EIGENSPACE RESTRUCTURING: A PRINCIPLE OF SPACE AND FREQUENCY IN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding the fundamental principles behind the massive success of neural networks is one of the most important open questions in deep learning. However, due to the highly complex nature of the problem, progress has been relatively slow. In this note, through the lens of infinite-width networks, a.k.a. neural kernels, we present one such principle resulting from hierarchical locality. It is well-known that the eigenstructure of infinite-width multilayer perceptrons (MLPs) depends solely on the concept *frequency*, which measures the order of interactions. We show that the topologies from convolutional networks (CNNs) restructure the associated eigenspaces into finer subspaces. In addition to frequency, the new structure also depends on the concept *space* — the distance among interaction terms, defined via the length of a minimum spanning tree containing them. The resulting fine-grained eigenstructure dramatically improves the network's learnability, empowering them to simultaneously model a much richer class of interactions, including long-range-low-frequency interactions, short-range-high-frequency interactions, and various interpolations and extrapolations in-between. Finally, we show that increasing the depth of a CNN can improve the inter/extrapolation resolution and, therefore, the network's learnability.

1 INTRODUCTION

Learning in high dimensions is commonly believed to suffer from the curse of dimensionality, in which the number of samples required to solve the problem grows rapidly (often polynomially) with the dimensionality of the input. Nevertheless, modern neural networks often exhibit an astonishing power to tackle a wide range of highly complex and high-dimensional real-world problems, many of which were thought to be out-of-scope of known methods (Krizhevsky et al., 2012; Vaswani et al., 2017; Devlin et al., 2018; Silver et al., 2016; Senior et al., 2020; Kaplan et al., 2020). What are the mathematical principles that govern the astonishing power of neural networks? This question perhaps is the most crucial research question in the theory of deep learning because such principles are also the keys to resolve fundamental questions in the practice of machine learning such as (out-of-distribution) generalization (Zhang et al., 2021), calibration (Ovadia et al., 2019), interpretability (Montavon et al., 2018), robustness (Goodfellow et al., 2014).

Unarguably, there can be more than one of such principles. They are related to one or more of the three basic ingredients of machine learning methods: the data, the model and the inference algorithm. Among them, the models, a.k.a. architectures of neural networks are the most crucial innovation in deep learning that set it apart from classical machine learning methods. More importantly, the current revolution in machine learning is initialized by the (re-)introduction of convolution-based architectures (Krizhevsky et al., 2012; Lecun, 1989), and subsequent breakthroughs are often driven by the discovery or application of novel architectures (Vaswani et al. (2017); Devlin et al. (2018)). As such, identifying and understanding fundamental roles of architectures are of great importance.

In this paper, we take a step forwards by leveraging recent developments in overparameterized networks (Poole et al. (2016); Daniely et al. (2016); Schoenholz et al. (2017); Lee et al. (2018); Matthews et al. (2018); Xiao et al. (2018); Jacot et al. (2018); Du et al. (2018); Novak et al. (2019); Lee et al. (2019) and many others.) These developments have discovered an important connection between neural networks and kernel machines: the Neural Network Gaussian Process (NNGP) kernels and the neural tangent kernels (NTKs). Under certain scaling limits, the former describes the

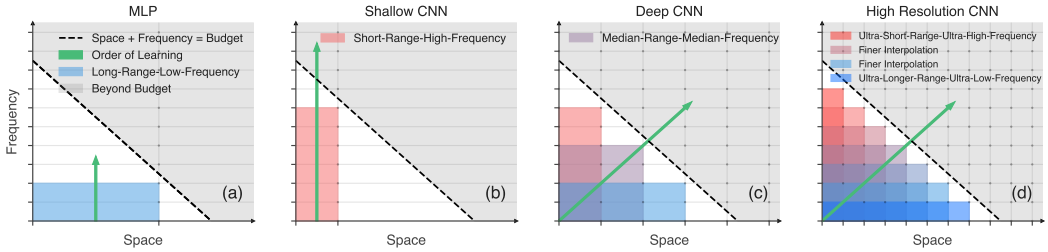


Figure 1: **Architectural Inductive Biases.** An demonstration of learnable functions vs architectures for four families of architectures. Each shaded box indicates the maximum learnable eigenspaces within a given compute budget (**Dashed Line.**) From left to right: (a) MLPs can model **Long-Range-Low-Frequency** interactions; (b) S-CNNs can model **Short-Range-High-Frequency** interactions; (c) Additionally, D-CNNs can also model interactions between **LRLF** and **SRHF**, a.k.a., **Median-Range-Median-Frequency** interactions. (d) Finally, HS-CNNs can additionally model interactions of **Ultra-Short-Range-Ultra-High-Frequency**, **Ultra-Long-Range-Ultra-Low-Frequency**, and **finer interpolations** in-between. The **Green Arrow** indicates the direction of expansion of learnable functions when increasing the compute budget.

distribution of the outputs of a randomly initialized network (a.k.a. *prior*), and the latter can describe the network's gradient descent dynamics. Although recent work (Ghorbani et al., 2019; Yang & Hu, 2020) has identified several limitations of using them in studying the feature learning dynamics of practical networks, we show that they do capture several crucial aspects of the architectural inductive biases.

Our main contribution is an *eigenspace restructuring* theorem. It characterizes a mathematical connection between a network's architecture and its learnability through a trade-off between *space* and *frequency*, providing novel insights behind the mystery power of deep CNNs (more generally, hierarchical locality (Deza et al., 2020).) We summarize our main contribution below; see Fig. 1.

1. The learning order (see **Green Arrow** in Fig. 1) of eigen-functions is governed by the learning index (LI), the sum of the frequency index (FI) and the spatial index (SI), which can be characterized precisely by the network's topology.
2. There is a trade-off between *space* and *frequency*: within a fixed (compute/data) budget, it is impossible to model generic Long-Range-High-Frequency interactions. MLPs can model **Long-Range-Low-Frequency (LRHF)** interactions but fail to model **Short-Range-High-Frequency (SRHF)**, while shallow CNNs (S-CNNs) are the opposite. Remarkably, deep CNNs (D-CNNs) can simultaneously model both and various interpolating interactions between them (e.g., **Median-Range-Median-Frequency (MRMF)**.)
3. In addition, *high-resolution* CNNs (HS-CNNs, EfficientNet-type of model scaling) further broaden the class of learnable functions to contain (1) **extrapolation**: **Ultra-Long-Range-Ultra-Low-Frequency** and the **dual** interactions and (2) **finer interpolating** interactions.
4. Finally, we verify the above claims empirically for neural kernel methods and finite-width networks using *SGD + Momentum* for dataset and networks of practical sizes.

2 Linear and Linearized Models

As a warm up exercise, we briefly go through the training dynamics of linear models. Let $(\mathcal{X}, \mathcal{Y})$ denote the inputs and labels, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}$. Assume $J : \mathbb{R}^d \to \mathbb{R}^n$ is a feature map and the task is to learn a linear function $f(x, \theta) = J(x)\theta$ to minimize the MSE objective $\frac{1}{2}\sum_{x\in\mathcal{X}} |f(x,\theta) - y|^2$. Let $\mathcal{R}(\mathcal{X}, \theta) = f(\mathcal{X}, \theta) - \mathcal{Y}$ be the residual of the predictions of $\mathcal{X}$. Then the gradient flow dynamics can be written as

$$\frac{d}{dt}\mathcal{R}(\mathcal{X}, \theta) = -J(\mathcal{X})J^T(\mathcal{X})\mathcal{R}(\mathcal{X}, \theta) \equiv -\mathcal{K}(\mathcal{X}, \mathcal{X})\mathcal{R}(\mathcal{X}, \theta) \tag{1}$$

Since the feature kernel $\mathcal{K}(\mathcal{X}, \mathcal{X}) = J(\mathcal{X})J^T(\mathcal{X})$ is constant in time, the above ODE can be solved in closed form. Let the $m = |\mathcal{X}|$ the cardinality of $\mathcal{X}$ and $\hat{\mathcal{K}}(j)/u_j$ be the j-th eigenvalue/eigenvector

of $\mathcal{K}(\mathcal{X}, \mathcal{X})$ in descending order. By initializing $\theta = 0$ at time $t = 0$ and denoting the projection by $\eta_j = u_j^T \mathfrak{R}(\mathcal{X}, 0)$, the dynamics of the residual and the loss can be reduced to

$$\mathfrak{R}(\mathcal{X}, \theta_t) = \sum_{j \in [m]} e^{-\hat{\mathcal{K}}(j)t} \eta_j u_j, \quad \mathcal{L}(\theta_t) = \frac{1}{2} \sum_{j \in [m]} e^{-2\hat{\mathcal{K}}(j)t} \eta_j^2 \tag{2}$$

Therefore, the amount of time needed for learning the j-th eigenfunction is about $t \sim \hat{\mathcal{K}}(j)^{-1}$. The larger $\hat{\mathcal{K}}(j)$ is, the shorter amount of time it takes to learn u_j.

Although simple, linear models provide us with the most useful intuition behind the relation between "eigenstructures" and learning dynamics.

2.1 Linearized Neural Networks: NNGP Kernels and NT Kernels

Let $f(\theta, x)$ be a general function, e.g. f is neural network parameterized by θ. Similarly,

$$\frac{d}{dt} \mathfrak{R}(\mathcal{X}, \theta) = -J(\mathcal{X}; \theta) J^T(\mathcal{X}; \theta) \mathfrak{R}(\mathcal{X}, \theta) \equiv -\mathcal{K}(\mathcal{X}, \mathcal{X}; \theta) \mathfrak{R}(\mathcal{X}, \theta) \,. \tag{3}$$

However, the kernel $\mathcal{K}(\mathcal{X}, \mathcal{X}; \theta)$ depends on θ via the Jacobian $J(\mathcal{X}; \theta)$ and evolves with time. The above system is unsolvable in general. However, under certain parameterization methods (e.g. Sohl-Dickstein et al. (2020)) and when the neural network is sufficient wide, this kernel does not change much during training and converges to a deterministic kernel called the NTK (Jacot et al., 2018),

$$\mathcal{K}(\mathcal{X}, \mathcal{X}; \theta) \to \Theta(\mathcal{X}, \mathcal{X}) \quad \text{as width} \to \infty. \tag{4}$$

The residual dynamics becomes a constant coefficient ODE again $\dot{\mathfrak{R}}(\mathcal{X}, \theta) = -\Theta(\mathcal{X}, \mathcal{X}) \mathfrak{R}(\mathcal{X}, \theta)$. To solve this system, we need the initial value of $\mathfrak{R}(\mathcal{X}, \theta)$. Since the parameters θ are often initialized using iid standard Gaussian variables, as the width approach infinity, the logits $f(\mathcal{X}; \theta)$ converge to a Gaussian process (GP), known as the neural network Gaussian process (NNGP). Specifically, $f(\mathcal{X}; \theta) \sim \mathcal{N}(\mathbf{0}; \mathfrak{K}(\mathcal{X}, \mathcal{X}))$, where $\mathfrak{K}$ is the NNGP kernel. Note that one can also treat infinite-width networks as Bayesian models, a.k.a. Bayesian Neural Networks, and apply Bayesian inference to compute the posteriors. This approach is equivalent to training *only* the network's classification layer (Lee et al., 2019) and the gradient descent dynamics is describe by the kernel $\mathfrak{K}$.

As such, there are two natural kernels, the NTK Θ and the NNGP kernel $\mathfrak{K}$, associated to infinite-width networks, whose training dynamics are governed by constant coefficient ODEs. To make progress, it is tempting to apply Mercer's Theorem to eigendecompose Θ and $\mathfrak{K}$, e.g.,

$$\mathfrak{K}(x, \bar{x}) = \sum \hat{\mathfrak{K}}(j) \phi_j(x) \phi_j(\bar{x}) \quad \text{and} \quad \Theta(x, \bar{x}) = \sum \hat{\Theta}(j) \psi_j(x) \psi_j(\bar{x}) \tag{5}$$

One advantage of applying this decomposition is that it has almost no constraint on the kernels and the inputs. However, this decomposition is too coarse to be useful since it can hardly provide fine-grained information about the eigenstructures. E.g, it is not clear what are the corrections to Eq. (5) when changing the architecture from a 2-layer CNN to a 4-layer CNNs. For this reason, we choose to work on "concrete" input spaces with richer mathematical structures. Our primary goal is to characterize the analytical dependence of the decomposition Eq. (5) on the network's topology.

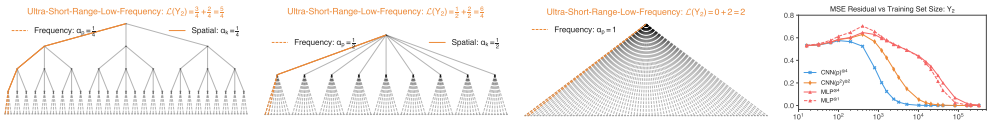
3 Neural computations on DAGs

We need some notation first. For a positive integer p, let $\mathbb{S}_{p-1}$ denote the unit sphere in $\mathbb{R}^p$ and $\overline{\mathbb{S}}_{p-1} = \sqrt{p}\mathbb{S}_{p-1}$, the sphere of radius $\sqrt{p}$ in $\mathbb{R}^p$. We introduce the normalized sum (integral)

$$\fint_{x \in X} f(x) \equiv |X|^{-1} \sum_{x \in X} f(x) \quad \left(\fint_{x \in X} f(x) \equiv \mu(X)^{-1} \int_{x \in X} f(x) \mu(dx) \right) \tag{6}$$

where X is a finite set (a measurable set with a finite positive measure μ).

We find it more convenient to express the computations in neural networks, and in neural kernels via directed acyclic graphs (DAGs) (Daniely et al., 2016), as both computations are of *recursive* nature. Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ denote a DAG, where $\mathcal{N}$ and $\mathcal{E}$ are the nodes and edges, resp. We always assume the



(a) CNN$(p)^{\otimes 4}$ (b) CNN$(p^2)^{\otimes 2}$ (c) MLP (d) MSE Residual

Figure 2: **Learning Index vs Architecture/DAG.**

graph to have a unique output node $o_{\mathcal{G}}$ and is an ancestor of all other nodes. Denote $\mathcal{N}_0 \subseteq \mathcal{N}$ the set of input nodes (leaves) of $\mathcal{G}$, i.e., the collection nodes without a child. Each node $u \in \mathcal{N}$ is associated with a pointwise function $\phi_u : \mathbb{R} \to \mathbb{R}$, which is normalized in the sense $\mathbb{E}_{z\in\mathcal{N}(0,1)}\phi_u^2(z) = 1$. It induces a function $\phi_u^* : I \equiv [-1, 1] \to I$ defined to be $\phi_u^*(t) = \mathbb{E}_{(z_1,z_2)\in\mathcal{N}_t}\phi_u(z_1)\phi_u(z_2)$. Here $\mathcal{N}_t$ denotes a pair of standard Gaussians with correlation t. We associate each $u \in \mathcal{N}$ a finite-dimensional Hilbert space $\mathbb{H}_u$, and each $uv \in \mathcal{E}$ a bounded linear operator $\mathcal{L}_{uv} : \mathbb{H}_v \to \mathbb{H}_u$. Let

$$\boldsymbol{\mathcal{X}} \equiv \prod_{u\in\mathcal{N}_0} \boldsymbol{\mathcal{X}}_u \equiv \prod_{u\in\mathcal{N}_0} \overline{\mathbb{S}}_{\dim(\mathbb{H}_u)-1} \subseteq \prod_{u\in\mathcal{N}_0} \mathbb{H}_u \quad \text{and} \quad \boldsymbol{I} = I^{|\mathcal{N}_0|}$$

be the input *tensors* and the input *correlations* to the graph $\mathcal{G}$, resp. We associate two types of computations to a DAG: finite-width neural network computation and kernel computation,

$$\mathcal{N}_{\mathcal{G}} : \boldsymbol{\mathcal{X}} \to \mathbb{H}_{o_{\mathcal{G}}} \quad \text{and} \quad \mathcal{K}_{\mathcal{G}} : \boldsymbol{I} \to I\,, \tag{7}$$

resp. They are defined recursively as follows

$$\mathcal{N}_u(\boldsymbol{x}) = \phi_u\left(\sum_{v:uv\in\mathcal{E}} \mathcal{L}_{uv}(\mathcal{N}_v(\boldsymbol{x}))\right) \qquad \text{if} \quad u \notin \mathcal{N}_0 \quad \text{else} \quad \mathcal{N}_u(\boldsymbol{x}) = x_u \tag{8}$$

$$\mathcal{K}_u(\boldsymbol{t}) = \phi_u^*\left(\fint_{v:uv\in\mathcal{E}} \mathcal{K}_v(\boldsymbol{t})\right) \qquad \text{if} \quad u \notin \mathcal{N}_0 \quad \text{else} \quad \mathcal{K}_u(\boldsymbol{t}) = \boldsymbol{t}_u \tag{9}$$

where $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$ and $\boldsymbol{t} \in \boldsymbol{I}$. The outputs of the computations are $\mathcal{N}_{\mathcal{G}}(\boldsymbol{x}) = \mathcal{N}o_{\mathcal{G}}(\boldsymbol{x})$ and $\mathcal{K}_{\mathcal{G}}(\boldsymbol{t}) = \mathcal{K}_{o_{\mathcal{G}}}(\boldsymbol{t})$. Note that $\mathcal{K}_{\mathcal{G}}$ is indeed the NNGP kernel. The NTK can also be written recursively as

$$\Theta_u(\boldsymbol{t}) = \dot{\phi}_u^*\left(\fint_{v:uv\in\mathcal{E}} \mathcal{K}_v(\boldsymbol{t})\right) \fint_{v:uv\in\mathcal{E}} (\mathcal{K}_v(\boldsymbol{t}) + \Theta_v(\boldsymbol{t})) \quad \text{with} \quad \Theta_{\mathcal{G}} = \Theta_{o_{\mathcal{G}}}\,. \tag{10}$$

Here, $\Theta_u = 0$ if $u \in \mathcal{N}_0$ and $\dot{\phi}_u^*$ is the derivative of ϕ_u^*.

