

Finite-Width Neural Tangent Kernels from Feynman Diagrams

by Philipp Misof

Department of Mathematical Sciences, Division of Algebra and Geometry

August 18, 2026



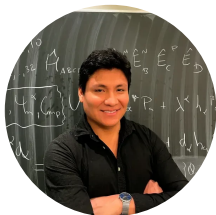
UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY

WASP | WALLENBERG AI
AUTONOMOUS SYSTEMS
AND SOFTWARE PROGRAM

Joint work with



Max Guillen



Jan E. Gerken

presented at the



Training Dynamics under Gradient Flow

Training Dynamics under Gradient Flow

$$\frac{d\theta_\mu(t)}{dt} = -\eta \frac{\partial \mathcal{L}}{\partial \theta_\mu} = -\eta \sum_{i=1}^N \frac{\partial f_\theta(\mathbf{x}_i)}{\partial \theta_\mu} \frac{\partial \ell(f_\theta(\mathbf{x}_i), y_i)}{\partial f_\theta(\mathbf{x}_i)}$$

Training Dynamics under Gradient Flow

parameters learning rate loss training data

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

network output

Training Dynamics under Gradient Flow

parameters learning rate loss training data

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

network output

by chain rule

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

Training Dynamics under Gradient Flow

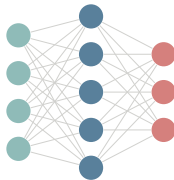
$$\frac{d\theta_\mu(t)}{dt} = -\underset{\text{learning rate}}{\eta} \frac{\partial \mathcal{L}}{\partial \theta_\mu} = -\eta \sum_{i=1}^N \frac{\partial f_\theta(x_i)}{\partial \theta_\mu} \frac{\partial \ell(\underset{\text{network output}}{f_\theta(x_i)}, \underset{\text{training data}}{y_i})}{\partial f_\theta(x_i)}$$

by chain rule

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

empirical Neural Tangent Kernel

Training Dynamics at Infinite-Width



Training Dynamics at Infinite-Width

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



(Jacot, Gabriel, and Hongler 2018)

Training Dynamics at Infinite-Width

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

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

frozen, deterministic NTK



(Jacot, Gabriel, and Hongler 2018)

Training Dynamics at Infinite-Width

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

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

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 $n \rightarrow \infty$

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

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

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

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

Loss of Feature Learning

- Implicitly changed parameterization

$$z_i^{(\ell)}(x) = \frac{1}{\sqrt{n_{\ell-1}}} \sum_{j=1}^{n_{\ell-1}} w_{ij}^{(\ell)} \sigma(z_j^{(\ell-1)}(x)) + b_i^{(\ell)}, \quad w_{ij}^{(\ell)} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{C}_w^{(\ell)})$$

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

Loss of Feature Learning

- Implicitly changed parameterization

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

- Change in activations during training

$$\mathbf{a}^{(\ell)}(x) := \sigma\left(\mathbf{z}^{(\ell)}(x)\right), \quad \Delta \mathbf{a}^{(\ell)}(x; t) := \mathbf{a}_t^{(\ell)}(x) - \mathbf{a}_0^{(\ell)}(x)$$

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

Loss of Feature Learning

- Implicitly changed parameterization

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

- Change in activations during training

$$\mathbf{a}^{(\ell)}(x) := \sigma(z^{(\ell)}(x)), \quad \Delta \mathbf{a}^{(\ell)}(x; t) := \mathbf{a}_t^{(\ell)}(x) - \mathbf{a}_0^{(\ell)}(x)$$

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

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

Different solutions to **recover feature learning**

Different solutions to **recover feature learning**

Changing the parameterization

⇒ E.g. Maximal update parameterization
(μ P)

Different solutions to **recover feature learning**

Changing the parameterization

⇒ E.g. Maximal update parameterization
(μ P)

Standard parameterization blows up

Different solutions to **recover feature learning**

Changing the parameterization

⇒ 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

⇒ 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

⇒ E.g. Maximal update parameterization
(μ P)

