# Graph Neural Tangent Kernel: Convergence on Large Graphs

**Sanjukta Krishnagopal**[*1 2 3] **Luana Ruiz**[*3 4]

## Abstract

Graph neural networks (GNNs) achieve remarkable performance in graph machine learning tasks but can be hard to train on large-graph data, where their learning dynamics are not well understood. We investigate the training dynamics of large-graph GNNs using graph neural tangent kernels (GNTKs) and graphons. In the limit of large width, optimization of an overparametrized NN is equivalent to kernel regression on the NTK. Here, we investigate how the GNTK evolves as another independent dimension is varied: the graph size. We use graphons to define limit objects—graphon NNs for GNNs, and graphon NTKs for GNTKs—, and prove that, on a sequence of graphs, the GNTKs converge to the graphon NTK. We further prove that the spectrum of the GNTK, which is related to the directions of fastest learning which becomes relevant during early stopping, converges to the spectrum of the graphon NTK. This implies that in the large-graph limit, the GNTK fitted on a graph of moderate size can be used to solve the same task on the large graph, and to infer the learning dynamics of the large-graph GNN. These results are verified empirically on node regression and classification tasks.

## 1. Introduction

Several real-world systems such as social-interactions, brain-connectome, epidemic spread, recommender systems, and traffic patterns are best represented by structured data in the form of large graphs. Graph neural networks (GNNs) are deep neural network architectures that leverage these graph structures to learn meaningful representations of node and edge data (Kipf & Welling, 2017; Hamilton et al., 2017; Defferrard et al., 2016; Gama et al., 2018). GNNs have shown remarkable empirical performance in a number of graph machine learning tasks, but can be hard to train on large-graph data. Recent research efforts have attempted to understand the large-graph behavior of GNNs, and in particular why GNNs trained on small graphs scale well to large networks (Ruiz et al., 2020; Levie et al., 2021; Keriven et al., 2020). However, the specific learning dynamics of a GNN trained directly on the large network, which are known to be challenging, are not as well understood.

Modern deep neural networks (DNNs) are typically overparametrized. The benefits of overparametrization include faster convergence (Allen-Zhu et al., 2019; Arora et al., 2018) and better generalization (Cao & Gu, 2019), but on the other hand the parameters can be more difficult to interpret and the learning dynamics harder to understand. A remarkable contribution of Jacot et al. (2018) was the observation that, in the infinite width limit, learning the weights of a DNN via gradient descent reduces to kernel regression with a deterministic and fixed kernel called the *neural tangent kernel* (NTK), which captures the first-order approximation of the neural network's evolution during gradient descent (Lee et al., 2019). Since its introduction, the NTK has been an important and widely-studied tool in the machine learning toolbox, and kernel regression using the NTK has shown strong performance on small datasets (Arora et al., 2019b).

In the graph case, it is straightforward to define the NTK associated with a GNN, or the graph neural tangent kernel (GNTK) (Du et al., 2019). The GNTK allows studying the training dynamics of the GNN when the number of features (the analog of width in the DNN) is large. Nonetheless, the effect of *graph size* on the GNN learning dynamics, which can itself be thought of as another 'width dimension', is not well understood, and there is limited research that investigates this rigorously. Herein, we propose to to understand the training dynamics of GNNs that are wide in both of these senses by combining the NTK formalism with the theory of graphons (Lovász, 2012; Borgs et al., 2008).

A graphon is a symmetric bounded measurable function $\mathbf{W} : [0,1]^2 \to [0,1]$ representing the limit of a sequence of dense graphs. It can also be interpreted as a random graph model, in which case we can use the graphon to sample stochastic graphs. The interpretation of $\mathbf{W}$ as both a graph limit and a

[*] Equal contribution [1] Dept. of Electrical Engineering and Computer Science [2] Dept. of Mathematics, UCLA [3] Work done in part while visiting the Simons Institute for the Theory of Computing. [4] MIT CSAIL. Correspondence to: Sanjukta Krishnagopal <sanjukta@berkeley.edu>, Luana Ruiz <ruizl@mit.edu>.

*Proceedings of the $40^{th}$ International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

random graph model makes it so that each graphon defines a family of similar graphs. Hence, one can expect properties of a graphon to generalize, in a probabilistic sense, the properties of graphs belonging to its family. Graphons have been used to study the limit behavior of GNNs, which converge to so-called graphon neural networks (WNNs) (Ruiz et al., 2020). The fact that GNNs have a limit on the graphon implies that they are transferable across graphs in the same family, thus allowing a GNN to be trained on a graph of moderate size and transferred to a larger graph.

### 1.1. Contributions

In this paper, our first contribution is to define the graphon NTK (WNTK) associated with the WNN (Sec. 4). We then prove, using mathematical induction, that the GNTK converges to the WNTK (Thm. 5.1). In practice, this implies that GNTKs, like GNNs, are transferable across graphs of different sizes associated with the same graphon. That is to say, one can subsample a small graph and the corresponding data from a large graph, then fit the subsampled data to the small-graph GNTK via kernel regression, and then transfer the fitted model to the large-graph GNTK, which is particularly important for large graphs.