3.1 Three Examples: MLPs, S-CNNs and D-CNNs.

To unpack the notation, we consider three concrete examples: an L-hidden layer MLP, a shallow convolutional network (S-CNN) that contains only one convolutional layer and a deep convolutional network (D-CNN) that contains $(1 + L)$ convolutional layers. The architectures are

MLP: $[\textit{Input}] \to [\textit{Dense-Act}]^{\otimes L} \to [\textit{Dense}]$ (11)

S-CNN: $[\textit{Input}] \to [\textit{Conv}(p)\textit{-Act}] \to [\textit{Flatten-Dense}]$ (12)

D-CNN: $[\textit{Input}] \to [\textit{Conv}(p)\textit{-Act}] \to [\textit{Conv}(k)\textit{-Act}]^{\otimes L} \to [\textit{Flatten-Dense-Act}] \to [\textit{Dense}]$ (13)

where p/k is the filter size of the first/hidden layers and Act means an activation layer. We choose the stride to be the same as the size of the filter for all convolutional layers and choose *flattening* as the readout strategy rather than pooling. See Fig. 2 (a, b, c) for the DAGs associated to a (1+3)-layer CNN (with $p = k = d^{\frac{1}{4}}$), a (1+1)-layer CNN (with $p = k = d^{\frac{1}{2}}$) and a 1-layer MLP.

MLPs. Let $\mathcal{G}$ be a linked list with $(L + 2)$ nodes, including the input/output nodes. Let $\mathcal{L}_{uv} \in \mathbb{R}^{n_u\times n_v}$, where $n_{u/v} = \dim(\mathbb{H}_{u/v})$ and the activations of the input/output nodes be the identity function. Then $\mathcal{N}_{\mathcal{G}}$ represents a L-hidden-layer MLP. In addition, let $\mathcal{L}_{uv}$ be initialized iid as

$$\mathcal{L}_{uv} = \frac{1}{\sqrt{n_v}}\,(\omega_{uv,ij})_{i\in[n_u],j\in[n_v]}\,, \quad \omega_{uv,ij} \sim \mathcal{N}(0, 1)\,. \tag{14}$$

Let $\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'} = \boldsymbol{x}^T\boldsymbol{x}'/n_u$ for $u \in \mathcal{N}_0$ and $n_v \to \infty$ for all hidden nodes, then the outputs $\mathcal{N}(\boldsymbol{\mathcal{X}})$ converge weakly to the GP $\mathcal{GP}(0, \mathcal{K}_{\mathcal{G}}(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'})_{\boldsymbol{x},\boldsymbol{x}'\in\boldsymbol{\mathcal{X}}})$ and $\Theta_{\mathcal{G}}(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'})_{\boldsymbol{x},\boldsymbol{x}'\in\boldsymbol{\mathcal{X}}}$ is the NTK in the sense

$$\mathbb{E}\mathcal{N}_{\mathcal{G}}(\boldsymbol{x})\mathcal{N}_{\mathcal{G}}(\boldsymbol{x}') \xrightarrow{\text{in prob.}} \mathcal{K}_{\mathcal{G}}(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'}) \quad \text{and} \quad \langle \nabla\mathcal{N}_{\mathcal{G}}(\boldsymbol{x}), \nabla\mathcal{N}_{\mathcal{G}}(\boldsymbol{x}')\rangle \xrightarrow{\text{in prob.}} \Theta_{\mathcal{G}}(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'}). \tag{15}$$

Indeed, note that $\deg(u) = 1$ for all $u \notin \mathcal{N}_0$. Eq. (9) and Eq. (10) become

$$\mathcal{K}_u(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'}) = \phi_u^*(\mathcal{K}_v(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'})) \quad \text{and} \quad \Theta_u(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'}) = \dot{\phi}_u^*(\mathcal{K}_v(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'}))(\mathcal{K}_v(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'}) + \Theta_v(\boldsymbol{t}_{\boldsymbol{x},\boldsymbol{x}'})) \tag{16}$$

which are the recursive formulas for the NNGP kernel and NTK; see e.g. Sec.E in Lee et al. (2019).

S-CNN. The input $\boldsymbol{\mathcal{X}} = (\overline{\mathbb{S}}_{p-1})^{1\times w} \subseteq \mathbb{R}^d$, where p is the patch size, w is the number of patches, $d = pw$ is the dimension of the inputs. Here, the inputs have been *pre-processed* by a patch extractor and then by a normalization operator. In words, the S-CNN has one convolutional layer with filter size p, followed by an activation function ϕ (e.g., Relu), and finally by a *flatten-dense* readout layer. Mathematically, by letting $n \in \mathbb{N}$ be the number of channels in the hidden layer, the output (i.e., logit) is given by

$$\textbf{Convolution + Activation:} \qquad z_{ij}(\boldsymbol{x}) = \phi\left(p^{-\frac{1}{2}} \sum_{\beta\in[p]} \omega_{1,j,\beta} x_{\beta,i}\right) \quad \text{for} \quad i \in [w], j \in [n] \tag{17}$$

$$\textbf{Flatten + Dense:} \qquad f(\boldsymbol{x}) = (wn)^{-\frac{1}{2}} \sum_{i\in[w],j\in[n]} \omega_{2,ij} z_{ij}(\boldsymbol{x}), \tag{18}$$

where $\omega_{1,i,\beta}$ and $\omega_{2,ij}$ are the parameters of the first and readout layers, resp.

We can associate a DAG $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ to the above S-CNN. Let the input, hidden and output nodes be $\mathcal{N}_0 = [1] \times [w]$, $\mathcal{N}_1 = [w]$ and $\mathcal{N}_2 = \{o_{\mathcal{G}}\} = \{\emptyset\}$, resp. and $\mathcal{N} = \mathcal{N}_0 \cup \mathcal{N}_1 \cup \mathcal{N}_2$. Moreover, $uv \in \mathcal{E}$ if $u = o_{\mathcal{G}}$ and $v \in \mathcal{N}_1$ or $u = (i,) \in \mathcal{N}_1$ and $v = (0, i) \in \mathcal{N}_0$. Let $\mathbb{H}_v = \mathbb{R}^p$ for $v \in \mathcal{N}_0$, $\mathbb{H}_u = \mathbb{R}^n$ if $u \in \mathcal{N}_1$ and $\mathbb{H}_{o_{\mathcal{G}}} = \mathbb{R}$. The associated linear operators are given by

$$\mathcal{L}_{uv} = p^{-\frac{1}{2}}(\omega_{1,j,\beta})_{j\in[n],\beta\in[p]} \in \mathbb{R}^{n\times p} \quad \text{for} \quad (u, v) \in \mathcal{N}_1 \times \mathcal{N}_0 \tag{19}$$

$$\mathcal{L}_{o_{\mathcal{G}}v} = (wn)^{-\frac{1}{2}}(\omega_{2,ij})_{j\in[n]} \in \mathbb{R}^n \quad \text{if} \quad v = (i,) \in \mathcal{N}_1 \tag{20}$$

Note that the weights are shared in the first layer but not in the readout layer. To compute the NNGP kernel and NTK, we initalize all parameters $\omega_{1,i,\beta}$ and $\omega_{2,ij}$ with iid Gaussian $\mathcal{N}(0, 1)$. Letting $n \to \infty$ and denoting $\boldsymbol{t}_v = \boldsymbol{x}_v^T\boldsymbol{x}_v'/p$ and $\boldsymbol{t} = (\boldsymbol{t}_v)_{v\in\mathcal{N}_0}$, we have

$$\mathcal{K}_{\mathcal{G}}(\boldsymbol{t}) = \fint_{v\in\mathcal{N}_0} \phi^*(\boldsymbol{t}_v) \quad \text{and} \quad \Theta_{\mathcal{G}}(\boldsymbol{t}) = \fint_{v\in\mathcal{N}_0} \phi^*(\boldsymbol{t}_v) + \dot{\phi}^*(\boldsymbol{t}_v) \tag{21}$$

D-CNN. The input space is $\boldsymbol{\mathcal{X}} = (\overline{\mathbb{S}}_{p-1})^{k^L\times w} \subseteq \mathbb{R}^{p\times 1\times k^L\times w}$, where p is the patch size of the input convolutional layer, k is the filter size in *hidden* layers, L is the number of *hidden* convolution layers and w is the spatial dimension of the penultimate layer. The total dimension of the input is $d = p \cdot k^L \cdot w$, and the number of input nodes is $|\mathcal{N}_0| = k^L \cdot w$. Since the stride is equal to the filter size for all convolutional layers, the *spatial* dimension is reduced by a factor of p in the first layer, a factor of k by each hidden layer, and is reduced to 1 by the *flatten-dense* layer. Similar to S-CNNs, one can associate a DAG to a D-CNN. Briefly, the input layer has $k^L \times w$ nodes and is reduced by a factor of k by each convolutional layer. The penultimate and output layers have w and 1 nodes, resp.

4 Main Results

The goal is to obtain a non-asymptotic charaterization as the input dimension $d \to \infty$ of the relation between eigenstructures of $\mathcal{K}$ / Θ and the DAG associated to the network architectures. As such we consider a sequence of graphs $\boldsymbol{\mathcal{G}} = \left(\mathcal{G}^{(d)}\right)_{d\in\mathbb{N}}$, where $\mathcal{G}^{(d)} = (\mathcal{N}^{(d)}, \mathcal{E}^{(d)})$. We associate a *finite* set of non-negative numbers $\Lambda_{\boldsymbol{\mathcal{G}}}$ to $\boldsymbol{\mathcal{G}}$, which is called the shape parameters of $\boldsymbol{\mathcal{G}}$,

$$0 \in \Lambda_{\boldsymbol{\mathcal{G}}} \subseteq [0, 1] \quad \text{and} \quad |\Lambda_{\boldsymbol{\mathcal{G}}}| < \infty. \tag{22}$$

We need several technical assumptions on $\boldsymbol{\mathcal{G}}$ regarding the asymptotic shapes of $\mathcal{G}^{(d)}$, which are summarized as **Assumption-$\boldsymbol{\mathcal{G}}$** in Sec.B of the appendix. We list two of them which are the most crucial ones. (1) For each *non-input* node $u \in \mathcal{N}^{(d)}$, there is $\alpha_u \in \Lambda_{\boldsymbol{\mathcal{G}}}$ with $\deg(u) \sim d^{\alpha_u}$.The weight associated to the edge $uv \in \mathcal{E}^{(d)}$ is defined to be $\omega_{uv} \equiv \alpha_u$. (2) For each input node v, there is $0 < \alpha_v \in \Lambda_{\boldsymbol{\mathcal{G}}}$ so that the input dimension $d_v \sim d^{\alpha_v}$. Here $a \sim b$ means $a/b \in [1/C, C]$ for some $C > 0$ independent of d. The main purpose of making these two assumptions is to remove non-leading terms when computing the spectra.

We say ϕ^* is semi-admissible if, for all $r \geq 1$, the r-th derivative of ϕ^* at zero is non-vanishing, i.e., $\phi^{*(r)}(0) > 0$. If, in addition, $\phi^*(0) = 0$ (i.e., the activation is centered), then we say ϕ is admissible. An activation ϕ is (semi-)admissible if ϕ^* is (semi-)admissible. Note that if ϕ^* is (semi-)admissible, then $\dot{\phi}^*$ is semi-admissible.

Assumption-ϕ. We make the following assumptions on the activations. (a.) If $u \in \mathcal{N}_0^{(d)}$, ϕ_u is the identity function. (b.) If $u \notin \mathcal{N}_0^{(d)} \cup \{o_{\mathcal{G}}\}$, ϕ_u is admissible. (c.) If $u = o_{\mathcal{G}}$, ϕ_u semi-admissible.

Next, we introduce the key concept which defines the *spatial distance* among nodes. It is the length of the minimum spanning tree (MST) of the nodes.

Definition 1 (Spatial Index of Nodes)**.** *Let $\mathfrak{n} \subseteq \mathcal{N}^{(d)}$. The spatial index of $\mathfrak{n}$ is defined to be*

$$\mathcal{S}(\mathfrak{n}) = \min_{\mathfrak{n} \subseteq \mathcal{T} \leq \mathcal{G}^{(d)}} \sum_{uv \in \mathcal{E}(\mathcal{T})} \omega_{uv} = \min_{\mathfrak{n} \subseteq \mathcal{T} \leq \mathcal{G}^{(d)}} \sum_{uv \in \mathcal{E}(\mathcal{T})} \deg(u; \mathcal{T}) \alpha_u \tag{23}$$

where $\mathfrak{n} \subseteq \mathcal{T} \leq \mathcal{G}^{(d)}$ means $\mathcal{T}$ is a sub-graph containing $\mathfrak{n}$ and $\deg(u; \mathcal{T})$ is the degree of u in $\mathcal{T}$. By default, $\mathcal{S}(\mathfrak{n}) = 0$ if $\mathfrak{n}$ contains only one or zero node.

Let $\boldsymbol{t}^{\boldsymbol{r}} : I^{|\mathcal{N}_0^{(d)}|} \to I$ be a monomial, where $\boldsymbol{r} : \mathcal{N}_0^{(d)} \to \mathbb{N}^{|\mathcal{N}_0^{(d)}|}$. We use $\mathfrak{n}(\boldsymbol{r}) = \{v \in \mathcal{N}_0^{(d)} : r_v \neq 0\}$ to denote the support of $\boldsymbol{r}$ and $\mathfrak{n}(\boldsymbol{r}; o_{\mathcal{G}}) = \mathfrak{n}(\boldsymbol{r}) \cup \{o_{\mathcal{G}}\}$.

