# A Rainbow in Deep Network Black Boxes

Florentin Guth
Shared by Du Junye. Supervised under Prof. Feng Long


Department of Statistics & Actuarial Science University of Hong Kong

Feb 25, 2025, Hong Kong

# A Rainbow in Deep Network Black Boxes

DNNs are usually trained by SGD from a random initialization, this randomness is a major challenge in analyzing the learned weight. Can we find some deterministic quantities independent of initialization and training?

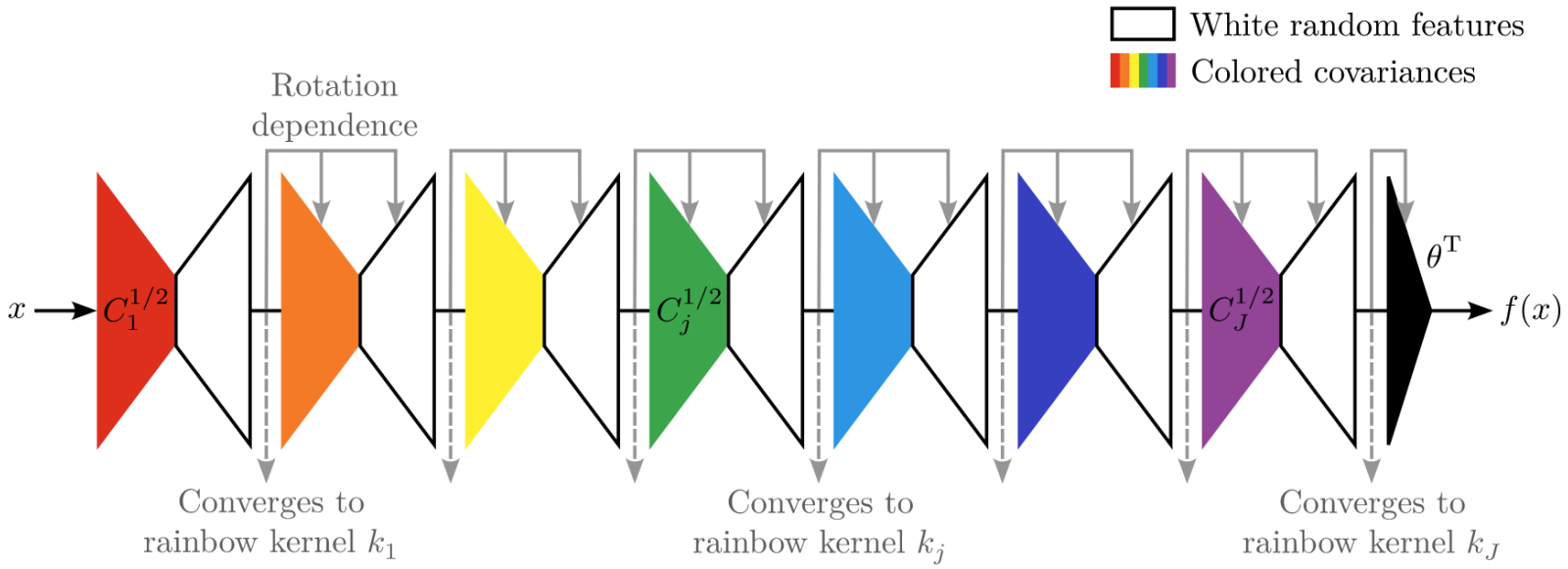


Figure: Rainbow Network Architecture

# OUTLINE

# Preliminaries

# Reproducing Kernel Hilbert Space (RKHS)

1. **Hilbert Space:** A Hilbert space is a complete inner product space:
   - **Inner Product Space:** A vector space with an inner product $\langle \cdot, \cdot \rangle$:
     - Conjugate Symmetry: $\langle f, g \rangle = \overline{\langle g, f \rangle}$
     - Linearity: $\langle af + bg, h \rangle = a\langle f, h \rangle + b\langle g, h \rangle$
     - Positive Definiteness: $\langle f, f \rangle \geqslant 0$, with equality iff $f = 0$
   - **Completeness:** Every Cauchy sequence converges to a point within space.
2. **Reproducing Kernel:** A reproducing kernel is a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$:
   - **Symmetry:** $K(x, y) = K(y, x)$
   - **Positive Definiteness:** For any finite set $\{x_1, x_2, \ldots, x_n\}$, the matrix $[K(x_i, x_j)]$ is positive semi-definite.
   - **Reproducing Property:** For any $f$ in the RKHS, $f(x) = \langle f, K(\cdot, x) \rangle$.
3. **Reproducing Kernel Hilbert Space (RKHS):** An RKHS is a Hilbert space with a reproducing kernel $K$:
   - **Uniqueness:** Each positive semi-definite kernel $K$ corresponds to a unique RKHS. (Theorem 12.11 in HDS)
   - **Density:** Functions in the RKHS can be approximated by linear combinations of the kernel function.