A more important implication of the convergence of the GNTK is that it is possible to understand the training dynamics of GNNs on large graphs by analyzing the behavior of the corresponding GNTK in graphs of moderate size. For instance, the eigenvalues of the NTK are associated with the speed of convergence along the corresponding eigendirections (Jacot et al., 2018). In Thm. 5.2, we show that the eigenvalues of the graph GNTK, which indicate the directions of fastest convergence of the GNN, converge to the eigenvalues of the WNTK. This allows these eigenvalues to be estimated from GNTKs associated with smaller graphs. Lastly, we verify our theoretical results in three numerical applications: prediction of opinion dynamics on random graphs, movie recommendation using the MovieLens dataset, and node classification on the Cora, CiteSeer and PubMed networks. We observe the convergence of the GNTK (Sec. 6.1), the effect of width in kernel regression and GNN training (Sec. 6.2), and the convergence of the GNTK eigenvalues on sequences of graphs (Sec. 6.3).

## 2. Related Work

**Neural tangent kernels.** The connection between infinitely wide DNNs and kernel methods (Gaussian processes) has been known since the 1990s (Neal & Neal, 1996; Williams, 1996), but a more theoretical formulation was presented by Jacot et al. (2018), which introduced the NTK and proved its constancy property in the infinite width limit, with Liu et al. (2020) later showing that constancy only holds for architectures with linear output layer. Several works have derived NTKs for a generalized classes of neural networks, including convolutional neural networks (Arora et al., 2019a; Li et al., 2019), ResNets (Huang et al., 2020) and, most closely related to our work, GNNs (Du et al., 2019).

**Graphons and size generalization in deep learning.** Graphons have been used to understand GNN convergence and transferability (Ruiz et al., 2020; 2021a; Maskey et al., 2023), to analyze the generalization properties of GNNs in large graphs (Maskey et al., 2022), and to propose more computationally efficient training algorithms for large-scale GNNs (Cerviño et al., 2023). More recently, Xia et al. (2022) proposed implicit graphon neural representations, which use neural networks to estimate graphons.

Yehudai et al. (2021) study graphs in which the local structure depends on the graph size—which is analogous to the dense graphs associated with graphons—, and find that GNNs are not guaranteed to scale to large graph sizes. However, their result foregoes the normalization of the adjacency matrix by the graph size, while we leverage this normalization to show operator norm convergence of the kernels associated with the graph convolution. It is also worth noting that our work is fundamentally different than GNN convergence or transferability results, because it relates not to the GNN architecture but to the GNN *learning dynamics*, and uses a mathematical induction proof as opposed to the spectral convergence argument typically used in GNN convergence proofs.

## 3. Graph and Graphon Neural Networks

Let $\mathbf{G}_n = (\mathcal{V}, \mathcal{E}, w)$ be a graph where $\mathcal{V}$, $|\mathcal{V}| = n$, is the set of nodes or vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges and $w : \mathcal{E} \to \mathbb{R}$ is a map assigning weights to the edges in $\mathcal{E}$. The *size-normalized* adjacency matrix of $\mathbf{G}_n$, denoted $\mathbf{A}_n \in \mathbb{R}^{n\times n}$, is given by $[\mathbf{A}]_{ij} = w(i,j)/n$. In this paper we focus on undirected graphs $\mathbf{G}_n$, with symmetric $\mathbf{A}_n$.

### 3.1. Graph Neural Networks

Supported on the graph $\mathbf{G}_n$ with $n$ nodes, we define node data $\mathbf{x}_n \in \mathbb{R}^n$—also called *graph signals* (Shuman et al., 2013)—where $[\mathbf{x}_n]_i$ is the value of the signal at node $i$. GNNs iteratively update each node's data by aggregating the data from its neighbors using *graph convolutions*. An order-$K$ graph convolution is defined as

$$\mathbf{y}_n = H(\mathbf{x}_n) = \sum_{k=0}^{K-1} h_k \mathbf{A}_n^k \mathbf{x}_n \tag{1}$$

where $h_0, \ldots, h_{K-1}$ are the convolution coefficients. When $K = 2$ and $\mathbf{A}_n$ is binary, (1) can be seen as an aggregation operation akin to the `AGGREGATE` operation in, e.g., (Xu et al., 2019; Hamilton et al., 2017), or as the message-passing operation in message-passing neural net-

*Table 1.* Description of variables.

| Variable | Description |
|---|---|
| $\mathbf{A}_n$ | Adjacency matrix of graph $\mathbf{G}_n$ |
| $\mathbf{x}_n$ | Graph signal on $\mathbf{G}_n$ |
| $\mathbf{W}_n$ | Graphon induced by $\mathbf{G}_n$ |
| $X_n$ | Graphon signal induced by $\mathbf{x}_n$ |
| $\mathbf{W}$ | Limiting graphon |
| $X$ | Limiting graphon signal |

works (MPNNs) (Gilmer et al., 2017).

More generally, let $\mathbf{X}_n \in \mathbb{R}^{n\times F}$ and $\mathbf{Y}_n \in \mathbb{R}^{n\times G}$ be $F$- and $G$-dimensional signals respectively, where the $f$th ($g$th) column is a *feature* $\mathbf{x}_n^f$ ($\mathbf{y}_n^g$). In this case, the graph convolution generalizes to

$$\mathbf{Y}_n = H(\mathbf{X}_n) = \sum_{k=0}^{K-1} \mathbf{A}_n^k \mathbf{X}_n \mathbf{H}_k \tag{2}$$

with weights $\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_{K-1} \in \mathbb{R}^{F\times G}$. Note that the number of parameters of the graph convolutions in (1) and (2), $K$ and $KFG$ respectively, are independent of $n$.

