Finite-Width Neural Tangent Kernels from Feynman Diagrams

by Philipp Misof

Department of Mathematical Sciences, Division of Algebra and Geometry

August 18, 2026

UNIVERSITY OF GOTHENBURG

CHALMERS UNIVERSITY OF TECHNOLOGY

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Joint work with

Max Guillen

Jan E. Gerken

presented at the

ICML
International Conference
On Machine Learning

Training Dynamics under Gradient Flow

Training Dynamics under Gradient Flow

$$\frac{\mathrm{d}\theta_\mu(t)}{\mathrm{d}t} = -\eta\,\frac{\partial \mathcal{L}}{\partial \theta_\mu} = -\eta \sum_{i=1}^{N} \frac{\partial f_\theta(x_i)}{\partial \theta_\mu}\,\frac{\partial \ell\big(f_\theta(x_i), y_i\big)}{\partial f_\theta(x_i)}$$

Training Dynamics under Gradient Flow

parameters

learning rate

loss

training data

$$\frac{\mathrm{d}\theta_\mu(t)}{\mathrm{d}t} = -\eta \frac{\partial \mathcal{L}}{\partial \theta_\mu} = -\eta \sum_{i=1}^{N} \frac{\partial f_\theta(x_i)}{\partial \theta_\mu} \frac{\partial \ell\big(f_\theta(x_i), y_i\big)}{\partial f_\theta(x_i)}$$

network output

Training Dynamics under Gradient Flow

learning rate

parameters loss training data

$$\frac{\mathrm{d}\theta_\mu(t)}{\mathrm{d}t} = -\eta\,\frac{\partial \mathcal{L}}{\partial \theta_\mu} = -\eta \sum_{i=1}^{N} \frac{\partial f_\theta(x_i)}{\partial \theta_\mu}\,\frac{\partial \ell\big(f_\theta(x_i), y_i\big)}{\partial f_\theta(x_i)}$$

network output

by chain rule

$$\frac{\mathrm{d}f_\theta(x)}{\mathrm{d}t} = \sum_{\mu} \frac{\partial f_\theta(x)}{\partial \theta_\mu}\,\frac{\mathrm{d}\theta_\mu(t)}{\mathrm{d}t} = -\eta \sum_{i=1}^{N} \sum_{\mu} \frac{\partial f_\theta(x)}{\partial \theta_\mu}\,\frac{\partial f_\theta(x_i)}{\partial \theta_\mu}\,\frac{\partial \ell\big(f_\theta(x_i), y_i\big)}{\partial f_\theta(x_i)}$$

Training Dynamics under Gradient Flow

learning rate

parameters loss training data

$$\frac{\mathrm{d}\theta_\mu(t)}{\mathrm{d}t} = -\eta \frac{\partial \mathcal{L}}{\partial \theta_\mu} = -\eta \sum_{i=1}^{N} \frac{\partial f_\theta(x_i)}{\partial \theta_\mu} \frac{\partial \ell\big(f_\theta(x_i), y_i\big)}{\partial f_\theta(x_i)}$$

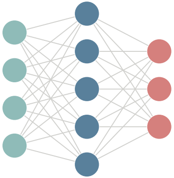
network output

by chain rule

$$\frac{\mathrm{d}f_\theta(x)}{\mathrm{d}t} = \sum_\mu \frac{\partial f_\theta(x)}{\partial \theta_\mu} \frac{\mathrm{d}\theta_\mu(t)}{\mathrm{d}t} = -\eta \sum_{i=1}^{N} \underbrace{\sum_\mu \frac{\partial f_\theta(x)}{\partial \theta_\mu} \frac{\partial f_\theta(x_i)}{\partial \theta_\mu}}_{\hat{\Theta}_t^{(L)}(x, x_i)} \frac{\partial \ell\big(f_\theta(x_i), y_i\big)}{\partial f_\theta(x_i)}$$

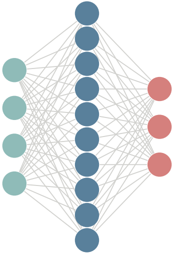
empirical Neural Tangent Kernel

Training Dynamics at Infinite-Width



Training Dynamics at Infinite-Width

- Taking hidden layer width $n \to \infty$



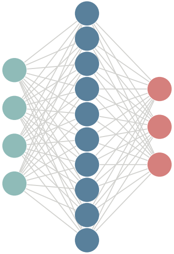
(Jacot, Gabriel, and Hongler 2018)

Training Dynamics at Infinite-Width

- Taking hidden layer width $n \to \infty$

$$\hat{\Theta}_t^{(L)}(x, x') \to \Theta^{(L)}(x, x')$$

frozen, deterministic NTK



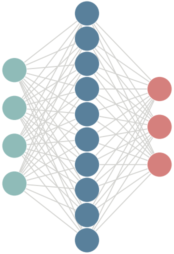
(Jacot, Gabriel, and Hongler 2018)

Training Dynamics at Infinite-Width

- Taking hidden layer width $n \to \infty$

$$\hat{\Theta}_t^{(L)}(x, x') \to \Theta^{(L)}(x, x')$$

  frozen, deterministic NTK
- **Gaussian process** over initializations with covariance $K^{(L)}(x, x')$ (NNGP kernel).



(Jacot, Gabriel, and Hongler 2018)

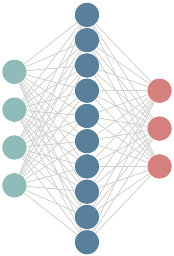
Training Dynamics at Infinite-Width

- Taking hidden layer width $n \to \infty$

$$\hat{\Theta}_t^{(L)}(x, x') \to \Theta^{(L)}(x, x')$$

frozen, deterministic NTK

- **Gaussian process** over initializations with covariance $K^{(L)}(x, x')$ (NNGP kernel).