# Corresponding RKHS of linear kernel

**Corresponding RKHS of linear kernel is the space of all linear functions:**

$$\mathcal{H} = \{f : f(\boldsymbol{x}) = \boldsymbol{\omega}^T x, \boldsymbol{\omega} \in \mathbb{R}^d\}$$

- **Definition of the inner product:** In RKHS, the inner product between two functions $f(\boldsymbol{x}) = \boldsymbol{\omega}^T \boldsymbol{x}$ and $g(\boldsymbol{x}) = v^T \boldsymbol{x}$ is defined as:

$$\langle f, g \rangle_{\mathcal{H}} = \boldsymbol{\omega}^T \boldsymbol{v}$$

- **Definition of the kernel:**

$$K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y} = \sum_{i=1}^{d} x_i y_i.$$

- **Properties:**
  - Symmetry: $K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{y}, \boldsymbol{x})$.
  - Positive Definiteness: $[K(\boldsymbol{x}_i, \boldsymbol{x}_j)]$ is positive semi-definite.
  - Reproducing Property: $f(\boldsymbol{x}) = \langle f, K(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}}$.
- **Reproducing Property:** For any function $f(\boldsymbol{x}) = \boldsymbol{\omega}^T \boldsymbol{x}$, we have:

$$\langle f, K(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}} = \langle \boldsymbol{\omega}^T \cdot, \boldsymbol{x}^T \cdot \rangle_{\mathcal{H}} = \boldsymbol{\omega}^T \boldsymbol{x} = f(\boldsymbol{x}).$$

# Kernel PCA Algorithm

**Given a data set $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and a kernel function $\boldsymbol{K}$**

1. Compute $\boldsymbol{K}_{ij} = K(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$ for all $i, j$
2. Compute $\boldsymbol{K}' = (\boldsymbol{I} - \mathbf{1}_N)\boldsymbol{K}(\boldsymbol{I} - \mathbf{1}_N)$ where $1_N$ is an $N \times N$ matrix where every entry is $\mathbf{1}/N$
   - The goal is to zero center data points in the feature space
3. Compute the $K$ leading eigenvectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$ of $\boldsymbol{K}'$ along with their eigenvalues $N\lambda_1, \ldots, N\lambda_K$
4. Compute the $k$-th PC of the projected data vector $\boldsymbol{z} \in \mathbb{R}^{K \times 1}$

$$z_k = \sum_{i=1}^{N} w_{ki} K(\boldsymbol{x}, \boldsymbol{x}^{(i)})$$

# Kernel related work

**For two different wide networks:**
A prominent example is the kernels defined by hidden activations of wide networks. That is, if we denote $\hat{\phi}_j(x)$ and $\hat{\phi}'_j(x)$ the $j$-th layer feature maps of two wide networks,

$$\langle \hat{\phi}_j(x), \hat{\phi}_j(x') \rangle \approx \langle \hat{\phi}'_j(x), \hat{\phi}'_j(x') \rangle, \quad \forall j, x, x'.$$

**Mean-field limit:**
For a one-hidden layer network, when the width goes to infinity, the kernel concentrates as a consequence of law of large numbers:

$$\langle \hat{\phi}_1(x), \hat{\phi}_1(x') \rangle = \frac{1}{d_1} \sum_{i=1}^{d_1} \sigma(\langle w_i, x \rangle) \sigma(\langle w_i, x' \rangle) \xrightarrow{d_1 \to \infty} \mathbb{E}_{w \sim \pi_1} [\sigma(\langle w, x \rangle) \sigma(\langle w, x' \rangle)].$$

# What this paper does?

Introduce the deep extension of the shallow random feature model to explain the kernel convergence in all layers.