Standard parameterization blows up

Dichotomy

No kernel regime allows feature learning

Finite-width corrections

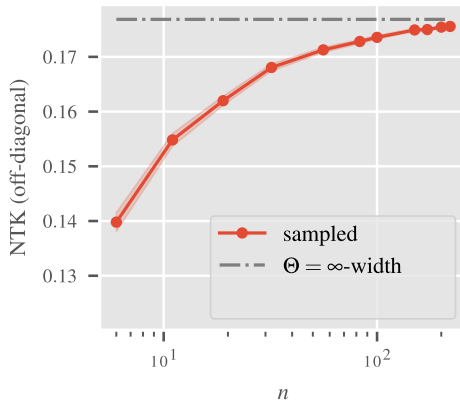
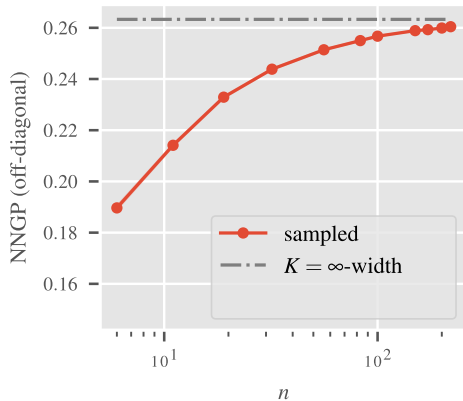
Extend NTK theory to capture finite-width effects

Advantage

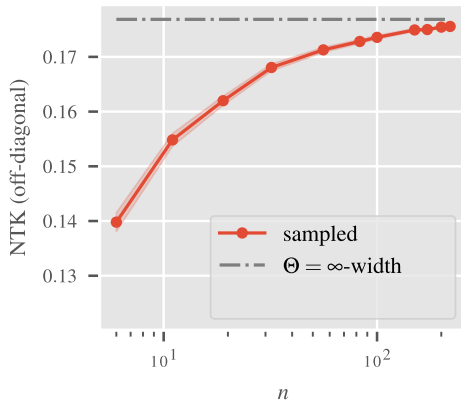
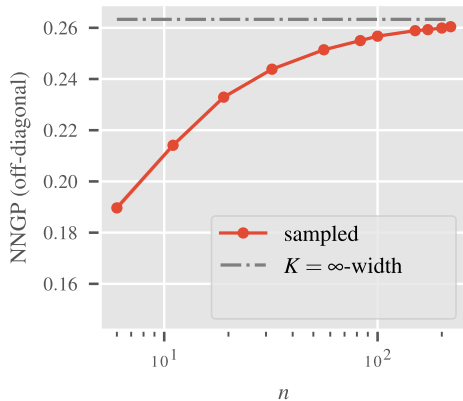
Converges to the well-understood limit
as $n \rightarrow \infty$

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

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

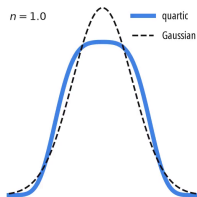
1/n-Corrections to the NTK Limit

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

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

1/n-Corrections to the NTK Limit

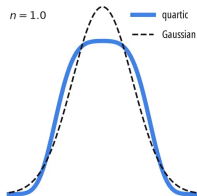
- When $n \rightarrow \infty$, z follows a **normal distribution** $\mathcal{N}(0, K)$.
- Make a **nearly-Gaussian ansatz** for finite n .



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

1/n-Corrections to the NTK Limit

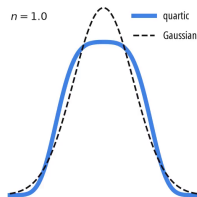
- When $n \rightarrow \infty$, z follows a **normal distribution** $\mathcal{N}(0, K)$.
- Make a **nearly-Gaussian ansatz** for finite n .