Definition 2 (Spatial, Frequency and Learning Indices)**.** *We say $\boldsymbol{r} \in \mathbb{N}^{|\mathcal{N}_0^{(d)}|}$ is $\mathcal{G}^{(d)}$-learnable, or learnable for short, if there is a common ancestor node u of $\mathfrak{n}(\boldsymbol{r})$ such that ϕ_u is semi-admissible. We use $\mathcal{A}(\mathcal{G}^{(d)}) \equiv \{\boldsymbol{r} \in \mathbb{N}^{\mathcal{N}_0^{(d)}} : \boldsymbol{r}$ is learnable$\}$. For $\boldsymbol{r} \in \mathcal{A}(\mathcal{G}^{(d)})$, the spatial index, frequency index and the learning index are defined to be,*

$$\mathcal{S}(\boldsymbol{r}) := \mathcal{S}(\mathfrak{n}(\boldsymbol{r}; o_{\mathcal{G}})), \quad \mathcal{F}(\boldsymbol{r}) := \sum_{v \in \mathcal{N}_0^{(d)}} r_v \alpha_v \quad \textit{and} \quad \mathcal{L}(\boldsymbol{r}) = \mathcal{S}(\boldsymbol{r}) + \mathcal{F}(\boldsymbol{r}), \tag{24}$$

resp. If $\boldsymbol{r} \notin \mathcal{A}(\mathcal{G}^{(d)})$, we set $\mathcal{S}(\boldsymbol{r}) = \mathcal{F}(\boldsymbol{r}) = \mathcal{L}(\boldsymbol{r}) = +\infty$. Let $\mathcal{L}(\mathcal{G}^{(d)})$ denote the sequence of learning indices in non-descending order, i.e.

$$\mathcal{L}(\mathcal{G}^{(d)}) \equiv \Big(\mathcal{L}(\boldsymbol{r}) : \boldsymbol{r} \in \mathcal{A}(\mathcal{G}^{(d)})\Big) \equiv (\cdots \leq r_j \leq r_{j+1} \leq \dots) \tag{25}$$

Finally, for each $u \in \mathcal{N}_0^{(d)}$, let $\{\overline{Y}_{r,l}\}_{l \in [N(d_u, r)], r \in \mathbb{N}}$ be the family of normalized spherical harmonics in $\overline{\mathbb{S}}_{d_u - 1}$, where $N(d_u, r)$ is the number of degree r spherical harmonics in $\mathbb{S}_{d_u - 1}$. Define

$$\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi}) = \prod_{u \in \mathcal{N}_0^{(d)}} \overline{Y}_{\boldsymbol{r}_u, \boldsymbol{l}_u}(\boldsymbol{\xi}_u), \quad \boldsymbol{l} = (\boldsymbol{l}_u)_{u \in \mathcal{N}_0^{(d)}} \in [\boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r})] \equiv \prod_{u \in \mathcal{N}_0^{(d)}} [N(d_u, \boldsymbol{r}_u)] \tag{26}$$

for $\boldsymbol{\xi} = (\boldsymbol{\xi}_u) \in \boldsymbol{\mathcal{X}}$. The following is our main theorem. It describe a connection between the architecture of a network and the eigenstructure of its inducing kernels.

Theorem 1 (**Eigenspace Restructuring**)**.** *Assume* **Assumption-$\boldsymbol{\mathcal{G}}$** *and* **Assumption-ϕ**. *We have the following eigen-decomposition for $\mathcal{K} = \mathcal{H}_{\mathcal{G}^{(d)}}$ or $\Theta_{\mathcal{G}^{(d)}}$. For $\boldsymbol{\xi}, \boldsymbol{\eta} \in \boldsymbol{\mathcal{X}}$*

$$\mathcal{K}(\boldsymbol{\xi}, \boldsymbol{\eta}) = \sum_{\boldsymbol{r} \in \mathbb{N}^{|\mathcal{N}_0^{(d)}|}} \lambda_{\mathcal{K}}(\boldsymbol{r}) \sum_{\boldsymbol{l} \in \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r})} \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi}) \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\eta}), \quad \textit{where} \quad \lambda_{\mathcal{K}}(\boldsymbol{r}) \sim d^{-\mathcal{L}(\boldsymbol{r})} \textit{ if } \boldsymbol{r} \neq \boldsymbol{0}. \tag{27}$$

Coupling with the observation in Eq. (2), Theorem 1 implies that within $t \sim d^r$ amount of time for gradient flow, when d is sufficiently large, only the eigenfunctions $\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}$ with $\mathcal{L}(\boldsymbol{r}) \leq r$ can be

learned. By leveraging an analytical result from Mei et al. (2021) (Sec. 3 Theorem 4), which says under certain regularity conditions on the eigenstructure, the corresponding kernel regression acts as a projection, Theorem 1 establishes a connection between architectures and generalization bounds of NNGP kernel/NTK.

Let $\boldsymbol{\sigma}$ be the uniform (product) probability measure on $\boldsymbol{\mathcal{X}}$ and denote $L^p(\boldsymbol{\mathcal{X}}) \equiv L^p(\boldsymbol{\mathcal{X}}, \boldsymbol{\sigma})$. For $X \subseteq \boldsymbol{\mathcal{X}}$ and $r \notin \mathcal{L}(\mathcal{G}^{(d)})$, define the regressor and the projection operator to be

$$\boldsymbol{R}_X(f)(x) = \mathcal{K}(x, X)\mathcal{K}(X, X)^{-1} f(X) \text{ and } \boldsymbol{P}_{>r}(f) = \sum_{\boldsymbol{r}:\mathcal{L}(\boldsymbol{r})>r} \sum_{\boldsymbol{l}\in\boldsymbol{N}(\boldsymbol{d},\boldsymbol{r})} \langle f, \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} \rangle_{L^2(\boldsymbol{\mathcal{X}})} \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} \,.$$

Theorem 2. *Let $\boldsymbol{\mathcal{G}} = \{\mathcal{G}^{(d)}\}_d$, where each $\mathcal{G}^{(d)}$ is a DAG associated to the D-CNN in Eq. (13). Let $r \notin \mathcal{L}(\mathcal{G}^{(d)})$ be fixed. Let $f \in L^2(\boldsymbol{\mathcal{X}})$ with $\mathbb{E}_{\boldsymbol{\sigma}} f = 0$. Then for $\epsilon > 0$,*

$$\left| \|\boldsymbol{R}_X(f) - f\|^2_{L^2(\boldsymbol{\mathcal{X}})} - \|\boldsymbol{P}_{>r}(f)\|^2_{L^2(\boldsymbol{\mathcal{X}})} \right| = c_{d,\epsilon} \|f\|^2_{L^{2+\epsilon}(\boldsymbol{\mathcal{X}})}, \tag{28}$$

where $c_{d,\epsilon} \to 0$ in probability as $d \to \infty$ over $X \sim \boldsymbol{\sigma}^{[d^r]}$.

In words, with $[d^r]$ many training examples where $r \notin \mathcal{L}(\mathcal{G}^{(d)})$, the NNGP kernel and the NTK are able to learn all $\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}$ with $\mathcal{L}(\boldsymbol{r}) < r$ but not any eigenfunctions with $\mathcal{L}(\boldsymbol{r}) > r$.

5 Interpretation of the Main Results

We say r is the budget index if (1) (**Finite Training Set**) the training set size $m \sim d^r$, or (2) (**Finite Compute Time**) the training set $X = \boldsymbol{\mathcal{X}}$ and the total number of training steps/time $t \sim d^r$.

MLPs and LRLF Interactions Fig. 1 (a). In this case, each $\mathcal{G}^{(d)}$ is a linked list. Let v be the input node. Clearly, we have $d_v = d$, i.e., $\alpha_v = 1$ and $\alpha_u = 0$ (since $\deg(u) = 1$) for all other nodes u. As such

$$\mathcal{A}(\mathcal{G}^{(d)}) = \mathbb{N}^d \quad \mathcal{F}(\boldsymbol{r}) = |\boldsymbol{r}|, \quad \mathcal{S}(\boldsymbol{r}) = 0, \quad \mathcal{L}(\boldsymbol{r}) = |\boldsymbol{r}| \quad \text{and} \quad \mathcal{L}_{\mathcal{G}^{(d)}} = (|\boldsymbol{r}| : \boldsymbol{r} \in \mathbb{N}^d). \tag{29}$$

There isn't any *spatial* structure since $\mathcal{S}(\boldsymbol{r}) = 0$. Given a budget index r, the kernels can only learn $\text{span}\{\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} : |\boldsymbol{r}| < r\}$. Therefore, MLPs are good at modeling LRLF interactions.

S-CNNs and SRHF Interactions Fig. 1 (b). For one-hidden layer convolutional networks, we have $d = p \times k^0 \times w$, (i.e. $L = 0$). The number of input nodes is $w \equiv d^{\alpha_w}$ and for each input node v we have $d_v = p \equiv d^{\alpha_p}$. We have $\alpha_p + \alpha_w = 1$. Since the activation function of the last layer is the identity function, $\mathcal{A}(\mathcal{G}^{(d)}) = \{\boldsymbol{r} \in \mathbb{N}^{|\mathcal{N}_0^{(d)}|}, |\boldsymbol{n}(\boldsymbol{r})| = 0 \text{ or } 1\}$. Therefore for $\boldsymbol{r} \in \mathcal{A}(\mathcal{G}^{(d)})$,

$$\mathcal{S}(\boldsymbol{r}) = \alpha_w \text{ if } |\boldsymbol{n}(\boldsymbol{r})| = 1 \text{ else } 0, \quad \mathcal{F}(\boldsymbol{r}) = |\boldsymbol{r}|\alpha_p, \quad \mathcal{L}_{\mathcal{G}} = \{0, \alpha_w + |\boldsymbol{r}|\alpha_p, \boldsymbol{r} \in \mathcal{A}(\mathcal{G}^{(d)})\} \tag{30}$$

Given a budget index r, the learnable $\boldsymbol{r}$ are the ones $\boldsymbol{r} \in \mathcal{A}(\mathcal{G}^{(d)})$ with

$$r > \alpha_w + |\boldsymbol{r}|\alpha_p = 1 + (|\boldsymbol{r}| - 1)\alpha_p \Rightarrow |\boldsymbol{r}| < 1 + (r-1)/\alpha_p \,. \tag{31}$$

When $\alpha_p = 1$ (and thus $\alpha_w = 0$), this S-CNN is essentially a shallow MLP and we have $|\boldsymbol{r}| < r$. On the other hand, if α_p is small (say $\alpha_p = 0.1$), this network can model interactions with much higher frequencies, at the cost of giving up long-range interactions. Therefore, there is a trade-off between space and frequency, and S-CNNs with small patch size are good at modeling SRHF interactions.

D-CNNs and Interterpolation Fig. 1 (c). Recall that $d = p \times k^L \times w$. Let $p = d^{\alpha_p}, k = d^{\alpha_k}$ and $w = d^{\alpha_w}$. Then we have the constraint $\alpha_p + L\alpha_k + \alpha_w = 1$. The learnable terms are the ones with

$$r > \mathcal{S}(\boldsymbol{r}) + \mathcal{F}(\boldsymbol{r}) = \mathcal{S}(\boldsymbol{r}) + |\boldsymbol{r}|\alpha_p. \tag{32}$$

D-CNNs can simultaneously model interpolations (including the two ends) between LRLF and SRHF, i.e. MRMF interactions. Let $|\boldsymbol{n}(\boldsymbol{r})| = \bar{r} \in [1, r]$ be fixed. Then the range of $\mathcal{S}(\boldsymbol{r})$ is

$$\mathcal{R}_{\bar{r}}(\alpha_p, \alpha_w, \alpha_k) \equiv \{a\alpha_w + b\alpha_k : a \in [1, \bar{r}],\ b \in [a(L-1) + \bar{r},\ \bar{r}L], \text{ and } a, b \in \mathbb{N}\} \tag{33}$$

The length can be as small as $(\alpha_w + L\alpha_k + (\bar{r} - 1)\alpha_k)$ and as large as $(\bar{r}(\alpha_w + L\alpha_k))$. The former/latter happens when $\boldsymbol{n}(\boldsymbol{r})$ is chosen in a way that the paths from $\boldsymbol{n}(\boldsymbol{r})$ to $o_{\mathcal{G}}$ is "maximally"/"minimally" shared (see e.g. Fig. 4 $\mathbf{Y}_5^*$/$\mathbf{Y}_5$). The former/latter leaves more/less budget for learning high-frequency interactions. In general, the set of learnable indices includes various interpolations between these two extreme spectra (see Fig. 1 (c)):

$$\{\boldsymbol{r} : a\alpha_w + b\alpha_k + |\boldsymbol{r}|\alpha_p < r, a \in [1, |\boldsymbol{n}(\boldsymbol{r})|],\ b \in [a(L-1) + |\boldsymbol{n}(\boldsymbol{r})|,\ |\boldsymbol{n}(\boldsymbol{r})|L],\ a, b \in \mathbb{N}\} \tag{34}$$

HS-CNNs Extrapolations and Finer Interpolations Fig. 1 (d). The resolution of the set Eq. (34) can be improved by decreasing α_k, α_p and increasing L accordingly, e.g. $\alpha_k \mapsto \alpha_k/2$, $\alpha_p \mapsto \alpha_p$ and $L \mapsto 2L$. This empowers the network to model finer-grained interpolating modes, and extrapolating modes, e.g. **Ultra-Short-Range-Ultra-High-Frequency** interactions (see $\mathbf{Y}_5^*$ in Fig. 3). We refer to such networks as high-resolution CNNs (HS-CNNs). For practical networks, e.g. ResNet (He et al., 2016), the filters/patches are already quite small and there isn't much room to reduce them. Equivalently, one increases the resolution of the input images instead, i.e. increasing d. From this point of view, HS-CNNs and, therefore, our theorem justify the additional performance gain from EfficientNet-type (Tan & Le, 2019) of model scaling.

6 Experiments

There are many practical consequences due to Theorem 1 and Theorem 2. We focus on two of them which are about *the impact of architectures to learning / generalization*:

1. **Order of Learning (Fig. 1 Green Arrow.)** The order of learning is restructured from frequency-based (MLPs) to space-and-frequency-based (CNNs).
2. **Learnability (Fig. 1 Cells under the budget line.)** With the same budget index, MLP-Learnable $\subsetneq$ D-CNN-Learnable $\subsetneq$ HS-CNN-Learnable. Moreover, the set differences between these learnable sets are captured as in Sec.5.

Overall, we see excellent agreements between predictions from our theorems and experimental results from both practical-size networks and kernel methods using NNGP/NT kernels, even when $d = 256$ is moderate-size. We detail the setup, results, corrections, etc. for the experiments below.

Setup. Set $d = p^4$ and the input $\boldsymbol{\mathcal{X}} = (\overline{\mathbb{S}}_{p-1})^{p^3} \subseteq \mathbb{R}^{p^4}$, where $p \in \mathbb{N}$. Note that $\alpha_p = 1/4$. The task is learning a function $Y \in L^2(\boldsymbol{\mathcal{X}})$ by minimizing the MSE, where

$$Y = \mathbf{Y_1} + \mathbf{Y_2} + \mathbf{Y_3} + \mathbf{Y_4} + \mathbf{Y_5} + \mathbf{Y_5^*} + \mathbf{Y_6} + \mathbf{Y_7} \tag{35}$$

and each eigenfunction Y_i is a normalized so that $\|Y_i\|_2^2 = \|Y\|_2^2/8 = 1$.

