Infinite Width, NTK, and Feature Learning

Student Meeting 1

Shreyas Kalvankar

September 17, 2025

TU Delft

Agenda

Goal of this talk

- Introduce the infinite-width lens, NTK, and the lazy training regime via a simple one-layer example.
- Situate these ideas within related work: baseline results at infinite width and viewpoints that go beyond linearization.

What I would like from you

- Conceptual clarifications and critiques of how I'm connecting the papers.
- Suggestions for key references I might be missing.
- Pointers to alternative frameworks worth comparing (mean-field, functional-analytic, optimization bias).

Why theory?

- Larger models keep improving, but we don't fully know *why*.

Why theory?

- Larger models keep improving, but we don't fully know *why*.
- **A solvable starting point:** infinite width gives a clean baseline; modern models are large-but-finite, deep, and do feature learning.

Why theory?

- Larger models keep improving, but we don't fully know *why*.
- **A solvable starting point:** infinite width gives a clean baseline; modern models are large-but-finite, deep, and do feature learning.
- **What can theory help in:** explanations of convergence/generalization, signals for when features move, and guidance on design choices (init, learning rate, normalization).

Motivation

- **Infinite width:** clean, analyzable baseline; randomness averages to a deterministic kernel.

Motivation

- **Infinite width:** clean, analyzable baseline; randomness averages to a deterministic kernel.
- **NTK:** first-order (linear) view of training; clear convergence intuition via a fixed kernel.

Motivation

- **Infinite width:** clean, analyzable baseline; randomness averages to a deterministic kernel.
- **NTK:** first-order (linear) view of training; clear convergence intuition via a fixed kernel.
- **Beyond the linear NTK picture:** when does it stop being accurate?
  - Does the *kernel* change during training?

Motivation

- **Infinite width:** clean, analyzable baseline; randomness averages to a deterministic kernel.
- **NTK:** first-order (linear) view of training; clear convergence intuition via a fixed kernel.
- **Beyond the linear NTK picture:** when does it stop being accurate?
  - Does the *kernel* change during training?
  - Do the model's internal features move?

Motivation

- **Infinite width:** clean, analyzable baseline; randomness averages to a deterministic kernel.
- **NTK:** first-order (linear) view of training; clear convergence intuition via a fixed kernel.
- **Beyond the linear NTK picture:** when does it stop being accurate?
  - Does the *kernel* change during training?
  - Do the model's internal features move?
  - Do training curves deviate from the NTK baseline prediction?

Motivation

- **Infinite width:** clean, analyzable baseline; randomness averages to a deterministic kernel.
- **NTK:** first-order (linear) view of training; clear convergence intuition via a fixed kernel.
- **Beyond the linear NTK picture:** when does it stop being accurate?
  - Does the *kernel* change during training?
  - Do the model's internal features move?
  - Do training curves deviate from the NTK baseline prediction?
  - Possible causes: higher-order effects, useful parameter re-organization that push the model out of the lazy regime.

Setup

Supervised learning: data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathbb{R}^d$.

Setup

Supervised learning: data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathbb{R}^d$.

Neural network $f : \mathbb{R}^d \times \Theta \to \mathbb{R}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$.

Setup

Supervised learning: data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathbb{R}^d$.

Neural network $f : \mathbb{R}^d \times \Theta \to \mathbb{R}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$.

Trained by (continuous-time) gradient flow on squared loss:

$$L(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{n} (f(\boldsymbol{x}_i; \boldsymbol{\theta}) - y_i)^2,$$

Setup

Supervised learning: data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathbb{R}^d$.

Neural network $f : \mathbb{R}^d \times \Theta \to \mathbb{R}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$.