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

1/n-Corrections to the NTK Limit

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

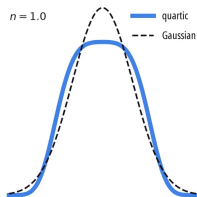


$$p(z) \propto \exp \left(-\frac{1}{2} \sum_{\alpha_1, \alpha_2 \in \mathcal{D}} g^{\alpha_1 \alpha_2} \sum_{i=1}^n z_{i; \alpha_1} z_{i; \alpha_2} + \frac{1}{8} \sum_{\alpha_1, \dots, \alpha_4 \in \mathcal{D}} v^{(\alpha_1 \alpha_2)(\alpha_3 \alpha_4)} \sum_{i_1, i_2=1}^n z_{i_1; \alpha_1} z_{i_1; \alpha_2}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

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

1/n-Corrections to the NTK Limit

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



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

- 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 $\hat{\Theta}$ up to $\mathcal{O}(1/n)$.

Cumulants Decompose into Tensors

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

A **pattern** emerges:

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

Cumulants Decompose into Tensors

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

A **pattern** emerges:

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

Cumulants Decompose into Tensors

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

A **pattern** emerges:

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

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

Cumulants Decompose into Tensors

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

A **pattern** emerges:

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

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

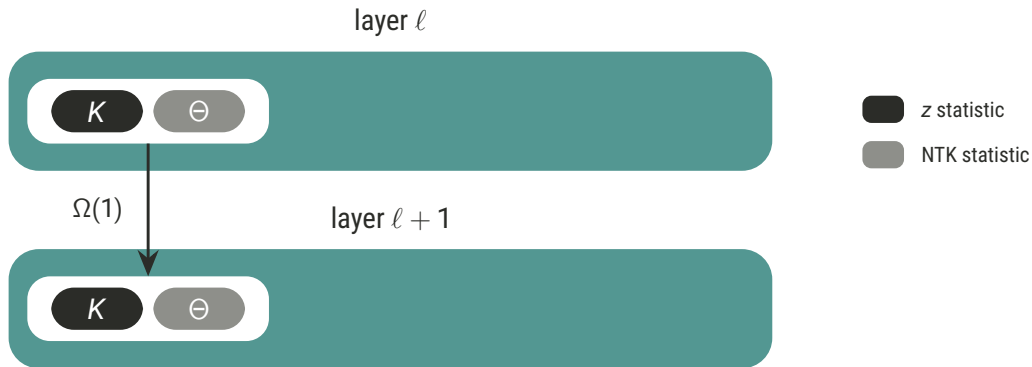
$$\begin{aligned} \mathbb{E}_{\theta}^c & \left[z_{i_1}^{(\ell+1)}(x_1), z_{i_2}^{(\ell+1)}(x_2), \widehat{\Delta\Theta}_{i_3 i_4}^{(\ell+1)}(x_3, x_4) \right] \\ &= \frac{1}{n_{\ell}} \left(D_{12\mathbf{3}4}^{(\ell+1)} \delta_{i_1 i_2} \delta_{i_3 i_4} + F_{1324}^{(\ell+1)} \delta_{i_1 i_3} \delta_{i_2 i_4} + F_{142\mathbf{3}}^{(\ell+1)} \delta_{i_1 i_4} \delta_{i_2 i_3} \right) \end{aligned}$$

