Power-Law Covariance

A simplifying structure for data and representations

Princeton 2026 Summer School in Machine Learning

Misha Gromov

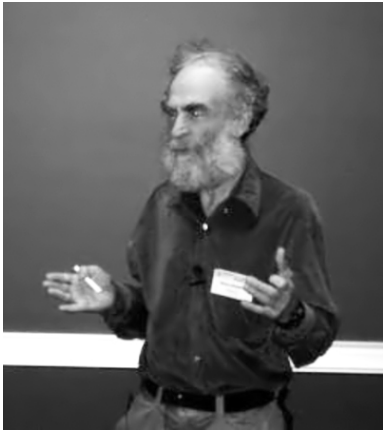


Photo: Gert-Martin Greuel / RedAndr, CC BY-SA 2.0 DE.

A theory of every neural network says almost nothing

With enough parameters, neural networks can represent an enormous range of input–output relationships.

A theory of every neural network says almost nothing

With enough parameters, neural networks can represent an enormous range of input–output relationships.

The data, target, architecture, and optimizer can all change the training story.

A theory of every neural network says almost nothing

With enough parameters, neural networks can represent an enormous range of input–output relationships.

The data, target, architecture, and optimizer can all change the training story.

A useful theory begins by choosing a structure that constrains the problem.

A theory of every neural network says almost nothing

With enough parameters, neural networks can represent an enormous range of input–output relationships.

The data, target, architecture, and optimizer can all change the training story.

A useful theory begins by choosing a structure that constrains the problem.

Here the structure is the **covariance eigenspectrum**.

Covariance and principal directions

For a random vector $X \in \mathbb{R}^d$,

$$\Sigma = \mathrm{Cov}(X) = \mathbb{E}[(X - \mu) \otimes (X - \mu)].$$

Covariance and principal directions

For a random vector $X \in \mathbb{R}^d$,

$$\Sigma = \mathrm{Cov}(X) = \mathbb{E}[(X - \mu) \otimes (X - \mu)].$$

For centered data,

$$\Sigma = \mathbb{E}[X \otimes X].$$

Covariance and principal directions

For a random vector $X \in \mathbb{R}^d$,

$$\Sigma = \mathrm{Cov}(X) = \mathbb{E}[(X - \mu) \otimes (X - \mu)].$$

For centered data,

$$\Sigma = \mathbb{E}[X \otimes X].$$

If u is a unit vector,

$$u^\top \Sigma u = \mathrm{Var}\langle u, X \rangle.$$

- eigenvectors = principal directions;
- eigenvalues = variance in those directions.

Covariance and principal directions

For a random vector $X \in \mathbb{R}^d$,

$$\Sigma = \mathrm{Cov}(X) = \mathbb{E}[(X - \mu) \otimes (X - \mu)].$$

For centered data,

$$\Sigma = \mathbb{E}[X \otimes X].$$

If u is a unit vector,

$$u^\top \Sigma u = \mathrm{Var}\langle u, X \rangle.$$

- eigenvectors = principal directions;
- eigenvalues = variance in those directions.

Covariance is a **linear** statistic, but in high dimension it can still carry rich geometry. Later we will ask whether nonlinear network transformations weakly preserve that geometry.

A power law is a soft notion of dimension

$$\lambda_j(\Sigma) \asymp j^{-\alpha} \qquad \Longleftrightarrow \qquad \log \lambda_j \approx C - \alpha \log j.$$

A power law is a soft notion of dimension

$$\lambda_j(\Sigma) \asymp j^{-\alpha} \qquad \Longleftrightarrow \qquad \log \lambda_j \approx C - \alpha \log j.$$

There is no sharp rank cutoff. Instead, count the directions visible above a resolution ε:

$$N(\varepsilon) = \#\{j : \lambda_j \geq \varepsilon\} \asymp \varepsilon^{-1/\alpha}.$$

A power law is a soft notion of dimension

$$\lambda_j(\Sigma) \asymp j^{-\alpha} \qquad \Longleftrightarrow \qquad \log \lambda_j \approx C - \alpha \log j.$$

There is no sharp rank cutoff. Instead, count the directions visible above a resolution ε:

$$N(\varepsilon) = \#\{j : \lambda_j \geq \varepsilon\} \asymp \varepsilon^{-1/\alpha}.$$

Larger α

Faster decay; fewer visible directions.

Smaller α

Heavier tail; more relevant directions.

A power law is a soft notion of dimension

$$\lambda_j(\Sigma) \asymp j^{-\alpha} \qquad \Longleftrightarrow \qquad \log \lambda_j \approx C - \alpha \log j.$$

There is no sharp rank cutoff. Instead, count the directions visible above a resolution ε:

$$N(\varepsilon) = \#\{j : \lambda_j \geq \varepsilon\} \asymp \varepsilon^{-1/\alpha}.$$

Larger α

Faster decay; fewer visible directions.

Smaller α

Heavier tail; more relevant directions.

Example: Matérn kernels connect α to smoothness and intrinsic dimension.

Image spectra survive random layers

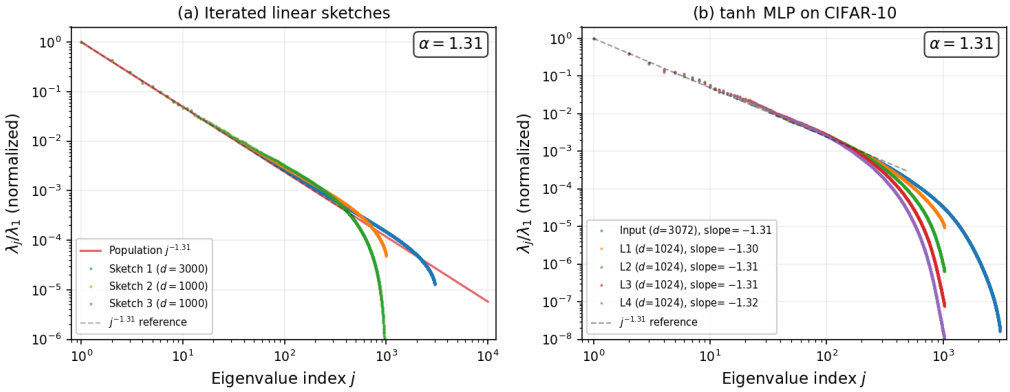
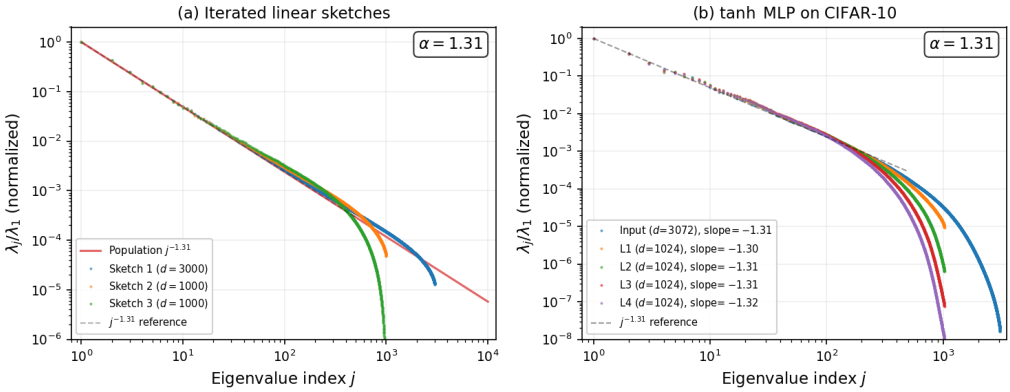


Image spectra survive random layers



The finite width changes the spectral cutoff much more than the fitted slope.

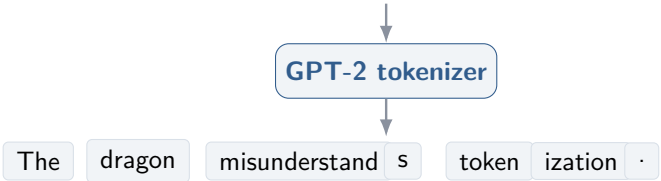
One sentence, three random vectors

The dragon misunderstands tokenization.

Start with a sentence.