A GNN consists of $L$ layers, each of which composes a graph convolution and a nonlinear activation function. Explicitly, the $l$th layer of a GNN can be written as

$$\begin{aligned}\mathbf{X}_{l,n} &= \sigma\left(\mathbf{U}_{l,n}\right)\\ \mathbf{U}_{l,n} &= H(\mathbf{X}_{l-1,n}) = \sum_{k=0}^{K-1} \mathbf{A}_n^k \mathbf{X}_{l-1,n} \mathbf{H}_{l,k}\end{aligned} \tag{3}$$

where $\mathbf{X}_{l-1,n} \in \mathbb{R}^{n\times F_{l-1}}$ is the layer input, $\mathbf{X}_{l,n} \in \mathbb{R}^{n\times F_l}$ is the layer output, $\mathbf{H}_{l,k} \in \mathbb{R}^{F_{l-1}\times F_l}$ are the layer weights and $\sigma$ is a pointwise nonlinear activation function, i.e., $[\sigma(\mathbf{X})]_{ij} = \sigma([\mathbf{X}]_{ij})$. Typical choices for $\sigma$ include ReLU, tanh, and sigmoid. At the first layer of the GNN, the input $\mathbf{X}_{0,n}$ is the input data $\mathbf{X}$. Similarly, the GNN output $\mathbf{Y}_n$ is given by the last layer output $\mathbf{X}_{n,L}$.

In the following, we represent the entire GNN consisting of the concatenation of $L$ layers like (3) as the parametric map $\mathbf{Y}_n = f(\mathbf{X}_n; \mathbf{A}_n, \mathcal{H})$, where $\mathcal{H} = \{\mathbf{H}_{l,k}\}_{l,k}$ groups the learnable weights $\mathbf{H}_{l,k}$ at all layers. This more concise representation highlights the independence between the adjacency matrix $\mathbf{A}_n$ (i.e., the graph) and the parameters $\mathcal{H}$, which the GNN inherits from the graph convolution. We note that while we describe a general class of GNNs here, the methods in our paper can be directly adapted to other types of GNNs.

### 3.2. Graphon Neural Networks

Graphons are bounded, symmetric, measurable functions $\mathbf{W}: [0,1]^2 \to [0,1]$ representing limits of sequences of dense graphs (Lovász, 2012; Borgs et al., 2008). A graph sequence $\{\mathbf{G}_n\}$ converges to a graphon in the sense that the densities of homomorphisms of any finite, unweighted and undirected graph $\mathbf{F} = (\mathcal{V}', \mathcal{E}')$ into $\mathbf{G}_n$ converge to the densities of homomorphisms of $\mathbf{F}$ into $\mathbf{W}$. The graphs $\mathbf{F}$ can be thought of as motifs, such as triangles, $k$-cycles, $k$-cliques, etc. Explicitly, let hom$(\mathbf{F}, \mathbf{G}_n)$ denote the total number of homomorphisms between $\mathcal{V}'$ and $\mathcal{V}_n$. The density of such homomorphisms is given by $t(\mathbf{F}, \mathbf{G}_n) = \text{hom}(\mathbf{F}, \mathbf{G}_n)/n^{|\mathcal{V}'|}$ and we can similarly define $t(\mathbf{F}, \mathbf{W})$ (see (Borgs et al., 2008)). Then, we say that $\mathbf{G}_n \to \mathbf{W}$ if and only if

$$\lim_{n\to\infty} t(\mathbf{F}, \mathbf{G}_n) = t(\mathbf{F}, \mathbf{W}) \tag{4}$$

for all simple $\mathbf{F}$ (Lovász & Szegedy, 2006).

Alternatively, the graphon can also be seen as a generative model for stochastic (also called $\mathbf{W}$-random) graphs. Nodes are picked by sampling points $u_i$, $1 \leq i \leq n$, from the unit interval and connecting edges between nodes $i$ and $j$ with probability $\mathbf{W}(u_i, u_j)$. Importantly, sequences of stochastic graphs generated in this way converge almost surely to the graphon (Lovász, 2012)[Cor. 10.4].

The notion of graph signals is extended to graphons by defining graphon signals, which are functions $X: [0,1] \to \mathbb{R}$ (Ruiz et al., 2021a). We restrict attention to graphon signals with finite energy, i.e., $X \in L^2([0,1])$.

Analogously to (2), given $F$- and $G$-dimensional graphon signals $X: [0,1] \to \mathbb{R}^F$ and $Y: [0,1] \to \mathbb{R}^G$, the graphon convolution is defined as (Ruiz et al., 2021a)

$$\begin{aligned}Y = T_H X &= \sum_{k=0}^{K-1} T_W^{(k)} X \mathbf{H}_k\\ T_W^{(k)} X &= \int_0^1 \mathbf{W}(u,v) T_W^{(k-1)} X(u) du\end{aligned} \tag{5}$$

where $T_W^{(0)} = \mathbf{I}$ is the identity and the convolution weights are collected in the matrices $\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_{K-1} \in \mathbb{R}^{F\times G}$.

The extension of the GNN to graphon data is the graphon neural network (WNN). Akin to the GNN, the WNN is formed by $L$ layers each of which composes a graphon convolution and a nonlinear activation function. Explicitly, the $l$th layer of the WNN is given by

$$\begin{aligned}X_l &= \sigma\left(U_l\right)\\ U_l &= T_{H_l} X_{l-1} = \sum_{k=0}^{K-1} T_W^{(k)} X_{l-1} \mathbf{H}_{l,k}\end{aligned} \tag{6}$$

where $X_{l-1}: [0,1] \to \mathbb{R}^{F_{l-1}}$ is the layer input, $X_l: [0,1] \to \mathbb{R}^{F_l}$ is its output, $\mathbf{H}_{l,k} \in \mathbb{R}^{F_{l-1}\times F_l}$ are its weights and $\sigma$ is a pointwise nonlinearity (e.g., the ReLU). The input of the first layer of the WNN is $X_0 = X$, and the output of the WNN is the last layer ouput, i.e., $Y = X_L$.

