].\end{aligned}$$

Here, we write $[v]$ for the element in projective space $\mathbb{P}^{k-1}$ corresponding to a nonzero vector $v \in \mathbb{R}^k$. The image of $\mu_{\mathbb{P}}$ is the projective neuromanifold $\mathbb{P}\mathcal{M}$.

We see that knowing the invariants in (1) fixes the scaling ambiguity up to signs:

Proposition 2. *Let $\delta_1, \ldots, \delta_{H-1} \in \mathbb{R}$, and consider non-zero filters $w_1 \in \mathbb{R}^{k_1}, \ldots, w_H \in \mathbb{R}^{k_H}$. Then, there are 2^{H-1} scalar tuples $(\lambda_1, \ldots, \lambda_H) \in \mathbb{R}^H$ such that $\lambda_1 \cdots \lambda_H = 1$ and*

$$\delta_i = \|\lambda_{i+1} w_{i+1}\|^2 - \|\lambda_i w_i\|^2, \quad \text{for } i = 1, \ldots, H-1.$$

These scalar tuples are all equal up to sign changes of the components.

The proof of Proposition 2 and all subsequent proofs can be found in the Appendix. In the following, we will make use of this standard fact:

Observation 3. Since μ is linear in each of the w_i, its derivative at a filter tuple $\theta = (w_1, \ldots, w_H)$ is the linear map

$$\begin{aligned}d_\theta \mu : \dot\theta = (\dot w_1, \ldots, \dot w_H) \mapsto &\mu(\dot w_1, w_2, \ldots, w_H) + \ldots\\ &+ \mu(w_1, \ldots, w_{H-1}, \dot w_H).\end{aligned}$$

It is represented by the Jacobian matrix $J = J_\theta$. It has the block columns structure $J = [J_1 | J_2 | \cdots | J_H]$, where J_i corresponds to taking the derivative with respect to w_i; in other words,

$$\begin{aligned} J_1 \dot w_1 &= \mu(\dot w_1, w_2, \ldots, w_H),\\ \vdots \quad &\vdots \qquad\qquad \vdots\\ J_H \dot w_H &= \mu(w_1, \ldots, w_{H-1}, \dot w_H).\end{aligned}$$

From the latter expressions, we see that the entries of J_i are linear in the entries of w_j (for $j \neq i$). We then conclude $K_\theta = JJ^\top = J_1 J_1^\top + \ldots + J_H J_H^\top$, where the entries of each term $K_i := J_i J_i^\top$ are homogeneous of degree 2 in the entries of w_j (for $j \neq i$).

2.1 All strides larger than one

We now show that the NTK only depends on the end-to-end filter $v \in \mathcal{M}$ and the invariants $\delta_1, \ldots, \delta_{H-1}$, when the strides in the layers – up to possibly the last one – are larger than one.

Theorem 4. *Let the strides $s_1, \ldots, s_{H-1}$ be larger than one. For any fixed $\delta = (\delta_1, \ldots, \delta_{H-1}) \in \mathbb{R}^{H-1}$ and any end-to-end function v in the smooth locus of the neuromanifold $\mathcal{M}$, there are 2^{H-1} many parameters $\theta = (w_1, \ldots, w_H)$ in $\mu^{-1}(v)$ that satisfy (1); they all have the same NTK K_θ.*

Hence, for any fixed choice of invariants $\delta = (\delta_1, \ldots, \delta_{H-1})$, the NTK only depends on the smooth point $v \in \mathcal{M}$ and we denote it by $K^{(\delta)}(v)$. See Example 15 in the Appendix.

The NTKs $K^{(\delta)}(v)$ are the inverse of the pushforward of the following Riemannian metric: Consider

$$\begin{aligned}\Theta_\delta := \{(w_1, \ldots, w_H) \in \mu^{-1}(\mathrm{Reg}(\mathcal{M})) \mid\\ \delta_i = \|w_{i+1}\|^2 - \|w_i\|^2 \text{ for } i = 1, \ldots, H-1\},\end{aligned}$$

where we recall that $\mathrm{Reg}(\mathcal{M})$ denotes the smooth locus of $\mathcal{M}$. This space Θ_δ is a smooth manifold of the same dimension as the neuromanifold $\mathcal{M}$. Its tangent space at any $\theta = (w_1, \ldots, w_H) \in \Theta_\delta$ is

$$\begin{aligned}T_\theta \Theta_\delta = \{(\dot w_1, \ldots, \dot w_H) \mid \langle w_{i+1}, \dot w_{i+1}\rangle = \langle w_1, \dot w_1\rangle\\ \text{for all } i = 1, \ldots, H-1\},\end{aligned} \tag{9}$$

where $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product.

Lemma 5. *For every $\theta \in \Theta_\delta$, the derivative $d_\theta \mu$ restricted to $T_\theta \Theta_\delta$ is a bijection onto $T_{\mu(\theta)}\mathcal{M}$.*

Corollary 6. *Let the strides $s_1, \ldots, s_{H-1}$ be larger than one. The network parametrization map μ restricted to Θ_δ is a submersion onto $\mathrm{Reg}(\mathcal{M})$ with finite fibers. The pushforward of the Euclidean metric on Θ_δ under μ is given by the bilinear forms $(K^{(\delta)}(v))^{-1}$ for $v \in \mathrm{Reg}(\mathcal{M})$.*

In particular, Corollary 6 shows that the $K^{(\delta)}(v)$, for any fixed choice of δ, form a Riemannian metric on the smooth locus of the neuromanifold $\mathcal{M}$. This metric can generally not be extended on the singular locus of $\mathcal{M}$. By Kohn et al. (2024), the singular locus of $\mathcal{M}$ consists precisely of those end-to-end filters v that either have several filter tuples (even up to scaling by constants) parametrizing it or such that the unique filter tuple θ (up to scaling) satisfies that the rank $d_\theta \mu$ is less than $\dim(\mathcal{M})$. In the latter case, the NTK K_θ drops rank. In the former case, the NTKs

K_{θ_1} and K_{θ_2} for $\mu(\theta_1) = \mu(\theta_2)$ are generally distinct as Example 16 in the Appendix demonstrates.

Moreover, for varying δ, the metrics $K^{(\delta)}$ on $\mathrm{Reg}(\mathcal{M})$ are generally distinct. See Ex. 17 in the Appendix.

2.2 Convergence of gradient flow

In this section, we focus on the squared-error loss. We still assume that the strides in the network layers are larger than one. We show that, for a sufficient amount of generic training data and generic initialization, gradient flow converges to a critical point θ in parameter space such that $\mu(\theta)$ is a critical point on the smooth locus of the neuromanifold $\mathcal{M}$.

Given training data $\mathcal{D} = \{(X_i, Y_i) \in \mathbb{R}^{d_0} \times \mathbb{R}^{d_H} \mid i = 1, 2, \ldots, N\}$, the squared-error loss of an end-to-end convolution α is $\ell_{\mathcal{D}}(\alpha) = \sum_{i=1}^{N} \|\alpha(X_i) - Y_i\|^2$. We denote the squared-error loss in parameter space by $\mathcal{L}_{\mathcal{D}} := \ell_{\mathcal{D}} \circ \mu$. Recall that we write k for the size of the end-to-end filters in $\mathcal{M}$.

Theorem 7. *Let the strides $s_1, \ldots, s_{H-1}$ be larger than one, and let $N \geq k$. For almost all data $\mathcal{D} \in (\mathbb{R}^{d_0} \times \mathbb{R}^{d_H})^N$ and almost all initializations, when minimizing the squared-error loss $\mathcal{L}_{\mathcal{D}}$, gradient flow converges to a critical point θ of $\mathcal{L}_{\mathcal{D}}$ such that $\mu(\theta)$ is a smooth point of $\mathcal{M}$ and a critical point of $\ell_{\mathcal{D}}|_{\mathrm{Reg}(\mathcal{M})}$.*

Proof. It is proven in Diederen et al. (2026) that gradient flow converges to a critical point of $\mathcal{L}_{\mathcal{D}}$. Moreover, under the assumptions that the strides are larger than one, $N \geq k$, and sufficiently generic data, (Kohn et al., 2024, Theorem 2.12) shows that every critical point θ of $\mathcal{L}_{\mathcal{D}}$ either satisfies $\mu(\theta) = 0$ or $\mu(\theta)$ is a smooth point of $\mathcal{M}$ that is a critical point of $\ell_{\mathcal{D}}$ restricted to that smooth locus. Hence, it is enough to exclude $\mu(\theta) = 0$ for generic initializations. We explain this separately in Proposition 8. □

Proposition 8. *Let $N \geq k$. For almost all data $\mathcal{D} \in (\mathbb{R}^{d_0} \times \mathbb{R}^{d_H})^N$ and almost all initializations, when minimizing $\mathcal{L}_{\mathcal{D}}$, gradient flow converges to a point θ with $\mu(\theta) \neq 0$.*

The proof of Prop. 8 makes use of *projective duality*, a standard construction from algebraic geometry.