One sentence, three random vectors

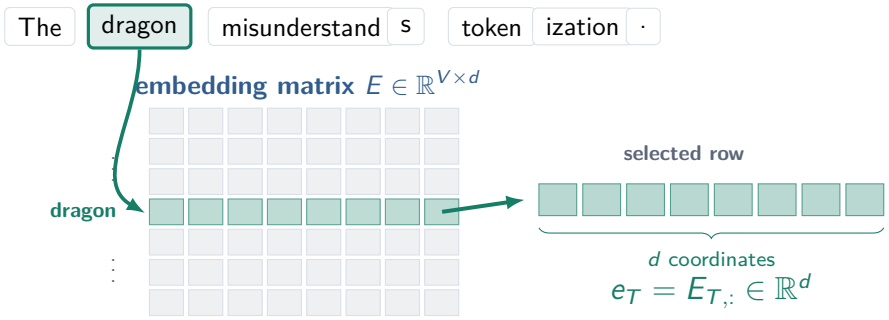
The dragon misunderstands tokenization.



seven GPT-2 BPE tokens

Tokenization chooses the discrete symbols the model will see.

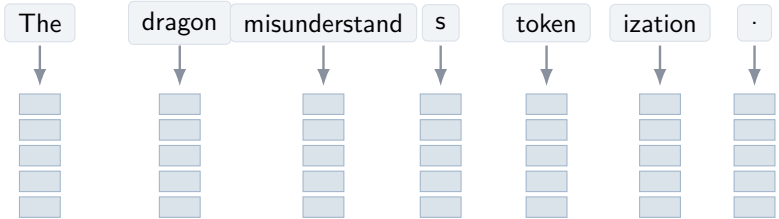
One sentence, three random vectors



For $T =$ dragon, take one row of E: this is e_T.

One sentence, three random vectors

A context is a sequence of token embeddings

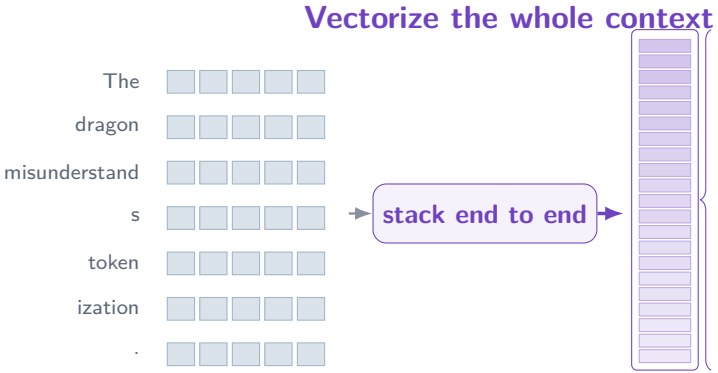


$e_{T_1}, e_{T_2}, \ldots, e_{T_L} \in \mathbb{R}^d$

one d-dimensional vector for every token

Keeping the vectors separate remembers which embedding belongs to which token.

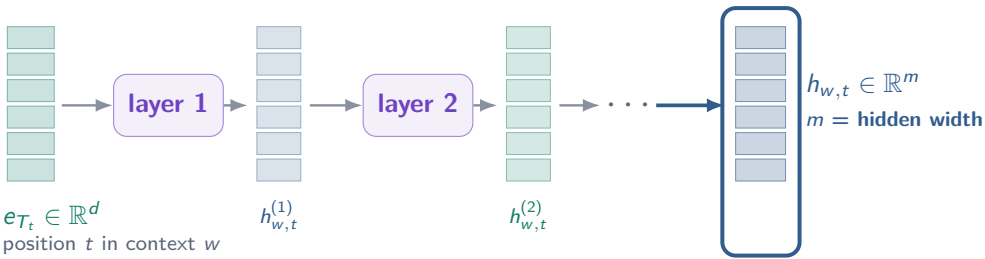
One sentence, three random vectors



Stacking the window produces a second random vector, C_L.

One sentence, three random vectors

The network creates new random vectors



After each transformation, the activation $h_{w,t}$ is another random vector.

Zipf frequencies determine token covariance

For a one-hot token u_T, with frequency vector p,

$$\mathrm{Cov}(u_T) = \mathrm{diag}(p) - pp^\top.$$

Zipf frequencies determine token covariance

For a one-hot token u_T, with frequency vector p,

$$\mathrm{Cov}(u_T) = \mathrm{diag}(p) - pp^\top.$$

If $p_{(1)} \geq p_{(2)} \geq \cdots$, rank-one interlacing gives

$$p_{(j)} \geq \lambda_j(\mathrm{Cov}(u_T)) \geq p_{(j+1)}.$$

Zipf frequencies determine token covariance

For a one-hot token u_T, with frequency vector p,

$$\mathrm{Cov}(u_T) = \mathrm{diag}(p) - pp^\top.$$

If $p_{(1)} \geq p_{(2)} \geq \cdots$, rank-one interlacing gives

$$p_{(j)} \geq \lambda_j(\mathrm{Cov}(u_T)) \geq p_{(j+1)}.$$

For an embedding matrix E,

$$\mathrm{Cov}(e_T) = E^\top(\mathrm{diag}(p) - pp^\top)E.$$

Zipf frequencies determine token covariance

For a one-hot token u_T, with frequency vector p,

$$\mathrm{Cov}(u_T) = \mathrm{diag}(p) - pp^\top.$$

If $p_{(1)} \geq p_{(2)} \geq \cdots$, rank-one interlacing gives

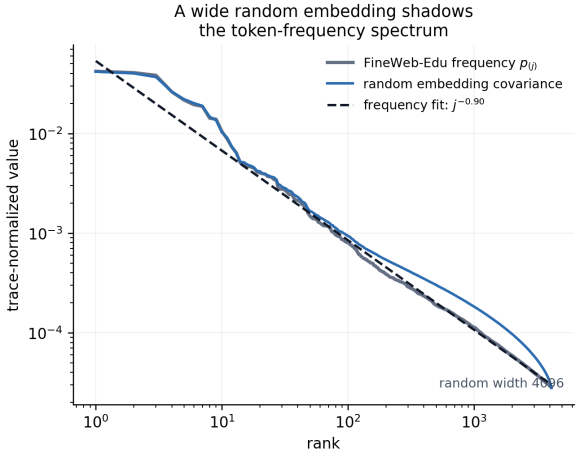
$$p_{(j)} \geq \lambda_j(\mathrm{Cov}(u_T)) \geq p_{(j+1)}.$$

For an embedding matrix E,

$$\mathrm{Cov}(e_T) = E^\top(\mathrm{diag}(p) - pp^\top)E.$$

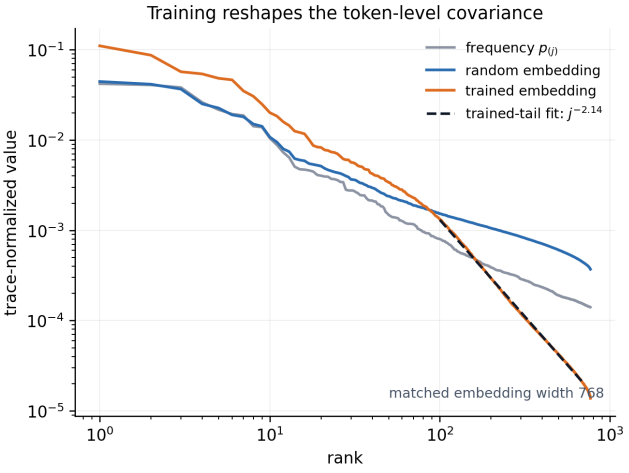
A wide random embedding sketches this covariance.

At initialization, frequency geometry survives the embedding



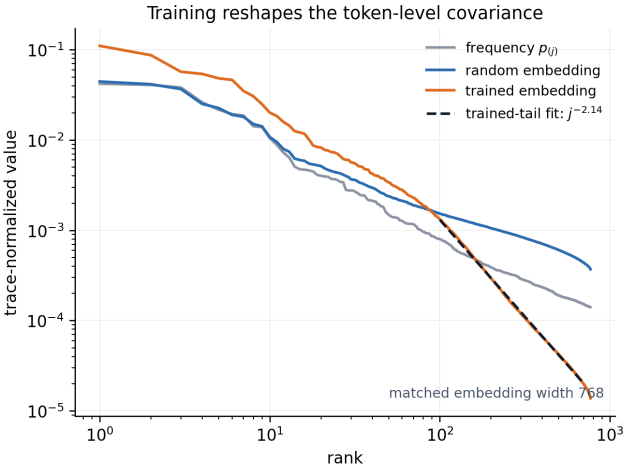
Held-out FineWeb-Edu tokens.