We can describe the WNN more compactly as the map $Y = f(X; \mathbf{W}, \mathcal{H})$, with $\mathcal{H} = \{\mathbf{H}_{l,k}\}_{l,k}$ the set of learnable parameters at all layers. Note that, if the weights $\mathcal{H}$ are the same, the WNN map $f(X; \mathbf{W}, \mathcal{H})$ is the same as the GNN map $f(X; \mathbf{A}_n, \mathcal{H})$ with $\mathbf{A}_n$ swapped with $\mathbf{W}$. This is important because it implies that, similarly to how graphons are generative models for graphs, *WNNs are generative models for GNNs*. Indeed, we can use the WNN $f(X; \mathbf{W}, \mathcal{H})$ to sample the GNN $\mathbf{y}_n = f(\mathbf{x}_n; \mathbf{A}_n, \mathcal{H})$ with

$$\begin{aligned} [\mathbf{A}_n]_{ij} &\sim \text{Ber}(\mathbf{W}(u_i, u_j)) \\ [\mathbf{x}_n]_i &= X(u_i) \end{aligned} \tag{7}$$

where the $u_i$ are sampled uniformly and independently at random from $[0, 1]$ and Ber() is the Bernoulli distribution.

As $n \to \infty$, sequences of GNNs sampled from the WNN as described above converge to the WNN. In fact, as long as $\mathcal{H}$ is the same, any GNN $f(\mathbf{x}_n; \mathbf{A}_n, \mathcal{H})$ applied to a sequence $\{(\mathbf{G}_n, \mathbf{x}_n)\}$ converging to $(\mathbf{W}, X)$ converges to $f(X; \mathbf{W}, \mathcal{H})$ (Ruiz et al., 2021a). A more important result in practice is that this convergence implies that GNNs are transferable across graphs associated with the same graphon, i.e., they can be trained on graphs $\mathbf{G}_n$ and executed on graphs $\mathbf{G}_m$ with an error that decreases asymptotically with $n$ and $m$ (Ruiz et al., 2021b; 2020).

In Sec. 5, we prove a similar convergence result for the neural tangent kernels (NTKs) associated with $f(\mathbf{x}_n; \mathbf{A}_n, \mathcal{H})$ and $f(X; \mathbf{W}, \mathcal{H})$, but before doing so, we need to introduce *induced WNNs*. The WNN induced by the GNN $f(\mathbf{x}_n; \mathbf{A}_n, \mathcal{H})$ is defined as $Y_n = f(X_n; \mathbf{W}_n, \mathcal{H})$, with

$$\begin{aligned} \mathbf{W}_n(u, v) &= \sum_{i=1}^{n} \sum_{j=1}^{n} [\mathbf{A}_n]_{ij} \mathbb{I}(u \in I_i) \mathbb{I}(v \in I_j), \\ X_n(u) &= \sum_{i=1}^{n} [\mathbf{x}_n]_i \mathbb{I}(u \in I_i) \end{aligned} \tag{8}$$

and where $\mathbb{I}$ is the indicator function, $I_i = [(i-1)/n, i/n)$ for $1 \leq i \leq n-1$, and $I_n = [(n-1)/n, 1]$. The graphon $\mathbf{W}_n$ is *induced by the graph* $\mathbf{G}_n$, and the graphon signals $X_n$ and $Y_n$ are *induced by the graph signals* $\mathbf{x}_n$ and $\mathbf{y}_n$. A succinct description of all graph and graphon variables is provided in Table 1.

# 4. Graph and Graphon Neural Tangent Kernel

Consider a general, fully-connected neural network $f(\mathbf{x}; \mathcal{H})$, with layers given by $\mathbf{x}_l = \sigma(\mathbf{H}_l \mathbf{x}_{l-1})$, input $\mathbf{x}_0 = \mathbf{x} \in \mathbb{R}^{d_0}$ and learnable parameters $\mathcal{H} = \{\mathbf{H}_l\}_l \in \mathbb{R}^{d_{l-1} \times d_l}$, $[\mathbf{H}_l]_{pq} = h_{l,pq}$. For a training set $\{\mathbf{x}_i, \tilde{\mathbf{y}}_i\}_{i=1}^{M}$, assume that the loss to be minimized is the mean squared error (MSE) or quadratic loss

$$\min_{\mathcal{H}} \ell(\mathcal{H}) = \min_{\mathcal{H}} \sum_{i=1}^{M} (f(\mathbf{x}_i; \mathcal{H}) - \tilde{\mathbf{y}}_i)^2. \tag{9}$$

As the training progresses, the output of the neural network for input $\mathbf{x}_i$ is updated as $\mathbf{y}_i(t) = f(\mathbf{x}_i; \mathcal{H}(t))$. As such, the weight update rule is given by $\frac{\partial \mathcal{H}}{\partial t} = -\nabla \ell(\mathcal{H}(t))$ and so the output evolves as

$$\begin{aligned} \frac{\partial \mathbf{y}_i(t)}{\partial t} &= \textstyle\sum_j \Theta(\mathbf{x}_i, \mathbf{x}_j; \mathcal{H}(t))(f(\mathbf{x}_i; \mathcal{H}(t)) - \hat{\mathbf{y}}_i), \\ \Theta(\mathbf{x}_i, \mathbf{x}_j; \mathcal{H}) &= \sum_{l,p,q} \frac{\partial f(\mathbf{x}_i; \mathcal{H})}{\partial h_{l,pq}} \frac{\partial f(\mathbf{x}_j; \mathcal{H})}{\partial h_{l,pq}}. \end{aligned} \tag{10}$$