2.3 Some strides equal to one

A key fact that enabled Thm. 4 and its proof is that a generic point v in the neuromanifold has a unique point (up to scaling) in its fiber $\mu^{-1}(v)$. In other words, the generic non-empty fiber of the projective network parametrization map $\mu_{\mathbb{P}}$ is a singleton. An algebraic map with that property is called *birational*; meaning that it is bijective almost everywhere.

When some of the strides $s_1, \ldots, s_{H-1}$ are equal to one, the projective network parametrization map $\mu_{\mathbb{P}}$ is no longer birational Kohn et al. (2024). This means that it is no longer true that a generic point v in the neuromanifold has a unique point (up to scaling) in its fiber $\mu^{-1}(v)$. In fact, the fibers of $\mu_{\mathbb{P}}$ are now generally finite but larger than one. This can be seen via the interpretation of μ as polynomial multiplication given in (8): If some stride s_l with $1 \leq l < H$ is equal to one, then $t_l = t_{l+1}$, and so the polynomial factors $\pi_{t_l}(w_l)$ and $\pi_{t_{l+1}}(w_{l+1})$ live in the same ring $\mathbb{R}[x^{t_l}, y^{t_l}]$, which means that we can swap some of their irreducible factors without changing the product in (8). This results in distinct (even up to scaling) filter tuples θ_1 and θ_2 parametrizing the same end-to-end filter, i.e., $\mu(\theta_1) = \mu(\theta_2)$. They can be scaled such that they satisfy the same invariants (1), but in general their NTKs remain distinct. See Example 19 in the Appendix.

3 CONVOLUTIONS ON HIGHER-DIMENSIONAL SIGNALS

A convolution on D-dimensional signals is given by a filter w that is a D-dimensional tensor of format $k^{(1)} \times \ldots \times k^{(D)}$ and by its stride tuple $\boldsymbol{s} = (s^{(1)}, \ldots, s^{(D)}) \in \mathbb{Z}^D_{>0}$. The convolution is a linear map $\alpha_{w,\boldsymbol{s}}$ that produces an output tensor of format $d^{(1)} \times \ldots \times d^{(D)}$ from an input tensor X of format $(s^{(1)}(d^{(1)}-1)+k^{(1)}) \times \ldots \times (s^{(D)}(d^{(D)}-1)+k^{(D)})$:

$$(\alpha_{w,\boldsymbol{s}}(X))_{i_1,\ldots,i_D} = \sum_{j_1=0}^{k^{(1)}-1} \cdots \sum_{j_D=0}^{k^{(D)}-1} w_{j_1,\ldots,j_D} \cdot X_{i_1 s^{(1)}+j_1,\ldots,i_D s^{(D)}+j_D} \tag{10}$$

for $i_m = 0, 1, \ldots, d^{(m)} - 1$. A linear convolutional network composes H such convolutions, which results again in a convolution on D-dimensional signals. Writing $s_l^{(m)}$ and $k_l^{(m)}$ for the strides and filter sizes in the l-th layer, the end-to-end convolution has stride $(s_1^{(1)} \cdots s_H^{(1)}, \ldots, s_1^{(D)} \cdots s_H^{(D)})$ and filter format $k^{(1)} \times \ldots \times k^{(D)}$, where $k^{(m)} := k_1^{(m)} + \sum_{l=2}^{H} (k_l^{(m)} - 1) \prod_{i=1}^{l-1} s_i^{(m)}$. The network parametrization map assigns to a tuple of filter tensors $(w_1, \ldots, w_H)$ the filter of the end-to-end convolution $\mu(w_1, \ldots, w_H)$:

$$\mathbb{R}^{k_1^{(1)} \times \ldots \times k_1^{(D)}} \times \ldots \times \mathbb{R}^{k_H^{(1)} \times \ldots \times k_H^{(D)}} \to \mathbb{R}^{k^{(1)} \times \ldots \times k^{(D)}}.$$

As in the case of 1-dimensional signals, the end-to-end filter can be equivalently computed via polynomial multiplication. Now, we need D pairs of variables $(\boldsymbol{x}, \boldsymbol{y}) := ((x_1, y_1), \dots, (x_D, y_D))$. For tuples $\boldsymbol{t} = (t^{(1)}, \dots, t^{(D)})$ and $\boldsymbol{k} = (k^{(1)}, \dots, k^{(D)})$ of positive integers, we write $\mathbb{R}[(\boldsymbol{x}, \boldsymbol{y})^{\boldsymbol{t}}]_{\boldsymbol{k}-\mathbf{1}}$ for the vector space of polynomials that are homogeneous of degree $k^{(m)} - 1$ in the variables $(x_m^{t^{(m)}}, y_m^{t^{(m)}})$ for $m = 1, \dots, D$. We can identify this space of polynomials with the space of tensors v in $\mathbb{R}^{k^{(1)} \times \dots \times k^{(D)}}$ via

$$\pi_{\boldsymbol{t}} : v \mapsto \sum_{i_1=0}^{k^{(1)}-1} \cdots \sum_{i_D=0}^{k^{(D)}-1} v_{i_1,\dots,i_D} x_1^{t^{(1)}(k^{(1)}-1-i_1)} y_1^{t^{(1)} i_1} \cdots x_D^{t^{(D)}(k^{(D)}-1-i_D)} y_D^{t^{(D)} i_D}.$$

Lemma 9. *Writing* $\mathbf{1} := (1, \dots, 1)$ *and* $t_l^{(m)} := s_1^{(m)} \cdots s_{l-1}^{(m)}$*, we have*

$$\pi_{\mathbf{1}}(\mu(w_1, \dots, w_H)) = \pi_{\boldsymbol{t}_H}(w_H) \cdots \pi_{\boldsymbol{t}_1}(w_1).$$

The above lemma lets us interpret the network parametrization map μ as the multiplication of polynomials, and we can think of the neuromanifold $\mathcal{M}$ as the set of all polynomials in the variables $x_1, y_1, \dots, x_D, y_D$ that can be factored as $Q_H \cdots Q_1$ with $Q_l \in \mathbb{R}[(\boldsymbol{x}, \boldsymbol{y})^{\boldsymbol{t}_l}]_{\boldsymbol{k}_l - \mathbf{1}}$. Similarly to the case of one-dimensional signals, we see that a filter size $k_l^{(m)}$ being 1 means that the variables x_m, y_m do not appear in the l-th layer.

Corollary 10. *Let* $D > 1$*, and assume that every layer has at least two filter sizes larger than one. Then the projectivization of* μ *is birational, up to possibly reordering some of the layer factors. In other words, almost every end-to-end filter in the neuromanifold* $\mathcal{M}$ *has a unique factorization into layer filters, up to scalar multiplication in each layer and possibly reordering some of the layers.*

Observation 11. a) The ordering of the filters in Cor. 10 is *not* uniquely determined if and only if the spaces $\mathbb{R}[(\boldsymbol{x}, \boldsymbol{y})^{\boldsymbol{t}_l}]_{\boldsymbol{k}_l - \mathbf{1}}$ and $\mathbb{R}[(\boldsymbol{x}, \boldsymbol{y})^{\boldsymbol{t}_L}]_{\boldsymbol{k}_L - \mathbf{1}}$ for two layers $l < L$ coincide. In this case,

$$\pi_{\boldsymbol{t}_l}(w_L) = \pi_{\boldsymbol{t}_L}(w_L),\ \pi_{\boldsymbol{t}_L}(w_l) = \pi_{\boldsymbol{t}_l}(w_l), \quad (11)$$

and so Lemma 9 implies that we can swap the filters w_l and w_L without affecting the image $\mu(w_1, \dots, w_H)$. Note that (11) requires $\boldsymbol{t}_l = \boldsymbol{t}_L$, which for $l < L$ can only happen if $D > 1$.