Tensor Recursions

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

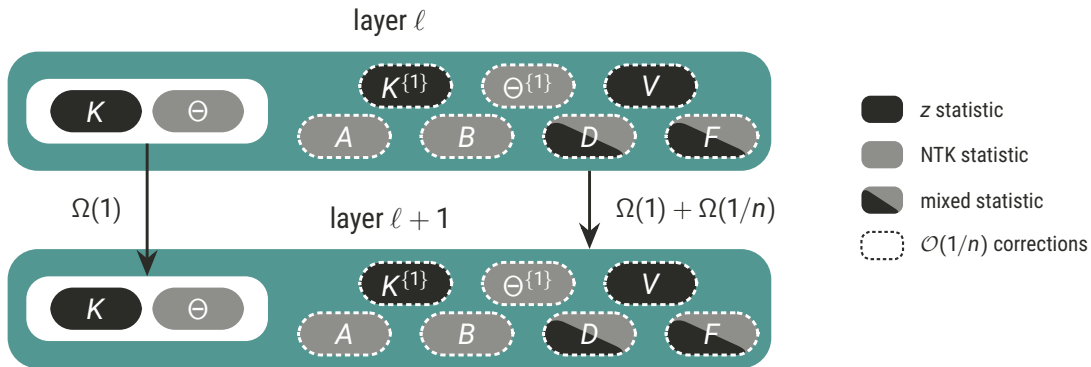
Tensor Recursions

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

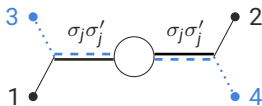
Feynman diagrams

- 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**



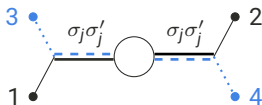
Feynman diagrams

- 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

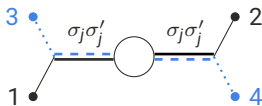
Feynman diagrams

- 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**

$$Z_\alpha \equiv \alpha \bullet \text{---} \qquad \widehat{\Delta\Theta}_{\alpha\beta} \equiv \begin{array}{c} \beta \bullet \text{.....} \\ \alpha \bullet \text{.....} \end{array}$$

Translating the Problem into Diagrams

- Start with **external vertices**

$$Z_\alpha \equiv \alpha \bullet \text{---} \quad \widehat{\Delta\Theta}_{\alpha\beta} \equiv \begin{array}{c} \beta \bullet \text{---} \\ \alpha \bullet \text{---} \end{array}$$

- Recall that the tensors are associated with cumulants

$$\begin{aligned} & \mathbb{E}_\theta^c[z_{i_1;1}^{(\ell+1)}, z_{i_2;2}^{(\ell+1)}, \widehat{\Delta\Theta}_{i_3 i_4; \mathbf{34}}^{(\ell+1)}] \\ &= \frac{1}{n_\ell} \left(D_{12\mathbf{34}}^{(\ell+1)} \delta_{i_1 i_2} \delta_{i_3 i_4} + F_{1\mathbf{324}}^{(\ell+1)} \delta_{i_1 i_3} \delta_{i_2 i_4} + F_{1\mathbf{423}}^{(\ell+1)} \delta_{i_1 i_4} \delta_{i_2 i_3} \right) \end{aligned}$$

Translating the Problem into Diagrams

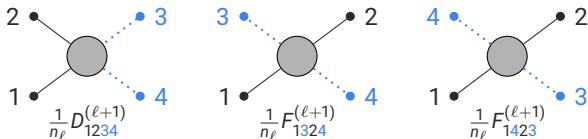
- Start with **external vertices**

$$Z_\alpha \equiv \alpha \bullet \text{---} \quad \widehat{\Delta\Theta}_{\alpha\beta} \equiv \begin{matrix} \beta \bullet \text{---} \\ \alpha \bullet \text{---} \end{matrix}$$

- Recall that the tensors are associated with cumulants

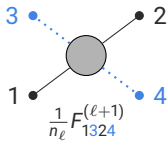
$$\begin{aligned} & \mathbb{E}_\theta^c[z_{i_1;1}^{(\ell+1)}, z_{i_2;2}^{(\ell+1)}, \widehat{\Delta\Theta}_{i_3 i_4; \mathbf{34}}^{(\ell+1)}] \\ &= \frac{1}{n_\ell} \left(D_{12\mathbf{34}}^{(\ell+1)} \delta_{i_1 i_2} \delta_{i_3 i_4} + F_{1\mathbf{324}}^{(\ell+1)} \delta_{i_1 i_3} \delta_{i_2 i_4} + F_{1\mathbf{423}}^{(\ell+1)} \delta_{i_1 i_4} \delta_{i_2 i_3} \right) \end{aligned}$$

- This defines the external vertices for each tensor



Deriving the F -Tensor Recursions

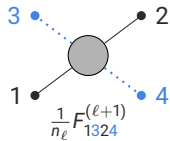
The goal: Replace the grey blob with other building blocks to derive a recursion for F .