Training introduces a second spectral regime



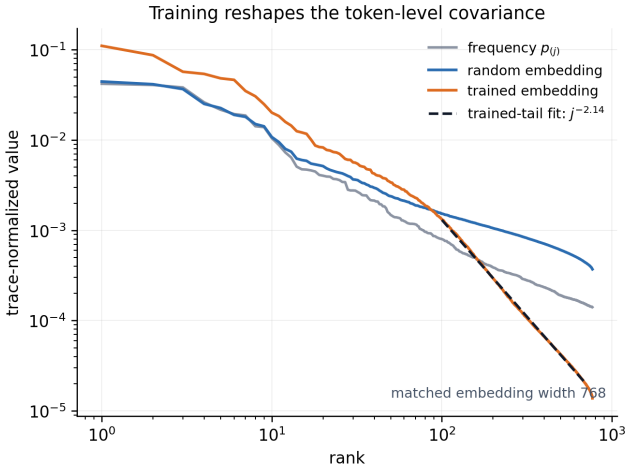
- The trained head is reshaped by semantic overlap between token embeddings.

Training introduces a second spectral regime



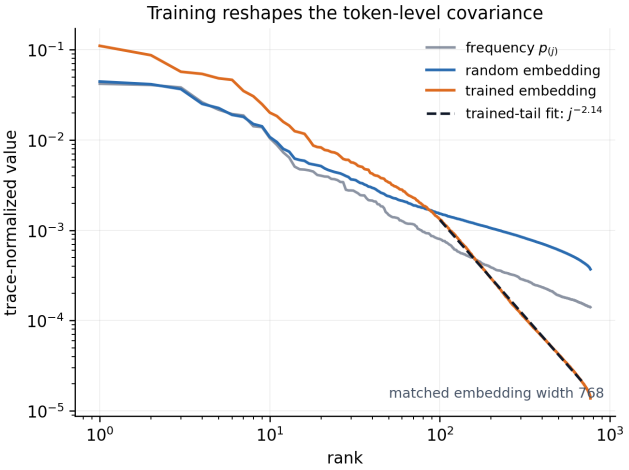
- The trained head is reshaped by semantic overlap between token embeddings.
- Redundant directions produce a sharper terminal cutoff.

Training introduces a second spectral regime



- The trained head is reshaped by semantic overlap between token embeddings.
- Redundant directions produce a sharper terminal cutoff.
- The crossover is a genuine **two-power-law** picture, not one global slope.

Training introduces a second spectral regime



- The trained head is reshaped by semantic overlap between token embeddings.
- Redundant directions produce a sharper terminal cutoff.
- The crossover is a genuine **two-power-law** picture, not one global slope.

Piecewise power laws give a tractable refinement of a single-exponent model.

Context covariance exposes token dependence

For a length-L window, let

$$C_L = \mathrm{vec}(e_{T_1} - \mu_1, \ldots, e_{T_L} - \mu_L).$$

Context covariance exposes token dependence

For a length-L window, let

$$C_L = \mathrm{vec}(e_{T_1} - \mu_1, \ldots, e_{T_L} - \mu_L).$$

Then

$$\Sigma_{\text{context}} = \begin{bmatrix} \Gamma_{11} & \cdots & \Gamma_{1L} \\ \vdots & \ddots & \vdots \\ \Gamma_{L1} & \cdots & \Gamma_{LL} \end{bmatrix}, \qquad \Gamma_{ab} = \mathbb{E}[(e_{T_a} - \mu_a) \otimes (e_{T_b} - \mu_b)].$$

Context covariance exposes token dependence

For a length-L window, let

$$C_L = \mathrm{vec}(e_{T_1} - \mu_1, \ldots, e_{T_L} - \mu_L).$$

Then

$$\Sigma_{\text{context}} = \begin{bmatrix} \Gamma_{11} & \cdots & \Gamma_{1L} \\ \vdots & \ddots & \vdots \\ \Gamma_{L1} & \cdots & \Gamma_{LL} \end{bmatrix}, \qquad \Gamma_{ab} = \mathbb{E}[(e_{T_a} - \mu_a) \otimes (e_{T_b} - \mu_b)].$$

- Γ_{aa}: token covariance at position a;

Context covariance exposes token dependence

For a length-L window, let

$$C_L = \mathrm{vec}(e_{T_1} - \mu_1, \ldots, e_{T_L} - \mu_L).$$

Then

$$\Sigma_{\text{context}} = \begin{bmatrix} \Gamma_{11} & \cdots & \Gamma_{1L} \\ \vdots & \ddots & \vdots \\ \Gamma_{L1} & \cdots & \Gamma_{LL} \end{bmatrix}, \qquad \Gamma_{ab} = \mathbb{E}[(e_{T_a} - \mu_a) \otimes (e_{T_b} - \mu_b)].$$

- Γ_{aa}: token covariance at position a;
- Γ_{ab}, $a \neq b$: cross-position dependence;

Context covariance exposes token dependence

For a length-L window, let

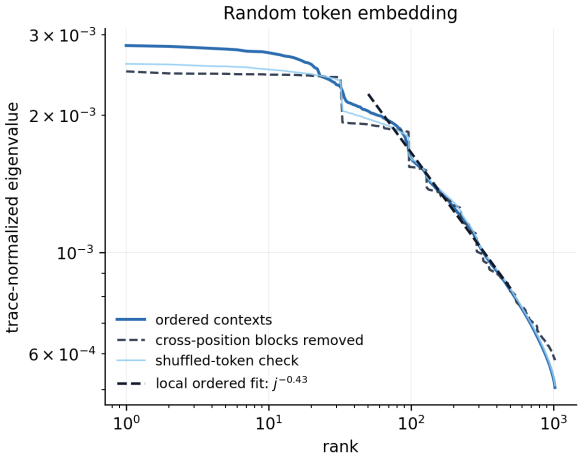
$$C_L = \mathrm{vec}(e_{T_1} - \mu_1, \ldots, e_{T_L} - \mu_L).$$

Then

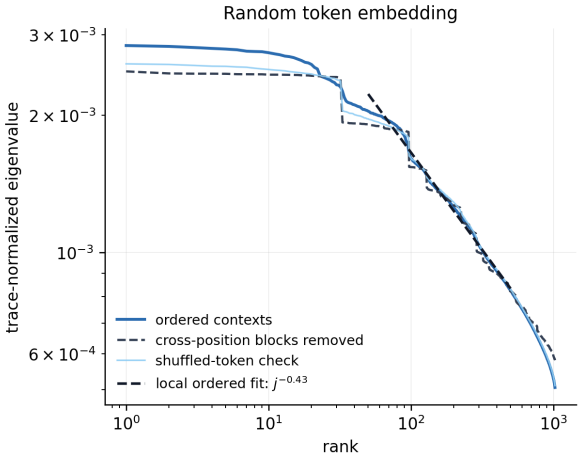
$$\Sigma_{\text{context}} = \begin{bmatrix} \Gamma_{11} & \cdots & \Gamma_{1L} \\ \vdots & \ddots & \vdots \\ \Gamma_{L1} & \cdots & \Gamma_{LL} \end{bmatrix}, \qquad \Gamma_{ab} = \mathbb{E}[(e_{T_a} - \mu_a) \otimes (e_{T_b} - \mu_b)].$$

- Γ_{aa}: token covariance at position a;
- Γ_{ab}, $a \neq b$: cross-position dependence;
- independent tokens would make every off-diagonal block vanish.

Context spectra begin with a flatter phase

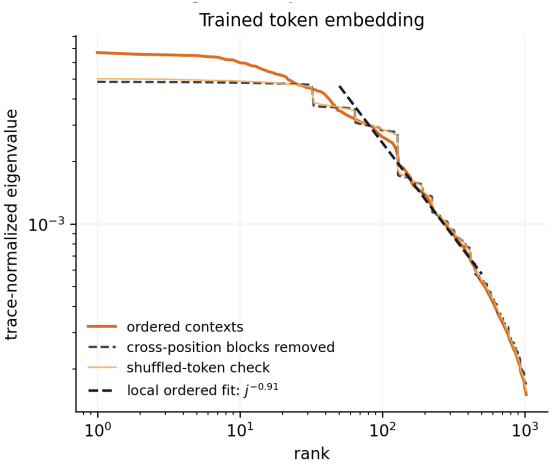


Context spectra begin with a flatter phase

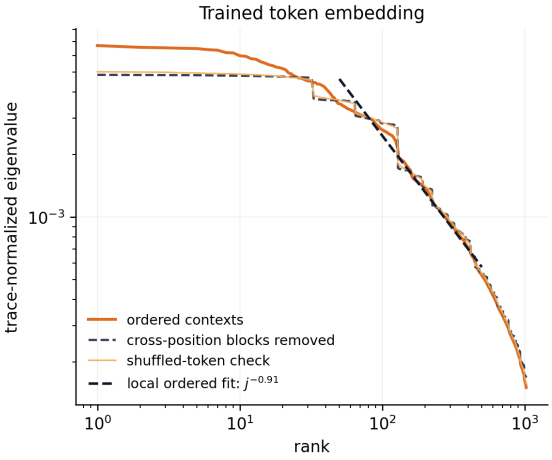


The initial flat regime gives way to a distinctly different spectral tail.