Trained by (continuous-time) gradient flow on squared loss:

$$L(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{n} (f(\boldsymbol{x}_i; \boldsymbol{\theta}) - y_i)^2, \qquad \frac{d}{dt}\boldsymbol{\theta}_t = -\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_t).$$

Linearization at Initialization

First-order Taylor around $\boldsymbol{\theta}_0$:

$$f(\boldsymbol{x};\boldsymbol{\theta}) \approx f(\boldsymbol{x};\boldsymbol{\theta}_0) + \underbrace{\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)^\top}_{:=\ \phi(\boldsymbol{x})^\top}(\boldsymbol{\theta} - \boldsymbol{\theta}_0).$$

Linearization at Initialization

First-order Taylor around $\boldsymbol{\theta}_0$:

$$f(\boldsymbol{x};\boldsymbol{\theta}) \approx f(\boldsymbol{x};\boldsymbol{\theta}_0) + \underbrace{\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)^\top}_{:=\ \phi(\boldsymbol{x})^\top}(\boldsymbol{\theta} - \boldsymbol{\theta}_0).$$

This induces the **Neural Tangent Kernel (NTK)** at $\boldsymbol{\theta}_0$:

$$K(\boldsymbol{x},\boldsymbol{x}') = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}') = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)^\top \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}';\boldsymbol{\theta}_0).$$

Ref: (Jacot et al., 2018)

Illustrative Model (One Hidden Layer, No Biases)

Consider a m-width neural network

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \, \sigma(\mathbf{w}_r^\top \boldsymbol{x}),$$

Illustrative Model (One Hidden Layer, No Biases)

Consider a m-width neural network

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \, \sigma(\mathbf{w}_r^\top \boldsymbol{x}), \quad a_r \sim \mathcal{N}(0, \sigma_a^2), \; \mathbf{w}_r \sim \mathcal{N}\Big(0, \tfrac{\sigma_w^2}{d} I\Big).$$

Illustrative Model (One Hidden Layer, No Biases)

Consider a m-width neural network

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \, \sigma(\mathbf{w}_r^\top \boldsymbol{x}), \quad a_r \sim \mathcal{N}(0, \sigma_a^2), \; \mathbf{w}_r \sim \mathcal{N}\Big(0, \tfrac{\sigma_w^2}{d} I\Big).$$

Gradients:

$$\nabla_{a_r} f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sigma(\mathbf{w}_r^\top \boldsymbol{x}),$$

Illustrative Model (One Hidden Layer, No Biases)

Consider a m-width neural network

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \, \sigma(\mathbf{w}_r^\top \boldsymbol{x}), \quad a_r \sim \mathcal{N}(0, \sigma_a^2), \; \mathbf{w}_r \sim \mathcal{N}\Big(0, \tfrac{\sigma_w^2}{d} I\Big).$$

Gradients:

$$\nabla_{a_r} f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sigma(\mathbf{w}_r^\top \boldsymbol{x}), \quad \nabla_{\mathbf{w}_r} f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \, a_r \, \sigma'(\mathbf{w}_r^\top \boldsymbol{x}) \, \boldsymbol{x}.$$

Illustrative Model (One Hidden Layer, No Biases)

Consider a m-width neural network

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \, \sigma(\mathbf{w}_r^\top \boldsymbol{x}), \quad a_r \sim \mathcal{N}(0, \sigma_a^2), \; \mathbf{w}_r \sim \mathcal{N}\Big(0, \tfrac{\sigma_w^2}{d} I\Big).$$

Gradients:

$$\nabla_{a_r} f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sigma(\mathbf{w}_r^\top \boldsymbol{x}), \quad \nabla_{\mathbf{w}_r} f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \, a_r \, \sigma'(\mathbf{w}_r^\top \boldsymbol{x}) \, \boldsymbol{x}.$$

Finite-width NTK:

$$\boxed{K_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m} \sum_{r=1}^{m} \sigma(\mathbf{w}_r^\top \boldsymbol{x}) \, \sigma(\mathbf{w}_r^\top \boldsymbol{x}') + \frac{1}{m} \sum_{r=1}^{m} a_r^2 \, \sigma'(\mathbf{w}_r^\top \boldsymbol{x}) \, \sigma'(\mathbf{w}_r^\top \boldsymbol{x}') \, \boldsymbol{x}^\top \boldsymbol{x}'.}$$