Show that the concentration of kernels is equivalent to the concentration of activations up to rotations. Which is:

$$\hat{\phi}_j(x) \approx \hat{A}_j \hat{\phi}'_j(x)$$

# Random feature network

# Random Feature Network

**One-Hidden Layer Network:** A one-hidden layer network computes a hidden activation layer with a matrix $W$ of size $d_1 \times d_0$ and a pointwise non-linearity $\sigma$:

$$\hat{\varphi}(x) = \sigma(Wx) \quad \text{for} \quad x \in \mathbb{R}^{d_0}.$$

**Random Feature Network (Rahimi and Recht, 2007):** The rows of $W$, which contain the weights of different neurons, are independent and have the same probability distribution $\pi$:

$$W = \langle w_i \rangle_{i \leqslant d_1} \quad \text{with} \quad i.i.d. \quad w_i \sim \pi.$$

**Learning in Random Feature Models:** In many random feature models, each row vector has a known distribution with uncorrelated coefficients. Learning is then reduced to calculating the output weights $\hat{\theta}$, which define:

$$\hat{f}(x) = \langle \hat{\theta}, \hat{\varphi}(x) \rangle.$$

**This paper consider general distribution of $\pi$ which is estimated from the weights, thus does not include any bias for simplicity.**

# Kernel Convergence

We now review the convergence properties of one-hidden layer random feature networks. This convergence is captured by the convergence of their kernel:

$$\hat{k}(x, x') = \langle \hat{\varphi}(x), \hat{\varphi}(x') \rangle = \frac{1}{d_1} \sum_{i=1}^{d_1} \sigma(\langle w_i, x \rangle) \sigma(\langle w_i, x' \rangle),$$

where the $d_1^{-1}$ is absurded in the normalization.

Since $w_i$ i.i.d. According to the law of large numbers, as $d_1$ goes to infinity, the empirical kernel converges to the asymptotic kernel:

$$k(x, x') = \mathbb{E}_{w \sim \pi} \left[ \sigma(\langle w, x \rangle) \sigma(\langle w, x' \rangle) \right]$$

# Infinite-dimensional deterministic feature

Let $\varphi(x)$ be an infinite-dimensional deterministic feature vector in a separable Hilbert space $H$, satisfies:

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_H$$

**Such feature vector always exists!**
**Examples**: We can let $\varphi(x) = \sigma(\langle w, x \rangle)_w$, the infinite-width limit of $\sigma W$. In this case, $H = L^2(\pi)$, the space of square-integrable functions with respect to $\pi$ with dot product $\langle f, g \rangle = \mathbb{E}_{w \sim \pi}[f(w)g(w)]$.

**Not unique!**: We can apply a unitary transformation to $\psi$ so that does not modify the dot product. So this paper choose the kernel PCA feature vector aso that the covariance matrix is a diagonal with decreasing values along the diagonal. So that $H = \ell^2(\mathbb{N})$.

# RKHS of the kernal k

Finally, we denote by $\mathcal{H}$ the reproducing kernel Hilbert space (RKHS) associated to the kernel $k$. It is the space of functions $f$ which can be written $f(x) = \langle \theta, \varphi(x) \rangle_H$, with norm $\|f\|_{\mathcal{H}} = \|\theta\|_H$. A random feature network defines approximations of functions in this RKHS. With $H = L^2(\pi)$, these functions can be written as:

$$f(x) = \mathbb{E}_{w \sim \pi}[\theta(w)\, \sigma(\langle w, x \rangle)] = \int \theta(w)\, \sigma(\langle w, x \rangle)\, d\pi(w).$$

**Note:** The $\theta$ is the minimum norm vector such that $f(x) = \langle \theta, \varphi(x) \rangle_H$.

# Rotational alignment

An informal derivation of the paper is that the empirical kernel could converge to the asymptotic kernel. We thus expect that for large widths there exists a rotation $\hat{A}$ so that $\hat{A}\hat{\varphi} \approx \varphi$. The network activations $\hat{\varphi}(x) \approx \hat{A}^T \varphi(x)$ are therefore random rotation of the deterministic feature vectors. **Note:** Here the rotation represents the orthogonal transformation, including rotation and reflection.