In the infinite-width limit as $d_l \to \infty$, Jacot et al. (2018); Liu et al. (2020) showed that, provided that the last layer has linear output (i.e., it does not have an activation $\sigma$), $\Theta(\mathbf{x}_i, \mathbf{x}_j; \mathcal{H})$ converges to a limiting kernel, the NTK. This kernel stays constant during training—a property called constancy—, which is equivalent to replacing the outputs of the neural network by their first-order Taylor expansion in the parameter space (Lee et al., 2019). Hence, in the infinite-width limit the training dynamics of (10) reduce to kernel ridge regression[1] on the NTK, which has a closed-form solution. This facilitates understanding the learning dynamics of overparametrized neural networks, which are notoriously difficult to study directly, by analysis of the corresponding NTK.

## 4.1. Graph Neural Tangent Kernel

For simplicity, we will consider a GNN with only one feature per layer; the generalization to multiple features is more involved but straightforward. Recall that the $L$-layer GNN supported on $\mathbf{A}_n \in \mathbb{R}^{n \times n}$ is written as

$$\begin{aligned} \mathbf{x}_{n,0} &= \mathbf{x}_n & \mathbf{u}_{l,n} &= H_l(\mathbf{x}_{l-1,n}) \\ \mathbf{x}_{l,n} &= \sigma(\mathbf{u}_{l,n}) & f(\mathbf{x}_n; \mathbf{A}_n, \mathcal{H}) &= \mathbf{x}_{L,n}. \end{aligned}$$

The graph NTK (GNTK) associated with this GNN is

$$\begin{aligned} \Theta(\mathbf{x}_n, \mathbf{x}'_n; \mathbf{A}_n, \mathcal{H}) =& \nabla_{\mathcal{H}} f(\mathbf{x}_n; \mathbf{A}_n, \mathcal{H})^T \nabla_{\mathcal{H}} f(\mathbf{x}'_n; \mathbf{A}_n, \mathcal{H}) \\ =& \textstyle\sum_k \sigma'(\mathbf{u}_{L,n}) \mathbf{A}_n^k \mathbf{x}_{L-1,n} \otimes \sigma'(\mathbf{u}'_{L,n}) \mathbf{A}_n^k \mathbf{x}'_{L-1,n} \\ &+ \sigma'(\mathbf{u}_{L,n})(\sigma'(\mathbf{u}_{L-1,n}) \mathbf{A}_n^k \mathbf{x}_{L-2,n}) \\ &\otimes \sigma'(\mathbf{u}'_{L,n}) H_L(\sigma'(\mathbf{u}'_{L-1,n}) \mathbf{A}_n^k \mathbf{x}'_{L-2,n}) + \ldots \end{aligned}$$

[1] In practice, it is not necessary to consider the MSE for this derivation. It suffices for the loss to be such that the norm of the training direction $f(\mathcal{H}_0) - f(\mathcal{H}^\star)$ is strictly decreasing during training (Jacot et al., 2018). E.g., if we consider the cross-entropy loss, the training dynamics reduce to kernel logistic regression.

where there are $L$ terms; the first term corresponds to the last layer, the second term to the second-last layer, and so on. We have used $\sigma'(\mathbf{u})$ to denote the diagonal matrix with $jj$th entry equal to $\sigma'([\mathbf{u}]_j)$, where $\sigma'$ is the derivative of $\sigma$. Note that $\nabla_{\mathcal{H}} f$ is a $|\mathcal{H}| \times n$ matrix, hence the GNTK is a matrix $\Theta_n(\mathbf{x}_n, \mathbf{x}'_n; \mathbf{A}_n, \mathcal{H}) \in \mathbb{R}^{n \times n}$.

### 4.2. Graphon Neural Tangent Kernel

To derive the WNTK, we will also consider a WNN with only one feature per layer for simplicity. Recall that the $L$-layer WNN associated with the graphon $\mathbf{W}$ is given by

$$\begin{aligned} X_0 &= X & U_l &= T_{H_l} X_{l-1} \\ X_l &= \sigma(U_l) & f(X; \mathbf{W}, \mathcal{H}) &= X_L. \end{aligned}$$

Therefore, the WNTK is given by

$$\begin{aligned} &\Theta(X, X'; \mathbf{W}, \mathcal{H}) = \\ &= \textstyle\sum_{l,k} \partial_{h_{l,k}} f(X; \mathbf{W}, \mathcal{H}) \otimes \partial_{h_{l,k}} f(X'; \mathbf{W}, \mathcal{H}). \end{aligned} \tag{11}$$

Calculating the derivative with respect to the $k$th weight in layer $l_j = L - j$, we get

$$\partial_{h_{l_j,k}} u_L = T_W^{(k)} X_{L-1}$$

for $j = 0$, and similarly for $0 < j < L$,

$$\partial_{h_{l_j,k}} u_L = T_{H_L} \sigma'(U_{L-1}) \dots T_{H_{l_j-1}} \sigma'(U_{L-j}) T_W^{(k)} X_{l_{j-1}}.$$