The mean of the network output evolves according to

$$\mu_t(x) = \Theta^{(L)}(x, X)\Theta^{(L)}(X, X)^{-1}\left(\mathbb{I} - e^{-\eta\,\Theta^{(L)}(X,X)t}\right) Y$$

(Jacot, Gabriel, and Hongler 2018)

Loss of Feature Learning

(Yang and Hu 2022; Lee et al. 2020)

Loss of Feature Learning

- Implicitly changed parameterization

$$z_i^{(\ell)}(x) = \frac{1}{\sqrt{n_{\ell-1}}} \sum_{j=1}^{n_{\ell-1}} W_{ij}^{(\ell)} \, \sigma\left(z_j^{(\ell-1)}(x)\right) + b_i^{(\ell)}, \qquad W_{ij}^{(\ell)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, C_W^{(\ell)}\right)$$

(Yang and Hu 2022; Lee et al. 2020)

Loss of Feature Learning

- Implicitly changed parameterization

$$z_i^{(\ell)}(x) = \frac{1}{\sqrt{n_{\ell-1}}} \sum_{j=1}^{n_{\ell-1}} W_{ij}^{(\ell)} \, \sigma\left(z_j^{(\ell-1)}(x)\right) + b_i^{(\ell)}, \quad W_{ij}^{(\ell)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, C_W^{(\ell)}\right)$$

(Yang and Hu 2022; Lee et al. 2020)

Loss of Feature Learning

- Implicitly changed parameterization

$$z_i^{(\ell)}(x) = \frac{1}{\sqrt{n_{\ell-1}}} \sum_{j=1}^{n_{\ell-1}} W_{ij}^{(\ell)} \sigma\left(z_j^{(\ell-1)}(x)\right) + b_i^{(\ell)}, \quad W_{ij}^{(\ell)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, C_W^{(\ell)}\right)$$

- Change in activations during training

$$a^{(\ell)}(x) \coloneqq \sigma\left(z^{(\ell)}(x)\right), \qquad \Delta a^{(\ell)}(x;t) \coloneqq a_t^{(\ell)}(x) - a_0^{(\ell)}(x)$$

(Yang and Hu 2022; Lee et al. 2020)

Loss of Feature Learning

- Implicitly changed parameterization

$$z_i^{(\ell)}(x) = \frac{1}{\sqrt{n_{\ell-1}}} \sum_{j=1}^{n_{\ell-1}} W_{ij}^{(\ell)} \, \sigma\left(z_j^{(\ell-1)}(x)\right) + b_i^{(\ell)}, \qquad W_{ij}^{(\ell)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, C_W^{(\ell)}\right)$$

- Change in activations during training

$$a^{(\ell)}(x) := \sigma\left(z^{(\ell)}(x)\right), \qquad \Delta a^{(\ell)}(x;t) := a_t^{(\ell)}(x) - a_0^{(\ell)}(x)$$

$$\boxed{\frac{1}{n_\ell} \left\| \Delta a^{(\ell)}(x;t) \right\|^2 \xrightarrow[n \to \infty]{\text{a.s.}} 0}$$

(Yang and Hu 2022; Lee et al. 2020)

Different solutions to **recover feature learning**

Different solutions to **recover feature learning**

Changing the parameterization

$\Rightarrow$ E.g. Maximal update parameterization
(μ P)

Different solutions to **recover feature learning**

Changing the parameterization

$\Rightarrow$ E.g. Maximal update parameterization (μ P)

Standard parameterization blows up

Different solutions to **recover feature learning**

Changing the parameterization

$\Rightarrow$ E.g. Maximal update parameterization (μ P)

Standard parameterization blows up

Dichotomy

No kernel regime allows feature learning

Different solutions to **recover feature learning**

Changing the parameterization

$\Rightarrow$ E.g. Maximal update parameterization (μ P)

Standard parameterization blows up

Dichotomy

No kernel regime allows feature learning

Finite-width corrections

Extend NTK theory to capture finite-width effects

Different solutions to **recover feature learning**

Changing the parameterization

$\Rightarrow$ E.g. Maximal update parameterization (μ P)

Standard parameterization blows up

Dichotomy

No kernel regime allows feature learning

Finite-width corrections

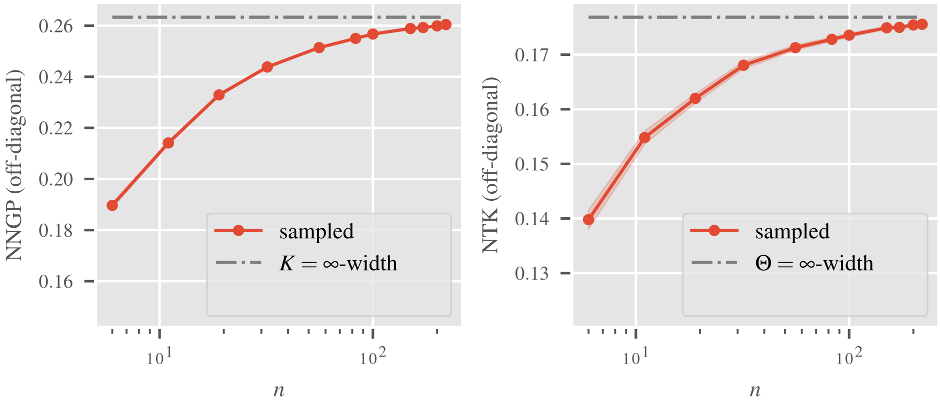
Extend NTK theory to capture finite-width effects

Advantage

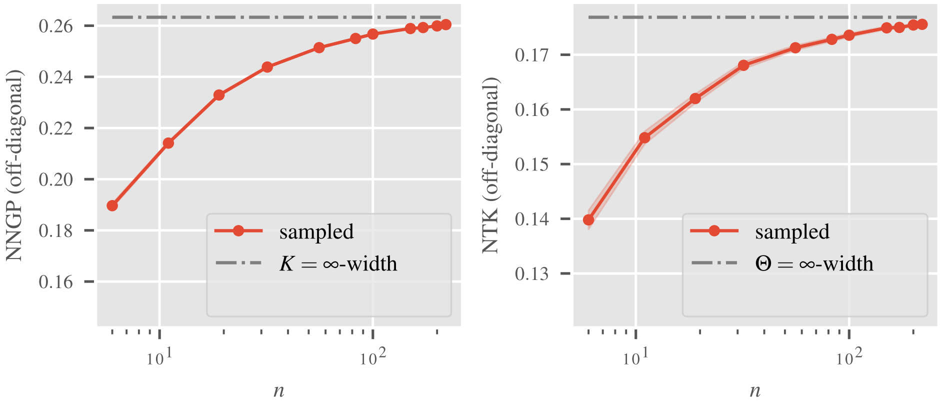
Converges to the well-understood limit as $n \to \infty$

The Goal: Capture the Finite-Width Correction at $\mathcal{O}(1/n)$

The Goal: Capture the Finite-Width Correction at $\mathcal{O}(1/n)$



The Goal: Capture the Finite-Width Correction at $\mathcal{O}(1/n)$



For now: Restrict to initialization

$1/n$-Corrections to the NTK Limit

$1/n$-Corrections to the NTK Limit

- When $n \to \infty$, z follows a **normal distribution** $\mathcal{N}(0, K)$.