Unber this settings, for any function $f(x) = \langle \theta, \varphi(x) \rangle_H$ in $\mathcal{H}$, if $\hat{\theta} = \hat{A}^T \theta$, then we have:

$$\hat{f}(x) = \langle \hat{A}^T \theta, \hat{\varphi}(x) \rangle_H = \langle \theta, \hat{A}\hat{\varphi}(x) \rangle_H \approx f(x)$$

The final layer coefficient $\hat{\theta}$ could cancel the random rotation!

# Rotational alignment

We write $\mathcal{O}(d_1)$ the set of linear operators $A$ from $\mathbb{R}^{d_1}$ to $H = \ell^2(\mathbb{N})$ which satisfy $\hat{A}^{\mathrm{T}}\hat{A} = \mathrm{Id}_{d_1}$. Each $\hat{A} \in \mathcal{O}(d_1)$ computes an isometric embedding of $\mathbb{R}^{d_1}$ into $H$, while $\hat{A}^{\mathrm{T}}$ is an orthogonal projection onto a $d_1$-dimensional subspace of $H$ which can be identified with $\mathbb{R}^{d_1}$. The alignment $\hat{A}$ of $\hat{\varphi}$ to $\varphi$ is defined as the minimizer of the mean squared error:

$$\hat{A} = \underset{\hat{A} \in \mathcal{O}(d_1)}{\arg\min}\ \mathbb{E}_x \left[ \| \hat{A}\hat{\varphi}(x) - \varphi(x) \|_H^2 \right]$$

This optimization problem has a closed form solution through SVD:

$$\hat{A} = UV^{\mathrm{T}} \text{ with } \mathbb{E}_x \left[ \varphi(x)\hat{\varphi}(x)^{\mathrm{T}} \right] = USV^{\mathrm{T}}.$$

# Theorem 1 for random feature network

**Theorem 1** *Assume that* $\mathbb{E}_x[\|x\|^2] < +\infty$, $\sigma$ *is Lipschitz continuous, and* $\pi$ *has finite fourth order moments. Then there exists a constant* $c > 0$ *which does not depend on* $d_0$ *nor* $d_1$ *such that*

$$\mathbb{E}_{W,x,x'}\left[|\hat{k}(x,x') - k(x,x')|^2\right] \leq c\, d_1^{-1},$$

*where* $x'$ *is an i.i.d. copy of* $x$. *Suppose that the sorted eigenvalues* $\lambda_1 \geq \cdots \geq \lambda_m \geq \cdots$ *of* $\mathbb{E}_x[\varphi(x)\,\varphi(x)^{\mathrm{T}}]$ *satisfy* $\lambda_m = O(m^{-\alpha})$ *with* $\alpha > 1$. *Then the alignment* $\hat{A}$ *defined in (5) satisfies*

$$\mathbb{E}_{W,x}\left[\|\hat{A}\,\hat{\varphi}(x) - \varphi(x)\|_H^2\right] \leq c\, d_1^{-\eta} \quad \textit{with} \quad \eta = \frac{\alpha - 1}{2(2\alpha - 1)} > 0.$$

*Finally, for any* $f(x) = \langle \theta, \varphi(x) \rangle_H$ *in* $\mathcal{H}$, *if* $\hat{\theta} = \hat{A}^{\mathrm{T}}\theta$ *then*

$$\mathbb{E}_{W,x}\left[|\hat{f}(x) - f(x)|^2\right] \leq c\, \|f\|_{\mathcal{H}}^2\, d_1^{-\eta}.$$

**Note:** Theorem 1 proves that there exists a rotation $\hat{A}$ so that could nearly aligns the hidden layer of a random feature network with any feature vector of the asymptotic kernel.