Training changes the context-spectrum crossover



Training changes the context-spectrum crossover



The trained representation again shows more than one spectral regime.

Caveat: eigenvalues do not determine the representation

For any orthogonal matrix Q,

$$Q\Sigma Q^\top$$

has exactly the same eigenvalues as Σ.

Caveat: eigenvalues do not determine the representation

For any orthogonal matrix Q,

$$Q\Sigma Q^\top$$

has exactly the same eigenvalues as Σ.

A permutation matrix is one special case.

Caveat: eigenvalues do not determine the representation

For any orthogonal matrix Q,

$$Q\Sigma Q^\top$$

has exactly the same eigenvalues as Σ.

A permutation matrix is one special case.

The basis is essential

Eigenvectors identify which combinations of features carry the variance.

Caveat: eigenvalues do not determine the representation

For any orthogonal matrix Q,

$$Q\Sigma Q^\top$$

has exactly the same eigenvalues as Σ.

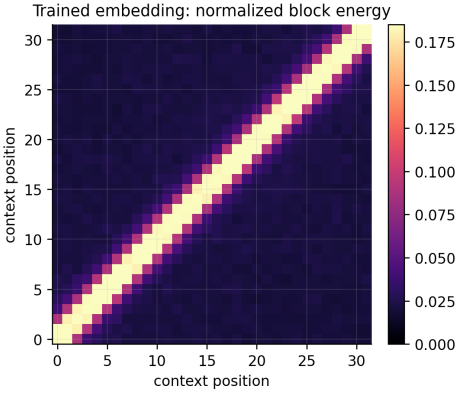
A permutation matrix is one special case.

The basis is essential

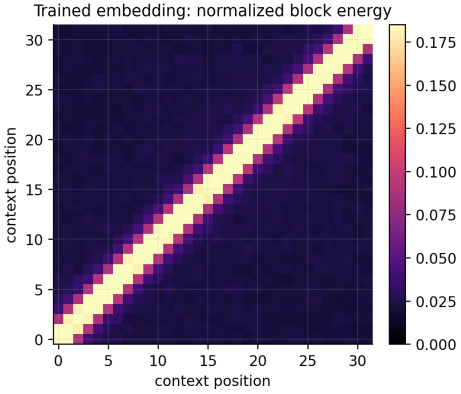
Eigenvectors identify which combinations of features carry the variance.

Permuting token positions can completely rearrange the block geometry while leaving every eigenvalue unchanged.

The spectrum does not show this positional structure

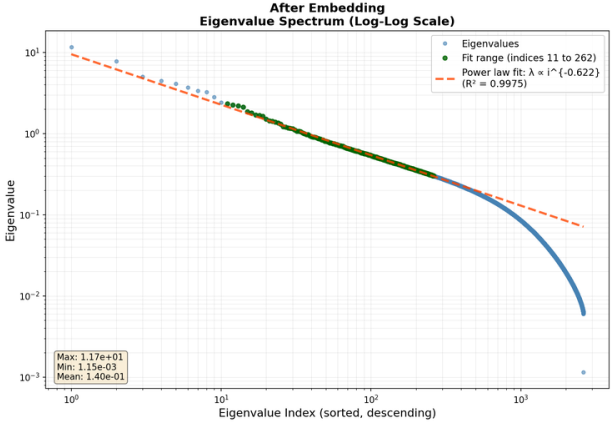


The spectrum does not show this positional structure

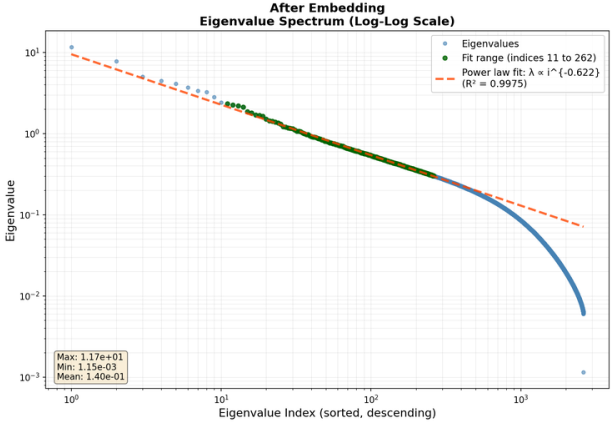


Permuting positions moves the diagonal band without changing the eigenvalues.

The embedding begins with an extended power-law regime

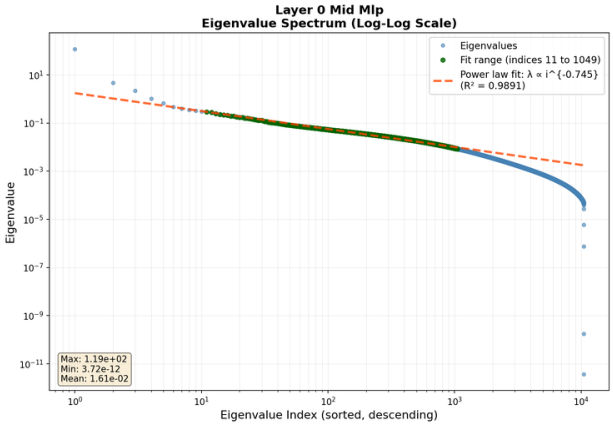


The embedding begins with an extended power-law regime

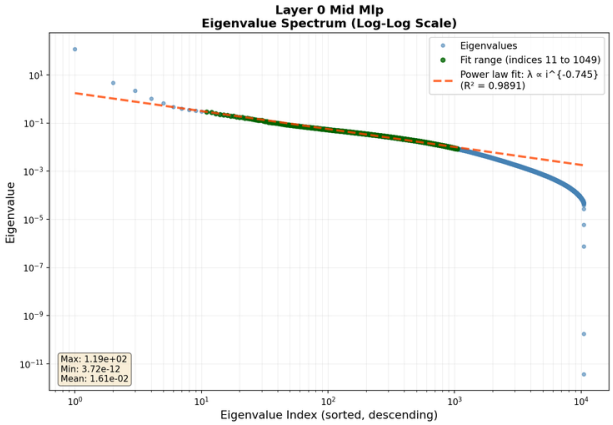


After the embedding, the fitted spectral exponent is approximately 0.62.

The power-law regime survives the first MLP

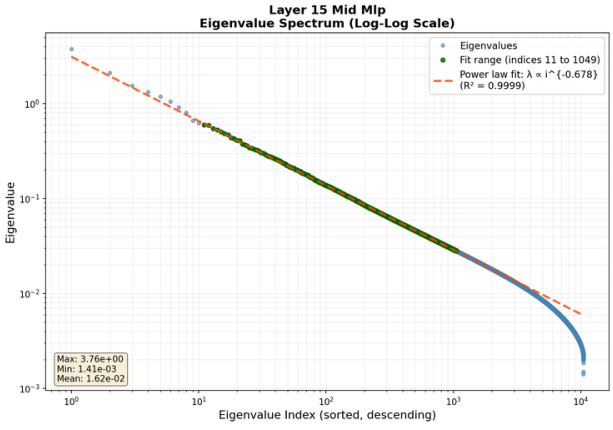


The power-law regime survives the first MLP

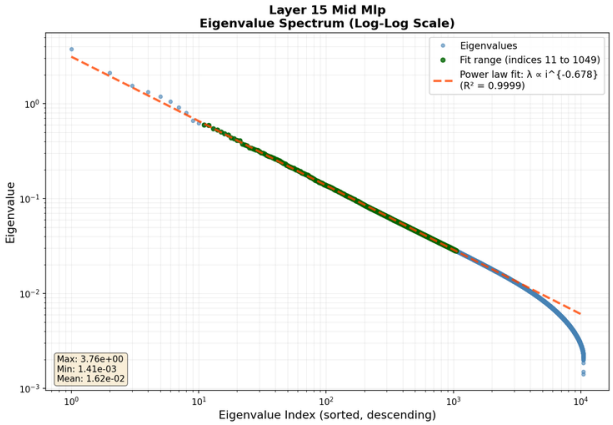


Inside layer 0, the fitted spectral exponent is approximately 0.75.