$$p(z) \propto \exp\left(-\frac{1}{2} \sum_{\alpha_1, \alpha_2 \in \mathcal{D}} K^{-1}_{\alpha_1 \alpha_2} \sum_{i=1}^{n} z_{i;\alpha_1} z_{i;\alpha_2}\right)$$

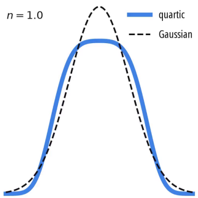
$1/n$-Corrections to the NTK Limit

- When $n \to \infty$, z follows a **normal distribution** $\mathcal{N}(0, K)$.
- Make a **nearly-Gaussian ansatz** for finite n.

$$p(z) \propto \exp\left(-\frac{1}{2} \sum_{\alpha_1,\alpha_2 \in \mathcal{D}} K^{-1}_{\alpha_1\alpha_2} \sum_{i=1}^{n} z_{i;\alpha_1} z_{i;\alpha_2}\right)$$

$1/n$-Corrections to the NTK Limit

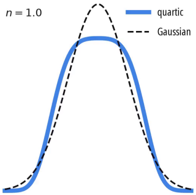
- When $n \to \infty$, z follows a **normal distribution** $\mathcal{N}(0, K)$.
- Make a **nearly-Gaussian ansatz** for finite n.



$$p(z) \propto \exp\left(-\frac{1}{2} \sum_{\alpha_1, \alpha_2 \in \mathcal{D}} g^{\alpha_1 \alpha_2} \sum_{i=1}^{n} z_{i;\alpha_1} z_{i;\alpha_2} + \frac{1}{8} \sum_{\alpha_1, \ldots, \alpha_4 \in \mathcal{D}} v^{(\alpha_1 \alpha_2)(\alpha_3 \alpha_4)} \sum_{i_1, i_2 = 1}^{n} z_{i_1;\alpha_1} z_{i_1;\alpha_2} z_{i_2;\alpha_3} z_{i_2;\alpha_4} \right)$$

$1/n$-Corrections to the NTK Limit

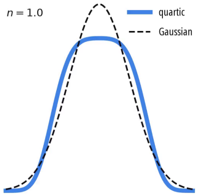
- When $n \to \infty$, z follows a **normal distribution** $\mathcal{N}(0, K)$.
- Make a **nearly-Gaussian ansatz** for finite n.



$$p(z) \propto \exp\left(-\frac{1}{2} \sum_{\alpha_1, \alpha_2 \in \mathcal{D}} g^{\alpha_1 \alpha_2} \sum_{i=1}^{n} z_{i;\alpha_1} z_{i;\alpha_2} + \frac{1}{8} \sum_{\alpha_1, \ldots, \alpha_4 \in \mathcal{D}} v^{(\alpha_1 \alpha_2)(\alpha_3 \alpha_4)} \sum_{i_1, i_2 = 1}^{n} z_{i_1;\alpha_1} z_{i_1;\alpha_2} z_{i_2;\alpha_3} z_{i_2;\alpha_4} \right)$$

$1/n$-Corrections to the NTK Limit

- When $n \to \infty$, z follows a **normal distribution** $\mathcal{N}(0, K)$.
- Make a **nearly-Gaussian ansatz** for finite n.

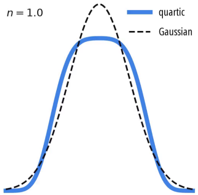


$$p(z) \propto \exp\left(-\frac{1}{2} \sum_{\alpha_1, \alpha_2 \in \mathcal{D}} g^{\alpha_1 \alpha_2} \sum_{i=1}^{n} z_{i;\alpha_1} z_{i;\alpha_2} + \frac{1}{8} \sum_{\alpha_1, \ldots, \alpha_4 \in \mathcal{D}} v^{(\alpha_1 \alpha_2)(\alpha_3 \alpha_4)} \sum_{i_1, i_2 = 1}^{n} z_{i_1;\alpha_1} z_{i_1;\alpha_2} z_{i_2;\alpha_3} z_{i_2;\alpha_4} \right)$$

- Layer by layer, compare with 2^{nd} and 4^{th} **cumulants** to determine $v^{(\ell)}$ and $g^{(\ell)}$ up to $\mathcal{O}(1/n)$.

$1/n$-Corrections to the NTK Limit

- When $n \to \infty$, z follows a **normal distribution** $\mathcal{N}(0, K)$.
- Make a **nearly-Gaussian ansatz** for finite n.



$$p(z) \propto \exp\left(-\frac{1}{2}\sum_{\alpha_1,\alpha_2\in\mathcal{D}} g^{\alpha_1\alpha_2}\sum_{i=1}^{n} z_{i;\alpha_1}z_{i;\alpha_2} + \frac{1}{8}\sum_{\alpha_1,\ldots,\alpha_4\in\mathcal{D}} v^{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}\sum_{i_1,i_2=1}^{n} z_{i_1;\alpha_1}z_{i_1;\alpha_2}z_{i_2;\alpha_3}z_{i_2;\alpha_4}\right)$$

- Layer by layer, compare with 2nd and 4th **cumulants** to determine $v^{(\ell)}$ and $g^{(\ell)}$ up to $\mathcal{O}(1/n)$.
- Can show inductively that $v = \mathcal{O}(1/n)$ and $g = K^{-1} + \mathcal{O}(1/n)$.

Cumulants Decompose into Tensors

Use the nearly-Gaussian distribution to **compute all (mixed) cumulants** of z and $\widehat{\Theta}$ up to $\mathcal{O}(1/n)$.