# Deep rainbow networks

# Infinite-width rainbow networks

In this part the paper consider the case of a deep fully-connected neural network with $J$ hidden layers, which iteratively transform the input data $x \in \mathbb{R}^{d_0}$ with weight matrices $W_j$ of size $d_j \times d_{j-1}$ and a pointwise non-linearity $\sigma$, to compute each activation layer of depth j:

$$\hat{\phi}_j(x) = \sigma W_j \cdots \sigma W_1 x$$

where $\sigma$ includes a division by $\sqrt{d_j}$. After $J$ non-linearities, the last layer outputs:

$$\hat{f}(x) = \langle \hat{\theta}, \hat{\phi}_J(x) \rangle$$

# Infinite-width rainbow networks

**Definition 1** *An infinite-width rainbow network has activation layers defined in a separable Hilbert space $H_j$ for any $j \leq J$ by*

$$\phi_j(x) = \varphi_j(\varphi_{j-1}(\dots \varphi_1(x)\dots)) \in H_j \quad \text{for} \quad x \in H_0 = \mathbb{R}^{d_0},$$

*where each $\varphi_j \colon H_{j-1} \to H_j$ is defined from a probability distribution $\pi_j$ on $H_{j-1}$ by*

$$\langle \varphi_j(z), \varphi_j(z') \rangle_{H_j} = \mathbb{E}_{w \sim \pi_j}\left[\sigma(\langle w, z\rangle_{H_{j-1}})\, \sigma(\langle w, z'\rangle_{H_{j-1}})\right] \quad \text{for} \quad z, z' \in H_{j-1}. \tag{8}$$

*It defines a rainbow kernel*

$$k_j(x, x') = \langle \phi_j(x), \phi_j(x') \rangle_{H_j}.$$

*For $\theta \in H_J$, the infinite-width rainbow network outputs*

$$f(x) = \langle \theta, \phi_J(x) \rangle_{H_J} \in \mathcal{H}_J,$$

*where $\mathcal{H}_J$ is the RKHS of the rainbow kernel $k_J$ of the last layer. If all probability distributions $\pi_j$ are Gaussian, then the rainbow network is said to be Gaussian.*

**Note:** Here the $\varphi_j$ is defined from the probaility distribution of $\pi_j$ on $H_{j-1}$, for example, it could just be the inifinite width limit of $\sigma W_j$ up to rotations.

# Infinite-width rainbow networks

Basing on above, we could arbitrarily rotate the feature vector $\varphi_j(z)$ so that it satisfy the above kernel equation:

$$\langle \varphi_j(z), \varphi_j(z') \rangle_{H_j} = \mathbb{E}_{w \sim \pi_j} \left[ \sigma(\langle w, z \rangle_{H_{j-1}}) \sigma(\langle w, z' \rangle_{H_{j-1}}) \right] \text{ for } z, z' \in H_{j-1}$$

This operation will also rotate the Hilbert space $H_j$ and $\phi_j(x)$. If the distribution $\pi_{j+1}$ at the next layter or the weight vector $\theta$ in the last layer is similarly rotated, the operation will preserve the dot products and therefore not affect the asymptotic rainbow kernels at each depth j and the ouput $f(x)$.

$$k_j(x, x') = \mathbb{E}_{w \sim \pi_j} \left[ \sigma(\langle w, \phi_{j-1}(x) \rangle_{H_{j-1}}) \sigma(\langle w, \phi_{j-1}(x') \rangle_{H_{j-1}}) \right]$$

**Note:** Similar to the random feature network, the paper fix the rotations by choosing KPCA feature vectors. So that $H_j = \ell^2(\mathbb{N})$ and the covariance matrix $\mathbb{E}[\phi_j(x)\phi_j(x)^T]$ is diagonal.

# Note of infinite dimension

Since for $j \geqslant 2$ the weight distribution $\pi_j$ is defined on the infinite space $H_j$, we need to be careful with the definition.

- $\pi$ has finite second moment iff $\mathbb{E}_{w\sim\pi}[ww^T]$ is bounded.
- $\phi$ has bounded fourth moment iff for every trace-class operator $T$, $\mathbb{E}_{w\sim\pi}[(w^T Tw)^2] \leqslant \infty$
- This paper generalize rainbow networks to cylindrical measures $\pi_j$, which define cylindrical random variable $w$.
- These cylindrical r.v $w$ are linear maps such that $w(z)$ is real r.v. The paper replace $w(z)$ by $\langle w, z\rangle$.

# Dimensionality reduction

For the rainbow network, the uncentered covariance matrix of the weight $C_j = \mathbb{E}_{w \sim \pi_j}[ww^T]$ could capture the linear dimensionality reductions of the networks.