We optimize finite-width networks by *SGD+Momentum* and infinite-width networks (NNGP and NTK) by kernel regression. We investigate three types of architectures: (1) Dense$^{\otimes 4}$, a four hidden layer MLP; (2) Conv$(p^2)^{\otimes 2}$, a "deep" CNN with filter size/stride $k = p^2$, and (3) Conv$(p)^{\otimes 4}$, a "HS"-CNN with filter size/stride $k = p$. See Fig.1 for a visualization of the associated DAGs. There is an activation ϕ in each *hidden* layer, which is chosen so that ϕ^* is the Gaussian kernel. For the CNNs, the readout layer(s) is *Flatten-Dense-Act-Dense*. We carefully chose the eigenfunctions $\{Y_i\}$ so that they cover a wide range of space-frequency combinations $(\mathcal{S}(Y_i), \mathcal{F}(Y_i))$ w.r.t. Conv$(p)^{\otimes 4}$. Under Conv$(p)^{\otimes 4}$, the corresponding learning indices are $\mathcal{L}(Y_i) = \mathcal{S}(Y_i) + \mathcal{F}(Y_i) = 3\alpha_p + i\alpha_p = (3+i)/4$. For the learning indices of Y_i under Conv$(p^2)^{\otimes 2}$ or Dense$^{\otimes 4}$, see the legends in Fig.3. The purpose of doing so is to create a "separation of learning" under Conv$(p)^{\otimes 4}$, since in the large p limit, learning Y_i requires $d^{(3+i)/4+\epsilon}$ examples/SGD steps. The relation between architectures and learning indices can be "visualized" in Fig. 2 for $\mathbf{Y_2}$, which is short-range-low-frequency ($\deg(\mathbf{Y_2}) = 2$). Fig. 2 (d) plots the test residual of $\mathbf{Y_2}$ vs training set size for NTK regression, as one example to showcase the relation among architectures, learning indices and generalization. See Fig. 4 in the appendix for other eigenfunctions.

In the experiments, the width/number of channels is set to 512 for all networks. We sample $m_t = 32 \times 10240$ ($m_v = 10240$) data points randomly from $\boldsymbol{\mathcal{X}}$ as training (test) set with $p = 4$ and $d = 256$. The SGD training configurations (batch size (=10240), learning rate (=1.), momentum (=0.9) etc.) are identical across architectures. To compute the kernels, we rely crucially on *NeuralTangents* (Novak et al., 2020) which is based on *JAX* (Bradbury et al., 2018).

In Fig.3, for each eigenfunction Y_i, we plot $\frac{1}{2}\mathbb{E}|\hat{Y}_i(x,t) - Y_i(x)|_2^2$ against t, where $\hat{Y}_i(x,t)$ is the projection of the prediction onto Y_i and t is either the training steps (SGD) or training set size (kernels). The expectation is taken over the test set. The budget index $r = \log(m_t)/\log(d) \approx 2.28$. As $d = 256$ ($p = 4$) is far from the asymptotic limit, we expect $r = 2.28$ being a *soft* cut-off between learnable and non-learnable indices. Although the theorems assume $d, p \to \infty$, they do provide good predictions even when d and p are far from ∞. We summarize several key observations below.

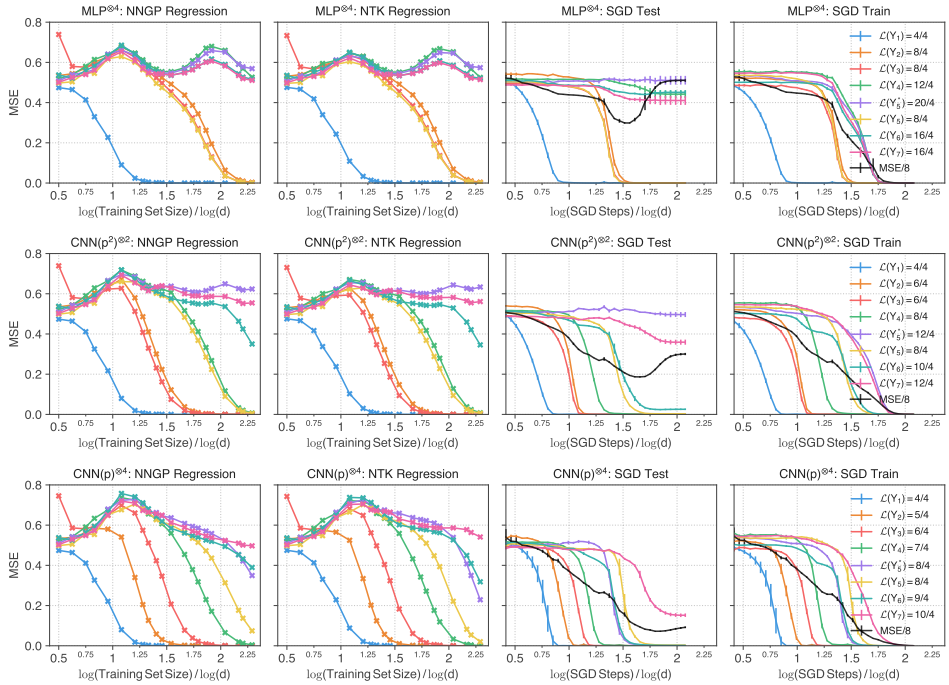


Figure 3: **Learning Dynamics vs Architectures vs Learning Indices.** We plot the learning/training dynamics of each eigenfunction Y_i. From top to bottom: a 4-layer MLP, a 2-layer CNN and a 4-layer CNN. From left to right: residual MSE (per eigenfunction) of NNGP/NTK regression, test/training MSE of SGD. The learning indices of Y_i in each architecture is shown in the legends.

1. **Dense**$^{\otimes 4}$ (1nd Row.) This architecture can capture all low-frequency interactions ($\deg = 1, 2$, $\mathbf{Y}_1$, $\mathbf{Y}_2$, $\mathbf{Y}_3$, $\mathbf{Y}_5$) but fail to learn $\deg \geq 3$ interactions, as expected. For MLPs, making the network deeper won't improve its learnability much; see Fig. 5.
2. **Conv**$(p^2)^{\otimes 2}$ (2nd Row.) Learning curves of $\mathbf{Y}_2$/$\mathbf{Y}_3$ are separated from $\mathbf{Y}_5$ because the spatial indices of them are different. Higher-frequency ($\deg = 3, 4$) shorter-range interactions ($\mathbf{Y}_4$, $\mathbf{Y}_6$) become (partially) learnable, as $\mathcal{L}(\mathbf{Y}_4) = \frac{8}{4}$, $\mathcal{L}(\mathbf{Y}_6) = \frac{10}{4} < r \approx 2.28$.
3. **Conv**$(p)^{\otimes 4}$ (3rd Row). We see $\mathcal{L}(Y_i)$ capture the order of learning very/reasonably well in the kernel/SGD setting. To test the ability of Conv$(p)^{\otimes 4}$ in modeling ultra-short-range-ultra-high-frequency interactions, we trace the learning progress of $\mathbf{Y}_5^*$ ($\deg(\mathbf{Y}_5^*) = 5, \mathcal{L}(\mathbf{Y}_5^*) = \frac{8}{4}$.) As expected, while other architectures completely fail to make progress, the NTK/NNGP of Conv$(p)^{\otimes 4}$ makes good progress and the SGD even completes the learning process. Interestingly, $\mathbf{Y}_5$[1] is learned faster than $\mathbf{Y}_5^*$ in the kernel setting but slower in the SGD setting (even slower than $\mathcal{L}(\mathbf{Y}_6) = \frac{10}{4}$), which is unexpected. We suspect it might be due to certain "implicit" effect of SGD. Further investigation is needed to understand it.

7 Conclusion

We establish a precise relation among networks' architectures, eigenstructures of the inducing kernels, and generalization of the corresponding kernel machines when the input dimension d is large. We show that deep convolutional networks restructure the eigenspaces of the inducing kernels, which empowers them to learn a dramatically broader class of functions, covering a wide range of space-frequency combinations. We believe our framework can be extended to study architectural inductive biases for other families of topologies, such as RNNs, GNNs, and self-attention. However, we have not covered the learning dynamics of SGD. In addition, it is of great interest and importance to study the combined effect of SGD and architectures in the future.

[1] Ultra-Long-Range-Low-Frequency under Conv$(p)^{\otimes 4}$, with $\mathcal{L}(\mathbf{Y}_5) = 8/4 = \mathcal{L}(\mathbf{Y}_5^*)$

REFERENCES

William Beckner. Sobolev inequalities, the poisson semigroup, and analysis on the sphere sn. *Proceedings of the National Academy of Sciences*, 89(11):4816–4819, 1992.

Alberto Bietti and Francis Bach. Deep equals shallow for relu networks in kernel regimes. *arXiv preprint arXiv:2009.14397*, 2020.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Arturo Deza, Qianli Liao, Andrzej Banburski, and Tomaso Poggio. Hierarchically compositional tasks and deep convolutional networks. *arXiv preprint arXiv:2006.13915*, 2020.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Christopher Frye and Costas J. Efthimiou. Spherical harmonics in p dimensions. 2012.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of lazy training of two-layers neural networks. *arXiv preprint arXiv:1906.08899*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Yann Lecun. Generalization and network design strategies. In *Connectionism in perspective*. Elsevier, 1989.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Sam Schoenholz, Jeffrey Pennington, and Jascha Sohl-dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

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*, 2019.

Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Generalization error of random features and kernel methods: hypercontractivity and kernel matrix concentration. *arXiv preprint arXiv:2101.10588*, 2021.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1g30j0qF7.

Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020. URL https://github.com/google/neural-tangents.

Yaniv Ovadia, Emily Fertig, J. Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances In Neural Information Processing Systems*, 2016.

Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *International Conference on Learning Representations*, 2017.

Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Jascha Sohl-Dickstein, Roman Novak, Samuel S. Schoenholz, and Jaehoon Lee. On the infinite width limit of neural networks with a standard parameterization, 2020.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

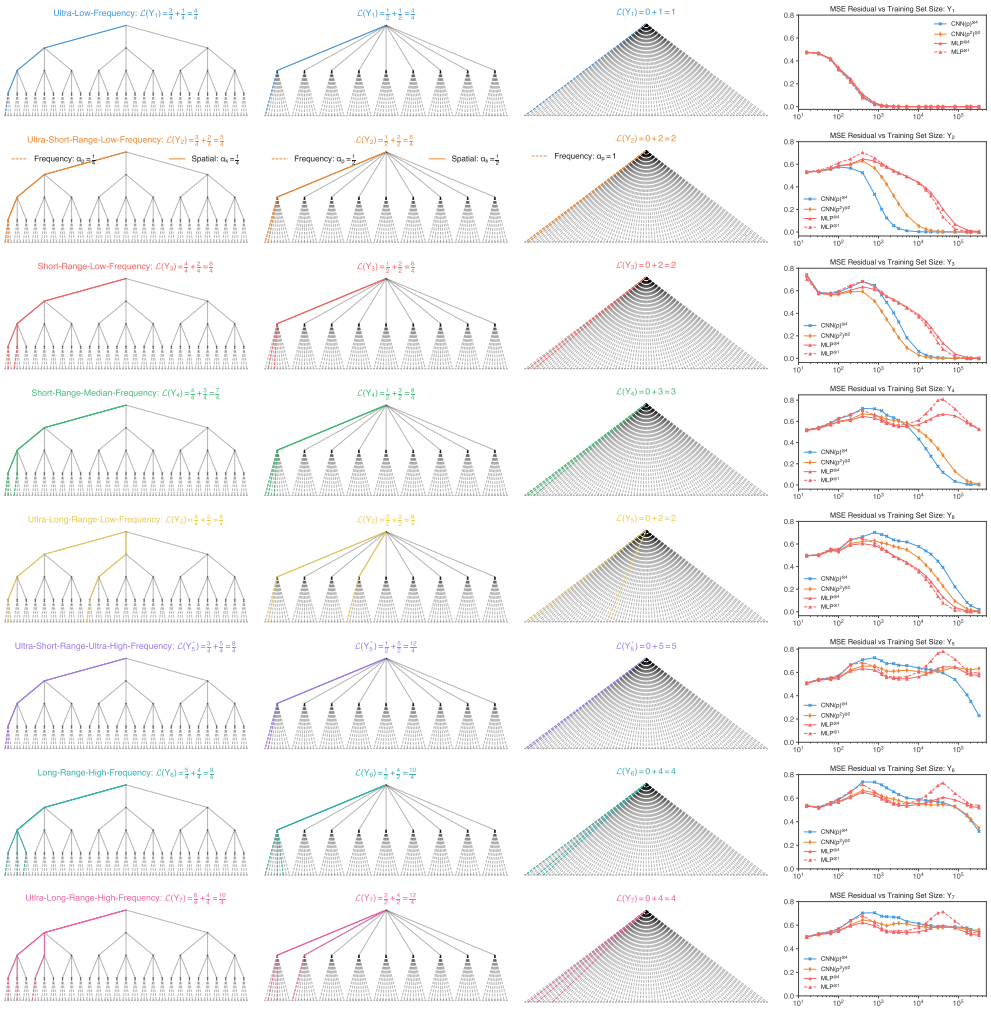
Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pp. 5393–5402, 2018.

Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

A FIGURE ZOO

A.1 VISUALIZATION OF EIGENFUNCTIONS AND LEARNING INDICES



(a) CNN$(p)^{\otimes 4}$ (b) CNN$(p^2)^{\otimes 2}$ (c) MLP (d) MSE Residual

Figure 4: **Eigenfunction vs Learning Index vs Architecture/DAG.** Rows: eigenfunctions with various space-frequency combinations. Columns: (a) "HS"-CNNs; our finest eigenstructures/architectures; (b) "D"-CNNs; finer eigenstructures/architectures; (c) MPL; coarse eigenstructures/architectures; (d) MSE of the residual of the corresponding eigenfunction using NTK-regression.

A.2 MLPS: DEPTH $\neq$ HIERARCHY

We compare a one hidden layer MLP and a four hidden layer MLP in Fig. 5. Unlike CNNs, increasing the number of layers does not improve the performance of MLPs much. This is consistent with a theoretical result from Bietti & Bach (2020).

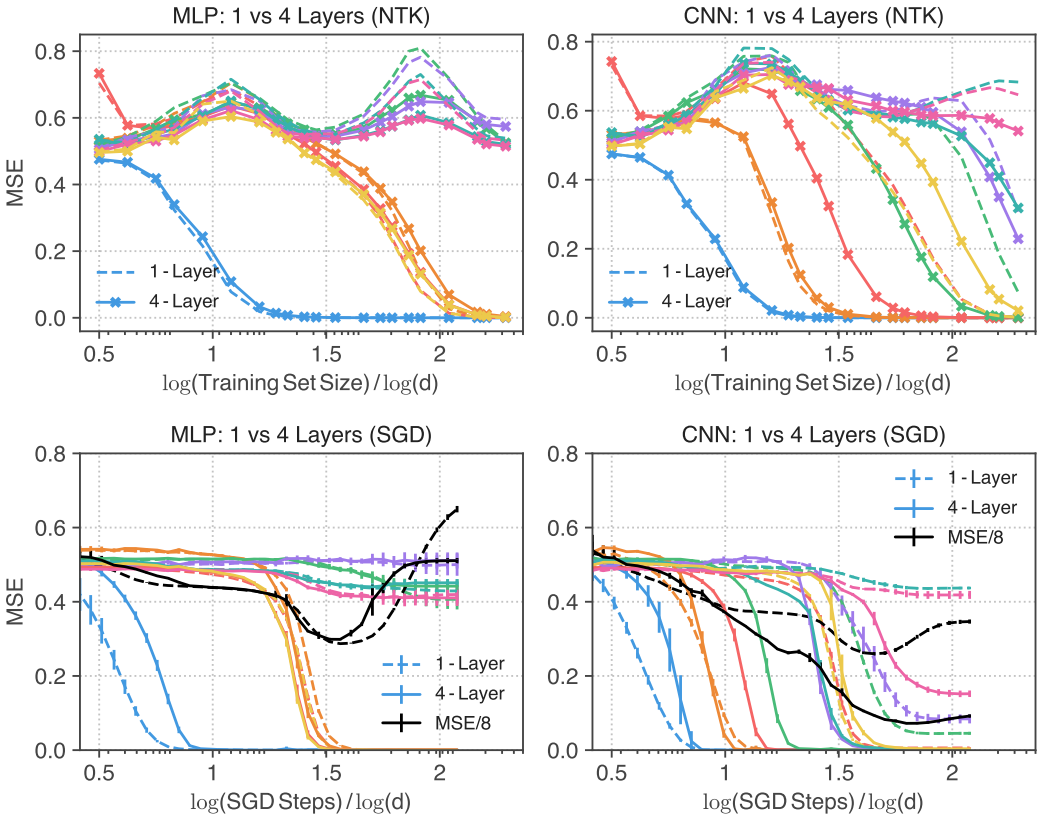


Figure 5: **MLPs do not benefits from having more layers.** We plot the learning dynamics vs training set size / SGD steps for each eigenfunction Y_i. Top: NTK regression and bottom: SGD + Momentum. Left: MLP; right: CNN. **Dashed lines / Solid lines** correspond to one-hidden/four-hidden layer networks. For both finite-width SGD training and infinite-width kernel regression, having more layers does not essentially improve performance of a MLP. This is in stark contrast to CNNs (right). By having more layers, the eigenstuctures of the kernels are refined.