Cumulants Decompose into Tensors

Use the nearly-Gaussian distribution to **compute all (mixed) cumulants** of z and $\widehat{\Theta}$ up to $\mathcal{O}(1/n)$.
A **pattern** emerges:

$$\begin{aligned}
&\mathbb{E}^c_\theta\left[z^{(\ell+1)}_{i_1}(x_1), z^{(\ell+1)}_{i_2}(x_2), z^{(\ell+1)}_{i_3}(x_3), z^{(\ell+1)}_{i_4}(x_4)\right] \\
&= \frac{1}{n_\ell}\left(V^{(\ell+1)}_{1234}\,\delta_{i_1 i_2}\delta_{i_3 i_4} + V^{(\ell+1)}_{1324}\,\delta_{i_1 i_3}\delta_{i_2 i_4} + V^{(\ell+1)}_{1423}\,\delta_{i_1 i_4}\delta_{i_2 i_3}\right)
\end{aligned}$$

Cumulants Decompose into Tensors

Use the nearly-Gaussian distribution to **compute all (mixed) cumulants** of z and $\widehat{\Theta}$ up to $\mathcal{O}(1/n)$.
A **pattern** emerges:

$$\mathbb{E}^c_\theta\left[z^{(\ell+1)}_{i_1}(x_1), z^{(\ell+1)}_{i_2}(x_2), z^{(\ell+1)}_{i_3}(x_3), z^{(\ell+1)}_{i_4}(x_4)\right]$$
$$= \frac{1}{n_\ell}\left(V^{(\ell+1)}_{1234}\,\delta_{i_1 i_2}\delta_{i_3 i_4} + V^{(\ell+1)}_{1324}\,\delta_{i_1 i_3}\delta_{i_2 i_4} + V^{(\ell+1)}_{1423}\,\delta_{i_1 i_4}\delta_{i_2 i_3}\right)$$

Cumulants Decompose into Tensors

Use the nearly-Gaussian distribution to **compute all (mixed) cumulants** of z and $\widehat{\Theta}$ up to $\mathcal{O}(1/n)$.

A **pattern** emerges:

$$\mathbb{E}^{c}_{\theta}\left[z^{(\ell+1)}_{i_1}(x_1), z^{(\ell+1)}_{i_2}(x_2), z^{(\ell+1)}_{i_3}(x_3), z^{(\ell+1)}_{i_4}(x_4)\right]$$
$$= \frac{1}{n_\ell}\left(V^{(\ell+1)}_{1234}\,\delta_{i_1 i_2}\delta_{i_3 i_4} + V^{(\ell+1)}_{1324}\,\delta_{i_1 i_3}\delta_{i_2 i_4} + V^{(\ell+1)}_{1423}\,\delta_{i_1 i_4}\delta_{i_2 i_3} \right)$$

Another mixed example: define $\widehat{\Delta\Theta}^{(\ell)}_{ij} = \widehat{\Theta}^{(\ell)}_{ij} - \mathbb{E}_{\theta}[\widehat{\Theta}^{(\ell)}_{ij}]$,

Cumulants Decompose into Tensors

Use the nearly-Gaussian distribution to **compute all (mixed) cumulants** of z and $\widehat{\Theta}$ up to $\mathcal{O}(1/n)$.
A **pattern** emerges:

$$\mathbb{E}^c_\theta\left[z^{(\ell+1)}_{i_1}(x_1), z^{(\ell+1)}_{i_2}(x_2), z^{(\ell+1)}_{i_3}(x_3), z^{(\ell+1)}_{i_4}(x_4)\right]$$
$$= \frac{1}{n_\ell}\left(V^{(\ell+1)}_{1234}\,\delta_{i_1 i_2}\delta_{i_3 i_4} + V^{(\ell+1)}_{1324}\,\delta_{i_1 i_3}\delta_{i_2 i_4} + V^{(\ell+1)}_{1423}\,\delta_{i_1 i_4}\delta_{i_2 i_3}\right)$$

Another mixed example: define $\widehat{\Delta\Theta}^{(\ell)}_{ij} = \widehat{\Theta}^{(\ell)}_{ij} - \mathbb{E}_\theta[\widehat{\Theta}^{(\ell)}_{ij}]$,

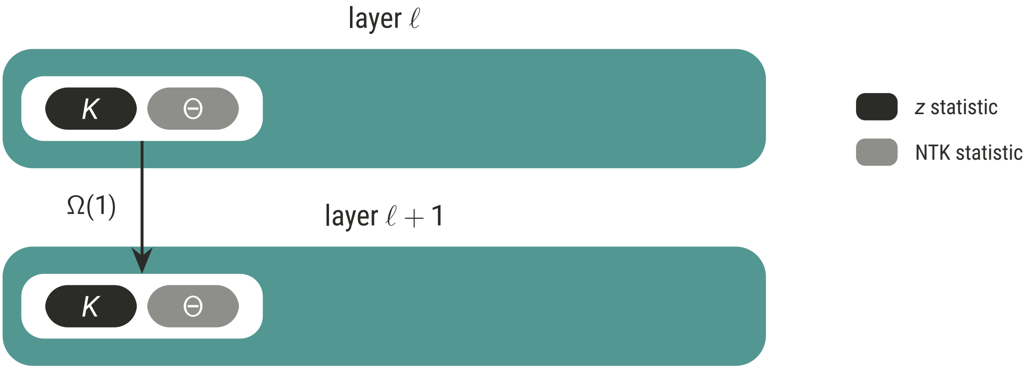
$$\mathbb{E}^c_\theta[z^{(\ell+1)}_{i_1}(x_1), z^{(\ell+1)}_{i_2}(x_2), \widehat{\Delta\Theta}^{(\ell+1)}_{i_3 i_4}(x_3, x_4)]$$
$$= \frac{1}{n_\ell}\left(D^{(\ell+1)}_{1234}\,\delta_{i_1 i_2}\delta_{i_3 i_4} + F^{(\ell+1)}_{1324}\,\delta_{i_1 i_3}\delta_{i_2 i_4} + F^{(\ell+1)}_{1423}\,\delta_{i_1 i_4}\delta_{i_2 i_3}\right)$$

Tensor Recursions

Obtain a **finite, closed set of tensor recursions** at $\mathcal{O}(1/n)$.

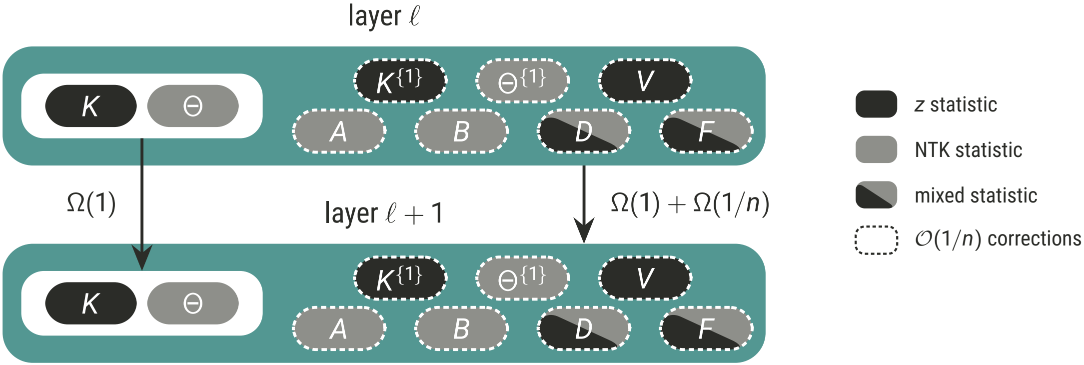
Tensor Recursions

Obtain a **finite, closed set of tensor recursions** at $\mathcal{O}(1/n)$.