The same structure persists deep in the network

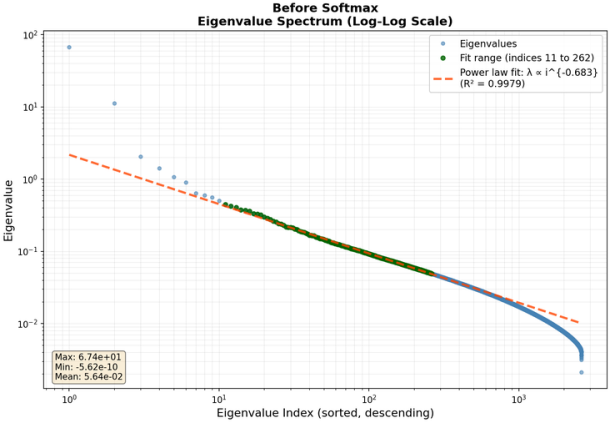


The same structure persists deep in the network

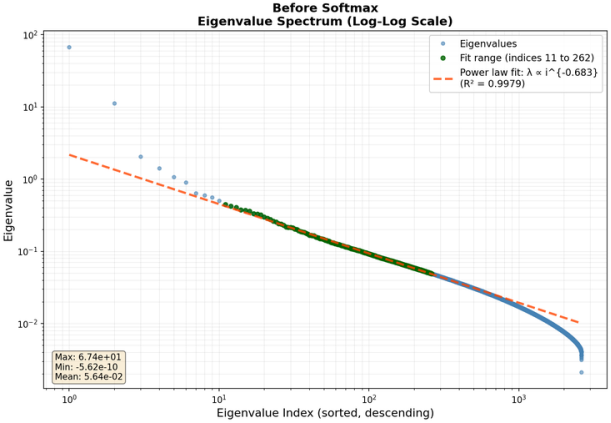


Inside layer 15, the fitted spectral exponent is approximately 0.68.

The final representation remains strongly anisotropic



The final representation remains strongly anisotropic



Immediately before the softmax, the fitted spectral exponent is approximately 0.68.

Theory: weak preservation

A linear statistic must survive nonlinear transformations

Covariance is a linear summary of the data, but a network repeatedly applies nonlinear transformations.

A linear statistic must survive nonlinear transformations

Covariance is a linear summary of the data, but a network repeatedly applies nonlinear transformations.

For the spectrum to remain useful inside a network, we want **weak preservation**:

$$\lambda_j(\Sigma) \asymp j^{-\alpha} \quad \Longrightarrow \quad \lambda_j(\Sigma_{\text{new}}) = j^{-\alpha} \times \text{slowly varying factors.}$$

A linear statistic must survive nonlinear transformations

Covariance is a linear summary of the data, but a network repeatedly applies nonlinear transformations.

For the spectrum to remain useful inside a network, we want **weak preservation**:

$$\lambda_j(\Sigma) \asymp j^{-\alpha} \quad \Longrightarrow \quad \lambda_j(\Sigma_{\text{new}}) = j^{-\alpha} \times \text{slowly varying factors.}$$

Constants, the eigenbasis, and logarithmic corrections may change while the polynomial exponent survives.

One-block stability can propagate through depth

For an MLP,

$$H^{(0)} = X, \qquad Z^{(\ell)} = (W^{(\ell)})^\top H^{(\ell-1)}, \qquad H^{(\ell)} = f_\ell(Z^{(\ell)}).$$

One-block stability can propagate through depth

For an MLP,

$$H^{(0)} = X, \qquad Z^{(\ell)} = (W^{(\ell)})^\top H^{(\ell-1)}, \qquad H^{(\ell)} = f_\ell(Z^{(\ell)}).$$

The layer-ℓ activation covariance is $\Sigma_\ell = \mathrm{Cov}(H^{(\ell)})$.

One-block stability can propagate through depth

For an MLP,

$$H^{(0)} = X, \qquad Z^{(\ell)} = (W^{(\ell)})^\top H^{(\ell-1)}, \qquad H^{(\ell)} = f_\ell(Z^{(\ell)}).$$

The layer-ℓ activation covariance is $\Sigma_\ell = \mathrm{Cov}(H^{(\ell)})$.

The question is whether

$$\lambda_j(\Sigma_{\ell-1}) \asymp j^{-\alpha} \quad \Longrightarrow \quad \lambda_j(\Sigma_\ell) \asymp j^{-\alpha}.$$

One-block stability can propagate through depth

For an MLP,

$$H^{(0)} = X, \qquad Z^{(\ell)} = (W^{(\ell)})^\top H^{(\ell-1)}, \qquad H^{(\ell)} = f_\ell(Z^{(\ell)}).$$

The layer-ℓ activation covariance is $\Sigma_\ell = \mathrm{Cov}(H^{(\ell)})$.

The question is whether

$$\lambda_j(\Sigma_{\ell-1}) \asymp j^{-\alpha} \quad \Longrightarrow \quad \lambda_j(\Sigma_\ell) \asymp j^{-\alpha}.$$

A uniform one-block statement propagates through finite depth by induction. The hard step is controlling one nonlinear transformation.

Random features isolate one nonlinear block

Freeze a Gaussian matrix W and study

$$h(X) = f(W^\top X).$$

Random features isolate one nonlinear block

Freeze a Gaussian matrix W and study

$$h(X) = f(W^\top X).$$

The population second-moment matrix is

$$K_{f,W} = \mathbb{E}_X\left[\frac{1}{d} f(W^\top X) f(W^\top X)^\top\right].$$

Random features isolate one nonlinear block

Freeze a Gaussian matrix W and study

$$h(X) = f(W^\top X).$$

The population second-moment matrix is

$$K_{f,W} = \mathbb{E}_X\left[\frac{1}{d} f(W^\top X) f(W^\top X)^\top\right].$$

Freezing W separates the representation question from the optimization of the final linear readout.

Random features isolate one nonlinear block

Freeze a Gaussian matrix W and study

$$h(X) = f(W^\top X).$$

The population second-moment matrix is

$$K_{f,W} = \mathbb{E}_X\left[\frac{1}{d} f(W^\top X) f(W^\top X)^\top\right].$$

Freezing W separates the representation question from the optimization of the final linear readout.

Maloney, Roberts, and Sully used this setting to identify approximate spectral preservation in a solvable scaling model (Maloney et al. 2022).

The exponent survives monomial random features

Theorem: power-law preservation for monomial random features

Let $X \sim N(0, \Sigma)$ with $\lambda_j(\Sigma) \asymp j^{-\alpha}$. Let W have independent Gaussian entries and let $f(y) = y^p$. Then, with high probability over W,

The exponent survives monomial random features

Theorem: power-law preservation for monomial random features

Let $X \sim N(0, \Sigma)$ with $\lambda_j(\Sigma) \asymp j^{-\alpha}$. Let W have independent Gaussian entries and let $f(y) = y^p$. Then, with high probability over W,

$$\lambda_j(K_{f,W}) \asymp \left(\frac{\log^{p-1}(j+1)}{j} \right)^{\alpha}, \qquad 1 \le j \le c \, \frac{d}{\log^{p+1} d}$$

The exponent survives monomial random features

Theorem: power-law preservation for monomial random features

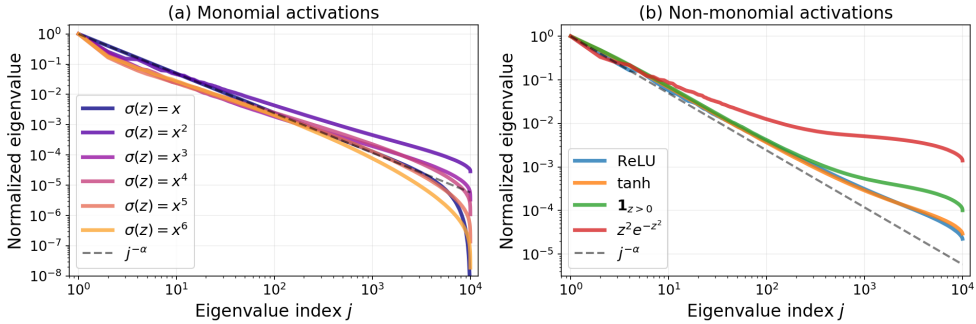
Let $X \sim N(0, \Sigma)$ with $\lambda_j(\Sigma) \asymp j^{-\alpha}$. Let W have independent Gaussian entries and let $f(y) = y^p$. Then, with high probability over W,

$$\lambda_j(K_{f,W}) \asymp \left(\frac{\log^{p-1}(j+1)}{j} \right)^{\alpha}, \qquad 1 \le j \le c \, \frac{d}{\log^{p+1} d}$$

P.–Ke Liang Xiao–Yizhe Zhu (2026): polynomial rate preserved; logarithmic correction added.