b) The NTK is not affected by the swap described in a). To see this, we consider $\theta = (w_1, \dots, w_H)$ and θ' that is obtained from θ by swapping w_l with w_L. Observation 3 also holds for convolutions on higher-dimensional signals as it only depends on the multilinearity of μ. So we consider the block structure of the Jacobian matrices $J_\theta = [J_{\theta,1} | \cdots | J_{\theta,H}]$ and $J_{\theta'} = [J_{\theta',1} | \cdots | J_{\theta',H}]$. By Observation 3 and Lemma 9, the matrix $J_{\theta,i}$ encodes the linear map that multiplies $\pi_{\boldsymbol{t}_i}(\cdot)$ with all polynomials $\pi_{\boldsymbol{t}_j}(w_j)$ for $j \neq i$. Therefore, (11) implies that $J_{\theta',l} = J_{\theta,L}$, $J_{\theta',L} = J_{\theta,l}$, and $J_{\theta',i} = J_{\theta,i}$ for $i \notin \{l, L\}$. Thus, we conclude $K_{\theta'} = \sum_{i=1}^H J_{\theta',i} J_{\theta',i}^\top = \sum_{i=1}^H J_{\theta,i} J_{\theta,i}^\top = K_\theta$.

c) $\mathbb{R}[(\boldsymbol{x}, \boldsymbol{y})^{\boldsymbol{t}_l}]_{\boldsymbol{k}_l - \mathbf{1}}$ coincides with $\mathbb{R}[(\boldsymbol{x}, \boldsymbol{y})^{\boldsymbol{t}_L}]_{\boldsymbol{k}_L - \mathbf{1}}$ if and only if $\boldsymbol{k}_l = \boldsymbol{k}_L$ and, for all $m = 1, \dots, D$ with $k_l^{(m)} > 1$, we have $t_l^{(m)} = t_L^{(m)}$. The latter condition means that the strides $s_l^{(m)}, \dots, s_{L-1}^{(m)}$ are equal to one.

As in the case of one-dimensional signals, the scaling ambiguity that appears in Corollary 10 is fixed (up to signs) by knowing the invariants in (1).

Proposition 12. *Let* $\delta_1, \dots, \delta_{H-1} \in \mathbb{R}$*, and consider non-zero filter tensors* $w_1 \in \mathbb{R}^{k_1^{(1)} \times \dots \times k_1^{(D)}}$*, …,* $w_H \in \mathbb{R}^{k_H^{(1)} \times \dots \times k_H^{(D)}}$*. There are* 2^{H-1} *scalar tuples* $(\lambda_1, \dots, \lambda_H) \in \mathbb{R}^H$ *such that* $\lambda_1 \cdots \lambda_H = 1$ *and*

$$\delta_i = \|\lambda_{i+1} w_{i+1}\|_F^2 - \|\lambda_i w_i\|_F^2, \quad \text{for } i = 1, \dots, H-1.$$

These scalar tuples are all equal up to signs.

Putting everything together, we can now conclude for convolutions on higher-dimensional signals that the NTK only depends on the end-to-end filter $v \in \mathcal{M}$ and the invariants $\delta_1, \dots, \delta_{H-1}$, no matter whether some strides are equal to one.

Theorem 13. *Let* $D > 1$*, and assume that every layer has at least two filter sizes larger than one. For any* $\delta = (\delta_1, \dots, \delta_{H-1}) \in \mathbb{R}^{H-1}$ *and any end-to-end function* v *in the smooth locus of the neuromanifold* $\mathcal{M}$*, there are* 2^{H-1} *many parameters* $\theta = (w_1, \dots, w_H)$ *in* $\mu^{-1}(v)$ *that satisfy* (1)*; they all have the same NTK* K_θ*. For fixed* δ*, these NTKs are a Riemannian metric on the smooth locus of* $\mathcal{M}$*.*

4 FULLY-CONNECTED NETWORKS

We now move to the case of fully-connected linear networks. As seen in Section 1.1, the NTK is parameter-independent for balanced initialization. Moreover, the polynomials (5) generate *all* algebraic relations

that stay constant along the gradient flow curve for generic initialization (Marcotte et al., 2024). Hence, a natural question is whether, like for convolutional networks with strides larger than 1, for a generic choice of Δ_i, the NTK is also parameter-independent. While the form (7) of the NTK suggests that this is not the case in general, we show this rigorously in Example 20 in the Appendix.

We now go back to the balanced case, and provide a geometric reason for why the NTK is parameter-independent in this case. The following result shows that the balanced weight matrices that lead to a fixed product matrix W differ only by orthogonal group action. The latter are isometries of the Euclidean metric on the parameter space and thus do not affect the NTK. Moreover, analogously to Corollary 6, the inverses of the NTK are the pushforward of the Euclidean metric on the space Θ_0 of balanced matrix tuples under the network parametrization map μ, which is simply the quotient metric under the orthogonal action.

Theorem 14. *Let $(W_H, \dots, W_1) \in GL_d(\mathbb{R})^H$ such that $W = W_H \cdots W_1$, and $\Delta_i = W_{i+1}^\top W_{i+1} - W_i W_i^\top = 0$ for all $i \in \{1, \dots, H-1\}$. Then the set of balanced weights with product W is equal to*

$$\mu^{-1}(W) \cap \Theta_0 = \big\{(W_H G_{H-1}, G_{H-1}^T W_{H-1} G_{H-2}, \dots, G_2^T W_2 G_1, G_1^T W_1) \text{ s.t. } G_{H-1}, \dots, G_1 \in \mathcal{O}_d(\mathbb{R})\big\}.$$

5 EXTENSION TO RELU NETWORKS

We illustrate in the case of shallow (i.e., two-layer) networks how the results presented in this work can be extended from linear to ReLU networks.

Shallow ReLU CNN. The NTK depends only on the end-to-end function, which we see as follows: The first observation is that the first convolutional layer is either zero or a linear map of full rank (Shahverdi et al., 2025, Lem. 4.1). Hence, for a nonzero network, there is a (Euclidean) open subset in the domain of the network where its first layer maps to the positive orthant, and so the network restricted to that domain subset is simply a shallow linear CNN. Hence, our factorization results on linear CNNs also apply to this ReLU case. In particular, for higher-dimensional convolutions, the filters in the two layers are generically uniquely identifiable (up to scaling) from the end-to-end function (as in Cor. 10). The second observation is that the δ invariants are still valid in the ReLU case, and - as in the linear case - they fix the scalings of the filters up to sign (as in Prop. 12). Finally, it is easy to see that the sign of the first layer cannot be flipped without changing the end-to-end function. All in all (except in degenerate cases of dimension 1 or stride 1), for a nonzero shallow ReLU CNN, there is only a single choice of parameters (i.e., filters) yielding the same end-to-end function. Hence, the NTK (trivially) only depends on the end-to-end function.

Shallow ReLU MLP. The argument above does not apply to fully-connected networks (MLPs). Already the first observation fails: Since fully-connected layers can have low rank, their range can completely miss the positive orthant. In fact, the study of parameter symmetries in ReLU MLPs is very subtle and not completely settled Grigsby et al. (2023, 2025). Either way, the NTK of ReLU MLPs is typically not parameter independent, even for gradient-flow invariants. To see this, we consider the shallow network in Example 20. By Marcotte et al. (2024), the conserved quantities via gradient flow for shallow ReLU networks are a subset of the linear ones, namely $\mathrm{diag}(W_2^\top W_2 - W_1 W_1^\top)$. Therefore, if we restrict the input space to positive X, then – as in the linear case – both U_2, U_1 and V_2, V_1 from Example 20 give rise to the same end-to-end function while also having equal conserved quantities. We have seen in Example 20 that they lead to different values for the linear NTK, and thus also their ReLU NTKs are different.

6 CONCLUSION

In this paper, we investigated how gradient flow on parameter space influences the dynamics of the end-to-end function in linear networks. We showed that, for a broad class of linear convolutional networks, the NTK – and thus the corresponding function-space ODE – can be expressed independently of the individual layer parameters, relying instead on a global quantity defined at initialization, leading to a Riemaniann gradient flow. In contrast, we demonstrated that this property does not generally extend to fully connected linear networks, highlighting a fundamental structural difference between the two architectures. We have further explained how these results on deep linear networks extend to shallow ReLU networks. Our findings raise intriguing questions about whether similar principles might emerge in the context of more practical, nonlinear networks.