Tensor Recursions

Obtain a **finite, closed set of tensor recursions** at $\mathcal{O}(1/n)$.



Tensor Recursions

Tensor Recursions

- Those tensors **fully determine the initialization statistics** at $\mathcal{O}(1/n)$.

Tensor Recursions

- Those tensors **fully determine the initialization statistics** at $\mathcal{O}(1/n)$.
- 6 more tensors needed to describe the **statistics of higher derivatives** relevant for training dynamics at $\mathcal{O}(1/n)$.

Tensor Recursions

- Those tensors **fully determine the initialization statistics** at $\mathcal{O}(1/n)$.
- 6 more tensors needed to describe the **statistics of higher derivatives** relevant for training dynamics at $\mathcal{O}(1/n)$.

Difficulty

Deriving the tensor recursions is tedious and error-prone.

Feynman diagrams

Feynman diagrams

- Used in quantum and statistical field theory to compute higher-order corrections to correlation functions.

Feynman diagrams

- Used in quantum and statistical field theory to compute higher-order corrections to correlation functions.

Central idea

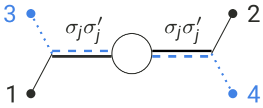
Replace algebraic manipulations with a set of diagrammatic rules

Feynman diagrams

- Used in quantum and statistical field theory to compute higher-order corrections to correlation functions.

Central idea
Replace algebraic manipulations with a set of diagrammatic rules

- Each connected diagram **represents a term**

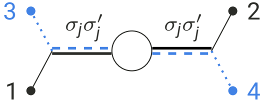


Feynman diagrams

- Used in quantum and statistical field theory to compute higher-order corrections to correlation functions.

Central idea
Replace algebraic manipulations with a set of diagrammatic rules

- Each connected diagram **represents a term**



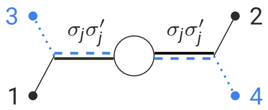
- Its individual components (**vertices, edges,** etc.) correspond to **factors** in the term

Feynman diagrams

- Used in quantum and statistical field theory to compute higher-order corrections to correlation functions.

Central idea
Replace algebraic manipulations with a set of diagrammatic rules

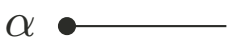
- Each connected diagram **represents a term**



- Its individual components (**vertices, edges,** etc.) correspond to **factors** in the term
- The sum of all diagrams gives the full correction

Translating the Problem into Diagrams

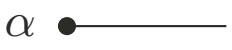
- Start with **external vertices**

$z_\alpha \equiv$ 

$\widehat{\Delta\Theta}_{\alpha\beta} \equiv$ 

Translating the Problem into Diagrams

- Start with **external vertices**

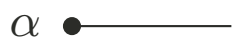
$$z_{\alpha} \equiv$$  $$\widehat{\Delta\Theta}_{\alpha\beta} \equiv$$ 

- Recall that the tensors are associated with cumulants

$$\mathbb{E}^{c}_{\theta}[z^{(\ell+1)}_{i_1;1}, z^{(\ell+1)}_{i_2;2}, \widehat{\Delta\Theta}^{(\ell+1)}_{i_3 i_4;34}]$$
$$= \frac{1}{n_\ell}\left(D^{(\ell+1)}_{1234}\,\delta_{i_1 i_2}\delta_{i_3 i_4} + F^{(\ell+1)}_{1324}\,\delta_{i_1 i_3}\delta_{i_2 i_4} + F^{(\ell+1)}_{1423}\,\delta_{i_1 i_4}\delta_{i_2 i_3} \right)$$

Translating the Problem into Diagrams

- Start with **external vertices**

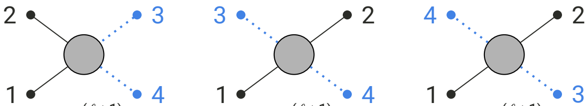
$z_\alpha \equiv$ 

$\widehat{\Delta\Theta}_{\alpha\beta} \equiv$ 

- Recall that the tensors are associated with cumulants

$$\mathbb{E}^c_\theta[z^{(\ell+1)}_{i_1;1}, z^{(\ell+1)}_{i_2;2}, \widehat{\Delta\Theta}^{(\ell+1)}_{i_3 i_4;34}]$$
$$= \frac{1}{n_\ell}\left(D^{(\ell+1)}_{1234}\,\delta_{i_1 i_2}\delta_{i_3 i_4} + F^{(\ell+1)}_{1324}\,\delta_{i_1 i_3}\delta_{i_2 i_4} + F^{(\ell+1)}_{1423}\,\delta_{i_1 i_4}\delta_{i_2 i_3} \right)$$

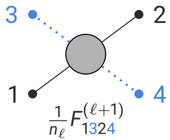
- This defines the external vertices for each tensor



$\frac{1}{n_\ell} D^{(\ell+1)}_{1234}$ $\frac{1}{n_\ell} F^{(\ell+1)}_{1324}$ $\frac{1}{n_\ell} F^{(\ell+1)}_{1423}$

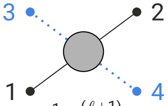
Deriving the F-Tensor Recursions

The goal: Replace the grey blob with other building blocks to derive a recursion for F.



Deriving the F-Tensor Recursions

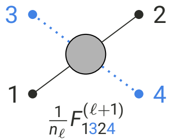
The goal: Replace the grey blob with other building blocks to derive a recursion for F.



$\frac{1}{n_\ell} F^{(\ell+1)}_{1324}$

Can choose from three categories

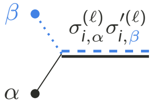
Deriving the F-Tensor Recursions



The goal: Replace the grey blob with other building blocks to derive a recursion for F.

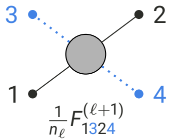
Can choose from three categories

Cubic vertices



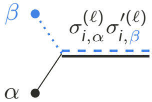
Deriving the F-Tensor Recursions

The goal: Replace the grey blob with other building blocks to derive a recursion for F.



Can choose from three categories

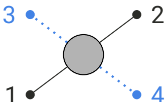
Cubic vertices



Propagator

Deriving the F-Tensor Recursions

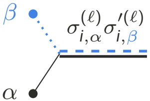
The goal: Replace the grey blob with other building blocks to derive a recursion for F.