Let $C_j^{\frac{1}{2}}$ be the symmetric square root of $C_j$, we can rewrite the kernel like:

$$\varphi_j(z) = \tilde{\varphi}_j\left(C_j^{1/2} z\right) \text{ with } \langle \tilde{\varphi}_j(z), \tilde{\varphi}_j(z') \rangle_{H_j} = \mathbb{E}_{w \sim \tilde{\pi}_j}\left[\sigma(\langle w, z \rangle)\sigma(\langle w, z' \rangle)\right]$$

where $\tilde{\pi}_j$ has an identity covariance. Rainbow network activations can thus be written

$$\phi_j(x) = \tilde{\varphi}_j\left(C_j^{1/2} \cdots \tilde{\varphi}_1\left(C_1^{1/2} x\right)\right).$$

**Note:** Empirical observations of trained deep networks show that they have approximately low-rank weight matrices. Each square root $C_j^{\frac{1}{2}}$ performs linear dimensionality reduction of its inputs.

# Gaussian Rainbow Network(Special Case)

In this part we consider the case when $\pi_j = \mathcal{N}(0, C_j)$. If $\sigma$ is a homogeneous non-linearity such as ReLU, the Gaussian rainbow network kernel can be written from a homogeneous dot-product:

$$k_j(x, x') = \|z_j(x)\|\|z_j(x')\|\kappa\left(\frac{\langle z_j(x), z_j(x')\rangle}{\|z_j(x)\|\|z_j(x')\|}\right) \text{ with } z_j(x) = C_j^{1/2}\phi_{j-1}(x)$$

where $\kappa$ is a scalar function depends on $\sigma$.
**Note:** This is due to that by using the $C_j$ matrix to do normalizationm, we could transform the kernel into the dot product.
Thus, we could get:

$$\langle\tilde{\varphi}_j(z), \tilde{\varphi}_j(z')\rangle_{H_j} = \|z\|\|z'\|\kappa\left(\frac{\langle z, z'\rangle}{\|z\|\|z'\|}\right)$$

# Finite-width rainbow networks

In this part we consider the general case of arbitrarily weight distribution $\pi_j$. We want to show that $\hat{A}_j\hat{\phi}_j \approx \phi_j$ where $\hat{A} : \mathbb{R}^{d_j} \to H_j$ is an alignment rotation.

Suppose that $W_1, \ldots, W_{j-1}$ have been defined. By induction, there exists an alignment rotation $\hat{A}_{j-1} : \mathbb{R}^{d_{j-1}} \to H_{j-1}$, defined by

$$\hat{A}_{j-1} = \underset{\hat{A} \in \mathcal{O}(d_{j-1})}{\arg\min} \ \mathbb{E}_x \left[ \|\hat{A}\hat{\phi}_{j-1}(x) - \phi_{j-1}(x)\|^2_{H_{j-1}} \right],$$

such that $\hat{A}_{j-1}\hat{\phi}_{j-1}(x) \approx \phi_{j-1}(x)$.

We wish to define $W_j$ so that $\hat{A}_j\hat{\phi}_j(x) \approx \phi_j(x)$. This can be achieved with a random feature approximation of $\varphi_j$ composed with alignment $\hat{A}_{j-1}$.

# Finite-width rainbow networks

Consider a (semi-infinite) random matrix $W_j'$ of $d_j$ i.i.d. rows in $H_{j-1}$ distributed according to $\pi_j$:

$$W_j' = (w_{ji}')_{i \leqslant d_j} \text{ with i.i.d. } w_{ji}' \sim \pi_j.$$

We then have $\hat{A}_j \sigma(W_j' x) \approx \varphi_j(x)$ for a suitably defined $\hat{A}_j$. Combining the two approximations, we obtain

$$\hat{A}_j \sigma\left(W_j' \hat{A}_{j-1} \hat{\phi}_{j-1}(x)\right) \approx \varphi_j(\phi_{j-1}(x)) = \phi_j(x).$$

We thus define the weight at layer $j$ with the aligned random features

$$W_j = W_j' \hat{A}_{j-1}$$

**Note:** semi-infinite means $d \times \infty$

# Finite-width apprximation

**Definition 2** *A finite-width rainbow network approximation of an infinite-width rainbow network with weight distributions $(\pi_j)_{j\le J}$ is defined for each $j \le J$ by a random weight matrix $W_j$ of size $d_j \times d_{j-1}$ which satisfies*

$$W_j = (\hat{A}_{j-1}^{\mathrm{T}} w'_{ji})_{i\le d_j} \quad \textit{with i.i.d.} \quad w'_{ji} \sim \pi_j, \tag{13}$$