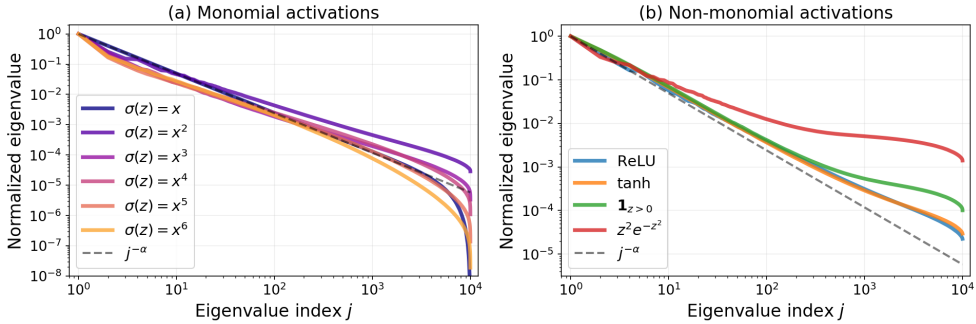
The phenomenon extends beyond the theorem

Median postactivation eigenvalue, 100 trials ($\alpha = 1.31$, $v = d = 10000$, $m = 100000$)



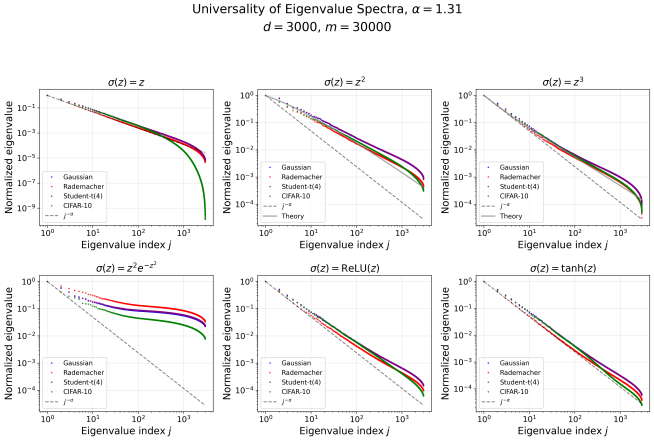
The phenomenon extends beyond the theorem

Median postactivation eigenvalue, 100 trials ($\alpha = 1.31$, $v = d = 10000$, $m = 100000$)

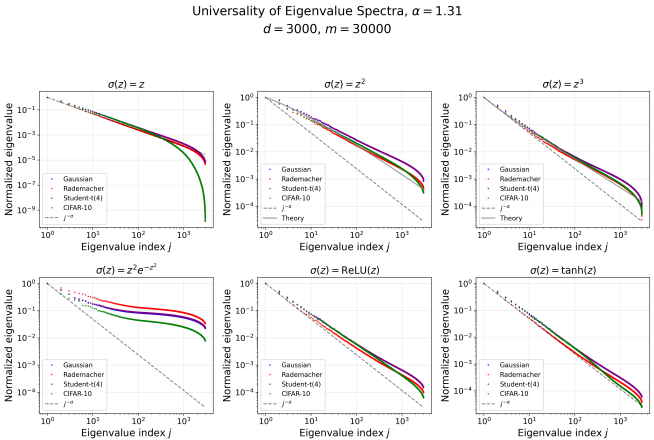


The theorem covers monomials. ReLU, tanh, and other activations show related behavior numerically—but not every activation preserves the same tail.

Preservation also appears beyond Gaussian inputs



Preservation also appears beyond Gaussian inputs



Similar slopes across four input distributions suggest a broader universality phenomenon.

Log corrections mimic new exponents

For $p = 2$, the theorem gives $\lambda_j \asymp \left(\frac{\log(j+1)}{j}\right)^{\alpha}$.

Log corrections mimic new exponents

For $p = 2$, the theorem gives $\lambda_j \asymp \left(\frac{\log(j+1)}{j}\right)^{\alpha}$.

What exponent would a local log–log fit report?

$$\alpha_{\mathrm{local}}(j) := -\frac{d \log \lambda_j}{d \log j} = \alpha \left(1 - \frac{j}{(j+1)\log(j+1)}\right).$$

Log corrections mimic new exponents

For $p = 2$, the theorem gives $\lambda_j \asymp \left(\frac{\log(j+1)}{j}\right)^{\alpha}$.

What exponent would a local log–log fit report?

$$\alpha_{\mathrm{local}}(j) := -\frac{d \log \lambda_j}{d \log j} = \alpha \left(1 - \frac{j}{(j+1)\log(j+1)}\right).$$

At $j = 3{,}000$

$$\alpha_{\mathrm{local}} \approx 0.875\alpha$$

The measured slope is 12.5% flatter.

Log corrections mimic new exponents

For $p = 2$, the theorem gives $\lambda_j \asymp \left(\frac{\log(j+1)}{j}\right)^{\alpha}$.

What exponent would a local log–log fit report?

$$\alpha_{\mathrm{local}}(j) := -\frac{d \log \lambda_j}{d \log j} = \alpha \left(1 - \frac{j}{(j+1)\log(j+1)}\right).$$

At $j = 3{,}000$

$$\alpha_{\mathrm{local}} \approx 0.875\alpha$$

The measured slope is 12.5% flatter.

To be within 5%

$$\alpha_{\mathrm{local}} \geq 0.95\alpha$$

$$j \gtrsim e^{20} \approx 4.9 \times 10^8$$

Log corrections mimic new exponents

For $p = 2$, the theorem gives $\lambda_j \asymp \left(\frac{\log(j+1)}{j}\right)^{\alpha}$.

What exponent would a local log–log fit report?

$$\alpha_{\text{local}}(j) := -\frac{d \log \lambda_j}{d \log j} = \alpha \left(1 - \frac{j}{(j+1)\log(j+1)}\right).$$

At $j = 3{,}000$

$\alpha_{\text{local}} \approx 0.875\alpha$

The measured slope is 12.5% flatter.

To be within 5%

$\alpha_{\text{local}} \geq 0.95\alpha$

$j \gtrsim e^{20} \approx 4.9 \times 10^8$

At realistic widths, the logarithm behaves like a different effective exponent.

The proof is a tensor covariance argument

1. **Center the activation** to isolate its highest Gaussian chaos.

The proof is a tensor covariance argument

1. **Center the activation** to isolate its highest Gaussian chaos.
2. **Lift into tensor space** so the kernel becomes a Gram matrix.

The proof is a tensor covariance argument

1. **Center the activation** to isolate its highest Gaussian chaos.
2. **Lift into tensor space** so the kernel becomes a Gram matrix.
3. **Count tensor-product eigenvalues** to obtain the logarithmic correction.

The proof is a tensor covariance argument

1. **Center the activation** to isolate its highest Gaussian chaos.
2. **Lift into tensor space** so the kernel becomes a Gram matrix.
3. **Count tensor-product eigenvalues** to obtain the logarithmic correction.
4. **Estimate a covariance from finite width** to transfer the population spectrum back to the random-feature matrix.

The proof is a tensor covariance argument

1. **Center the activation** to isolate its highest Gaussian chaos.
2. **Lift into tensor space** so the kernel becomes a Gram matrix.
3. **Count tensor-product eigenvalues** to obtain the logarithmic correction.
4. **Estimate a covariance from finite width** to transfer the population spectrum back to the random-feature matrix.

For the quadratic activation, every step is already visible in the second Wiener chaos.