$\frac{1}{n_\ell} F^{(\ell+1)}_{1324}$

Can choose from three categories

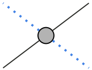
Cubic vertices



$\sigma^{(\ell)}_{i,\alpha} \sigma'^{(\ell)}_{i,\beta}$

Propagator

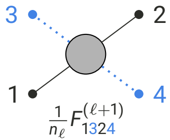
Internal quartic vertices (tensor from previous layer)



$\frac{1}{n_{\ell-1}} F^{(\ell)}_4$

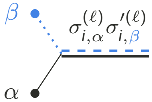
Deriving the F-Tensor Recursions

The goal: Replace the grey blob with other building blocks to derive a recursion for F.



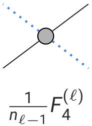
Can choose from three categories

Cubic vertices



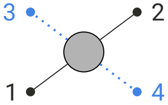
Propagator

Internal quartic vertices (tensor from previous layer)

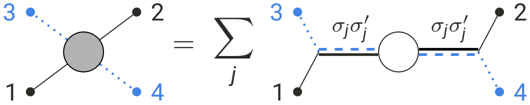


The propagator  connects only to internal lines and is restricted by **selection rules**.

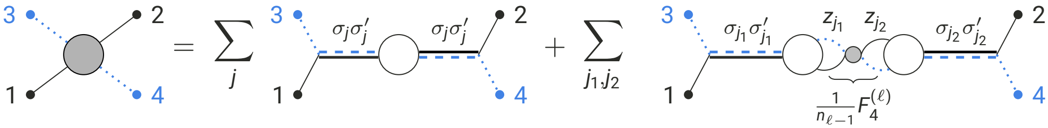
Deriving the F-Tensor Recursions



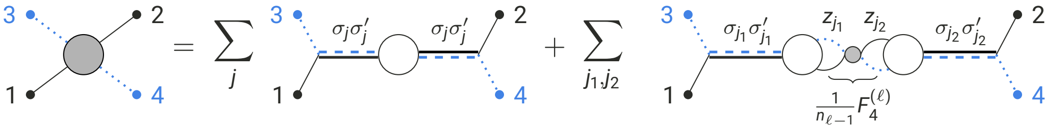
Deriving the F-Tensor Recursions



Deriving the F-Tensor Recursions



Deriving the F-Tensor Recursions



This translates to

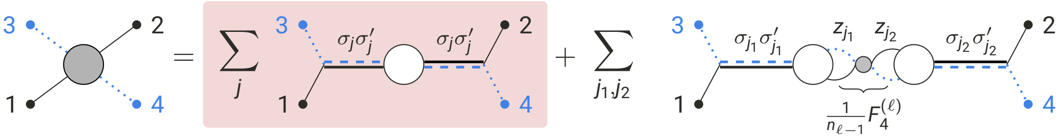
Gaussian expectation

$$F^{(\ell+1)}_{1324} = C_W^2 \langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_3'^{(\ell)} \sigma_4'^{(\ell)} \rangle_{K^{(\ell)}} \Theta^{(\ell)}_{34}$$

$$+ \frac{n_\ell}{n_{\ell-1}} C_W^2 \sum_{\alpha,\beta,\gamma,\delta=1}^{4} \langle \sigma_1^{(\ell)} \sigma_3'^{(\ell)} z_\alpha^{(\ell)} \rangle_{K^{(\ell)}} \langle \sigma_2^{(\ell)} \sigma_4'^{(\ell)} z_\beta^{(\ell)} \rangle_{K^{(\ell)}} K^{\alpha\gamma}_{(\ell)} K^{\beta\delta}_{(\ell)} F^{(\ell)}_{\gamma 3 \delta 4}$$

$$+ \mathcal{O}(1/n)$$

Deriving the F-Tensor Recursions

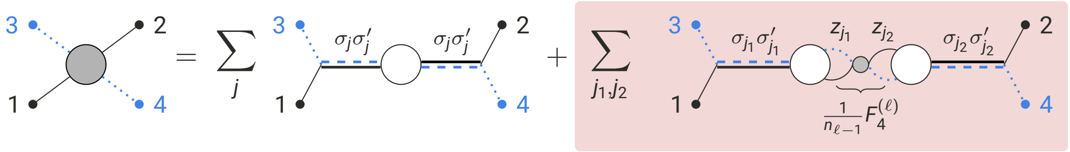


This translates to

Gaussian expectation

$$
\begin{aligned}
F^{(\ell+1)}_{1324} = {} & C_W^2 \langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_3'^{(\ell)} \sigma_4'^{(\ell)} \rangle_{K^{(\ell)}} \Theta^{(\ell)}_{34} \\
& + \frac{n_\ell}{n_{\ell-1}} C_W^2 \sum_{\alpha,\beta,\gamma,\delta=1}^{4} \langle \sigma_1^{(\ell)} \sigma_3'^{(\ell)} z_\alpha^{(\ell)} \rangle_{K^{(\ell)}} \langle \sigma_2^{(\ell)} \sigma_4'^{(\ell)} z_\beta^{(\ell)} \rangle_{K^{(\ell)}} K^{\alpha\gamma}_{(\ell)} K^{\beta\delta}_{(\ell)} F^{(\ell)}_{\gamma 3 \delta 4} \\
& + \mathcal{O}\left(1/n\right)
\end{aligned}
$$

Deriving the F-Tensor Recursions



This translates to

Gaussian expectation

$$F^{(\ell+1)}_{1324} = C_W^2 \langle \sigma_1^{(\ell)}\sigma_2^{(\ell)}\sigma_3^{\prime(\ell)}\sigma_4^{\prime(\ell)}\rangle_{K^{(\ell)}}\Theta^{(\ell)}_{34}$$

$$+ \frac{n_\ell}{n_{\ell-1}} C_W^2 \sum_{\alpha,\beta,\gamma,\delta=1}^{4} \langle \sigma_1^{(\ell)}\sigma_3^{\prime(\ell)} z_\alpha^{(\ell)}\rangle_{K^{(\ell)}} \langle \sigma_2^{(\ell)}\sigma_4^{\prime(\ell)} z_\beta^{(\ell)}\rangle_{K^{(\ell)}} K^{\alpha\gamma}_{(\ell)} K^{\beta\delta}_{(\ell)} F^{(\ell)}_{\gamma 3\delta 4}$$

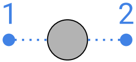
$$+ \mathcal{O}(1/n)$$

NTK Mean Corrections

Using the Feynman diagram approach, we can derive the $\mathcal{O}(1/n)$ corrections to the NTK mean