Deriving the F -Tensor Recursions

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

Can choose from three categories

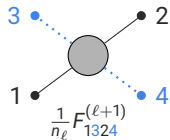
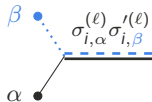


Deriving the F -Tensor Recursions

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

Can choose from three categories

Cubic vertices

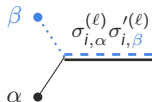


Deriving the F -Tensor Recursions

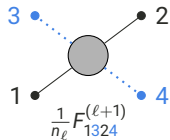
The goal: Replace the grey blob with other building blocks to derive a recursion for F .

Can choose from three categories

Cubic vertices



Propagator

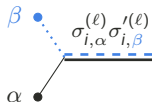


Deriving the F -Tensor Recursions

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

Can choose from three categories

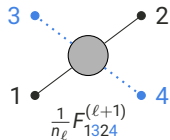
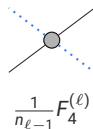
Cubic vertices



Propagator

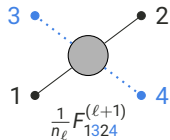


Internal quartic vertices
(tensor from previous layer)



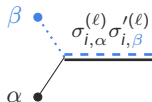
Deriving the F -Tensor Recursions

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



Can choose from three categories

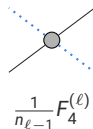
Cubic vertices



Propagator

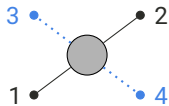


Internal quartic vertices
(tensor from previous layer)



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

Deriving the F -Tensor Recursions



Deriving the F -Tensor Recursions

The diagram illustrates a recursion for the F -tensor. On the left, a gray circle (the F -tensor) has four external legs: a solid black line labeled 1 (bottom-left), a solid black line labeled 2 (top-right), a dotted blue line labeled 3 (top-left), and a dotted blue line labeled 4 (bottom-right). This is equal to a summation over j of a diagram where a white circle is connected to the gray circle via two horizontal lines. The left horizontal line is solid black and labeled $\sigma_j \sigma'_j$, and the right horizontal line is dotted blue and labeled $\sigma_j \sigma'_j$. The white circle has four external legs: a solid black line labeled 1 (bottom-left), a dotted blue line labeled 3 (top-left), a solid black line labeled 2 (top-right), and a dotted blue line labeled 4 (bottom-right).

$$\text{Diagram 1} = \sum_j \text{Diagram 2}$$

Deriving the F -Tensor Recursions

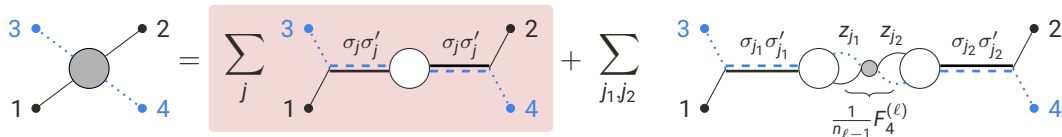
$$\begin{aligned}
 & \text{Diagram with grey circle and legs } 1, 2, 3, 4 \\
 &= \sum_j \text{Diagram with white circle and legs } 1, 2, 3, 4 \text{ and index } j \\
 &+ \sum_{j_1, j_2} \text{Diagram with two white circles and a central grey circle, labeled } \frac{1}{n_{l-1}} F_4^{(l)}
 \end{aligned}$$

This translates to

Gaussian expectation

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

Deriving the F -Tensor Recursions



This translates to

Gaussian expectation

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

Deriving the F -Tensor Recursions

$$\begin{aligned}
 & \text{Diagram with grey circle and legs 1, 2, 3, 4} = \sum_j \text{Diagram with white circle and legs 1, 2, 3, 4} + \sum_{j_1, j_2} \text{Diagram with two white circles and a grey circle} \\
 & \text{The grey circle in the second term is labeled } \frac{1}{n_{l-1}} F_4^{(l)}
 \end{aligned}$$

This translates to

Gaussian expectation

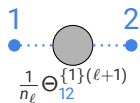
$$\begin{aligned}
 F_{1324}^{(\ell+1)} &= C_W^2 \langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_3'^{(\ell)} \sigma_4'^{(\ell)} \rangle_{K^{(\ell)}} \Theta_{34}^{(\ell)} \\
 &+ \frac{n_\ell}{n_{\ell-1}} C_W^2 \sum_{\alpha, \beta, \gamma, \delta=1}^4 \langle \sigma_1^{(\ell)} \sigma_3'^{(\ell)} z_\alpha^{(\ell)}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