Acknowledgements

We thank Antonio Lerario and Giovanni Luca Marchetti for helpful discussions. KK was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. EMA and HR acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project number 442047500 through the Collaborative Research Center "Sparsity and Singular Structures" (SFB 1481). EMA acknowledges funding by UM6P as well.

References

Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

Yossi Arjevani, Joan Bruna, Joe Kileel, Elzbieta Polak, and Matthew Trager. Geometry and optimization of shallow polynomial networks. *arXiv:2501.06074*, 2025.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253. PMLR, 2018.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332, 2019a.

Sanjeev Arora, Noah Golowich, Nadav Cohen, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2019b.

Bubacarr Bah, Holger Rauhut, Ulrich Terstiege, and Michael Westdickenberg. Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers. *Information and Inference: A Journal of the IMA*, 11(1):307–353, 2022.

Ayush Bharadwaj and Serkan Hoşten. Complex critical points of deep linear neural networks. *arXiv:2301.12651*, 2023.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv:2104.13478*, 2021.

Anna Choromanska, MIkael Henaff, Michael Mathieu, Gerard Ben Arous, and Yann LeCun. The Loss Surfaces of Multilayer Networks. In *International Conference on Artificial Intelligence and Statistics*, pages 192–204, 2015.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999, 2016.

Jona-Maria Diederen, Holger Rauhut, and Ulrich Terstiege. Convergence of gradient flow for learning linear convolutional neural networks. *arXiv:2601.08547*, 2026.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

Bella Finkel, Jose Israel Rodriguez, Chenxi Wu, and Thomas Yahl. Activation degree thresholds and expressiveness of polynomial neural networks. *Algebraic Statistics*, 16:113–130, 2025.

Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020.

Israel M Gelfand, Mikhail M Kapranov, and Andrei V Zelevinsky. *Discriminants, resultants, and multidimensional determinants*. Birkhäuser, 1994.

Giacomo Graziani. On the stable euclidean distance degree of algebraic layers. *arXiv:2601.16071*, 2026.

Elisenda Grigsby, Kathryn Lindsey, and David Rolnick. Hidden symmetries of ReLU networks. In *International Conference on Machine Learning*, pages 11734–11760, 2023.

J. Elisenda Grigsby, Kathryn Lindsey, Robert Meyerhoff, and Chenxi Wu. Functional dimension of feedforward relu neural networks. *Advances in Mathematics*, 482, 2025.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro.

Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Benjamin D Haeffele and René Vidal. Global optimality in tensor factorization, deep learning, and beyond. *arXiv:1506.07540*, 2015.

Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. In *International Conference on Machine Learning*, pages 4542–4551, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019.

Kenji Kawaguchi. Deep learning without poor local minima. *Advances in Neural Information Processing Systems*, 2016.

Joe Kileel, Matthew Trager, and Joan Bruna. On the expressive power of deep polynomial neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Kathlén Kohn, Bernt Ivar Utstøl Nødland, and Paolo Tripoli. Secants, bitangents, and their congruences. In *Combinatorial Algebraic Geometry*, pages 87–112. Springer, 2017.

Kathlén Kohn, Thomas Merkh, Guido Montúfar, and Matthew Trager. Geometry of linear convolutional networks. *SIAM Journal on Applied Algebra and Geometry*, 6(3):368–406, 2022.

Kathlén Kohn, Guido Montúfar, Vahid Shahverdi, and Matthew Trager. Function space and critical points of linear convolutional networks. *SIAM Journal on Applied Algebra and Geometry*, 8:333–362, 2024.

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

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), 2016.

Giovanni Luca Marchetti, Vahid Shahverdi, Stefano Mereta, Matthew Trager, and Kathlén Kohn. Position: Algebra unveils deep learning-an invitation to neuroalgebraic geometry. In *International Conference on Machine Learning*, 2025.

Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyré. Abide by the law and follow the flow: Conservation laws for gradient flows. *Advances in Neural Information Processing Systems*, 36, 2024.

Alex Massarenti and Massimiliano Mella. The alexander-hirschowitz theorem for neurovarieties. *arXiv:2511.19703*, 2025.

Dhagash Mehta, Tianran Chen, Tingting Tang, and Jonathan D Hauenstein. The loss surface of deep linear networks viewed through the algebraic geometry lens. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:5664–5680, 2021.

Edward Moroshko, Blake E Woodworth, Suriya Gunasekar, Jason D Lee, Nati Srebro, and Daniel Soudry. Implicit bias in deep linear classification: Initialization scale vs training accuracy. *Advances in Neural Information Processing Systems*, 33:22182–22193, 2020.

Gabin Maxime Nguegnang, Holger Rauhut, and Ulrich Terstiege. Convergence of gradient descent for learning linear neural networks. *Advances in Continuous and Discrete Models*, 23, 2024.

Quynh N Nguyen and Marco Mondelli. Global convergence of deep networks with one wide layer followed by pyramidal topology. *Advances in Neural Information Processing Systems*, 33:11961–11972, 2020.

Ioannis Panageas and Georgios Piliouras. Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions. *arXiv:1605.00405*, 2016.

A Saxe, J McClelland, and S Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Represenatations*, 2014.

Hanie Sedghi, Vineet Gupta, and Philip M Long. The singular values of convolutional layers. *arXiv:1805.10408*, 2018.

Vahid Shahverdi. Algebraic complexity and neurovariety of linear convolutional networks. *Acta Universitatis Sapientiae, Mathematica*, 17, 2025.

Vahid Shahverdi, Giovanni Luca Marchetti, and Kathlén Kohn. On the geometry and optimization of polynomial convolutional networks. In *International Conference on Artificial Intelligence and Statistics*, 2025.

Vahid Shahverdi, Giovanni Luca Marchetti, and Kathlén Kohn. Learning on a razor's edge: Identifiability and singularity of polynomial neural networks. In *International Conference on Learning Representations*, 2026.

Konstantin Usevich, Ricardo Augusto Borsoi, Clara Dérand, and Marianne Clausel. Identifiability of deep polynomial neural networks. In *Conference on Neural Information Processing Systems*, 2025.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673, 2020.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Material: The Riemannian Geometry Associated to Gradient Flows of Linear Convolutional Networks

7 PROOFS AND EXAMPLES

7.1 Proofs and Examples of Section 2

7.1.1 Proof of Proposition 2

Proof. This proof is inspired by the proof of (Kohn et al., 2022, Corollary 5.14). Define $C := \prod_{l=1}^{H} \|w_l\|^2$ and $\beta_l := \|\lambda_l w_l\|^2 = \lambda_l^2 \|w_l\|^2$ for all $l = 1, \dots, H$. Then we have reformulated our problem in that we want to show that the conditions

$$\prod_{l=1}^{H} \beta_l = C \text{ and } \beta_{i+1} - \beta_i = \delta_i \text{ for } i = 1, \dots, H-1 \tag{12}$$

have only one solution $\beta_1, \dots, \beta_H$ where all β_l are positive. All β_l can be computed from β_1 via $\beta_l = \beta_1 + \sum_{i=1}^{l-1} \delta_i$. Hence, the conditions (12) can be rewritten as a single equation

$$\beta_1 \cdot (\beta_1 + \delta_1) \cdots (\beta_1 + \delta_1 + \ldots + \delta_{H-1}) = C. \tag{13}$$

Now we choose a sufficiently large $n_1 > 0$ such that $n_l := n_1 + \sum_{i=1}^{l-1} \delta_i \geq 0$ for all $l = 2, \dots, H$. We define the new unknown $x := \beta_1 - n_1$ such that the equality (13) becomes

$$(x + n_1)(x + n_2) \cdots (x + n_H) = C. \tag{14}$$

Defining $n := \min\{n_1, \dots, n_H\} \geq 0$, we observe that the left-hand side of (14) is positive and monotonically increasing in the range $(-n, +\infty)$. Since the left-hand side is zero for $x = -n$ and converges to $+\infty$ for $x \to \infty$ and $C > 0$, the intermediate value theorem tells us that there is precisely one x in the range $(-n, +\infty)$ that yields the equality in (14). By definition of n, this is the only solution where all $\beta_l = x + n_l$ are positive. □