B ASSUMPTIONS

Assumption-$\boldsymbol{\mathcal{G}}$. Let $\boldsymbol{\mathcal{G}} = (\mathcal{G}^{(d)})_d$. There are absolute constants $c, C > 0$ such that

(a.) For each *non-input* node $u \in \mathcal{N}^{(d)}$, there is $\alpha_u \in \Lambda_{\boldsymbol{\mathcal{G}}}$ such that

$$cd^{\alpha_u} \leq \deg(u) \leq Cd^{\alpha_u} \,. \tag{36}$$

For each edge $uv \in \mathcal{E}^{(d)}$, its weight is defined to be $\omega_{uv} \equiv \alpha_u$.

(b.) For each *input* node v, there are $d_v \in \mathbb{N}$ and $0 < \alpha_v \in \Lambda_{\boldsymbol{\mathcal{G}}}$ such that

$$cd^{\alpha_v} \leq d_v \leq Cd^{\alpha_v} \quad \text{and} \quad \sum_{v \in \mathcal{N}_0^{(d)}} \alpha_v = 1. \tag{37}$$

(c.) Let $\mathcal{N}_1^{(d)} \equiv \{u : \exists v \in \mathcal{N}_0^{(d)} \text{ s.t. } uv \in \mathcal{E}^{(d)}\}$ be the collection of nodes in the *first* hidden layer. We assume that for every $u \in \mathcal{N}_1^{(d)}$, $\alpha_u = 0$ and all children of u are input nodes.

(d.) For every $v \in \mathcal{N}^{(d)}$, $|\{u : uv \in \mathcal{E}^{(d)}\}| \leq C$. Moreover, the number of *layers* is uniformly bounded, namely, for any node u, any path from u to $o_{\mathcal{G}}$ contains at most C edges.

The first two assumptions help to create spectral gaps between eigenspaces. When d is not large, “finite-width" effect is no longer negligible and we expect that the spectra decay more smoothly. Assumption (c.) says there is no “skip" connections from the input layer to other layers except the first hidden layer.

C PROOF.

Lemma 1. *Let $\boldsymbol{r} \in \mathcal{N}_0^{(d)}$. Then*

$$\mathcal{K}_{\mathcal{G}}(0), \quad \Theta_{\mathcal{G}}(0) = 0 \tag{38}$$

and for $\boldsymbol{r} \neq 0$, for d large enough,

$$\mathcal{K}_{\mathcal{G}}^{(\boldsymbol{r})}(0), \quad \Theta_{\mathcal{G}}^{(\boldsymbol{r})}(0) \sim d^{-\mathcal{S}(\boldsymbol{r})} \quad \textit{and} \quad \|\mathcal{K}_{\mathcal{G}}^{(\boldsymbol{r})}\|_\infty,\ \|\Theta_{\mathcal{G}}^{(\boldsymbol{r})}\|_\infty \lesssim d^{-\mathcal{S}(\boldsymbol{r})} \tag{39}$$

Proof. Recall that the recursion formulas for $\mathcal{K}$ and Θ are

$$\mathcal{K}_u(\boldsymbol{t}) = \phi_u^*\left(\fint_{v:uv\in\mathcal{E}} \mathcal{K}_v(\boldsymbol{t})\right) \tag{40}$$

$$\Theta_u(\boldsymbol{t}) = \dot{\phi}_u^*\left(\fint_{v:uv\in\mathcal{E}} \mathcal{K}_v(\boldsymbol{t})\right) \fint_{v:uv\in\mathcal{E}} (\mathcal{K}_v(\boldsymbol{t}) + \Theta_v(\boldsymbol{t})) \tag{41}$$

For convenience, define $\overline{\mathcal{K}}$, $\overline{\Theta}$, $\dot{\mathcal{K}}$ as follows

$$\overline{\mathcal{K}}_u(\boldsymbol{t}) = \fint_{v:uv\in\mathcal{E}} \mathcal{K}_v(\boldsymbol{t}) \tag{42}$$

$$\overline{\Theta}_u(\boldsymbol{t}) = \fint_{v:uv\in\mathcal{E}} \Theta_v(\boldsymbol{t}) \tag{43}$$

$$\dot{\mathcal{K}}_u(\boldsymbol{t}) = \dot{\phi}_u^* \circ \overline{\mathcal{K}}_u(\boldsymbol{t})\,. \tag{44}$$

Note that

$$\mathcal{K}_u(\boldsymbol{t}) = \phi_u^* \circ \overline{\mathcal{K}}_u(\boldsymbol{t}) \quad \text{and} \quad \Theta(\boldsymbol{t}) = \dot{\mathcal{K}}_u(\boldsymbol{t})(\overline{\mathcal{K}}_u(\boldsymbol{t}) + \overline{\Theta}_u(\boldsymbol{t}))\,.$$

The equalities $\mathcal{K}_u(\mathbf{0}) = \Theta_u(\mathbf{0}) = 0$ follow easily from recursion and the fact $\phi_u^*(0) = 0$ for all $u \in \mathcal{N}^{(d)}$.

We induct on the tuple $(h, |\boldsymbol{r}|)$, where $h \geq 0$ is the number of *hidden* layers in $\mathcal{G}^{(d)}$ and $|\boldsymbol{r}|$ is the total degree of $\boldsymbol{r}$. We begin with the proof of the NNGP kernel $\mathcal{K}$.

Base Case I: $|\boldsymbol{r}| = 1$ and $h \geq 0$ is any integer. Let $u \in \mathcal{N}_0$ be such that $\boldsymbol{r} = \boldsymbol{e}_u$, where $\{\boldsymbol{e}_u\}_{u\in\mathcal{N}_0^{(d)}}$ is the standard basis. Then

$$\partial_{\boldsymbol{t}_u}\mathcal{K}_{\mathcal{G}}(\boldsymbol{t}) = \sum_{\text{path}\in\mathcal{P}(u\to o_{\mathcal{G}})} \prod_{v\in\text{path}} \deg(v)^{-1}\dot{\phi}_v^* \circ \overline{\mathcal{K}}_v(\boldsymbol{t}) \sim \sum_{\text{path}\in\mathcal{P}(u\to o_{\mathcal{G}})} \prod_{v\in\text{path}} d^{-\alpha_v}\dot{\phi}_v^* \circ \overline{\mathcal{K}}_v(\boldsymbol{t}) \tag{45}$$

Here $\mathcal{P}(u \to u')$ represents the set of paths from u to u'. By **Assumption-$\mathcal{G}$** and **Assumption-ϕ**, $|\mathcal{P}(u \to o_{\mathcal{G}})|$ is uniformly bounded and $\dot{\phi}_v^*(0) > 0$ for all hidden nodes v. Therefore

$$\begin{aligned}
\partial_{\boldsymbol{t}_u}\mathcal{K}_{\mathcal{G}}(\mathbf{0}) &\sim \sum_{\text{path}\in\mathcal{P}(u\to o_{\mathcal{G}})} d^{-\sum_{v\in\text{path}}\alpha_v}\dot{\phi}_v^* \circ \overline{\mathcal{K}}_v(\mathbf{0}) \\
&= \sum_{\text{path}\in\mathcal{P}(u\to o_{\mathcal{G}})} d^{-\sum_{v\in\text{path}}\alpha_v}\dot{\phi}_v^*(0) \\
&\sim \max_{\text{path}\in\mathcal{P}(u\to o_{\mathcal{G}})} d^{-\sum_{v\in\text{path}}\alpha_v} \\
&= d^{-\mathcal{S}(\boldsymbol{e}_u)} = d^{-\mathcal{S}(\boldsymbol{r})}
\end{aligned}$$

In the above, we have used $\overline{\mathcal{K}}_v(\mathbf{0}) = 0$ (which is due to $\phi^*(0) = 0$.) The second estimate $\|\partial_{t_u}\mathcal{K}_{\mathcal{G}}\|_\infty \lesssim d^{-\mathcal{S}(\boldsymbol{r})}$ follows from $|\dot{\phi}_v^* \circ \overline{\mathcal{K}}_v(\boldsymbol{t})| \lesssim 1$.

Base Case II: $h = 0$ and $|\boldsymbol{r}| \geq 1$ is any number. Note that $\mathcal{G}^{(d)}$ has no hidden layer and all input nodes are linked to the output node $o_{\mathcal{G}}$. The case when the activation $\phi_{o_{\mathcal{G}}}$ is the identity function is obvious and we assume $\phi_{o_{\mathcal{G}}}$ is admissible.

$$\partial_{\boldsymbol{t}}^{\boldsymbol{r}}\mathcal{K}_{\mathcal{G}}(\boldsymbol{t}) = \deg(o_{\mathcal{G}})^{-|\boldsymbol{r}|}\phi_{o_{\mathcal{G}}}^{*\,(|\boldsymbol{r}|)}\left(\fint_{u\in\mathcal{N}_0^{(d)}} \boldsymbol{t}_u\right)\,.$$

This implies Eq. (39) since $\mathcal{S}(\boldsymbol{r}) = 0$, $\deg(o_{\mathcal{G}}) \lesssim 1$ by **Assumption-$\mathcal{G}$** and $\phi^{*}_{o_{\mathcal{G}}}{}^{(|\boldsymbol{r}|)}(0) > 0$ by **Assumption-ϕ**.

Induction: $|\boldsymbol{r}| \geq 2$, $h \geq 1$ and $\boldsymbol{r} \in \mathcal{A}(\mathcal{G}^{(d)})$. We only prove the first estimate in Eq. (39) since the other one can be proved similarly. WLOG, we assume $\phi_{o_{\mathcal{G}}}$ is not the identity function and hence is semi-admissible. Let $u \in \mathcal{N}_0^{(d)}$ be such that $\boldsymbol{r}_u \geq 1$ and denote $\bar{\boldsymbol{r}} = \boldsymbol{r} - \boldsymbol{e}_u$. Then

$$\partial_{\boldsymbol{t}}^{\boldsymbol{r}} \mathcal{K}_{\mathcal{G}}(\boldsymbol{t})\Big|_{\boldsymbol{t}=\boldsymbol{0}} = \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}}(\partial_{\boldsymbol{t}}^{\boldsymbol{e}_u}\mathcal{K}_{\mathcal{G}}(\boldsymbol{t}))\Big|_{\boldsymbol{t}=\boldsymbol{0}} = \sum_{\substack{o_{\mathcal{G}}v\in\mathcal{E}\\ \partial_{\boldsymbol{t}}^{\boldsymbol{e}_u}\mathcal{K}_v\neq 0}} \deg(o_{\mathcal{G}})^{-1}\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}}\left(\dot{\mathcal{K}}_{o_{\mathcal{G}}}(\boldsymbol{t})\partial_{\boldsymbol{t}}^{\boldsymbol{e}_u}\mathcal{K}_v(\boldsymbol{t})\right)\Big|_{\boldsymbol{t}=\boldsymbol{0}} \tag{46}$$

$$= \sum_{\substack{o_{\mathcal{G}}v\in\mathcal{E}\\ \partial_{\boldsymbol{t}}^{\boldsymbol{e}_u}\mathcal{K}_v\neq 0}} \sum_{\bar{\boldsymbol{r}}_1+\bar{\boldsymbol{r}}_2=\bar{\boldsymbol{r}}} \deg(o_{\mathcal{G}})^{-1}\left(\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_1}\dot{\mathcal{K}}_{o_{\mathcal{G}}}(\boldsymbol{t})\,\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2+\boldsymbol{e}_u}\mathcal{K}_v(\boldsymbol{t})\right)\Big|_{\boldsymbol{t}=\boldsymbol{0}} \tag{47}$$

$$\sim \sum_{\substack{o_{\mathcal{G}}v\in\mathcal{E}\\ \partial_{\boldsymbol{t}}^{\boldsymbol{e}_u}\mathcal{K}_v\neq 0}} \sum_{\bar{\boldsymbol{r}}_1+\bar{\boldsymbol{r}}_2=\bar{\boldsymbol{r}}} \deg(o_{\mathcal{G}})^{-1} d^{-\mathcal{S}(\bar{\boldsymbol{r}}_1)} d^{-\mathcal{S}(n(\bar{\boldsymbol{r}}_2+\boldsymbol{e}_u;v))} \tag{48}$$

$$\sim \sum_{\substack{o_{\mathcal{G}}v\in\mathcal{E}\\ \partial_{\boldsymbol{t}}^{\boldsymbol{e}_u}\mathcal{K}_v\neq 0}} \sum_{\bar{\boldsymbol{r}}_1+\bar{\boldsymbol{r}}_2=\bar{\boldsymbol{r}}} d^{-\mathcal{S}(\bar{\boldsymbol{r}}_1)} d^{-\left(\alpha_{o_{\mathcal{G}}}+\mathcal{S}(n(\bar{\boldsymbol{r}}_2+\boldsymbol{e}_u;v))\right)} \tag{49}$$

$$\sim \sup_{\substack{o_{\mathcal{G}}v\in\mathcal{E}\\ \partial_{\boldsymbol{t}}^{\boldsymbol{e}_u}\mathcal{K}_v\neq 0}} \sup_{\bar{\boldsymbol{r}}_1+\bar{\boldsymbol{r}}_2=\bar{\boldsymbol{r}}} d^{-\left(\mathcal{S}(\bar{\boldsymbol{r}}_1)+\alpha_{o_{\mathcal{G}}}+\mathcal{S}(n(\bar{\boldsymbol{r}}_2+\boldsymbol{e}_u;v))\right)} \tag{50}$$

We have applied induction twice in Eq. (48): one to obtain the estimate $\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_1}\dot{\mathcal{K}}^{*}_{o_{\mathcal{G}}}(\boldsymbol{0}) \sim d^{-\mathcal{S}(\bar{\boldsymbol{r}}_1)}$ (with $|\bar{\boldsymbol{r}}_1| < |\boldsymbol{r}|$ and $\dot{\phi}^{*}_{o_{\mathcal{G}}}$ semi-admissible) and one to $\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2+\boldsymbol{e}_u}\mathcal{K}_v(\boldsymbol{t}) \sim d^{-\mathcal{S}(n(\bar{\boldsymbol{r}}_2+\boldsymbol{e}_u;v))}$, in which the subgraph with v as the output node has depth at most $(h-1)$. The last line follows from that both the cardinality of the tuple $(\bar{\boldsymbol{r}}_1, \bar{\boldsymbol{r}}_2)$ with $\bar{\boldsymbol{r}}_1, \bar{\boldsymbol{r}}_2 \geq \boldsymbol{0}$ and $\bar{\boldsymbol{r}}_1 + \bar{\boldsymbol{r}}_2 = \bar{\boldsymbol{r}}$ and the cardinality of $v \in \mathcal{N}_0^{(d)}$ with $o_{\mathcal{G}}v \in \mathcal{E}$ and $\partial_{\boldsymbol{t}}^{\boldsymbol{e}_u}\mathcal{K}_v \neq 0$ are finite and independent of d. From the definition of MST, it is clear that for all $(\bar{\boldsymbol{r}}_1, \bar{\boldsymbol{r}}_2)$

$$\mathcal{S}(\bar{\boldsymbol{r}}_1) + \alpha_{o_{\mathcal{G}}} + \mathcal{S}(n(\bar{\boldsymbol{r}}_2 + \boldsymbol{e}_u; v)) \geq \mathcal{S}(\bar{\boldsymbol{r}}_1) + \mathcal{S}(\bar{\boldsymbol{r}}_2 + \boldsymbol{e}_u) \geq \mathcal{S}(\boldsymbol{r}) \tag{51}$$