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

The diagram shows an equality between two Feynman diagrams. On the left, a blue dot labeled '1' is connected by a dotted line to a grey circle, which is then connected by another dotted line to a blue dot labeled '2'. Below the grey circle is the label $\frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)}$. On the right, a blue dot labeled '1' is connected by a dotted line to a blue dot labeled '2'. Above this line is a white circle, which is connected to a grey dot. Above the grey dot is the label $\frac{1}{n_{\ell-1}} \Theta^{\{1\}(\ell)}$. Between the white circle and the dotted line connecting dots 1 and 2 are two vertical double lines labeled $\sigma'_j \sigma'_j$.

$$\frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)} = \frac{1}{n_{\ell-1}} \Theta^{\{1\}(\ell)} \sigma'_j \sigma'_j$$

NTK Mean Corrections

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

The diagram illustrates the decomposition of a two-point function into two diagrams representing $\mathcal{O}(1/n)$ corrections. On the left, a blue dot labeled '1' is connected by a dotted line to a grey circle, which is then connected by another dotted line to a blue dot labeled '2'. Below the grey circle is the label $\frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)}$. This is equal to the sum of two diagrams. The first diagram on the right has a blue dot '1' connected by a dotted line to a blue dot '2'. Above this line is a grey circle connected to the line by two vertical dotted lines, with the label $\sigma'_j \sigma'_j$ below it. Above the grey circle is a dotted loop, and above that is a grey dot with the label $\frac{1}{n_{\ell-1}} \Theta^{\{1\}(\ell)}$. The second diagram on the right also has a blue dot '1' connected by a dotted line to a blue dot '2'. Above this line is a grey circle connected to the line by a wavy line, with the label $\widehat{\Delta\Omega}_{j,12}$ below it. Above the grey circle is a solid loop, and above that is a grey dot with the label $\frac{1}{n_{\ell-1}} K^{\{1\}(\ell)}$ and a label z_j next to the loop.

$$\frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)} = \frac{1}{n_{\ell-1}} \Theta^{\{1\}(\ell)} \sigma'_j \sigma'_j + \frac{1}{n_{\ell-1}} K^{\{1\}(\ell)} z_j \widehat{\Delta\Omega}_{j,12}$$

NTK Mean Corrections

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

The diagram shows an equation where a two-point function on the left is equal to the sum of three diagrams on the right. Each diagram has two external blue dots labeled 1 and 2 connected by a horizontal dotted line.

- Left side:** A grey circle is connected to the horizontal line. Below it is the label $\frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)}$.
- First term on the right:** A grey circle is connected to the horizontal line via a vertical dotted line. Above it is a dashed loop with a grey dot. The label is $\frac{1}{n_{\ell-1}} \Theta^{\{1\}(\ell)}$. Below the vertical line are two small vertical dotted lines labeled $\sigma'_j \sigma'_j$.
- Second term on the right:** A grey circle is connected to the horizontal line via a wavy line. Above it is a loop with a grey dot. The label is $\frac{1}{n_{\ell-1}} K^{\{1\}(\ell)}$. Below the wavy line is a wavy line labeled $\widehat{\Delta\Omega}_{j,12}$.
- Third term on the right:** A grey circle is connected to the horizontal line via a wavy line. Above it is a loop with a grey dot. The label is $\frac{1}{n_{\ell-1}} V_4^{(\ell)}$. Below the wavy line is a wavy line labeled $\widehat{\Delta\Omega}_{j,12}$.