Hence, (11) has $L$ terms in total, explicitly

$$\begin{aligned} &\Theta(X, X'; \mathbf{W}, \mathcal{H}) = \\ &\quad = \textstyle\sum_k \sigma'(U_L) T_W^{(k)} X_{L-1} \otimes \sigma'(U'_L) T_W^{(k)} X'_{L-1} \\ &\quad + \sigma'(U_L) T_{H_L} (\sigma'(U_{L-1}) T_W^{(k)} X_{L-2}) \\ &\quad \otimes \sigma'(U'_L) T_{H_L} (\sigma'(U'_{L-1}) T_W^{(k)} X'_{L-2}) + \dots \end{aligned}$$

where $\sigma'(U_L)$ is now evaluated and multiplied pointwise. Note that $\Theta$ is a linear operator on graphon signals, i.e., given a signal $Y \in L^2([0,1])$, the WNTK applied to this signal yields

$$\begin{aligned} \Theta(X, X'; \mathbf{W}, \mathcal{H}) Y = &\textstyle\int_0^1 \sum_k \sigma'(U_L(u)) T_W^{(k)} X_{L-1}(u) \\ &\times \sigma'(U'_L(v)) T_W^{(k)} X'_{L-1}(v) Y(v) \\ &+ \sigma'(U_L(u)) T_{H_L} (\sigma'(U_{L-1}) T_W^{(k)} X_{L-2})(u) \\ &\times \sigma'(U_L(v)) T_{H_L} (\sigma'(U_{L-1}) T_W^{(k)} X_{L-2})(v) Y(v) + \dots\, dv \end{aligned}$$

where the first term corresponds to the last layer, the second corrresponds to the second-last layer and so on.

For a graph $\mathbf{A}_n$ sampled from $\mathbf{W}$ with associated induced graphon $\mathbf{W}_n$, and for input features $\mathbf{x}_n$ sampled from a graphon signal $X$ and associated induced graphon signal $X_n$ [cf. (8)], the induced WNTK is given by the same formula as (11) but with $\mathbf{W}$, $X$ replaced by $\mathbf{W}_n$, $X_n$.

## 5. Graph NTK Converges to Graphon NTK

Next, we consider a sequence of graph signals $\{(\mathbf{G}_n, \mathbf{x}_n)\}$ converging to a graphon signal $(\mathbf{W}, X)$ to evaluate the convergence of the GNTK $\Theta(\mathbf{x}_n, \mathbf{x}'_n; \mathbf{A}_n, \mathcal{H})$ to the WNTK $\Theta(X, X'; \mathbf{W}, \mathcal{H})$. First, for each pair of signals $X, X'$, the WNTK $\Theta(X, X'; \mathbf{W}, \mathcal{H})$—which we denote $\Theta(X, X')$ to simplify notation—is a linear operator on functions in $L^2$. Hence, to prove convergence to this operator we introduce the operator norm

$$||\Theta(X, X')|| = \sup_{Y \neq 0} \frac{||\Theta(X, X')Y||}{||Y||}. \tag{12}$$

Let $\mathbf{W}$ be a graphon and $X, X'$ be fixed graphon signals. Consider a sequence of graph signals $\{(\mathbf{G}_n, \mathbf{x}_n)\}$ (respectively $\{(\mathbf{G}_n, \mathbf{x}'_n)\}$) converging to $(\mathbf{W}, X)$ (respectively $(\mathbf{W}, X')$) in the sense of (Ruiz et al., 2021a)[Def. 2]. The notion of convergence of graph signals on a sequence of dense graphs is simple; $\mathbf{G}_n$ converges to $\mathbf{W}$ in the homomorphism density sense [cf. (4)], and $X_n$, the graphon signal induced by $\mathbf{x}_n$, converges to $X$ in the $L^2$ norm, i.e.

$$\|X_n - X\| \to 0 \tag{13}$$

up to node relabelings (see (Ruiz et al., 2021a) for further details, and in particular Lemma 2 for the relationship between $(\mathbf{G}_n, \mathbf{x}_n)$ and the induced signal $(\mathbf{W}_n, X_n)$).

Let $(\mathbf{W}_n, X_n)$ be the graphon signal induced by the graph signal $(\mathbf{G}_n, \mathbf{x}_n)$ [cf. (8)]. The main result of this paper is the following theorem that shows that the WNTK induced by the GNTK converges to the limiting WNTK. The proof is deferred to Appendix A.

**Theorem 5.1.** *Let* $\mathbf{W}$ *be a graphon and* $X, X' \in L^2([0,1])$ *be arbitrary graphon signals. Suppose that* $\{\mathbf{G}_n\}$ *is a sequence of graphs converging to* $\mathbf{W}$ *[cf.* (4)*] and* $\{\mathbf{x}_n\}, \{\mathbf{x}'_n\}$ *are sequences of graph signals converging to* $X$ *[cf.* (13)*]. Then, for any L-layer GNN with finite* $K$*, fixed weights* $\mathcal{H}$ *and* 1*-Lipschitz nonlinearity* $\sigma$*, the associated GNTKs* $\Theta(\mathbf{x}_n, \mathbf{x}'_n; \mathbf{A}_n, \mathcal{H})$ *converge in the operator norm:*

$$\lim_{n \to \infty} ||\Theta(X_n, X'_n; \mathbf{W}_n, \mathcal{H}) - \Theta(X, X'; \mathbf{W}, \mathcal{H})|| = 0$$

*where* $\Theta(X_n, X'_n; \mathbf{W}_n, \mathcal{H})$ *is the WNTK induced by the GNTK* $\Theta(\mathbf{x}_n, \mathbf{x}'_n; \mathbf{A}_n, \mathcal{H})$*.*

The convergence of the GNTK to a limit object—the WNTK—has two important implications for machine learning on large-scale graphs. First, consider that we have