NTK Mean Corrections

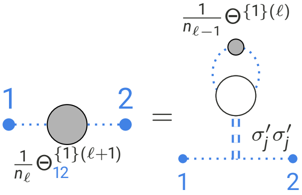
Using the Feynman diagram approach, we can derive the $\mathcal{O}(1/n)$ corrections to the NTK mean



$$\frac{1}{n_\ell}\Theta^{\{1\}(\ell+1)}_{12}$$

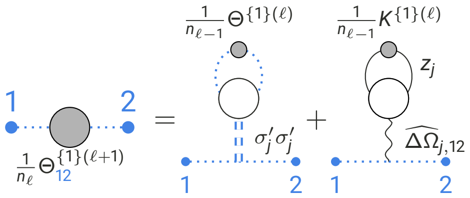
NTK Mean Corrections

Using the Feynman diagram approach, we can derive the $\mathcal{O}(1/n)$ corrections to the NTK mean



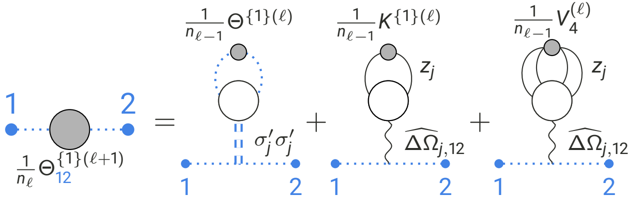
NTK Mean Corrections

Using the Feynman diagram approach, we can derive the $\mathcal{O}(1/n)$ corrections to the NTK mean



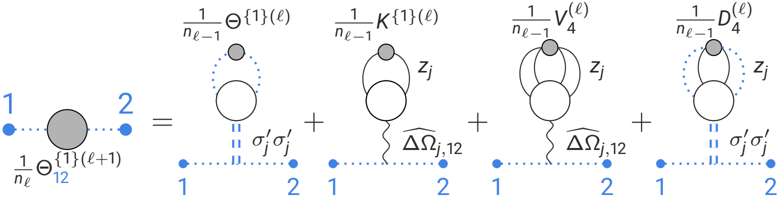
NTK Mean Corrections

Using the Feynman diagram approach, we can derive the $\mathcal{O}(1/n)$ corrections to the NTK mean



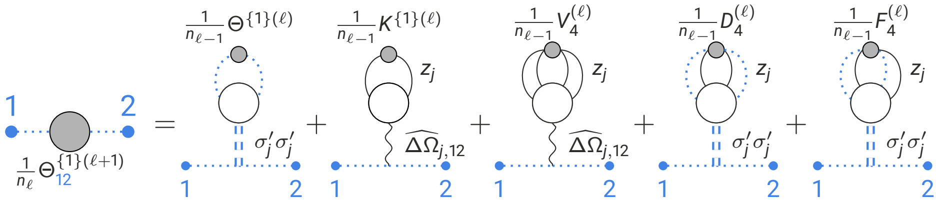
NTK Mean Corrections

Using the Feynman diagram approach, we can derive the $\mathcal{O}(1/n)$ corrections to the NTK mean

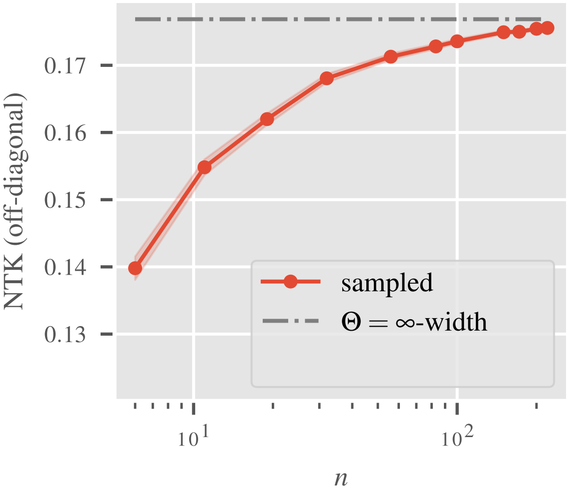


NTK Mean Corrections

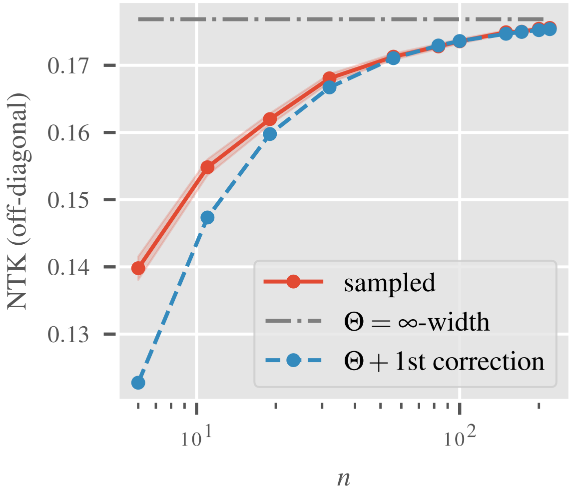
Using the Feynman diagram approach, we can derive the $\mathcal{O}(1/n)$ corrections to the NTK mean