It remains to show that there exists at least one pair $(\bar{\boldsymbol{r}}_1, \bar{\boldsymbol{r}}_2)$ such that the above can be an equality. Let $\mathcal{T} \subseteq \mathcal{G}^{(d)}$ be a MST containing all nodes in $n(\boldsymbol{r})$. If $o_{\mathcal{G}}$ has only one child v in $\mathcal{T}$, then we choose $\bar{\boldsymbol{r}}_1 = 0$ and notice that

$$\mathcal{S}(\bar{\boldsymbol{r}}_1) + \alpha_{o_{\mathcal{G}}} + \mathcal{S}(n(\bar{\boldsymbol{r}}_2 + \boldsymbol{e}_u; v)) = 0 + \mathcal{S}(\boldsymbol{r}_2 + \boldsymbol{e}_u) = \mathcal{S}(\boldsymbol{r}) \tag{52}$$

since $\mathcal{S}(\bar{\boldsymbol{r}}_1) = 0$ and $\bar{\boldsymbol{r}}_2 + \boldsymbol{e}_u = \boldsymbol{r}$. Else, $\mathcal{T}$ contains at least two children and therefore at least two disjoint branches split from $o_{\mathcal{G}}$. Let $\mathcal{T}_u \subseteq \mathcal{T}$ be the branch that contains u and choose $\bar{\boldsymbol{r}}_2 \leq \bar{\boldsymbol{r}}$ be such that all the nodes of $(\bar{\boldsymbol{r}}_2 + \boldsymbol{e}_u)$ are contained in $\mathcal{T}_u$ and all the nodes of $\bar{\boldsymbol{r}}_1 \equiv \boldsymbol{r} - (\bar{\boldsymbol{r}}_2 + \boldsymbol{e}_u)$ are contained in $\mathcal{T}\backslash\mathcal{T}_u$. Clearly

$$\mathcal{S}(\boldsymbol{r}) = \mathcal{S}(\bar{\boldsymbol{r}}_1) + \mathcal{S}(\bar{\boldsymbol{r}}_2 + \boldsymbol{e}_u) = \mathcal{S}(\bar{\boldsymbol{r}}_1) + \mathcal{S}(n(\bar{\boldsymbol{r}}_2 + \boldsymbol{e}_u; v)) + \alpha_{o_{\mathcal{G}}}, \tag{53}$$

where v is the unique child of $o_{\mathcal{G}}$ in $\mathcal{T}_u$.

This completes the proof of the NNGP kernel $\mathcal{K}$. As the proof of the NTK part is quite similar, we will be brief and focus only on the induction step.

Induction Step of Θ: $|\boldsymbol{r}| \geq 2$, $h \geq 1$ and $\boldsymbol{r} \in \mathcal{A}(\mathcal{G}^{(d)})$. Recall that the formula of Θ is

$$\Theta_u(\boldsymbol{t}) = \dot{\phi}^{*}_u\left(\fint_{v:uv\in\mathcal{E}} \mathcal{K}_v(\boldsymbol{t})\right) \fint_{v:uv\in\mathcal{E}} (\mathcal{K}_v(\boldsymbol{t}) + \Theta_v(\boldsymbol{t})) \tag{54}$$

For $\boldsymbol{r} \in \mathcal{N}_0^{(d)}$,

$$\partial_{\boldsymbol{t}}^{\boldsymbol{r}}\Theta_{o_{\mathcal{G}}}(\boldsymbol{t}) = \sum_{\bar{\boldsymbol{r}}_1+\bar{\boldsymbol{r}}_2=\boldsymbol{r}} \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_1}\dot{\phi}^{*}_{o_{\mathcal{G}}}\left(\fint_{v:o_{\mathcal{G}}v\in\mathcal{E}} \mathcal{K}_v(\boldsymbol{t})\right) \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} \fint_{v:o_{\mathcal{G}}v\in\mathcal{E}} (\mathcal{K}_v(\boldsymbol{t}) + \Theta_v(\boldsymbol{t})) \tag{55}$$

Note that $\dot{\phi}^*_{o_\mathcal{G}}$ is semi-admissible. We apply the result of $\mathcal{K}$ to conclude that

$$\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_1} \dot{\phi}^*_{o_\mathcal{G}} \left(\fint_{v: o_\mathcal{G} v \in \mathcal{E}} \mathcal{K}_v(\boldsymbol{t}) \right) \Bigg|_{\boldsymbol{t}=\mathbf{0}} \sim d^{-\mathcal{S}(\boldsymbol{r}_1)} \tag{56}$$

$$\left\| \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_1} \dot{\phi}^*_{o_\mathcal{G}} \left(\fint_{v: o_\mathcal{G} v \in \mathcal{E}} \mathcal{K}_v(\boldsymbol{t}) \right) \right\|_\infty \lesssim d^{-\mathcal{S}(\bar{\boldsymbol{r}}_1)} \tag{57}$$

and the inductive step to conclude that

$$\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} \fint_{v: o_\mathcal{G} v \in \mathcal{E}} (\mathcal{K}_v(\boldsymbol{t}) + \Theta_v(\boldsymbol{t})) \Bigg|_{\boldsymbol{t}=\mathbf{0}} \sim \sum_{\substack{v: o_\mathcal{G} v \in \mathcal{E} \\ \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} (\mathcal{K}_v + \Theta_v) \not\equiv 0}} d^{-\mathcal{S}(\bar{\boldsymbol{r}}_2)} \quad \text{if} \quad \bar{\boldsymbol{r}}_2 \neq \mathbf{0} \quad \text{else} \quad 0 \tag{58}$$

$$\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} \fint_{v: o_\mathcal{G} v \in \mathcal{E}} (\mathcal{K}_v(\boldsymbol{t}) + \Theta_v(\boldsymbol{t})) \lesssim \sum_{\substack{v: o_\mathcal{G} v \in \mathcal{E} \\ \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} (\mathcal{K}_v + \Theta_v) \not\equiv 0}} d^{-\mathcal{S}(\bar{\boldsymbol{r}}_2)} \,. \tag{59}$$

Note that

$$|\{v : o_\mathcal{G} v \in \mathcal{E}^{(d)} \;\text{ and }\; \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} (\mathcal{K}_v + \Theta_v) \not\equiv 0\}| \lesssim 1 \,.$$

Thus

$$\|\partial_{\boldsymbol{t}}^{\boldsymbol{r}} \Theta_{o_\mathcal{G}}(\boldsymbol{t})\|_\infty \lesssim \sum_{\bar{\boldsymbol{r}}_1 + \bar{\boldsymbol{r}}_2 = \boldsymbol{r}} d^{-\mathcal{S}(\bar{\boldsymbol{r}}_1) - \mathcal{S}(\bar{\boldsymbol{r}}_2)} \lesssim d^{-\mathcal{S}(\boldsymbol{r})}. \tag{60}$$

To control the lower bound, let $\mathcal{T}$ be a MST containing $n(\boldsymbol{r})$. If $\deg(o_\mathcal{G}; \mathcal{T}) = 1$, then we can choose $\bar{\boldsymbol{r}}_1 = 0$ and $\bar{\boldsymbol{r}}_2 = \boldsymbol{r} \neq \mathbf{0}$. Notice that there is at least one child node v of $o_\mathcal{G}$ with $\partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} (\mathcal{K}_v + \Theta_v) \not\equiv 0$. Therefore

$$\sum_{v: \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} (\mathcal{K}_v + \Theta_v) \not\equiv 0} d^{-\mathcal{S}(\bar{\boldsymbol{r}}_2)} \gtrsim d^{-\mathcal{S}(\bar{\boldsymbol{r}}_2)} = d^{-\mathcal{S}(\boldsymbol{r})} \tag{61}$$

Combining with

$$\dot{\phi}^*_{o_\mathcal{G}} \left(\fint_{v: o_\mathcal{G} v \in \mathcal{E}} \mathcal{K}_v(\mathbf{0}) \right) = \dot{\phi}^*_{o_\mathcal{G}}(0) > 0 \tag{62}$$

we have

$$\partial_{\boldsymbol{t}}^{\boldsymbol{r}} \Theta_{o_\mathcal{G}}(\mathbf{0}) \gtrsim d^{-\mathcal{S}(\boldsymbol{r})}$$

It remains to handle the $\deg(o_\mathcal{G}; \mathcal{T}) > 1$ case. We choose $(\bar{\boldsymbol{r}}_1, \bar{\boldsymbol{r}}_2)$ such that one branch of $\mathcal{T}$ is the MST that contains $n(\bar{\boldsymbol{r}}_2)$ and the $o_\mathcal{G}$ and the remaining branch(es) is a MST that contains $n(\bar{\boldsymbol{r}}_1)$ and the $o_\mathcal{G}$. Then

$$\begin{aligned} \partial_{\boldsymbol{t}}^{\boldsymbol{r}} \Theta_{o_\mathcal{G}}(\mathbf{0}) \gtrsim & \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_1} \dot{\phi}^*_{o_\mathcal{G}} \left(\fint_{v: o_\mathcal{G} v \in \mathcal{E}} \mathcal{K}_v(\boldsymbol{t}) \right) \partial_{\boldsymbol{t}}^{\bar{\boldsymbol{r}}_2} \fint_{v: o_\mathcal{G} v \in \mathcal{E}} (\mathcal{K}_v(\boldsymbol{t}) + \Theta_v(\boldsymbol{t})) \Bigg|_{\boldsymbol{t}=\mathbf{0}} \\ \gtrsim & d^{-\mathcal{S}(\boldsymbol{r}_1) - \mathcal{S}(\boldsymbol{r}_2)} = d^{-\mathcal{S}(\boldsymbol{r})} \end{aligned}$$

□

C.1 Legendre Polynomials, Spherical Harmonics and their Tensor Products.

Our notation follows closely from (Frye & Efthimiou, 2012).

Legendre Polynomials. Let $d_{\text{in}} \in \mathbb{N}^*$ and $\omega_{d_{\text{in}}}$ be the measure defined on the interval $I = [-1, 1]$

$$\omega_{d_{\text{in}}}(t) = (1 - t^2)^{(d_{\text{in}} - 3)/2} \tag{63}$$

The Legendre polynomials[2] $\{P_r(t) : r \in \mathbb{N}\}$ is an orthogonal basis for the Hilbert space $L^2(I, \omega_{d_{\text{in}}})$, i.e.

$$\int_I P_r(t)P_{r'}(t)\omega_{d_{\text{in}}}(t)dt = 0 \quad \text{if} \quad r \neq r' \quad \text{else} \quad N(d_{\text{in}}, r)^{-1}\left(\frac{|\mathbb{S}_{d_{\text{in}}-1}|}{|\mathbb{S}_{d_{\text{in}}-2}|}\right) \tag{64}$$

Here $P_r(t)$ is a degree r polynomials with $P_r(1) = 1$ that satisfies the formula below, $N(d_{\text{in}}, r)$ is the cardinality of degree r spherical harmonics in $\mathbb{R}^{d_{\text{in}}}$ and $|\mathbb{S}_{d_{\text{in}}-1}|$ is the measure of $\mathbb{S}_{d_{\text{in}}-1}$.

Lemma 2 (Rodrigues Formula. Proposition 4.19 (Frye & Efthimiou, 2012))**.**

$$P_r(t) = c_r\omega_{d_{in}}^{-1}(t)\left(\frac{d}{dt}\right)^r(1-t^2)^{r+(d_{in}-3)/2}\,, \tag{65}$$

where

$$c_r = \frac{(-1)^r}{2^r(r+(d_{in}-3)/2)_r} \tag{66}$$

In the above lemma, $(x)_l$ denotes the falling factorial

$$(x)_l \equiv x(x-1)\cdots(x-l+1) \tag{67}$$
$$(x)_0 \equiv 1 \tag{68}$$

Spherical Harmonics. Let $d\mathbb{S}_{d_{\text{in}}-1}$ define the (un-normalized) uniform measure on the unit sphere $\mathbb{S}_{d_{\text{in}}-1}$. Then

$$|\mathbb{S}_{d_{\text{in}}-1}| \equiv \int_{\mathbb{S}_{d_{\text{in}}-1}} d\mathbb{S}_{d_{\text{in}}-1} = \frac{2\pi^{d_{\text{in}}/2}}{\Gamma(\frac{d_{\text{in}}}{2})}\,. \tag{69}$$

The normalized measure on this sphere is defined to be

$$d\sigma_{d_{\text{in}}} = \frac{1}{|\mathbb{S}_{d_{\text{in}}-1}|}d\mathbb{S}_{d_{\text{in}}-1} \quad \text{and} \quad \int_{\mathbb{S}_{d_{\text{in}}-1}} d\sigma_{d_{\text{in}}} = 1\,. \tag{70}$$

The spherical harmonics $\{Y_{r,l}\}_{r,l}$ in $\mathbb{R}^{d_{\text{in}}}$ are homogeneous harmonic polynomials that form an orthonormal basis in $L^2(\mathbb{S}_{d_{\text{in}}-1}, \sigma_{d_{\text{in}}})$

$$\int_{\xi\in\mathbb{S}_{d_{\text{in}}-1}} Y_{r,l}(\xi)Y_{r',l'}(\xi)d\sigma_{d_{\text{in}}} = \delta_{(r,l)=(r',l')}\,. \tag{71}$$

Here $Y_{r,l}$ denotes the l-th spherical harmonic whose degree is r, where $r \in \mathbb{N}$, $l \in [N(d_{\text{in}}, r)]$ and

$$N(d_{\text{in}}, r) = \frac{2r + d_{\text{in}} - 2}{r}\binom{d_{\text{in}} + r - 3}{r-1} \sim (d_{\text{in}})^r/r! \quad \text{as} \quad d_{\text{in}} \to \infty\,. \tag{72}$$

The Legendre polynomials and spherical harmonics are related through the addition theorem.

Lemma 3 (Addition Theorem. Theorem 4.11 (Frye & Efthimiou, 2012))**.**

$$P_r(\xi^T\eta) = \frac{1}{N(d_{in}, r)}\sum_{l\in[N(d_{in},r)]} Y_{r,l}(\xi)Y_{r,l}(\eta), \quad \xi, \eta \in \mathbb{S}_{d_{in}-1}\,. \tag{73}$$

Tensor Products. Let $p = |\mathcal{N}_0^{(d)}|$, $\boldsymbol{d} = (d_u)_{|u\in\mathcal{N}_0^{(d)}|}$, $\boldsymbol{r} \in \mathbb{N}^{|\mathcal{N}_0^{(d)}|} \simeq \mathbb{N}^p$, $I^{|\mathcal{N}_0^{(d)}|} \simeq I^p = [-1, 1]^p$ and $\boldsymbol{\omega} = \bigotimes_{u\in\mathcal{N}_0^{(d)}} \omega_{d_u}^p$ be the product measure on I^p. Then the (product of) Legendre polynomials

$$\boldsymbol{P_r}(\boldsymbol{t}) = \prod_{u\in\mathcal{N}_0^{(d)}} P_{r_u}(t_u)\,, \quad \boldsymbol{t} = (t_u)_{u\in\mathcal{N}_0^{(d)}} \in I^p\,, \tag{74}$$

[2]More accurate, this should be called Gegenbauer Polynomials. However, we decide to stick to the terminology in (Frye & Efthimiou, 2012)

which form an orthogonal basis for the Hilbert space $L^2(I^p, \boldsymbol{\omega}) = \bigotimes_{u \in \mathcal{N}_0^{(d)}} L^2(I, \omega_{d_u})$. Similarly, the tensor product of spherical harmonics

$$\boldsymbol{Y}_{\boldsymbol{r},\boldsymbol{l}} = \prod_{u \in \mathcal{N}_0^{(d)}} Y_{r_u, l_u}, \quad \boldsymbol{l} = (l_u)_{u \in \mathcal{N}_0^{(d)}} \in [\boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r})] \equiv \prod_{u \in \mathcal{N}_0^{(d)}} [N(d_u, r)] \tag{75}$$

form an orthonormal basis for the product space

$$L^2(\boldsymbol{\mathcal{X}}, \boldsymbol{\sigma}) \equiv \bigotimes_{u \in \mathcal{N}_0^{(d)}} L^2(\mathbb{S}_{d_u - 1}, \sigma_{d_u}) \tag{76}$$

Elements in the set $\{\boldsymbol{Y}_{\boldsymbol{r},\boldsymbol{l}}\}_{\boldsymbol{l} \in [\boldsymbol{N}(\boldsymbol{d},\boldsymbol{r})]}$ are called degree (order) $\boldsymbol{r}$ spherical harmonics in $L^2(\boldsymbol{\mathcal{X}}, \boldsymbol{\sigma})$ and also degree r spherical harmonics if $|\boldsymbol{r}| = r \in \mathbb{N}$.