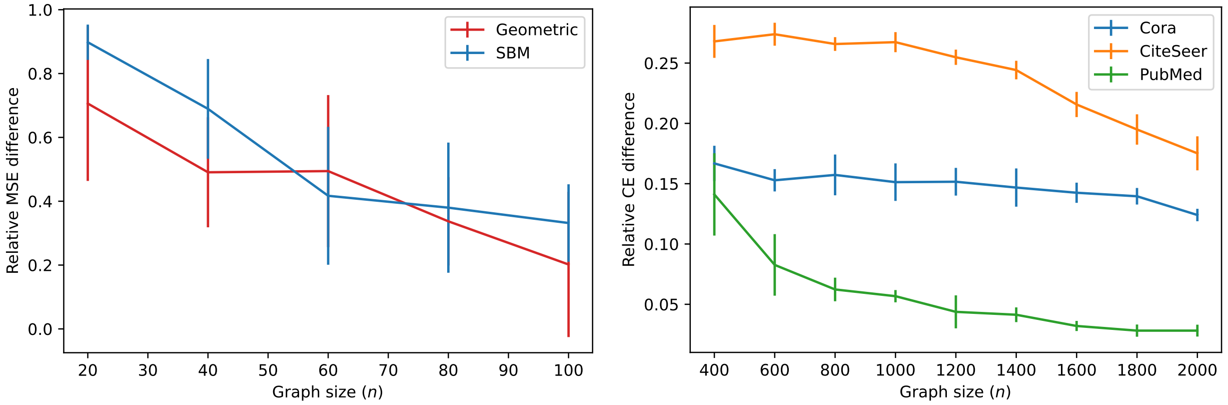

*Figure 1.* Difference between the test error achieved by the GNTK fitted on the $n$-node graph when used for prediction the $n$-node graph, and the test error achived by the same GNTK when used for predicition on the $N$-node graph. (left) Opinion dynamics on geometric and SBM graphs, where the test error is the MSE $N = 300$. (right) Node classification on Cora, CiteSeer and Pubmed, where the test error is the CE and $N = 2708$, 3327, and 10000 respectively.

a large graph $\mathbf{G}_N$ on which we want to predict signals $\mathbf{y}_N$ using the GNTK, but that we do not have enough computational resources to compute the full-sized GNTK $\Theta(\mathbf{x}_N, \mathbf{x}'_N; \mathbf{A}_N, \mathcal{H})$. This is expected, since calculating the kernel regression weights on this graph requires inverting a matrix with dimension proportional to $M$, the number of training samples, and the graph size $N$, see Sec. 4. Thm. 5.1 implies that we can subsample the GNTK and the labels $Y_N$ on a smaller graph $\mathbf{G}_n$, $n \ll N$—as $\Theta(\mathbf{x}_n, \mathbf{x}'_n; \mathbf{A}_n, \mathcal{H})$ and $\mathbf{y}_n$ respectively—and fit the data to the GNTK on this smaller graph. Once the kernel regression weights are obtained, they can then be *transferred* to make predictions on $\mathbf{G}_N$. Naturally, there will be an error associated with this transference; however, due to convergence, this error vanishes asymptotically in $n$.

The precise convergence bounds for different graphs can be computed by plugging into (26). For instance, for graphs sampled from the graphon as in (Ruiz et al., 2021b)[Def. 1], the bound is given by

$$||\Theta(X_n, X'_n) - \Theta(X, X')|| \leq C(K^{4+L}\frac{2A_w}{n} + K^{2+L}\frac{A_x}{n}).$$

where $C$ is a constant, and $A_w$ and $A_x$ are the Lipschitz constants of the graphon and the graphon signal respectively.

In practice, transferability is useful in a variety of real-world applications, especially on large graphs. For instance, one may be interested in the learning dynamics of a GNN trained to provide ad recommendations on a growing, dense social media network. Our theoretical results show that one can calculate the GNTK on a smaller subgraph, e.g., the same social media network at an earlier time, and then transfer it to the large target graph with theoretical performance guarantees. This works because dense networks growing under the same underlying 'rule' or 'process' should have similar homomorphism densities, and converge to similar graphons; or, inversely, they can be thought of as being sampled from similar graphons. The more similar these limiting graphons are (under a metric such as the cut metric or the $L^2$ norm), the lower the error when transferring the GNTK across graphs.

A second implication of Thm. 5.1, which is perhaps more important, is that GNTK convergence implies that a GNTK fitted on a smaller graph can be used to infer details about the *training dynamics* of wide/overparametrized GNNs on larger graphs. This follows from Thm. 5.1 in conjunction with theoretical results by Liu et al. (2020), which proves that in the infinite-width limit the training dynamics of general convolutional models (including GNNs) reduces to kernel ridge regression on the corresponding NTK; and empirical results by Sabanayagam et al. (2021) showing that even when the output layer is nonlinear, constancy of the GNTK can still be observed.

An example of property of a large-graph GNTK $\Theta(\mathbf{x}_N, \mathbf{x}'_N; \mathbf{A}_N, \mathcal{H})$ that can be estimated from a small-graph GNTK $\Theta(\mathbf{x}_n, \mathbf{x}'_n; \mathbf{A}_n, \mathcal{H})$ is its eigenvalue spectrum. As we show in Thm. 5.2, the spectrum of $\Theta(X_n, X'_n; \mathbf{W}_n, \mathcal{H})$ converges to the the spectrum of $\Theta(X, X'; \mathbf{W}, \mathcal{H})$. The proof is deferred to Appendix B.