7.1.2 Proof of Theorem 4 and Example

Proof. Under the assumption that $s_l > 1$ for all $l = 1, \dots, H-1$, every point $[v]$ in the smooth locus of $\mathbb{P}\mathcal{M}$ has a unique point in its fiber $\mu_{\mathbb{P}}^{-1}([v])$ (Kohn et al., 2024, Section 5). In other words, for every end-to-end filter v in the smooth part of $\mathcal{M}$, there is a unique filter tuple, up to scaling, that parametrizes v. Together with Proposition 2, this shows that, for fixed $\delta_1, \dots, \delta_{H-1}$, every end-to-end filter v in the smooth locus of the neuromanifold $\mathcal{M}$ has a unique filter tuple $\theta = (w_1, \dots, w_H)$, up to signs, such that $\mu(\theta) = v$ and the invariants in (1) are satisfied. The signs do not affect the neural tangent kernel K_θ since, due to Observation 3, K_θ is the sum of matrices K_i that are homogeneous of degree two in the filters in θ. □

Example 15. The following network architecture will serve as a running example. We consider a network with two layers whose filters have sizes $k_1 = 3, k_2 = 2$ and with strides $s_1 = 2, s_2 = 1$. For ease of notation we denote the first filter by $a = (a_0, a_1, a_2) \in \mathbb{R}^3$ and the second filter by $b = (b_0, b_1) \in \mathbb{R}^2$. The filter of the end-to-end convolution is

$$\mu(a, b) = (a_0 b_0, a_1 b_0, a_2 b_0 + a_0 b_1, a_1 b_1, a_2 b_1).$$

According to (Kohn et al., 2022, Example 4.12), the neuromanifold is

$$\mathcal{M} = \{v \in \mathbb{R}^5 \mid v_0 v_3^2 + v_1^2 v_4 - v_1 v_2 v_3 = 0,\ v_2^2 - 4 v_0 v_4 \geq 0\}.$$

Thus, its Zariski closure is the hypersurface in $\mathbb{R}^5$ defined by $v_0 v_3^2 + v_1^2 v_4 - v_1 v_2 v_3 = 0$, that is singular at all points of the form $v = (v_0, 0, v_2, 0, v_4)$.

Now, we consider a smooth point $v \in \mathcal{M}$ (i.e., $v_1 \neq 0$ or $v_3 \neq 0$). As in the proof of Theorem 4, there is – up to scaling – only one choice of filters (a, b) such that $\mu(a, b) = v$. If both $v_1 \neq 0$ and $v_3 \neq 0$, then this choice is

$$a = (\frac{v_0}{v_1}, 1, \frac{v_4}{v_3}) \quad \text{and} \quad b = (v_1, v_3). \tag{15}$$

If precisely one of v_1 and v_3 is zero, then the preimage filters are $a = (v_2, v_3, v_4)$ and $b = (0, 1)$ for $v_1 = 0$, or $a = (v_0, v_1, v_2)$ and $b = (1, 0)$ for $v_3 = 0$.

Next, for a fixed $\delta \in \mathbb{R}$, we compute the possible scalars λ such that $\delta = \|\frac{1}{\lambda} b\|^2 - \|\lambda a\|^2$. We follow the proof of Proposition 2. For 2-layer networks, we see that (13) becomes a quadratic equation in λ^2:

$$\lambda^2 \|a\|^2 \cdot (\lambda^2 \|a\|^2 + \delta) = \|a\|^2 \|b\|^2.$$

Its only positive solution is $\lambda^2 = \frac{-\delta + \sqrt{\delta^2 + 4\|a\|^2 \|b\|^2}}{2\|a\|^2}$.

Theorem 4 tells us that we can compute $K^{(\delta)}(v)$ as $K_{(\lambda a, \frac{1}{\lambda} b)} = \frac{1}{\lambda^2} K_1 + \lambda^2 K_2$, where

$$K_1 = \begin{pmatrix} b_0^2 & 0 & b_0 b_1 & 0 & 0 \\ 0 & b_0^2 & 0 & b_0 b_1 & 0 \\ b_0 b_1 & 0 & b_0^2 + b_1^2 & 0 & b_0 b_1 \\ 0 & b_0 b_1 & 0 & b_1^2 & 0 \\ 0 & 0 & b_0 b_1 & 0 & b_1^2 \end{pmatrix} \text{ and } K_2 = \begin{pmatrix} a_0^2 & a_0 a_1 & a_0 a_2 & 0 & 0 \\ a_0 a_1 & a_1^2 & a_1 a_2 & 0 & 0 \\ a_0 a_2 & a_1 a_2 & a_0^2 + a_2^2 & a_0 a_1 & a_0 a_2 \\ 0 & 0 & a_0 a_1 & a_1^2 & a_1 a_2 \\ 0 & 0 & a_0 a_2 & a_1 a_2 & a_2^2 \end{pmatrix}.$$

This only depends on v and δ. For instance, if we assume that both $v_1 \neq 0$ and $v_3 \neq 0$, then K_1 and K_2 can be expressed only in terms of v using the substitution (15). Similarly, λ^2 can be written in terms of v and δ. In the balanced case, this simplifies to $\lambda^2 = \frac{\|b\|}{\|a\|} = \sqrt{\frac{v_1^2 + v_3^2}{\frac{v_0^2}{v_1^2} + 1 + \frac{v_4^2}{v_3^2}}}$. ♢

7.1.3 Proof of Lemma 5

Proof. Since $\mu(\theta)$ is a smooth point of $\mathcal{M}$, the derivative $d_\theta \mu$ is surjective onto $T_{\mu(\theta)}\mathcal{M}$, and its kernel is spanned by $\dot{\theta}_1 := (-w_1, w_2, 0, \ldots, 0), \ldots, \dot{\theta}_{H-1} := (0, \ldots, 0, -w_{H-1}, w_H)$, see Kohn et al. (2024). Due to $\dim(\Theta_\delta) = \dim(\mathcal{M})$, it is sufficient to show that this kernel intersects the tangent space $T_\theta \Theta_\delta$ only at 0. For that, we consider a point $\lambda_1 \dot{\theta}_1 + \ldots + \lambda_{H-1} \dot{\theta}_{H-1}$ that is contained in $T_\theta \Theta_\delta$. Our aim is to show that all λ_i must be equal to 0.

We begin by proving inductively that each λ_i is of the form $\lambda_1 \cdot c_i$ for some positive scalar c_i. This is immediate for $i = 1$. For $i = 2$, we plug in the point $\sum_j \lambda_j \dot{\theta}_j$ into the first condition from (9). This yields the equality $(\lambda_1 - \lambda_2)\|w_2\|^2 = -\lambda_1 \|w_1\|^2$. Therefore, $\lambda_2 = \lambda_1 \cdot c_2$, where $c_2 := \frac{\|w_1\|^2 + \|w_2\|^2}{\|w_2\|^2} > 0$. Similarly, for $i > 2$, the conditions in (9) yield $(\lambda_{i-1} - \lambda_i)\|w_i\|^2 = -\lambda_1 \|w_1\|^2$, which can be rewritten as $\lambda_i = \frac{1}{\|w_i\|^2}(\lambda_{i-1}\|w_i\|^2 + \lambda_1 \|w_1\|^2)$. Applying the induction hypothesis, we obtain $\lambda_i = \lambda_1 \cdot c_i$, where $c_i := \frac{1}{\|w_i\|^2}(c_{i-1}\|w_i\|^2 + \|w_1\|^2) > 0$.

On the other hand, the last condition in (9) gives us that $\lambda_{H-1}\|w_H\|^2 = -\lambda_1 \|w_1\|^2$. Thus, we have $\lambda_1 \cdot c_{H-1} = \lambda_{H-1} = -\lambda_1 \cdot \frac{\|w_1\|^2}{\|w_H\|^2}$, which shows that λ_1 must be 0, and so all λ_i are 0. □

7.1.4 Proof of Corollary 6 and Examples

Proof. The first part of the claim follows immediately from Lemma 5 and Theorem 4. More concretely, Lemma 5 states that the derivatives of $\mu_\delta := \mu|_{\Theta_\delta}$ are bijective. For such a submersion with finite fibers, the pushforward metric can be computed via averaging over the metrics on the fibers:

$$\begin{aligned}\|\dot{v}\|_{\mu_\delta}^2 :=& \frac{1}{|\mu_\delta^{-1}(v)|}\sum_{\theta\in\mu_\delta^{-1}(v)} \|(d_\theta\mu_\delta)^{-1}(\dot{v})\|^2 \\ =& \frac{1}{|\mu_\delta^{-1}(v)|}\sum_{\theta\in\mu_\delta^{-1}(v)} ((d_\theta\mu_\delta)^{-1}(\dot{v}))^\top (d_\theta\mu_\delta)^{-1}(\dot{v}) \\ =& \frac{1}{|\mu_\delta^{-1}(v)|}\sum_{\theta\in\mu_\delta^{-1}(v)} \dot{v}^\top (J_\theta J_\theta^\top)^{-1}\dot{v} = \frac{1}{|\mu_\delta^{-1}(v)|}\sum_{\theta\in\mu_\delta^{-1}(v)} \dot{v}^\top K_\theta^{-1}\dot{v},\end{aligned}$$

where $\dot{v}\in T_v\mathcal{M}$ for some $v\in\mathrm{Reg}(\mathcal{M})$. By Theorem 4, all $\theta\in\mu_\delta^{-1}(v)$ have the same neural tangent kernel $K_\theta = K^{(\delta)}(v)$, and so we conclude that the pushforward metric is $\|\dot{v}\|_{\mu_\delta}^2 = \dot{v}^\top(K^{(\delta)}(v))^{-1}\dot{v}$. □

Example 16. We consider the network architecture from Example 15. Recall that the singular points of the neuromanifold $\mathcal{M}$ are of the form $v = (v_0, 0, v_2, 0, v_4)$. They are parametrized by filters of the form $a = (a_0, 0, a_2)$ and $b = (b_0, b_1)$.