Theorem 3. *We have the following, for $\mathcal{K} = \mathcal{K}_{\mathcal{G}^{(d)}}$ or $\mathcal{K} = \Theta_{\mathcal{G}^{(d)}}$*

$$\mathcal{K}(\boldsymbol{t}) = \sum_{\boldsymbol{r} \in \mathbb{N}^{\mathcal{N}_0^{(d)}}} \hat{\mathcal{K}}(\boldsymbol{r}) \boldsymbol{P}_{\boldsymbol{r}}(\boldsymbol{t}) \qquad \textit{with} \qquad \hat{\mathcal{K}}(\boldsymbol{r}) \sim d^{-\mathcal{S}(\boldsymbol{r})} \quad \textit{if} \quad \boldsymbol{r} \neq 0 \quad \textit{else} \quad 0. \tag{77}$$

Note that Theorem 1 follows from this theorem and the addition theorem.

Proof of Theorem 1. Assume $\boldsymbol{r} \neq \boldsymbol{0}$. Indeed, setting

$$\boldsymbol{\xi} = (\boldsymbol{\xi}_u)_{u \in \mathcal{N}_0^{(d)}} \in \boldsymbol{\mathcal{X}}, \quad \boldsymbol{\eta} = (\boldsymbol{\eta}_u)_{u \in \mathcal{N}_0^{(d)}} \in \boldsymbol{\mathcal{X}} \; and \quad \boldsymbol{t} = (\boldsymbol{t}_u)_{u \in \mathcal{N}_0^{(d)}} = (\boldsymbol{\xi}_u^T \boldsymbol{\eta}_u / \boldsymbol{d}_u)_{u \in \mathcal{N}_0^{(d)}},$$

we have

$$\boldsymbol{P}_{\boldsymbol{r}}(\boldsymbol{t}) = \prod_{u \in \mathcal{N}_0^{(d)}} P_{\boldsymbol{r}_u}(\boldsymbol{t}_u) = \prod_{u \in \mathcal{N}_0^{(d)}} N(\boldsymbol{d}_u, \boldsymbol{r}_u)^{-1} \sum_{\boldsymbol{l}_u \in N(\boldsymbol{d}_u, \boldsymbol{r}_u)} Y_{\boldsymbol{r}_u, \boldsymbol{l}_u}(\boldsymbol{\xi}_u / \sqrt{\boldsymbol{d}_u}) Y_{\boldsymbol{r}_u, \boldsymbol{l}_u}(\boldsymbol{\eta}_u / \sqrt{\boldsymbol{d}_u}) \tag{78}$$

$$= \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r})^{-1} \sum_{\boldsymbol{l} \in [\boldsymbol{N}(\boldsymbol{d},\boldsymbol{r})]} \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi}) \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\eta}). \tag{79}$$

Then Theorem 1 follows by noticing

$$\mathcal{K}(\boldsymbol{r}) \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r})^{-1} \sim d^{-\mathcal{S}(\boldsymbol{r})} d^{-\sum_{u \in \mathcal{N}_0^{(d)}} \boldsymbol{r}_u \alpha_u} = d^{-\mathcal{L}(\boldsymbol{r})} \tag{80}$$

□

Proof of Theorem 3. From the orthogonality,

$$\hat{\mathcal{K}}(\boldsymbol{r}) = \langle \mathcal{K}, \boldsymbol{P}_{\boldsymbol{r}} \rangle / \|\boldsymbol{P}_{\boldsymbol{r}}\|_{L^2(I^p, \boldsymbol{\omega})}{}^2 \tag{81}$$

We begin with the denominator. Note that

$$\|\boldsymbol{P}_{\boldsymbol{r}}\|_{L^2(I^p, \boldsymbol{\sigma})}{}^2 = \prod_{u \in \mathcal{N}_0^{(d)}} \|P_{\boldsymbol{r}_u}\|^2_{L^2(I, \omega_{\boldsymbol{d}_u})} = \boldsymbol{N}(\boldsymbol{d}; \boldsymbol{r})^{-1} \prod_{u \in \mathcal{N}_0^{(d)}} (|\mathbb{S}_{\boldsymbol{d}_u - 1}| / |\mathbb{S}_{\boldsymbol{d}_u - 2}|) \tag{82}$$

By applying Lemma 2, integration by parts and continuity of $\mathcal{K}^{(\boldsymbol{r})}$ on the boundary ∂I^p

$$\langle \mathcal{K}, \boldsymbol{P}_{\boldsymbol{r}} \rangle_{L^2(I^p, \boldsymbol{\omega})} = c_{\boldsymbol{r}} \int_{I^p} \mathcal{K}(\boldsymbol{t}) \left(\frac{d}{d\boldsymbol{t}} \right)^{\boldsymbol{r}} \left(1 - \boldsymbol{t}^2\right)^{\boldsymbol{r} + (\boldsymbol{d} - 3)/2} d\boldsymbol{t} \tag{83}$$

$$= (-1)^{\boldsymbol{r}} c_{\boldsymbol{r}} \int_{I^p} \mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{t}) \left(1 - \boldsymbol{t}^2\right)^{\boldsymbol{r} + (\boldsymbol{d} - 3)/2} d\boldsymbol{t} \tag{84}$$

$$= (-1)^{\boldsymbol{r}} c_{\boldsymbol{r}} \left(\mathcal{M}(\mathcal{K}, \boldsymbol{d}) + \epsilon(\mathcal{K}, \boldsymbol{d}) \right) \tag{85}$$

where $\mathcal{K}^{(\boldsymbol{r})}$ is the $\boldsymbol{r}$ derivative of $\mathcal{K}$, the coefficient $c_{\boldsymbol{r}}$ is given by Lemma 2

$$c_{\boldsymbol{r}} = \prod_{u \in \mathcal{N}_0^{(d)}} c_{\boldsymbol{r}_u} = \prod_{u \in \mathcal{N}_0^{(d)}} \frac{(-1)^{\boldsymbol{r}_u}}{2^{\boldsymbol{r}_u} (\boldsymbol{r}_u + (\boldsymbol{d} - 3)/2)_{\boldsymbol{r}_u}} \sim \prod_{u \in \mathcal{N}_0^{(d)}} (-1)^{\boldsymbol{r}_u} \boldsymbol{d}_u^{-\boldsymbol{r}_u} = (-1)^{\boldsymbol{r}} \boldsymbol{d}^{-\boldsymbol{r}} \tag{86}$$

and the major and error terms are given by

$$\mathcal{M}(\mathcal{K}, \boldsymbol{d}) \equiv \mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{0}) \int_{I^p} \left(1 - \boldsymbol{t}^2\right)^{\boldsymbol{r}+(\boldsymbol{d}-3)/2} d\boldsymbol{t} = \mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{0}) \prod_{u \in \mathcal{N}_0^{(d)}} \frac{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-1}|}{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-2}|} \tag{87}$$

$$\epsilon(\mathcal{K}, \boldsymbol{d}) \equiv \int_{I^p} \left(\mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{t}) - \mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{0})\right) \left(1 - \boldsymbol{t}^2\right)^{\boldsymbol{r}+(\boldsymbol{d}-3)/2} d\boldsymbol{t} \tag{88}$$

The mean value theorem gives

$$|\mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{t}) - \mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{0})| \leq \sum_{u \in \mathcal{N}_0^{(d)}} \|\mathcal{K}^{(\boldsymbol{r}+\boldsymbol{e}_u)}\|_{L^\infty(I^p)} |\boldsymbol{t}_u| \tag{89}$$

and the error term $|\epsilon(\mathcal{K}, \boldsymbol{d})|$ is bounded above by

$$\int_{I^p} \left(1 - \boldsymbol{t}^2\right)^{\boldsymbol{r}+(\boldsymbol{d}-3)/2} d\boldsymbol{t} \sum_{u \in \mathcal{N}_0^{(d)}} \|\mathcal{K}^{(\boldsymbol{r}+\boldsymbol{e}_u)}\|_{L^\infty(I^p)} \left(\frac{\int_I |\boldsymbol{t}_u| \left(1 - \boldsymbol{t}_u^2\right)^{\boldsymbol{r}_u+(\boldsymbol{d}_u-3)/2} d\boldsymbol{t}_u}{\int_I \left(1 - \boldsymbol{t}_u^2\right)^{\boldsymbol{r}_u+(\boldsymbol{r}_u-3)/2} d\boldsymbol{t}_u} \right) \tag{90}$$

$$\sim \prod_{u \in \mathcal{N}_0^{(d)}} \frac{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-1}|}{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-2}|} \sum_{u \in \mathcal{N}_0^{(d)}} \|\mathcal{K}^{(\boldsymbol{r}+\boldsymbol{e}_u)}\|_{L^\infty(I^p)} \boldsymbol{d}_u^{-1} \left(\frac{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-1}|}{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-2}|} \right)^{-1} . \tag{91}$$

Since for any $\alpha \in \mathbb{N}$, as $\boldsymbol{d}_u \to \infty$,

$$\frac{|\mathbb{S}_{\alpha+\boldsymbol{d}_u-1}|}{|\mathbb{S}_{\alpha+\boldsymbol{d}_u-2}|} = \pi^{\frac{1}{2}} \Gamma((\alpha + \boldsymbol{d}_u - 1)/2)/\Gamma((\alpha + \boldsymbol{d}_u)/2) \sim \pi^{\frac{1}{2}} (\boldsymbol{d}_u/2)^{-\frac{1}{2}} \sim (\boldsymbol{d}_u)^{-\frac{1}{2}} , \tag{92}$$

we have

$$|\epsilon(\mathcal{K}, \boldsymbol{d})| \lesssim \sum_{u \in \mathcal{N}_0^{(d)}} \|\mathcal{K}^{(\boldsymbol{r}+\boldsymbol{e}_u)}\|_{L^\infty(I^p)} \boldsymbol{d}_u^{-\frac{1}{2}} \prod_{u \in \mathcal{N}_0^{(d)}} \frac{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-1}|}{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-2}|} . \tag{93}$$

We claim that (which will be proved later)

$$\sum_{u \in \mathcal{N}_0^{(d)}} \|\mathcal{K}^{(\boldsymbol{r}+\boldsymbol{e}_u)}\|_{L^\infty(I^p)} \lesssim d^{-\mathcal{S}(\boldsymbol{r})} \tag{94}$$

which implies

$$\langle \mathcal{K}, \boldsymbol{P}_{\boldsymbol{r}} \rangle_{L^2(I^p, \omega^p_{d_{\text{in}}})} = c_{\boldsymbol{r}} \left(\mathcal{K}^{(\boldsymbol{r})}(0) + \mathcal{O}\left(d^{-\mathcal{S}(\boldsymbol{r})} (\min_{u \in \mathcal{N}_0^{(d)}} \boldsymbol{d}_u)^{-\frac{1}{2}} \right) \right) \prod_{u \in \mathcal{N}_0^{(d)}} \frac{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-1}|}{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-2}|} \tag{95}$$

Plugging back to Eq. (81), we have

$$\hat{\mathcal{K}}(\boldsymbol{r}) = (-1)^{\boldsymbol{r}} c_{\boldsymbol{r}} \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r}) \left(\mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{0}) + \mathcal{O}\left(d^{-\mathcal{S}(\boldsymbol{r})} (\min_{u \in \mathcal{N}_0^{(d)}} \boldsymbol{d}_u)^{-\frac{1}{2}} \right) \right) \left(\prod_{u \in \mathcal{N}_0^{(d)}} \frac{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-1}|}{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-2}|} \left(\frac{|\mathbb{S}_{\boldsymbol{d}_u-1}|}{|\mathbb{S}_{\boldsymbol{d}_u-2}|} \right)^{-1} \right) \tag{96}$$

Since, for $\boldsymbol{r}$ fixed and as $\boldsymbol{d}_u \to \infty$ for all $u \in \mathcal{N}_0^{(d)}$

$$\frac{c_{\boldsymbol{r}}}{(-1)^{\boldsymbol{r}} \boldsymbol{d}^{-\boldsymbol{r}}} \to 1 \quad \text{and} \quad \frac{\boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r})}{\boldsymbol{d}^{\boldsymbol{r}}/\boldsymbol{r}!} \to 1 \quad \text{and} \quad \left(\prod_{u \in \mathcal{N}_0^{(d)}} \frac{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-1}|}{|\mathbb{S}_{2\boldsymbol{r}_u+\boldsymbol{d}_u-2}|} \left(\frac{|\mathbb{S}_{\boldsymbol{d}_u-1}|}{|\mathbb{S}_{\boldsymbol{d}_u-2}|} \right)^{-1} \right) \to 1 \tag{97}$$

and thus

$$\hat{\mathcal{K}}(\boldsymbol{r}) \sim \boldsymbol{r}!^{-1} \left(\mathcal{K}^{(\boldsymbol{r})}(\boldsymbol{0}) + \mathcal{O}\left(d^{-\mathcal{S}(\boldsymbol{r})} (\min_{u \in \mathcal{N}_0^{(d)}} \boldsymbol{d}_u)^{-\frac{1}{2}} \right) \right) \tag{98}$$

It remains to verify Eq. (94). By Lemma 1, we only need to show that

$$\sum_{u\in\mathcal{N}_0^{(d)}} d^{-\mathcal{S}(\boldsymbol{r}+\boldsymbol{e}_u)} \lesssim d^{-\mathcal{S}(\boldsymbol{r})} \tag{99}$$

We prove this by induction on the number of hidden layers of $\mathcal{G}^{(d)}$. The base case is obvious. Now suppose the depth of $\mathcal{G}^{(d)}$ is h. Let $\mathcal{C}(\boldsymbol{r})$ be the set of children of $o_{\mathcal{G}}$ that are ancestors of at least one node of $n(\boldsymbol{r})$. We split $\mathcal{N}_0^{(d)}$ into two disjoint sets

$$\mathcal{Q}(\boldsymbol{r}) \equiv \{u \in \mathcal{N}_0^{(d)} : \exists v \in \mathcal{C}(\boldsymbol{r}) \text{ s.t. } \mathcal{P}(u \to v) \neq \emptyset\} \quad \text{and} \quad \mathcal{N}_0^{(d)} \backslash \mathcal{Q}_0(\boldsymbol{r})\,.$$