Infinite-Width NTK

The two sums are empirical averages of i.i.d. terms. Since a_r and w_r are independent with finite moments, the (strong) law of large numbers gives, almost surely, as width $m \to \infty$:

Infinite-Width NTK

The two sums are empirical averages of i.i.d. terms. Since a_r and w_r are independent with finite moments, the (strong) law of large numbers gives, almost surely, as width $m \to \infty$:

$$\frac{1}{m}\sum_{r=1}^{m}\sigma(w_r^\top x)\,\sigma(w_r^\top x') \;\longrightarrow\; \mathbb{E}_w\big[\sigma(w^\top x)\,\sigma(w^\top x')\big],$$

Infinite-Width NTK

The two sums are empirical averages of i.i.d. terms. Since a_r and w_r are independent with finite moments, the (strong) law of large numbers gives, almost surely, as width $m \to \infty$:

$$\frac{1}{m}\sum_{r=1}^{m} \sigma(w_r^\top x)\,\sigma(w_r^\top x') \;\longrightarrow\; \mathbb{E}_w\big[\sigma(w^\top x)\,\sigma(w^\top x')\big],$$

$$\frac{1}{m}\sum_{r=1}^{m} a_r^2\,\sigma'(w_r^\top x)\,\sigma'(w_r^\top x') \;\longrightarrow\; \sigma_a^2\,\mathbb{E}_w\big[\sigma'(w^\top x)\,\sigma'(w^\top x')\big].$$

Infinite-Width NTK

The two sums are empirical averages of i.i.d. terms. Since a_r and w_r are independent with finite moments, the (strong) law of large numbers gives, almost surely, as width $m \to \infty$:

$$\frac{1}{m}\sum_{r=1}^{m} \sigma(w_r^\top x)\,\sigma(w_r^\top x') \;\longrightarrow\; \mathbb{E}_w\big[\sigma(w^\top x)\,\sigma(w^\top x')\big],$$

$$\frac{1}{m}\sum_{r=1}^{m} a_r^2\,\sigma'(w_r^\top x)\,\sigma'(w_r^\top x') \;\longrightarrow\; \sigma_a^2\,\mathbb{E}_w\big[\sigma'(w^\top x)\,\sigma'(w^\top x')\big].$$

Thus, in the infinite-width limit, the empirical NTK converges almost surely to a deterministic kernel

$$\boxed{K_\infty(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}_{\mathbf{w}}\big[\sigma(\mathbf{w}^\top\boldsymbol{x})\sigma(\mathbf{w}^\top\boldsymbol{x}')\big] + \sigma_a^2\,\boldsymbol{x}^\top\boldsymbol{x}'\,\mathbb{E}_{\mathbf{w}}\big[\sigma'(\mathbf{w}^\top\boldsymbol{x})\sigma'(\mathbf{w}^\top\boldsymbol{x}')\big].}$$

Infinite-Width NTK

The two sums are empirical averages of i.i.d. terms. Since a_r and w_r are independent with finite moments, the (strong) law of large numbers gives, almost surely, as width $m \to \infty$:

$$\frac{1}{m}\sum_{r=1}^{m} \sigma(w_r^\top x)\,\sigma(w_r^\top x') \;\longrightarrow\; \mathbb{E}_w\big[\sigma(w^\top x)\,\sigma(w^\top x')\big],$$

$$\frac{1}{m}\sum_{r=1}^{m} a_r^2\,\sigma'(w_r^\top x)\,\sigma'(w_r^\top x') \;\longrightarrow\; \sigma_a^2\,\mathbb{E}_w\big[\sigma'(w^\top x)\,\sigma'(w^\top x')\big].$$