If (a_0, a_2) and (b_0, b_1) are linearly independent, then the resulting end-to-end filter v has two distinct (up to scaling) parametrizations, namely $v = \mu(a_0, 0, a_2, b_0, b_1) = \mu(b_0, 0, b_1, a_0, a_2)$. At both parameter tuples, the derivative of μ is of maximal rank 4, and so is the neural tangent kernel. However, in general, $K_{(a_0,0,a_2,b_0,b_1)} \neq K_{(b_0,0,b_1,a_0,a_2)}$, even for fixed δ-invariant. One concrete balanced (i.e., $\delta = 0$) example is $(a_0, a_2) = (1, 2)$ and $(b_0, b_1) = (2, 1)$. In this case,

$$K_{(a_0,0,a_2,b_0,b_1)} = \begin{pmatrix} 5 & 0 & 4 & 0 & 0 \\ 0 & 4 & 0 & 2 & 0 \\ 4 & 0 & 10 & 0 & 4 \\ 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 4 & 0 & 5 \end{pmatrix} \text{ and } K_{(b_0,0,b_1,a_0,a_2)} = \begin{pmatrix} 5 & 0 & 4 & 0 & 0 \\ 0 & 1 & 0 & 2 & 0 \\ 4 & 0 & 10 & 0 & 4 \\ 0 & 2 & 0 & 4 & 0 \\ 0 & 0 & 4 & 0 & 5 \end{pmatrix}$$

If (a_0, a_2) and (b_0, b_1) are linearly dependent, they are the same up to scaling. This means that, up to scaling, v has only one preimage under μ. That preimage $(a_0, 0, a_2, b_0, b_1)$ is a critical point of the map μ, i.e., the rank of the neural tangent kernel $K_{(a_0,0,a_2,b_0,b_1)}$ is less than the generic rank 4. ♢

Example 17. We consider the same network architecture as in Example 15. Suppose $v\in\mathcal{M}$ is such that v_0 and v_1 are not equal to 0. Such a v is in particular a smooth point of the neuromanifold $\mathcal{M}$. We show that for different δ we obtain different $K^{(\delta)}(v)$.

We fix filters a and b such that $\mu(a, b) = v$. Our assumptions on v imply that $a_0 \neq 0$ and $a_1 \neq 0$. By Example 15, we have $K^{(\delta)}(v) = K_{(\lambda a, \frac{1}{\lambda}b)}$, where $\lambda^2 = \frac{-\delta+\sqrt{\delta^2+4\|a\|^2\|b\|^2}}{2\|a\|^2}$. In particular, the entry of $K^{(\delta)}(v)$ at position $(2, 1)$ is equal to $\lambda^2 a_0 a_1 \neq 0$. The function which to $\delta\in\mathbb{R}$ associates $\frac{-\delta+\sqrt{\delta^2+4\|a\|^2\|b\|^2}}{2\|a\|^2}$ is strictly decreasing and therefore bijective. Hence, distinct δ's yield distinct values for λ^2 and thus for the $(2, 1)$-entry of $K^{(\delta)}(v)$. ♢

7.1.5 Proof of Proposition 8 and Example

To prove Proposition 8, we start by rewriting the squared-error loss in terms of filters instead of convolutions. Under the assumption $N \geq k$, it is shown in (Kohn et al., 2024, Corollary 7.2) that, for almost all input training data $X_1, \ldots, X_N \in \mathbb{R}^{d_0}$, minimizing $\ell_{\mathcal{D}}(\alpha)$ is equivalent to minimizing

$$\ell_{A,u}(w) := (w - u)^\top A(w - u),$$

where w is the filter of the convolution α, u is a data filter that depends on the training data $\mathcal{D}$, and A is a symmetric positive-definite matrix that depends on the input training data $X_1, \dots, X_N$. We denote the corresponding loss in parameter space by $\mathcal{L}_{A,u} := \ell_{A,u} \circ \mu$.

When the data $\mathcal{D}$ is generic, the vector Au is generic as well (see (Kohn et al., 2024, Section 7)). We use this to prove the technical Lemma 18, that is based on the following standard construction from algebraic geometry. Given a k-dimensional vector space V, its projectivization $\mathbb{P}(V)$ is a $(k-1)$-dimensional projective space. The projectivization $\mathbb{P}(V^*)$ of the dual vector space V^* consists of all the hyperplanes in $\mathbb{P}(V)$. Given a subvariety $X \subseteq \mathbb{P}(V)$, we say that a hyperplane $H \subseteq \mathbb{P}(V)$ is tangent to X if there is a smooth point $x \in X$ such that H contains the embedded tangent space $T_x X \subseteq \mathbb{P}(V)$. Thinking of the hyperplanes in $\mathbb{P}(V)$ as points of the dual projective space $\mathbb{P}(V^*)$, the set of hyperplanes that are tangent to X is a subset of $\mathbb{P}(V^*)$. Its Zariski closure is a proper subvariety of $\mathbb{P}(V^*)$, called the *dual variety* of X [Chapter 1]Gelfand et al. (1994).

Lemma 18. *Let $\theta = (w_1, \dots, w_H)$ with $w_i \in \mathbb{R}^{k_i} \setminus \{0\}$, and let $N \geq k$. For almost all N-tuples of data $\mathcal{D}$, the vector Au is not orthogonal to the image of the derivative $d_\theta \mu$.*

Proof. Let us first consider the case that $\mu(\theta)$ is a smooth point of the neuromanifold $\mathcal{M}$. Then, the image of $d_\theta \mu$ is the tangent space of $\mathcal{M}$ at $\mu(\theta)$. A vector $v = (v_0, \dots, v_{k-1})$ is orthogonal to this tangent space if and only if the hyperplane $P_v := \{(h_0, \dots, h_{k-1}) \mid \sum_i v_i h_i = 0\}$ contains the tangent space. Due to the genericity of the data $\mathcal{D}$, the vector Au and the hyperplane P_{Au} are generic. Since the dual variety of the Zariski closure $\mathbb{P}\overline{\mathcal{M}}$ of the projective neuromanifold $\mathbb{P}\mathcal{M}$ is a proper subvariety of $(\mathbb{P}^{k-1})^*$, the generic hyperplane P_{Au}, considered as a point in the dual space $(\mathbb{P}^{k-1})^*$, is not contained in the dual variety of $\mathbb{P}\overline{\mathcal{M}}$. Hence, the hyperplane P_{Au} is not tangent to $\mathbb{P}\overline{\mathcal{M}}$, and so the vector Au is not orthogonal to $T_{\mu(\theta)}\mathcal{M} = \mathrm{im}(d_\theta \mu)$.