For $u \notin \mathcal{Q}(\boldsymbol{r})$, we have $\mathcal{S}(\boldsymbol{r}+\boldsymbol{e}_u) = \mathcal{S}(\boldsymbol{r}) + \mathcal{S}(\boldsymbol{e}_u)$ and hence

$$\sum_{u\notin\mathcal{Q}(\boldsymbol{r})} d^{-\mathcal{S}(\boldsymbol{r}+\boldsymbol{e}_u)} = \sum_{u\notin\mathcal{Q}(\boldsymbol{r})} d^{-\mathcal{S}(\boldsymbol{r})-\mathcal{S}(\boldsymbol{e}_u)} = d^{-\mathcal{S}(\boldsymbol{r})} \sum_{u\notin\mathcal{Q}(\boldsymbol{r})} d^{-\mathcal{S}(\boldsymbol{e}_u)} \lesssim d^{-\mathcal{S}(\boldsymbol{r})}\,. \tag{100}$$

In the last inequality above, we have used

$$\sum_{u\notin\mathcal{Q}(\boldsymbol{r})} d^{-\mathcal{S}(\boldsymbol{e}_u)} \leq \sum_{u\in\mathcal{N}_0^{(d)}} d^{-\mathcal{S}(\boldsymbol{e}_u)} \sim 1\,. \tag{101}$$

To estimate the remaining, we use induction. Note that $|\mathcal{C}(\boldsymbol{r})|$ is finite and independent of d. Then

$$\sum_{u\in\mathcal{Q}(\boldsymbol{r})} d^{-\mathcal{S}(\boldsymbol{r}+\boldsymbol{e}_u)} \leq \sum_{v\in\mathcal{C}(\boldsymbol{r})} \sum_{u\in\mathcal{Q}(\boldsymbol{r})} d^{-\alpha_{o_{\mathcal{G}}}-\mathcal{S}(n(\boldsymbol{r}+\boldsymbol{e}_u;v))} \tag{102}$$

$$= d^{-\alpha_{o_{\mathcal{G}}}} \sum_{v\in\mathcal{C}(\boldsymbol{r})} \sum_{u\in\mathcal{Q}(\boldsymbol{r})} d^{-\mathcal{S}(n(\boldsymbol{r}+\boldsymbol{e}_u;v))} \tag{103}$$

$$\lesssim |\mathcal{C}(\boldsymbol{r})| d^{-\alpha_{o_{\mathcal{G}}}} \max_{v\in\mathcal{C}(\boldsymbol{r})} d^{-\mathcal{S}(n(\boldsymbol{r};v))} \sim d^{-\mathcal{S}(\boldsymbol{r})} \tag{104}$$

We have used induction on the sub-graph with v as the output node. □

D Proof of Theorem 2

Let $\mathcal{G}^{(d)}$ be a DAG associated to the convolutional networks whose filter sizes in the l-th layer is $k_l = [d^{\alpha_l}]$, for $0 \leq l \leq L+1$, in which we treat the *flatten-dense* readout layer as a convolution with filter size $[d^{\alpha_{L+1}}]$. Note that we have set $\alpha_p = \alpha_0$ and $\alpha_w = \alpha_{L+1}$. We also assume an *activation* layer after the *flatten-dense* layer, which does not essentially alter the topology of the DAG.

We need the following dimension counting lemma.

Lemma 4. *Let $r \in \mathcal{L}(\mathcal{G}^{(d)})$. Then*

$$\dim\left(\operatorname{span}\left\{\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} : \quad \mathcal{L}(\boldsymbol{r}) = r,\ \boldsymbol{l}\ \in \boldsymbol{N}(\boldsymbol{d},\boldsymbol{r})\right\}\right) \sim d^r \tag{105}$$

To prove Theorem 2, we only need to verify the assumptions of Theorem 4 in Mei et al. (2021). They follow directly from Theorem 1, Lemma 4 and the hypercontractivity of spherial harmonics Beckner (1992).

It is convenient to group eigenspaces together according to the learning indices $\mathcal{L}(\mathcal{G}^{(d)})$. Recall that $\mathcal{L}(\mathcal{G}^{(d)}) = (r_1 \leq r_2 \leq r_3 \dots)$. Let

$$E_i = \operatorname{span}\{\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} : \mathcal{L}(\boldsymbol{r}) = r_i\} \tag{106}$$

Then by Theorem 1 and Lemma 4,

$$\dim(E_i) \sim d^{r_i} \quad \text{and} \quad \lambda(g) \sim d^{-r_i} \quad \forall g \in E_i,\ g \neq \mathbf{0}. \tag{107}$$

We proceed to verify the assumptions of Theorem 4 in Mei et al. (2021).

D.1 VERIFYING ASSUMPTION 4 IN MEI ET AL. (2021)

Assumption 4 (a) Hypercontractivity. Let $D_s = \text{span}\{\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} : |\mathcal{L}(\boldsymbol{r})| < s\}$, where $s = \inf\{u \in \mathcal{L}(\mathcal{G}^{(d)}) : u > 2r + 100\}$. Let $\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} \in D_s$. By hypercontractivity of spherical harmonics,

$$\|\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}\|_{L^{2q}}^2 = \prod_{u \in \mathcal{N}_0^{(d)}} \|\overline{Y}_{\boldsymbol{r}_u,\boldsymbol{l}_l}\|_{L^{2q}}^2 \tag{108}$$

$$\leq \prod_{u:\boldsymbol{r}_u \neq 0} (q-1)^{\boldsymbol{r}_u} \|\overline{Y}_{\boldsymbol{r}_u,\boldsymbol{l}_l}\|_2^2 = (q-1)^{|\boldsymbol{r}|} \|\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}\|_{L^2}^2 \tag{109}$$

$$\leq (q-1)^{s/\alpha_0} \|\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}\|_{L^2}^2 \tag{110}$$

Let $h = \sum_{\boldsymbol{r},\boldsymbol{l}} a_{\boldsymbol{r},\boldsymbol{l}} \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} \in D_s$. Then by the triangle inequality and the L^2-orthogonality

$$\|h\|_{L^{2q}}^2 = \|\sum_{\boldsymbol{r},\boldsymbol{l}} a_{\boldsymbol{r},\boldsymbol{l}} \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}\|_{L^{2q}}^2 \leq (q-1)^{s/\alpha_0} (\sum_{\boldsymbol{r},\boldsymbol{l}} a_{\boldsymbol{r},\boldsymbol{l}} \|\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}\|_{L^2})^2 = (q-1)^{s/\alpha_0} \|\sum_{\boldsymbol{r},\boldsymbol{l}} a_{\boldsymbol{r},\boldsymbol{l}} \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}\|_{L^2}^2 \tag{111}$$

Assumption 4 (b) Properly decaying eigenvalues. We list all eigenvalues $\lesssim d^{-s}$ by $\{\lambda_j\}_{j \geq j_0}$. By Eq. (107), for $l > 1$,

$$\sum_{j \geq j_0} \lambda_j^l \sim \sum_{i: r_i \geq s} \dim(E_i) d^{-l r_i} \sim \sum_{i: r_i \geq s} d^{r_i} d^{-l r_i} = d^{-(l-1)s} \tag{112}$$

and for $l = 1$,

$$\sum_{j \geq j_0} \lambda_j^l \sim 1. \tag{113}$$

Thus

$$\frac{(\sum_{j \geq j_0} \lambda_j^l)^2}{\sum_{j \geq j_0} \lambda_j^{2l}} \sim \frac{d^{-2s(l-1)}}{d^{-s(2l-1)}} = d^s > d^{2r+1}. \tag{114}$$

Assumption 4 (c) Concentration of diagonal elements. We have

$$\mathbb{E}_{\boldsymbol{\eta}} \sum_{\boldsymbol{r},\boldsymbol{l}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 |\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi}) \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\eta})|^2 \tag{115}$$

$$= \sum_{\boldsymbol{r},\boldsymbol{l}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 |\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi})|^2 \tag{116}$$

$$= \sum_{\boldsymbol{r}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 \sum_{\boldsymbol{l}} |\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi})|^2 \tag{117}$$

$$= \sum_{\boldsymbol{r}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r}) \boldsymbol{P}_{\boldsymbol{r}}(\boldsymbol{1}) = \sum_{\boldsymbol{r}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r}) \,. \tag{118}$$

Similarly,

$$\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{\eta}} \sum_{\boldsymbol{r},\boldsymbol{l}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 |\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi}) \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\eta})|^2 = \sum_{\boldsymbol{r}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r}) \,. \tag{119}$$

Thus

$$\mathbb{E}_{\boldsymbol{\eta}} \sum_{\boldsymbol{r},\boldsymbol{l}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 |\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi}) \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\eta})|^2 - \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{\eta}} \sum_{\boldsymbol{r},\boldsymbol{l}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r})^2 |\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi}) \overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\eta})|^2 = 0 \,. \tag{120}$$

For the diagonal terms,

$$\sum_{\boldsymbol{r},\boldsymbol{l}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r}) |\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}(\boldsymbol{\xi})|^2 = \sum_{\boldsymbol{r}: \mathcal{L}(\boldsymbol{r}) > r} \lambda_{\mathcal{K}}(\boldsymbol{r}) \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r}) \,, \tag{121}$$

which is deterministic.

D.2 VERIFYING ASSUMPTION 5 AND SPECTRAL GAPS

Recall that $|X| \sim d^r$ and $r_j < r < r_{j+1}$. Assumption 5(a) follows from Eq. (107). Indeed, for $l > 1$

$$(d^{-r_{j+1}})^{-l} \sum_{i:r_i \geq r_{j+1}} \dim(E_i) d^{-lr_i} \sim d^{lr_{j+1}} \sum_{i:r_i \geq r_{j+1}} d^{r_i} d^{-lr_i} = d^{r_{j+1}} > |X|^{1+\epsilon} \tag{122}$$

for some $\epsilon > 0$ since $r < r_{j+1}$. For $l = 1$,

$$(d^{-r_{j+1}})^{-1} \sum_{i:r_i \geq r_{j+1}} \sum_{\boldsymbol{r},\boldsymbol{l}:\mathcal{L}(\boldsymbol{r})=r_i} \lambda_{\mathcal{K}}(\boldsymbol{r}) \sim (d^{-r_{j+1}})^{-1} \cdot 1 = d^{r_{j+1}} \,. \tag{123}$$

Assumption 5(b) holds since $r > r_j$. To verify the spectral gap property, we need for some $\epsilon > 0$,

$$(d^{-r_j})^{-1} \sum_{i:r_i \geq r_{j+1}} \sum_{\boldsymbol{r},\boldsymbol{l}:\mathcal{L}(\boldsymbol{r})=r_i} \lambda_{\mathcal{K}}(\boldsymbol{r}) < |X|^{1-\epsilon} < |X|^{1+\epsilon} < (d^{-r_{j+1}})^{-1} \sum_{i:r_i \geq r_{j+1}} \sum_{\boldsymbol{r},\boldsymbol{l}:\mathcal{L}(\boldsymbol{r})=r_i} \lambda_{\mathcal{K}}(\boldsymbol{r}) \tag{124}$$

i.e.

$$d^{r_j} < |X|^{1-\epsilon} < |X|^{1+\epsilon} < d^{r_{j+1}}$$

which is obvious since $r_j < r < r_{j+1}$.

E PROOF OF LEMMA 4

Proof. The lemma can be proved by induction. **Base case:** $L = 0$**.** The network is a S-CNN. WLOG, assume $\alpha_0 \neq 0$ and $\alpha_1 \neq 0$. For $r \in \mathcal{L}(\mathcal{G}^{(d)})$, we know that r can be written as a combination of α_0 and α_1, i.e. $r = k_0\alpha_0 + k_1\alpha_1$ for some $k_0, k_1 \geq 0$. We say a tuple (k_0, k_1) is r-feasible if in addition, there exists $\boldsymbol{r}$ with $\mathcal{S}(\boldsymbol{r}) = k_1\alpha_1$ and $\mathfrak{F}(\boldsymbol{r}) = k_0\alpha_0$. Consider the set of all r-feasible tuple

$$F(r) \equiv \{(k_0, k_1) : r\text{-feasible}\} \,. \tag{125}$$

Clearly, $F(r)$ is finite. It suffices to prove that for each r-feasible tuple (k_0, k_1),

$$\dim\left(\text{span}\left\{\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} : \mathfrak{F}(\boldsymbol{r}) = k_0\alpha_0 \quad \mathcal{S}(\boldsymbol{r}) = k_1\alpha_1,\ \boldsymbol{l} \in \boldsymbol{N}(\boldsymbol{d}, \boldsymbol{r})\right\}\right) \sim d^r \tag{126}$$

Note that there are $\sim\ (d^{\alpha_1})^{k_1}$ many ways to choose k_1 nodes in the penultimate layer. Then the dimension of the above set is about

$$(d^{\alpha_1})^{k_1} \sum_{\substack{(k_{0,1},\ldots,k_{0,k_1}) \\ k_{0,1}+\cdots+k_{0,k_1}=k_0}} \prod_{j=1}^{k_1} N(d^{\alpha_0}, k_{0,j}) \sim (d^{\alpha_1})^{k_1} \sum_{\substack{(k_{0,1},\ldots,k_{0,k_1}) \\ k_{0,1}+\cdots+k_{0,k_1}=k_0}} \prod_{j=1}^{k_1} (d^{\alpha_0})^{k_{0,j}} \tag{127}$$

$$\sim d^{k_0\alpha_0 + k_1\alpha_1} = d^r \,, \tag{128}$$

since the cardinality of the set

$$\{(k_{0,1}, \ldots, k_{0,k_1}) : k_{0,1} + \cdots + k_{0,k_1} = k_0, \quad k_{0,j} \geq 1 \quad k_{0,j} \in \mathbb{N}\}$$

is finite.

Induction step: $L \geq 1$**.** For $\boldsymbol{r}$ with $\mathcal{L}(\boldsymbol{r}) = r$, let k be the number of children of a MST of $\mathfrak{n}(\boldsymbol{r}; o_{\mathcal{G}})$. Clearly, $k \in [1, [r/\alpha_{L+1}]]$. Then we can classify $\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}$ into at most $[r/\alpha_{L+1}]$ bins: $\{\Omega_k\}_{k=1,\ldots,[r/\alpha_{L+1}]}$ depending on the number of children of $o_{\mathcal{G}}$ in a MST. Let Ω_k be non-empty. We only need to prove the number of $\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}}$ in Ω_k is d^r. Note that there are $\sim (d^{\alpha_{L+1}})^k$ many ways to choose k children from $o_{\mathcal{G}}$. Let $\{u_j\}_{j=1,\ldots,k}$ be one fixed choice and $\{\mathcal{G}_j\}$ be the subgraphs with $\{u_j\}$ as the output nodes. Next, we partition $(r - k\alpha_{L+1})$ into k components,

$$r - k\alpha_{L+1} = r_1 + \cdots + r_k$$

so that each r_j is a combination of $\{\alpha_j\}_{0 \leq j \leq L}$. The cardinality of such partition is also finite. We fix one of such partition $(r_1, \ldots, r_k)$ so that each r_j is a learning index of $\mathcal{G}_j$. We can apply induction to

each $(\mathcal{G}_j, r_j)$ to conclude that the cardinality of $\overline{\boldsymbol{Y}}_{\boldsymbol{r}_j,\boldsymbol{l}_j}$ with $\mathcal{L}_{\mathcal{G}_j}(\boldsymbol{r}_j) = r_j$ is $\sim d^{r_j}$, where $\mathcal{L}_{\mathcal{G}_j}(\boldsymbol{r}_j)$ is the learning index of $\boldsymbol{r}_j$ of $\mathcal{G}_j$. Therefore, we have

$$\dim\left(\text{span}\left\{\overline{\boldsymbol{Y}}_{\boldsymbol{r},\boldsymbol{l}} : \quad \mathcal{L}(\boldsymbol{r}) = r,\ \boldsymbol{l} \in \boldsymbol{N}(\boldsymbol{d},\boldsymbol{r})\right\}\right) \sim (d^{\alpha_{L+1}})^k \prod_{j\in[k]} d^{r_j} = d^r\,. \tag{129}$$

□