NTK Mean Corrections

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

The diagram illustrates the expansion of a NTK mean into four Feynman diagrams representing $\mathcal{O}(1/n)$ corrections. The left side shows a diagram with two external nodes labeled 1 and 2, connected by a dashed line. A grey circle is attached to this line, with a label $\frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)}$ below it. This is equal to the sum of four diagrams on the right:

- Diagram 1: A grey circle with a dashed loop, labeled $\frac{1}{n_{\ell-1}} \Theta^{\{1\}(\ell)}$ above and $\sigma'_j \sigma'_j$ below.
- Diagram 2: A grey circle with a solid loop, labeled $\frac{1}{n_{\ell-1}} K^{\{1\}(\ell)}$ above and $\widehat{\Delta\Omega}_{j,12}$ below.
- Diagram 3: A grey circle with a solid loop and a wavy line, labeled $\frac{1}{n_{\ell-1}} V_4^{(\ell)}$ above and $\widehat{\Delta\Omega}_{j,12}$ below.
- Diagram 4: A grey circle with a dashed loop and a wavy line, labeled $\frac{1}{n_{\ell-1}} D_4^{(\ell)}$ above and $\sigma'_j \sigma'_j$ below.

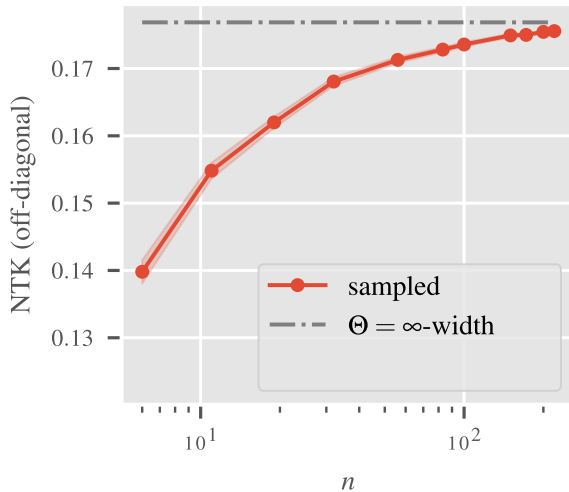
NTK Mean Corrections

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

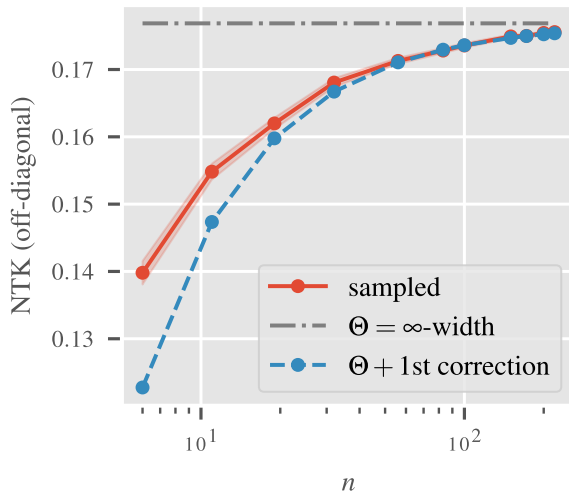
The diagram illustrates the expansion of a two-point function into five terms representing different types of corrections. The left side shows a two-point function with external indices 1 and 2, a grey vertex, and a label $\frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)}$. The right side is a sum of five terms, each with a coefficient and a diagram:

- Term 1: Coefficient $\frac{1}{n_{\ell-1}} \Theta^{\{1\}(\ell)}$. Diagram: A grey vertex with a dashed blue loop, connected to a white vertex, which is connected to a wavy line labeled $\widehat{\Delta\Omega}_{j,12}$.
- Term 2: Coefficient $\frac{1}{n_{\ell-1}} K^{\{1\}(\ell)}$. Diagram: A grey vertex with a solid black loop, connected to a white vertex, which is connected to a wavy line labeled $\widehat{\Delta\Omega}_{j,12}$.
- Term 3: Coefficient $\frac{1}{n_{\ell-1}} V_4^{(\ell)}$. Diagram: A grey vertex with two solid black loops, connected to a white vertex, which is connected to a wavy line labeled $\widehat{\Delta\Omega}_{j,12}$.
- Term 4: Coefficient $\frac{1}{n_{\ell-1}} D_4^{(\ell)}$. Diagram: A grey vertex with a dashed blue loop and a solid black loop, connected to a white vertex, which is connected to a wavy line labeled $\widehat{\Delta\Omega}_{j,12}$.
- Term 5: Coefficient $\frac{1}{n_{\ell-1}} F_4^{(\ell)}$. Diagram: A grey vertex with a dashed blue loop and a solid black loop, connected to a white vertex, which is connected to a wavy line labeled $\widehat{\Delta\Omega}_{j,12}$.