Centering isolates the pure second chaos

Write $W = [w_1 \ \cdots \ w_d] \in \mathbb{R}^{v \times d}$, so $w_a \in \mathbb{R}^v$ for $a \in [d] := \{1, \ldots, d\}$. Set

$$g_a = w_a^\top X, \qquad q_a = \mathbb{E}_X[g_a^2] = w_a^\top \Sigma w_a.$$

Centering isolates the pure second chaos

Write $W = [w_1 \ \cdots \ w_d] \in \mathbb{R}^{v \times d}$, so $w_a \in \mathbb{R}^v$ for $a \in [d] := \{1, \dots, d\}$. Set

$$g_a = w_a^\top X, \qquad q_a = \mathbb{E}_X[g_a^2] = w_a^\top \Sigma w_a.$$

The centered quadratic feature is

$$: g_a^2 := g_a^2 - q_a.$$

Centering isolates the pure second chaos

Write $W = [w_1 \ \cdots \ w_d] \in \mathbb{R}^{v \times d}$, so $w_a \in \mathbb{R}^v$ for $a \in [d] := \{1, \dots, d\}$. Set

$$g_a = w_a^\top X, \qquad q_a = \mathbb{E}_X[g_a^2] = w_a^\top \Sigma w_a.$$

The centered quadratic feature is

$$:g_a^2: = g_a^2 - q_a.$$

If $G = W^\top \Sigma W$ and $q = \mathrm{diag}(G)$, then

$$K_{2,W}^{\mathrm{raw}} = \frac{1}{d} q q^\top + \frac{2}{d} G^{\odot 2}, \qquad K_{2,W}^{\mathrm{chaos}} = \frac{2}{d} G^{\odot 2}.$$

Centering isolates the pure second chaos

Write $W = [w_1 \ \cdots \ w_d] \in \mathbb{R}^{v \times d}$, so $w_a \in \mathbb{R}^v$ for $a \in [d] := \{1, \dots, d\}$. Set

$$g_a = w_a^\top X, \qquad q_a = \mathbb{E}_X[g_a^2] = w_a^\top \Sigma w_a.$$

The centered quadratic feature is

$$:g_a^2: = g_a^2 - q_a.$$

If $G = W^\top \Sigma W$ and $q = \mathrm{diag}(G)$, then

$$K_{2,W}^{\mathrm{raw}} = \frac{1}{d} q q^\top + \frac{2}{d} G^{\odot 2}, \qquad K_{2,W}^{\mathrm{chaos}} = \frac{2}{d} G^{\odot 2}.$$

The two matrices differ by one rank-one term; interlacing preserves the tail.

Quadratic features live in symmetric tensor space

Let $X = \Sigma^{1/2} Z$, where $Z \sim N(0, I_v)$, and let $h(Z)$ collect the orthonormal basis functions $(Z_r^2 - 1)/\sqrt{2}$ and $Z_r Z_s$ for $r < s$. Then

$$
\begin{aligned}
: g_a^2 : &= \langle \phi_a, h(Z) \rangle, \\
\phi_a &= \left((\sqrt{2}\, \lambda_r w_{ra}^2)_r,\ (2\sqrt{\lambda_r \lambda_s}\, w_{ra} w_{sa})_{r<s} \right).
\end{aligned}
$$

Quadratic features live in symmetric tensor space

Let $X = \Sigma^{1/2} Z$, where $Z \sim N(0, I_v)$, and let $h(Z)$ collect the orthonormal basis functions $(Z_r^2 - 1)/\sqrt{2}$ and $Z_r Z_s$ for $r < s$. Then

$$
\begin{aligned}
:g_a^2: &= \langle \phi_a, h(Z) \rangle, \\
\phi_a &= \Big((\sqrt{2}\,\lambda_r w_{ra}^2)_r,\ (2\sqrt{\lambda_r \lambda_s}\, w_{ra} w_{sa})_{r<s} \Big).
\end{aligned}
$$

Orthonormality gives $\mathbb{E}_X[:g_a^2 :: g_b^2:] = \langle \phi_a, \phi_b \rangle$.

Quadratic features live in symmetric tensor space

Let $X = \Sigma^{1/2} Z$, where $Z \sim N(0, I_v)$, and let $h(Z)$ collect the orthonormal basis functions $(Z_r^2 - 1)/\sqrt{2}$ and $Z_r Z_s$ for $r < s$. Then

$$
\begin{aligned}
:g_a^2: &= \langle \phi_a, h(Z) \rangle, \\
\phi_a &= \Big((\sqrt{2}\,\lambda_r w_{ra}^2)_r,\ (2\sqrt{\lambda_r \lambda_s}\, w_{ra} w_{sa})_{r<s} \Big).
\end{aligned}
$$

Orthonormality gives $\mathbb{E}_X[:g_a^2 :: g_b^2:] = \langle \phi_a, \phi_b \rangle$.

Thus, with $\Phi = [\phi_1 \ \cdots \ \phi_d] \in \mathbb{R}^{v(v+1)/2 \times d}$,

$$
\boxed{K_{2,W}^{\text{chaos}} = \frac{1}{d} \Phi^\top \Phi}
$$

Tensor eigenvalues are pair products

The principal population eigenvalues in tensor space are

$$4\lambda_r\lambda_s, \qquad r < s.$$

Tensor eigenvalues are pair products

The principal population eigenvalues in tensor space are

$$4\lambda_r\lambda_s, \qquad r < s.$$

If $\lambda_r = r^{-\alpha}$, then the number above ε is

$$\begin{aligned} N_{\text{off}}(\varepsilon) &= \#\{r < s : 4\lambda_r\lambda_s \geq \varepsilon\} \\ &= \#\{r < s : rs \leq (4/\varepsilon)^{1/\alpha}\}. \end{aligned}$$

Tensor eigenvalues are pair products

The principal population eigenvalues in tensor space are

$$4\lambda_r\lambda_s, \qquad r < s.$$

If $\lambda_r = r^{-\alpha}$, then the number above ε is

$$\begin{aligned} N_{\mathrm{off}}(\varepsilon) &= \#\{r < s : 4\lambda_r\lambda_s \geq \varepsilon\} \\ &= \#\{r < s : rs \leq (4/\varepsilon)^{1/\alpha}\}. \end{aligned}$$

Thus the spectral counting problem is a lattice-point problem under a hyperbola.

The divisor count creates the logarithm

The divisor count gives

$$\#\{r < s : rs \leq R\} \sim \frac{1}{2} R \log R.$$

The divisor count creates the logarithm

The divisor count gives

$$\#\{r < s : rs \leq R\} \sim \frac{1}{2} R \log R.$$

Hence $j \asymp R \log R$, so

$$\lambda_j \asymp \left(\frac{\log(j+1)}{j} \right)^{\alpha}.$$

Width becomes sample size in tensor space

The nonzero eigenvalues of

$$\frac{1}{d}\Phi^\top\Phi \qquad \text{and} \qquad \widehat{C}_{\text{chaos}} := \frac{1}{d}\sum_{a=1}^{d}\phi_a \otimes \phi_a = \frac{1}{d}\Phi\Phi^\top$$

are identical.

Width becomes sample size in tensor space

The nonzero eigenvalues of

$$\frac{1}{d}\Phi^\top\Phi \qquad \text{and} \qquad \widehat{C}_{\text{chaos}} := \frac{1}{d}\sum_{a=1}^{d}\phi_a \otimes \phi_a = \frac{1}{d}\Phi\Phi^\top$$

are identical.

The random-feature width d is the sample size in an empirical covariance-estimation problem for tensor features.

The proof starts with a head–tail split

Let H_k index the k largest distinct-pair scales $4\lambda_r\lambda_s$. For each sample $a \in [d]$, define

$$D_k := \mathrm{diag}(4\lambda_r\lambda_s)_{(r,s)\in H_k},$$
$$\xi_a := (w_{ra}w_{sa})_{(r,s)\in H_k} \in \mathbb{R}^k, \qquad \phi_a = \underbrace{D_k^{1/2}\xi_a}_{\text{head}} \oplus \underbrace{\phi_a^{\text{tail}}}_{\text{tail}}.$$

The proof starts with a head–tail split

Let H_k index the k largest distinct-pair scales $4\lambda_r\lambda_s$. For each sample $a \in [d]$, define

$$D_k := \operatorname{diag}(4\lambda_r\lambda_s)_{(r,s)\in H_k},$$
$$\xi_a := (w_{ra}w_{sa})_{(r,s)\in H_k} \in \mathbb{R}^k, \qquad \phi_a = \underbrace{D_k^{1/2}\xi_a}_{\text{head}} \oplus \underbrace{\phi_a^{\text{tail}}}_{\text{tail}}.$$

Head covariance: $\widehat{C}_k = D_k^{1/2}\left(\frac{1}{d}\sum_{a=1}^{d}\xi_a \otimes \xi_a\right)D_k^{1/2}.$

The proof starts with a head–tail split

Let H_k index the k largest distinct-pair scales $4\lambda_r\lambda_s$. For each sample $a \in [d]$, define

$$D_k := \mathrm{diag}(4\lambda_r\lambda_s)_{(r,s)\in H_k},$$
$$\xi_a := (w_{ra}w_{sa})_{(r,s)\in H_k} \in \mathbb{R}^k, \qquad \phi_a = \underbrace{D_k^{1/2}\xi_a}_{\text{head}} \oplus \underbrace{\phi_a^{\text{tail}}}_{\text{tail}}.$$