Thus, in the infinite-width limit, the empirical NTK converges almost surely to a deterministic kernel

$$\boxed{K_\infty(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\mathbf{w}}\big[\sigma(\mathbf{w}^\top \boldsymbol{x})\sigma(\mathbf{w}^\top \boldsymbol{x}')\big] + \sigma_a^2\,\boldsymbol{x}^\top \boldsymbol{x}'\,\mathbb{E}_{\mathbf{w}}\big[\sigma'(\mathbf{w}^\top \boldsymbol{x})\sigma'(\mathbf{w}^\top \boldsymbol{x}')\big].}$$

Key point: in the wide limit, $K_t \approx K_0$ remains *essentially constant* during training (lazy regime).

Refs: (Neal, 1996; Lee et al., 2019)

Training dynamics: mini-derivation

Gradient flow & chain rule: Let $f_t(x_i) := f(x_i; \theta_t)$. Then

$$\frac{d}{dt} f_t(x_i) = \nabla_\theta f(x_i; \theta_t)^\top \dot{\theta}_t = -\nabla_\theta f(x_i; \theta_t)^\top \nabla_\theta L(\theta_t).$$

Training dynamics: mini-derivation

Gradient flow & chain rule: Let $f_t(x_i) := f(x_i;\theta_t)$. Then

$$\frac{d}{dt} f_t(x_i) = \nabla_\theta f(x_i;\theta_t)^\top \dot{\theta}_t = -\,\nabla_\theta f(x_i;\theta_t)^\top \nabla_\theta L(\theta_t).$$

Loss gradient (squared loss):

$$L(\theta) = \tfrac{1}{2}\sum_{j=1}^{n} \big(f(x_j;\theta) - y_j\big)^2, \qquad \nabla_\theta L(\theta_t) = \sum_{j=1}^{n} \big(f_t(x_j) - y_j\big)\, \nabla_\theta f(x_j;\theta_t).$$

Training dynamics: constant kernel

Substitute and define the (time–dependent) NTK.

$$\frac{d}{dt} f_t(x_i) = -\sum_{j=1}^{n} \underbrace{\nabla_\theta f(x_i;\theta_t)^\top \nabla_\theta f(x_j;\theta_t)}_{=:\ K_t(x_i,x_j)} \big(f_t(x_j) - y_j\big).$$

Training dynamics: constant kernel

Substitute and define the (time–dependent) NTK.

$$\frac{d}{dt} f_t(x_i) = -\sum_{j=1}^{n} \underbrace{\nabla_\theta f(x_i;\theta_t)^\top \nabla_\theta f(x_j;\theta_t)}_{=:\ K_t(x_i,x_j)} \big(f_t(x_j) - y_j\big).$$

Vectorized form:

$$\boxed{\dot{f}_t \ = \ -K_t\,(f_t - y).}$$

Training dynamics: constant kernel

Substitute and define the (time–dependent) NTK.

$$\frac{d}{dt} f_t(x_i) = -\sum_{j=1}^{n} \underbrace{\nabla_\theta f(x_i;\theta_t)^\top \nabla_\theta f(x_j;\theta_t)}_{=:\ K_t(x_i,x_j)} \big(f_t(x_j) - y_j\big).$$

Vectorized form:

$$\boxed{\dot{f}_t \ = \ -K_t\,(f_t - y).}$$

Constant-kernel (NTK) regime. If $K_t \approx K_0 \equiv K$ (infinite width / lazy),

$$\dot{f}_t = -K(f_t - y) \quad \Rightarrow \quad \boxed{f_t = y + e^{-Kt}\,(f_0 - y).}$$

Refs: (Jacot et al., 2018; Lee et al., 2019)

Lazy Training

Lazy regime: parameter drift is small, $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\| \ll \|\boldsymbol{\theta}_0\|$.

- Features $\phi(\boldsymbol{x})$ and kernel K stay (approximately) constant.
- Training reduces to kernel regression with fixed K.