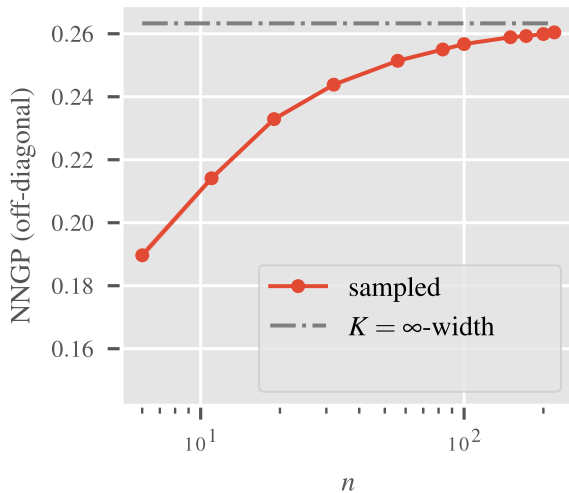
Numerical NTK Mean



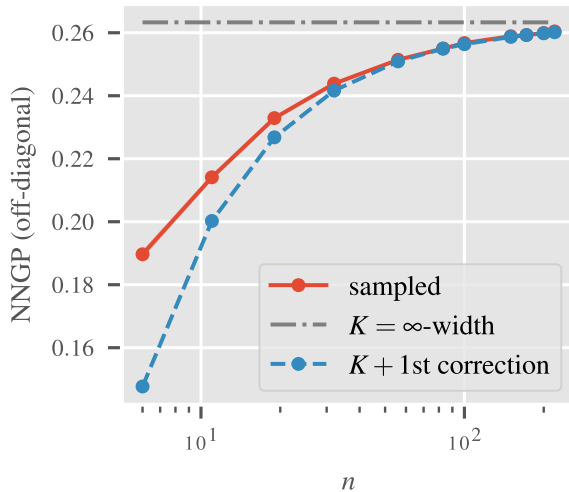
Numerical NTK Mean



Numerical NNGP Mean



Numerical NNGP Mean



Applications

Applications

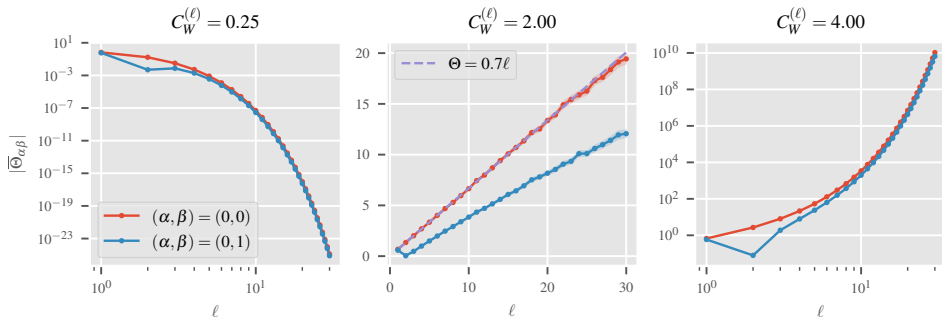
Gradient Stability

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

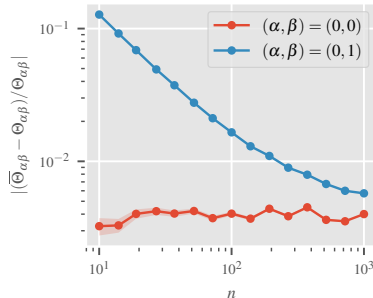
Definition [scale-invariant activation functions]: $\sigma(\lambda x) = \lambda \sigma(x)$. E.g. ReLU

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*)



arXiv:2508.11522