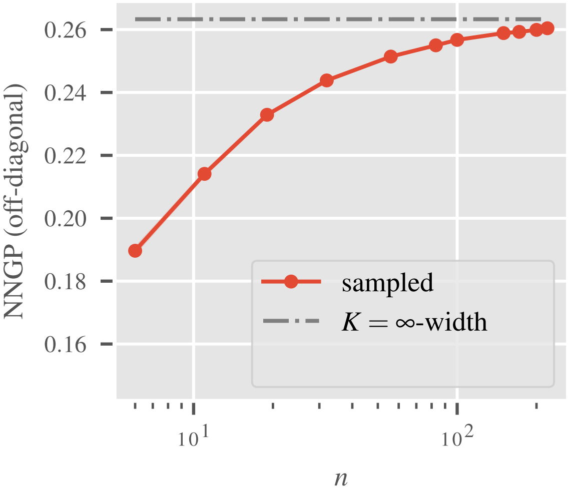
Numerical NTK Mean



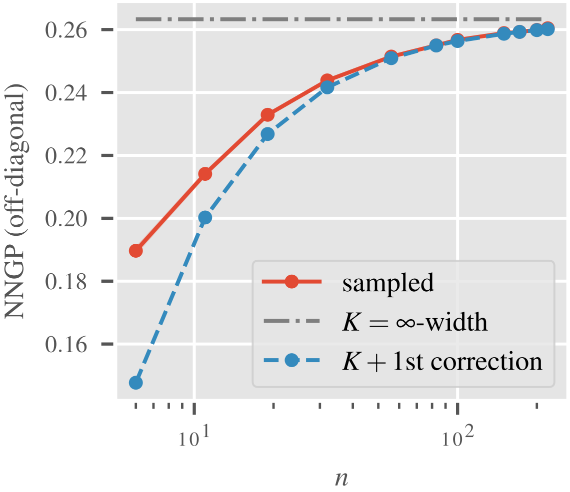
Numerical NTK Mean



Numerical NNGP Mean



Numerical NNGP Mean



Applications

Applications

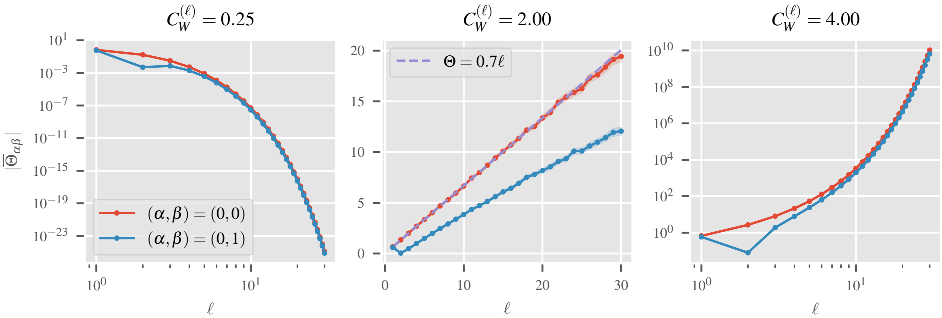
Gradient Stability

If the NNGP is **critical**, then so is any cumulant involving the NTK

Applications

Gradient Stability

If the NNGP is **critical**, then so is any cumulant involving the NTK



Applications

Definition [scale-invariant activation functions]: $\sigma(\lambda x) = \lambda\sigma(x)$. E.g. ReLU

Applications

Definition [scale-invariant activation functions]: $\sigma(\lambda x) = \lambda\sigma(x)$. E.g. ReLU

NTK diagonal of scale-invariant activations

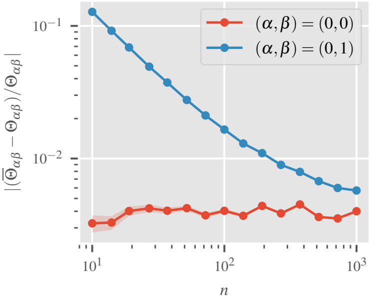
For scale-invariant activation functions, the **diagonal of the NTK is exact** does not receive corrections.

Applications

Definition [scale-invariant activation functions]: $\sigma(\lambda x) = \lambda\sigma(x)$. E.g. ReLU

NTK diagonal of scale-invariant activations

For scale-invariant activation functions, the **diagonal of the NTK is exact** does not receive corrections.



NTK-Unlimited Package

NTK-Unlimited Package

- Code to compute $\mathcal{O}(1/n)$ **recursions** and **sampling** of finite-width tensors available on GitHub

NTK-Unlimited Package

- Code to compute $\mathcal{O}(1/n)$ **recursions** and **sampling** of finite-width tensors available on GitHub
- Recursions implemented with custom **SymPy** classes $\Rightarrow$ easy extension to more recursions

NTK-Unlimited Package

- Code to compute $\mathcal{O}(1/n)$ **recursions** and **sampling** of finite-width tensors available on GitHub
- Recursions implemented with custom **SymPy** classes $\Rightarrow$ easy extension to more recursions
- Sampling heavily parallelized in **JAX**

Takeaways

Takeaways

- Finite-width corrections to the NTK limit **recover feature learning**

Takeaways

- Finite-width corrections to the NTK limit **recover feature learning**
- Feynman diagrams **provide a shortcut** to derive the $\mathcal{O}(1/n)$ corrections to the preactivation-NTK statistics

Takeaways

- Finite-width corrections to the NTK limit **recover feature learning**
- Feynman diagrams **provide a shortcut** to derive the $\mathcal{O}(1/n)$ corrections to the preactivation-NTK statistics
- First-order corrections already in **close agreement already at low width** of $n \gtrsim 20$

Takeaways

- Finite-width corrections to the NTK limit **recover feature learning**
- Feynman diagrams **provide a shortcut** to derive the $\mathcal{O}(1/n)$ corrections to the preactivation-NTK statistics
- First-order corrections already in **close agreement already at low width** of $n \gtrsim 20$

Outlook

Takeaways

- Finite-width corrections to the NTK limit **recover feature learning**
- Feynman diagrams **provide a shortcut** to derive the $\mathcal{O}(1/n)$ corrections to the preactivation-NTK statistics
- First-order corrections already in **close agreement already at low width** of $n \gtrsim 20$

Outlook

- Implement higher-derivative tensors as well to **predict training at finite width**

Takeaways

- Finite-width corrections to the NTK limit **recover feature learning**
- Feynman diagrams **provide a shortcut** to derive the $\mathcal{O}(1/n)$ corrections to the preactivation-NTK statistics
- First-order corrections already in **close agreement already at low width** of $n \gtrsim 20$

Outlook

- Implement higher-derivative tensors as well to **predict training at finite width**
- **Other initialization schemes**, e.g. orthogonal weights *(Guillen and Gerken 2026)*

Takeaways

- Finite-width corrections to the NTK limit **recover feature learning**
- Feynman diagrams **provide a shortcut** to derive the $\mathcal{O}(1/n)$ corrections to the preactivation-NTK statistics
- First-order corrections already in **close agreement already at low width** of $n \gtrsim 20$

Outlook

- Implement higher-derivative tensors as well to **predict training at finite width**
- **Other initialization schemes**, e.g. orthogonal weights *(Guillen and Gerken 2026)*
- Extension to other **architectures**, e.g. (G)CNNs *(Misof, Kessel, and Gerken 2025)*

Takeaways

- Finite-width corrections to the NTK limit **recover feature learning**
- Feynman diagrams **provide a shortcut** to derive the $\mathcal{O}(1/n)$ corrections to the preactivation-NTK statistics
- First-order corrections already in **close agreement already at low width** of $n \gtrsim 20$

Outlook

- Implement higher-derivative tensors as well to **predict training at finite width**
- **Other initialization schemes**, e.g. orthogonal weights *(Guillen and Gerken 2026)*
- Extension to other **architectures**, e.g. (G)CNNs *(Misof, Kessel, and Gerken 2025)*
- Influence of **Deep Kernel Shaping** *(He et al. 2022)*