Head covariance: $\widehat{C}_k = D_k^{1/2}\left(\frac{1}{d}\sum_{a=1}^{d}\xi_a \otimes \xi_a\right)D_k^{1/2}.$

Concentration target: $\left\|\frac{1}{d}\sum_{a=1}^{d}\xi_a \otimes \xi_a - I_k\right\| \le \delta.$

The proof starts with a head–tail split

Let H_k index the k largest distinct-pair scales $4\lambda_r\lambda_s$. For each sample $a \in [d]$, define

$$D_k := \mathrm{diag}(4\lambda_r\lambda_s)_{(r,s)\in H_k},$$
$$\xi_a := (w_{ra}w_{sa})_{(r,s)\in H_k} \in \mathbb{R}^k, \qquad \phi_a = \underbrace{D_k^{1/2}\xi_a}_{\text{head}} \oplus \underbrace{\phi_a^{\text{tail}}}_{\text{tail}}.$$

Head covariance: $\widehat{C}_k = D_k^{1/2}\left(\frac{1}{d}\sum_{a=1}^{d}\xi_a \otimes \xi_a\right)D_k^{1/2}.$

Concentration target: $\left\|\frac{1}{d}\sum_{a=1}^{d}\xi_a \otimes \xi_a - I_k\right\| \le \delta.$

Head: preserve D_k multiplicatively.

Tail: control spectral leakage block by block.

The tensor features are heavy-tailed

Coordinates of ξ_a are products $w_{ra} w_{sa}$. For a unit vector u, Gaussian hypercontractivity gives

$$\|\langle \xi_a, u \rangle\|_{L^q} \le (q-1)\|\langle \xi_a, u \rangle\|_{L^2}, \qquad q \ge 2.$$

The tensor features are heavy-tailed

Coordinates of ξ_a are products $w_{ra} w_{sa}$. For a unit vector u, Gaussian hypercontractivity gives

$$\|\langle \xi_a, u \rangle\|_{L^q} \le (q-1)\|\langle \xi_a, u \rangle\|_{L^2}, \qquad q \ge 2.$$

This is **sub-exponential**, not sub-Gaussian, growth.

The tensor features are heavy-tailed

Coordinates of ξ_a are products $w_{ra} w_{sa}$. For a unit vector u, Gaussian hypercontractivity gives

$$\|\langle \xi_a, u \rangle\|_{L^q} \le (q-1)\|\langle \xi_a, u \rangle\|_{L^2}, \qquad q \ge 2.$$

This is **sub-exponential**, not sub-Gaussian, growth.

Large individual tensor-feature vectors can dominate the ordinary empirical covariance, so the clean $d \asymp k$ sub-Gaussian theorem does not apply. The concentration step uses bounds for sums of heavy-tailed random matrices (Jirak et al. 2026).

The tensor features are heavy-tailed

Coordinates of ξ_a are products $w_{ra} w_{sa}$. For a unit vector u, Gaussian hypercontractivity gives

$$\|\langle \xi_a, u \rangle\|_{L^q} \le (q-1)\|\langle \xi_a, u \rangle\|_{L^2}, \qquad q \ge 2.$$

This is **sub-exponential**, not sub-Gaussian, growth.

Large individual tensor-feature vectors can dominate the ordinary empirical covariance, so the clean $d \asymp k$ sub-Gaussian theorem does not apply. The concentration step uses bounds for sums of heavy-tailed random matrices (Jirak et al. 2026).

For degree p, the current argument gives a sharp range

$$k_\star \asymp \frac{d}{\log^{p+1} d}; \qquad p = 2: \quad k_\star \asymp \frac{d}{\log^3 d}.$$

Two logarithms come from different mechanisms

Structural

$$\lambda_j \asymp \left(\frac{\log j}{j}\right)^{\alpha}$$

Comes from counting pair products $rs \leq R$.

It is already present in the **population tensor spectrum**.

Two logarithms come from different mechanisms

Structural

$$\lambda_j \asymp \left(\frac{\log j}{j}\right)^{\alpha}$$

Comes from counting pair products $rs \leq R$.

It is already present in the **population tensor spectrum**.

Statistical

$$k_\star \asymp \frac{d}{\log^3 d}$$

Comes from estimating a covariance with heavy-tailed tensor samples.

It limits the current **finite-width proof range**.

Open problems

- **ReLU and tanh.** Can we prove sharp preservation beyond monomials?

Open problems

- **ReLU and tanh.** Can we prove sharp preservation beyond monomials?
- **Remove $k_\star$.** Can a finite-width theorem reach the full spectrum?

Open problems

- **ReLU and tanh.** Can we prove sharp preservation beyond monomials?
- **Remove $k_\star$.** Can a finite-width theorem reach the full spectrum?
- **Other models.** Where else does the same mechanism appear?

Open problems

- **ReLU and tanh.** Can we prove sharp preservation beyond monomials?
- **Remove $k_\star$.** Can a finite-width theorem reach the full spectrum?
- **Other models.** Where else does the same mechanism appear?
- **Residual connections.** What spectral structure do skip connections preserve?

Open problems

- **ReLU and tanh.** Can we prove sharp preservation beyond monomials?
- **Remove $k_\star$.** Can a finite-width theorem reach the full spectrum?
- **Other models.** Where else does the same mechanism appear?
- **Residual connections.** What spectral structure do skip connections preserve?
- **Sequences and attention.** If dependence decays along the sequence, what structure survives attention and depth?

Selected references

1. Elliot Paquette, Ke Liang Xiao, and Yizhe Zhu. *Power-Law Spectrum of the Random Feature Model* (2026). arXiv:2603.14578.

Selected references

1. Elliot Paquette, Ke Liang Xiao, and Yizhe Zhu. *Power-Law Spectrum of the Random Feature Model* (2026). arXiv:2603.14578.

2. Moritz Jirak, Stanislav Minsker, Yiqiu Shen, and Martin Wahl. *Concentration and Moment Inequalities for Sums of Independent Heavy-Tailed Random Matrices*. *Probability Theory and Related Fields* 194 (2026), 1917–1944. doi:10.1007/s00440-025-01412-6.

Selected references

1. Elliot Paquette, Ke Liang Xiao, and Yizhe Zhu. *Power-Law Spectrum of the Random Feature Model* (2026). arXiv:2603.14578.

2. Moritz Jirak, Stanislav Minsker, Yiqiu Shen, and Martin Wahl. *Concentration and Moment Inequalities for Sums of Independent Heavy-Tailed Random Matrices*. *Probability Theory and Related Fields* 194 (2026), 1917–1944. doi:10.1007/s00440-025-01412-6.

3. Alexander Maloney, Daniel A. Roberts, and James Sully. *A Solvable Model of Neural Scaling Laws* (2022). arXiv:2210.16859.

The spectrum is a useful first coordinate system

1. Covariance eigenvalues quantify effective dimension.

The spectrum is a useful first coordinate system

1. Covariance eigenvalues quantify effective dimension.
2. The spectrum is not the whole story: eigenvectors encode important structural information about the data.

The spectrum is a useful first coordinate system

1. Covariance eigenvalues quantify effective dimension.
2. The spectrum is not the whole story: eigenvectors encode important structural information about the data.
3. Real spectra can cross over between several power-law regimes.

The spectrum is a useful first coordinate system

1. Covariance eigenvalues quantify effective dimension.
2. The spectrum is not the whole story: eigenvectors encode important structural information about the data.
3. Real spectra can cross over between several power-law regimes.
4. Random monomial features preserve the polynomial exponent, up to a structural logarithmic correction.

The spectrum is a useful first coordinate system

1. Covariance eigenvalues quantify effective dimension.
2. The spectrum is not the whole story: eigenvectors encode important structural information about the data.
3. Real spectra can cross over between several power-law regimes.
4. Random monomial features preserve the polynomial exponent, up to a structural logarithmic correction.
5. Finite width turns the proof into heavy-tailed covariance estimation.

The spectrum is a useful first coordinate system

1. Covariance eigenvalues quantify effective dimension.
2. The spectrum is not the whole story: eigenvectors encode important structural information about the data.
3. Real spectra can cross over between several power-law regimes.
4. Random monomial features preserve the polynomial exponent, up to a structural logarithmic correction.
5. Finite width turns the proof into heavy-tailed covariance estimation.

In Module 2, we will explore how spectra affect learning curves.

We will introduce how this spectral structure can affect scaling laws.