*where $\hat{A}_{j-1}$ is the rotation defined in (12). The last layer weight vector is $\hat{\theta} = \hat{A}_J^{\mathrm{T}}\theta$ where $\theta$ is the last layer weight of the infinite-width rainbow network.*

**Note:** The random weights of $W_j$ of a finite network are defined as rotations and finite-dimensional projections of $d_j$ infinite-dimensional random vectors $w'_{ji}$

# Rotation and projection

The dependence of the of $W_j$ on the previous layers is captured by the rotation $\hat{A}_{j-1}$. The rows in $W_j$ are not independent, but are independent when conditioned on the $(W_\ell)_{\ell<j}$. As such, the conditional covariance of $W_j$ is given by:

$$\hat{C}_j = \hat{A}_{j-1}^{\mathrm{T}} C_j \hat{A}_{j-1}$$

$W_j$ can then be factorized as the product of a white random random feature matrix $\tilde{W}_j$ with the covariance square root:

$$W_j = \tilde{W}_j \hat{C}_j^{1/2} \text{ with i.i.d. } \tilde{w}_{ji} \text{ conditionally on } (W_\ell)_{\ell<j}$$

**Example:** For the Gaussian case, $\hat{W}_j$ is a white Gaussian matrix with i.i.d. normal entries independent of the previous layers.

# Convergence to infinite-width networks

**Theorem 2** *Assume that* $\mathbb{E}_x[\|x\|^2] < +\infty$ *and* $\sigma$ *is Lipschitz continuous. Let* $(\phi_j)_{j \leq J}$ *be the activation layer of an infinite-width rainbow network with distributions* $(\pi_j)_{j \leq J}$ *with bounded second- and fourth-order moments, and an output* $f(x)$*. Let* $(\hat{\phi}_j)_{j \leq J}$ *be the activation layers of sizes* $(d_j)_{j \leq J}$ *of a finite-width rainbow network approximation, with an output* $\hat{f}(x)$*. Let* $k_j(x, x') = \langle \phi_j(x), \phi_j(x') \rangle$ *and* $\hat{k}_j(x, x') = \langle \hat{\phi}_j(x), \hat{\phi}_j(x') \rangle$*. Suppose that the sorted eigenvalues of* $\mathbb{E}_x[\phi_j(x)\,\phi_j(x)^{\mathrm{T}}]$ *satisfy* $\lambda_{j,m} = O(m^{-\alpha_j})$ *with* $\alpha_j > 1$*. Then there exists* $c > 0$ *which does not depend upon* $(d_j)_{j \leq J}$ *such that*

$$\mathbb{E}_{W_1,\dots,W_j,x,x'}\left[|\hat{k}_j(x,x') - k_j(x,x')|^2\right] \leq c\left(\varepsilon_{j-1} + d_j^{-1/2}\right)^2$$

$$\mathbb{E}_{W_1,\dots,W_j,x}\left[\|\hat{A}_j\,\hat{\phi}_j(x) - \phi_j(x)\|_{H_j}^2\right] \leq c\,\varepsilon_j^2$$

$$\mathbb{E}_{W_1,\dots,W_J,x}\left[|\hat{f}(x) - f(x)|^2\right] \leq c\,\|f\|_{\mathcal{H}_J}^2\,\varepsilon_J^2,$$

*where*

$$\varepsilon_j = \sum_{\ell=1}^{j} d_\ell^{-\eta_\ell/2} \quad \text{with} \quad \eta_\ell = \frac{\alpha_\ell - 1}{2(2\alpha_\ell - 1)} > 0.$$

# Notes of Theorem 2

- At each layer, a finite-width rainbow network has a kernel which converges in mean-square to the deterministic kernel $k_j$
- After alignment, each activation layer $\hat{\phi}_j$ also converges to the infinite $\phi_j$
- The finite rainbow output $\hat{f}$ converges to the function $f$ in the RKHS $\mathcal{H}_J$.

# Simulation Study

- Convergence of Activations
- Properties of Learned Weight Covariances
- Gaussian Rainbow Approximations

# Thanks! Questions?