Now, we consider arbitrary θ where all layer filters are nonzero. We denote by r the rank of the derivative $d_\theta \mu$. Since $d_\theta \mu(\theta) = H \cdot \mu(\theta) \neq 0$, we have that $r \geq 1$. We further denote by X_r the set of all filter tuples θ' such that the rank of $d_{\theta'} \mu$ is r, and by Y_r the Zariski closure of $\mu(X_r)$. Note that $Y_{\dim \mathcal{M}} = \overline{\mathcal{M}}$. If $\mu(\theta)$ is a smooth point of Y_r, then $\mathrm{im}(d_\theta \mu)$ is the tangent space $T_{\mu(\theta)} Y_r$. Otherwise, since $\mathrm{im}(d_\theta \mu)$ has dimension r, this linear space is contained in the Zariski closure of the set of all tangent spaces at smooth points of Y_r. Therefore, the dual variety of $\mathbb{P}Y_r$ contains all hyperplanes that contain the r-dimensional $\mathrm{im}(d_\theta \mu)$. Since, as above, the hyperplane P_{Au} is a generic point in $(\mathbb{P}^{k-1})^*$, it is not contained in the dual variety of $\mathbb{P}Y_r$. Thus, it does not contain $\mathrm{im}(d_\theta \mu)$, and so Au is not orthogonal to $\mathrm{im}(d_\theta \mu)$. □

Proof of Proposition 8. For generic weight initialization, the δ_i (for $i = 1, \dots, H-1$) are generic as well, and also the $\sum_{i=k}^{l} \delta_i$. Since the invariants (1) stay constant during gradient flow, at the point $\theta = (w_1, \dots, w_H)$ of convergence, at most one of the filters w_l can be zero. Indeed, if two filters w_k and w_l are equal to zero then $\sum_{i=k}^{l} \delta_i$ has to be zero, which is not the case because of genericity. If none of the filters is zero, we are done. Therefore, we assume now that precisely one of the filters, say w_l, is zero.

We now consider the Hessian matrix $\mathcal{H}$ at θ of the loss function $\mathcal{L}_{A,u}$ that is equivalent to the squared-error loss. The Hessian matrix consists of blocks, each corresponding to a pair of filters (w_i, w_j). We denote these blocks by $\mathcal{H}_{(w_i, w_j)}$. Those blocks encode bilinear functions

$$d^2_{w_i, w_j} \mathcal{L}_{A,u} : (\dot{w}_i, \dot{w}_j) \mapsto 2 \left(\mu_i(\dot{w}_i)^\top A \mu_j(\dot{w}_j) - \mu_{i,j}(\dot{w}_i, \dot{w}_j)^\top Au \right),$$

where $\mu_i(\dot{w}_i) = \mu(w_1, \dots, w_{i-1}, \dot{w}_i, w_{i+1}, \dots, w_H)$, μ_j is analogously defined, and similarly $\mu_{i,j}$ substitutes both w_i and w_j by $\dot{w}_i$ and $\dot{w}_j$ (if $i = j$, then $\mu_{i,i} = 0$). In the case that neither i nor j are equal to l, then our assumption $w_l = 0$ implies that $d^2_{w_i, w_j} \mathcal{L}_{A,u} = 0$. Thus, also the corresponding Hessian block $\mathcal{H}_{(w_i, w_j)}$ is zero.

Now we show that not all Hessian blocks with $j = l$ and $i \neq l$ are zero. Those blocks encode bilinear forms $d^2_{w_i, w_l} \mathcal{L}_{A,u} = -2\mu_{i,l}^\top Au$. We assume for contradiction that all those forms are zero. We pick a nonzero filter $c \in \mathbb{R}^{k_l}$ and consider the map $\tilde{\mu} : (w'_1, \dots, w'_{l-1}, w'_{l+1}, \dots, w'_H) \mapsto \mu(w'_1, \dots, w'_{l-1}, c, w'_{l+1}, \dots, w'_H)$. We also write $\tilde{\theta}$ for the filter tuple that is obtained from θ by omitting w_l. Then, $d_{\tilde{\theta}} \tilde{\mu}(\dot{w}_1, \dots, \dot{w}_{l-1}, \dot{w}_{l+1}, \dots, \dot{w}_H)^\top Au = \sum_{i \neq l} \mu_{i,l}(\dot{w}_i, c)^\top Au = 0$ for all $\dot{w}_i \in \mathbb{R}^{k_i}$. This means that the vector Au is orthogonal to the image of the

derivative $d_{\tilde{\theta}}\tilde{\mu}$. However, this contradicts Lemma 18 applied to the $(H-1)$-layer network parametrization map $\tilde{\mu}$.

Therefore, one of the blocks $\mathcal{H}_{(w_i,w_l)}$ with $i \neq l$ has a non-zero entry. We denote the coordinates of that entry in the Hessian matrix $\mathcal{H}$ by (s,t) (such that the index t corresponds to the filter w_l). We consider the s-th and t-th standard basis vectors e_s and e_t of $\mathbb{R}^{\sum_{j=1}^{H} k_j}$, and the vector $v = \alpha e_s + e_t$ for some scalar $\alpha \in \mathbb{R}$. We have

$$v^\top \mathcal{H} v = \alpha^2 \mathcal{H}_{ss} + 2\alpha \mathcal{H}_{st} + \mathcal{H}_{tt} = 2\alpha \mathcal{H}_{st} + \mathcal{H}_{tt}.$$

Since $\mathcal{H}_{st} \neq 0$, we can choose α such that $v^\top \mathcal{H} v < 0$. Hence, θ is a critical point for which the Hessian has at least one negative eigenvalue. This is called a *strict saddle*, and we know from Bah et al. (2022) that, for almost all initializations, gradient flow avoids strict saddles. Thus, θ with one of the layer filters equal to zero could not have been the point of convergence. □

Example 19. We change the previous running example to both layers having stride one. Given two layers $a = (a_0, a_1, a_2)$ and $b = (b_0, b_1)$, the end-to-end filter is $\mu(a,b) = (a_0b_0, a_0b_1 + a_1b_0, a_1b_1 + a_2b_0, a_2b_1)$. It corresponds to a cubic polynomial $a_0b_0x^3 + (a_0b_1 + a_1b_0)x^2y + (a_1b_1 + a_2b_0)xy^2 + a_2b_1y^3$ that factors into a quadratic term $a_0x^2 + a_1xy + a_2y^2$ and a linear one $b_0x + b_1y$. Every cubic polynomial has such a factorization, so the neuromanifold is $\mathcal{M} = \mathbb{R}^4$. Cubic polynomials with only one real root have precisely one such factorization (up to scaling), while cubics with three distinct real roots have three distinct factorizations (up to scaling).

For instance, the filter $v = (0,1,1,0)$ that corresponds to the cubic $x^2y + xy^2 = xy(x+y)$ has the following factorizations according to the network architecture:

1. $a = \sqrt[4]{2} \cdot (0,1,0)$ and $b = \frac{1}{\sqrt[4]{2}} \cdot (1,1)$
2. $a = \frac{1}{\sqrt[4]{2}} \cdot (1,1,0)$ and $b = \sqrt[4]{2} \cdot (0,1)$
3. $a = \frac{1}{\sqrt[4]{2}} \cdot (0,1,1)$ and $b = \sqrt[4]{2} \cdot (1,0)$

Each of these filter pairs has been scaled such that $\|b\|^2 = \|a\|^2$, i.e., $\delta = 0$.

The neural tangent kernel for this network architecture is

$$K_{a,b} = \begin{pmatrix} a_0^2 + b_0^2 & a_0a_1 + b_0b_1 & a_0a_2 & 0 \\ a_0a_1 + b_0b_1 & a_0^2 + a_1^2 + b_0^2 + b_1^2 & a_0a_1 + a_1a_2 + b_0b_1 & a_0a_2 \\ a_0a_2 & a_0a_1 + a_1a_2 + b_0b_1 & a_1^2 + a_2^2 + b_0^2 + b_1^2 & a_1a_2 + b_0b_1 \\ 0 & a_0a_2 & a_1a_2 + b_0b_1 & a_2^2 + b_1^2 \end{pmatrix}.$$

For the three factorizations of $v = (0,1,1,0)$ above, the matrices $K_{a,b}$ are

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad \text{and} \quad \frac{1}{\sqrt{2}} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

showing that the neural tangent kernel does not only depend on the end-to-end filter v and the invariant δ. ♢