Limitation: suppresses *feature learning* (representation change).

Aim: quantify *when* lazy holds/breaks and *how* to model beyond it.

Diagnosing the transition to feature learning

Some signals to consider

- **Kernel drift:** Does the NTK matrix K_t change during training? Compare K_t to K_0 on the (same) data. Bigger change $\Rightarrow$ more feature learning.

Diagnosing the transition to feature learning

Some signals to consider

- **Kernel drift:** Does the NTK matrix K_t change during training? Compare K_t to K_0 on the (same) data. Bigger change $\Rightarrow$ more feature learning.
- **Feature drift:** Do the tangent features $\phi_t(x) = \nabla_\theta f(x;\theta_t)$ move on a small fixed set $\mathcal{S}$? Track the average change from $t = 0$.

Diagnosing the transition to feature learning

Some signals to consider

- **Kernel drift:** Does the NTK matrix K_t change during training? Compare K_t to K_0 on the (same) data. Bigger change $\Rightarrow$ more feature learning.
- **Feature drift:** Do the tangent features $\phi_t(x) = \nabla_\theta f(x; \theta_t)$ move on a small fixed set $\mathcal{S}$? Track the average change from $t = 0$.

How to (often) push out of lazy

- larger learning rate
- smaller width
- more depth / biases / normalization

Beyond linearization I: quadratic / higher-order

$$f(\boldsymbol{x};\boldsymbol{\theta}) \approx f(\boldsymbol{x};\boldsymbol{\theta}_0) + \phi(\boldsymbol{x})^\top \Delta\boldsymbol{\theta} + \tfrac{1}{2}\,\Delta\boldsymbol{\theta}^\top H_f(\boldsymbol{x})\,\Delta\boldsymbol{\theta}, \qquad \Delta\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}_0.$$

- **Mechanism (Bai and Lee (2020)):** construct regimes where the linear term is suppressed so the *quadratic* term governs the dynamics; extendable to $k > 2$ (“higher-order NTKs”).
- **Findings:** with the linear term suppressed, progress comes from *feature changes*; this adaptive regime is easy to optimize and can beat NTK in sample use on simple tasks.

See: “Beyond Linearization: On Quadratic and Higher-Order Approximation of Wide Neural Networks.”

Beyond linearization II: adaptive / time-varying kernels

- Zhang et al. (2024) replace the fixed NTK with a *time-varying* kernel K_t that evolves during training (“kernel drift”).
- Features adapt during training and increasingly align with label-relevant directions (growing alignment).
- They provide a prototype of an over-parameterized Gaussian sequence model to analyze feature learning beyond the NTK picture.

See: “Towards a Statistical Understanding of Neural Networks: Beyond the NTK Theories.”

Summary & discussion

- Infinite width as a clean baseline; NTK via linearization at initialization.
- Constant–kernel training dynamics: $\dot{f}_t = -K(f_t - y)$ with solution $f_t = y + e^{-Kt}(f_0 - y)$.
- Lazy training $\Rightarrow$ features (and K) stay essentially fixed.
- How to spot leaving lazy: *kernel drift* ($K_t \neq K_0$) and *feature drift* on a probe set.
- Beyond NTK in the literature:
  - *Quadratic / higher-order* near init (Bai and Lee, 2020).
  - *Adaptive / time-varying kernels* and alignment (Zhang et al., 2024).

References

Y. Bai and J. D. Lee. Beyond linearization: On quadratic and higher-order approximation of wide neural networks. In *International Conference on Learning Representations*, 2020.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, 2019.

R. M. Neal. *Priors for infinite networks*. PhD thesis, University of Toronto, 1996.

H. Zhang, J. Lai, Y. Li, Q. Lin, and J. S. Liu. Towards a statistical understanding of neural networks: Beyond the neural tangent kernel theories. *arXiv preprint arXiv:2412.18756*, 2024.