7.2 Proofs of Section 3

7.2.1 Proof of Lemma 9

Proof. We abbreviate (10) by $(\alpha_{w,\boldsymbol{s}}(X))_{\boldsymbol{i}} = \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{k}-\boldsymbol{1}} w_{\boldsymbol{j}} X_{\boldsymbol{i}\boldsymbol{s}+\boldsymbol{j}}$. The assertion follows inductively from composing two such convolutions:

$$\begin{aligned}(\alpha_{w_2,\boldsymbol{s}_2} \circ \alpha_{w_1,\boldsymbol{s}_1})(X)_{\boldsymbol{i}} &= \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{k}_2-\boldsymbol{1}} w_{2,\boldsymbol{j}} \alpha_{w_1,\boldsymbol{s}_1}(X)_{\boldsymbol{i}\boldsymbol{s}_2+\boldsymbol{j}} = \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{k}_2-\boldsymbol{1}} w_{2,\boldsymbol{j}} \sum_{\boldsymbol{l}=\boldsymbol{0}}^{\boldsymbol{k}_1-\boldsymbol{1}} w_{1,\boldsymbol{l}} X_{(\boldsymbol{i}\boldsymbol{s}_2+\boldsymbol{j})\boldsymbol{s}_1+\boldsymbol{l}} \\ &= \sum_{\boldsymbol{m}=0}^{(\boldsymbol{k}_2-\boldsymbol{1})\boldsymbol{s}_1+\boldsymbol{k}_1-\boldsymbol{1}} v_{\boldsymbol{m}} X_{\boldsymbol{i}\boldsymbol{s}_2\boldsymbol{s}_1+\boldsymbol{m}},\end{aligned}$$

where $I_{\boldsymbol{m}} := \{(\boldsymbol{j},\boldsymbol{l}) \mid \boldsymbol{0} \le \boldsymbol{j} < \boldsymbol{k}_2,\ \boldsymbol{0} \le \boldsymbol{l} < \boldsymbol{k}_1,\ \boldsymbol{j}\boldsymbol{s}_1 + \boldsymbol{l} = \boldsymbol{m}\}$ and $v_{\boldsymbol{m}} := \sum_{(\boldsymbol{j},\boldsymbol{l})\in I_{\boldsymbol{m}}} w_{2,\boldsymbol{j}} w_{1,\boldsymbol{l}}$. Note that $v_{\boldsymbol{m}}$ coincides with the coefficient of the polynomial $\pi_{\boldsymbol{s}_1}(w_2)\pi_{\boldsymbol{1}}(w_1)$, where the degrees of the variables $y_1, \dots, y_D$ are $m_1, \dots, m_D$. □

7.2.2 Proof of Corollary 10

Proof. Our assumption means that, for every layer l, at least two variable pairs (x_{m_1}, y_{m_1}) and (x_{m_2}, y_{m_2}) appear in the polynomials in $\mathbb{R}[(\boldsymbol{x},\boldsymbol{y})^{\boldsymbol{t}_l}]_{\boldsymbol{k}_l-\boldsymbol{1}}$ that correspond to the l-th layer filter. Hence, the polynomials in $\mathbb{R}[(\boldsymbol{x},\boldsymbol{y})^{\boldsymbol{t}_l}]_{\boldsymbol{k}_l-\boldsymbol{1}}$ are generically irreducible. (This is a standard fact from algebraic geometry, following from Bertini's theorem; see the proof of (Kohn et al., 2022, Corollary 4.16) for a formal argument.) Therefore, for a generic filter tuple $\theta = (w_1, \dots, w_H)$, the unordered set of filters $\{w_1, \dots, w_H\}$ can be recovered (up to scalars) from the end-to-end filter $\mu(\theta)$ by computing the factorization of $\pi_{\boldsymbol{1}}(\mu(\theta))$ into its irreducible factors $\{\pi_{\boldsymbol{t}_1}(w_1), \dots, \pi_{\boldsymbol{t}_H}(w_H)\}$. □

7.2.3 Proof of Proposition 12

Proof. The same proof as for Proposition 2 applies. □

7.2.4 Proof of Theorem 13

Proof. We see from Lemma 9 that every fiber of the projectivization $\mu_{\mathbb{P}}$ of μ is finite. Moreover, Corollary 10 says that, after possibly modding out some permutations of the input layers (cf. Observation 11), the map $\mu_{\mathbb{P}}$ becomes birational (i.e., its generic fiber is a singleton). For such a map, a standard fact from algebraic geometry (Kohn et al., 2017, Lemma 3.2) tells us that the image of $\mu_{\mathbb{P}}$ (over the complex numbers) is smooth precisely at those points that have a single preimage under $\mu_{\mathbb{P}}$ (modulo the necessary layer permutations) and where the derivative of $\mu_{\mathbb{P}}$ at that preimage has maximal rank. In other words, for every end-to-end filter v in the smooth locus of $\mathcal{M}$, there is a unique filter tuple θ, up to scalars and possibly some layer permutations, that parametrizes v; moreover, $d_\theta\mu$ has maximal rank. By Proposition 12, the invariants δ fix the scalars in the filter tuple θ, up to signs. The signs and the possible layer permutations do not affect the neural tangent kernel K_θ, by Observations 3 and 11. Finally, the proofs of Lemma 5 and Corollary 6 hold verbatim. □

7.3 Example and Proof of Section 4

Example 20. For $\Delta_1 = diag(0, 15/4)$, the neural tangent kernel is parameter-dependent. Indeed, let the end-to-end product matrix be equal to the identity $W = I$. Consider $V_2 = \mathrm{diag}(1, 2)$ and $V_1 = \mathrm{diag}(1, 1/2)$. Hence $V_2V_1 = I_2 = W$ and the balancedness constant matrix is indeed equal to $V_2^TV_2 - V_1V_1^T = \mathrm{diag}(0, 15/4) = \Delta_1$. Consider then $U_2 = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}$ and $U_1 = \begin{pmatrix} 0 & 1 \\ 1/2 & 0 \end{pmatrix}$, we have again $U_2U_1 = I_2$ and $U_2^TU_2 - U_1U_1^T = \mathrm{diag}(0, 15/4) = \Delta_1$. However, using the formula (7) for the NTK in this case, we have $K_{V_2,V_1}^{(\Delta_1)} =$

$\nabla_W \ell(W) V_1^T V_1 + V_2 V_2^T \nabla_W \ell(W) \neq \nabla_W \ell(W) U_1^T U_1 + U_2 U_2^T \nabla_W \ell(W) = K_{U_2,U_1}^{(\Delta_1)}$. Hence the neural tangent kernel is parameter-dependent. Using Remark 1, we conclude that this is the case for almost all Δ_1. We can also easily generalize this example to any depth and width of the network.

7.3.1 Proof of Theorem 14

Proof. Consider the map $\mu : GL_d(\mathbb{R})^H \to GL_d(\mathbb{R})$ defined by $\mu(W_H, \dots, W_1) = W_H \cdots W_1$. Then the fibers of a generic W can be written as

$$(W_H G_{H-1}, G_{H-1}^{-1} W_{H-1} G_{H-2}, \dots, G_2^{-1} W_2 G_1, G_1^{-1} W_1),$$

where $(W_H, \dots, W_1)$ are any fixed matrices such that their product is equal to W. Let us prove that the fiber such that all the layers are balanced leads then to all G_i being orthogonal. It was shown in Arora et al. (2018) that we have for all balanced weights $(W'_H, \dots, W'_1)$ in the fiber of W, for all $j \in \{1, \dots, H\}$ the following equations

$$W'_H \cdots W'_j (W'_H \cdots W'_j)^T = (WW^T)^{\frac{H-j+1}{H}}.$$

Let $(W_H, \dots, W_1)$ be balanced matrices in the fiber of W, and $G_1, \dots, G_{H-1}$ invertible matrices such that $(W_H G_{H-1}, G_{H-1}^{-1} W_{H-1} G_{H-2}, \dots, G_2^{-1} W_2 G_1, G_1^{-1} W_1)$ are also balanced matrices (in the fiber of W). We have, for $j \in \{1, \dots, H-1\}$,

$$W_H \cdots W_{j+1} G_j G_j^T (W_H \cdots W_{j+1})^T = (WW^T)^{\frac{H-j}{H}}$$
$$\text{and} \qquad W_H \cdots W_{j+1} (W_H \cdots W_{j+1})^T = (WW^T)^{\frac{H-j}{H}}.$$

Hence by unicity of the symmetric PSD root of a symmetric PSD matrix we have

$$W_H \cdots W_{j+1} G_j G_j^T (W_H \cdots W_{j+1})^T = W_H \cdots W_{j+1} (W_H \cdots W_{j+1})^T.$$

Therefore, since all weight matrices are invertible we obtain $G_j G_j^T = I$ so that the G_j, $j = 1, \dots, H-1$ are orthogonal. Conversely if some weight layers are balanced and in the fiber of W, then one can easily check that for G_j orthogonal, the layers are also balanced. □