**Theorem 5.2.** *Let $\{\mathbf{x}_{i,n}\}_{i=1}^M$ be a set of $M$ signals on the graph $\mathbf{G}_n$, and $\{X_{i,n}\}_{i=1}^M$ the corresponding induced signals on the induced graphon $\mathbf{W}_n$. Assume that $\mathbf{G}_n \to \mathbf{W}$ [cf. (4)] and $X_{i,n} \to X$ [cf. (13)] for all $i$. Define the operators $\mathbf{\Theta}_n$ and $\mathbf{\Theta}$ where $[\mathbf{\Theta}_n]_{ij} = \Theta(X_{i,n}, X_{j,n}; \mathbf{W}_n, \mathcal{H})$ and $[\mathbf{\Theta}]_{ij} = \Theta(X_i, X_j; \mathbf{W}, \mathcal{H})$. Let $\lambda_p(T)$, $p \in \mathbb{Z} \setminus \{0\}$, denote the pth eigenvalue of a compact, self-adjoint operator $T$, with $\lambda_{-1} < \lambda_{-2} < \ldots \leq 0 \leq \ldots < \lambda_2 < \lambda_1$. If the*

*Table 2.* Movie recommendation results. Relative difference between the test MSE achieved by the GNTK fitted on the $n$-node graph when used for prediction the $n$-node graph, and the test error achived by the same GNTK when used for predicition on the $N$-node graph, where $N = 1000$.

| Number of nodes | 100 | 200 | 300 | 400 | 500 |
|---|---|---|---|---|---|
| Mean (%) | 6.80 | 0.78 | 0.09 | 0.03 | 0.03 |
| Std. dev. (p.p.) | 3.51 | 1.16 | 0.03 | 0.01 | 0.02 |

*weights in $\mathcal{H}$ are bounded, then, for all $p$,*

$$\lim_{n\to\infty} |\lambda_p(\mathbf{\Theta}_n) - \lambda_p(\mathbf{\Theta})| \to 0.$$

*Moreover, the corresponding eigenspaces converge in the sense that the spectral projectors converge.*

This is an important result because, as proved by Jacot et al. (2018), the convergence of kernel gradient descent follows the kernel principal components. Thus, if the GNN $f(\mathcal{H})$ is sufficiently wide, and if $f(\mathcal{H}_0) - f(\mathcal{H}^\star)$ is aligned with the $i$th GNTK principal component, we can expect gradient descent to converge with rate proportional to $\lambda_i$, the $i$th eigenvalue of the GNTK. Thm. 5.2 shows that the GNTK eigenvalues converge to the eigenvalues of the WNTK. As such, we can estimate the speed of convergence of a large-graph GNN $f(\mathbf{x}_N; \mathbf{A}_N, \mathcal{H})$ from the eigenvalues of a small-graph GNTK $\Theta(\mathbf{x}_n, \mathbf{x}_n'; \mathbf{A}_n, \mathcal{H})$. A numerical example of this application of Thms. 5.1 and 5.2 is given in Sec. 6.3.

Spectral convergence of the GNTK is particularly relevant for practical applications on large graphs. Convergence of the spectrum implies convergence of the directions of learning. Intuitively, one can calculate the GNTK on the smaller graph, then transfer it to the large graph with the guarantee that the dominant directions of fastest convergence are being captured. We can learn GNN architectures that harness this property by pretraining on a smaller graph, and making adjustments on the large graph; or, by iterating between different resolutions of graph size.

## 6. Numerical Results

In the following, we illustrate the convergence of the GNTK to the WNTK on a simulated opinion dynamics model on random graphs; on a movie recommendation graph; and on citation networks (Sec. 6.1). In the opinion dynamics problem, we further vary the width of the network to analyze the large width behavior of the GNTK and how it relates to the behavior of the GNN (Sec. 6.2). We conclude with an example of how the convergence of the GNTK can be used in practice to estimate the eigenvalues of the GNTK, and thus the speed of GNN training along its principal components, on large-scale graphs (Sec. 6.3).

**Opinion dynamics.** This is a node-level task modeling the outcomes of a mathematical model for studying the evolution of ideologies, affiliations, and opinions in society (Lorenz, 2007), including topics of important practical interest such as political ideologies and misinformation spread. On an undirected $n$-node graph $\mathbf{G} = (\mathcal{V}, \mathcal{E})$, we consider an opinion dynamics process $\mathbf{x}_t \in \mathbb{R}^n$. The node data $[\mathbf{x}_t]_i$ is the opinion of individual $i$ under a standard bounded-confidence model described by Hegselmann et al. (2002):

$$[\mathbf{x}_t]_i = \sum_{j \in \mathcal{S}_{i,t-1}} c[\mathbf{x}_{t-1}]_j. \tag{14}$$

where $c > 0$ is the so-called influence parameter and $\mathcal{S}_{i,t-1} = \mathcal{N}_i \cap \mathcal{X}_{i,t-1}$ is the intersection of $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}$, the neighborhood of $i$, and $\mathcal{X}_{i,t} = \{j \in \mathcal{V} \mid |[\mathbf{x}_t]_j - [\mathbf{x}_t]_i| \leq \epsilon\}$, the set of nodes whose opinion at time $t$ diverges by at most $0 \leq \epsilon \leq 1$ of the opinion of node $i$. This model reduces to the classic De-Groot (DeGroot, 1974) opinion model for $\epsilon = 1$. We fix $\epsilon = 0.3$ and $c = 0.1$.

After drawing the initial opinions $\mathbf{x}_0$ from a Gaussian distribution with $\boldsymbol{\mu} = \mathbf{0}$ and $\mathbf{\Sigma} = 2\mathbf{I}$, we run the process (14) until convergence (for at most $T = 1000$ iterations) to obtain the final opinions $\mathbf{x}_T$. The goal of the learning problem is then to predict $\mathbf{y} = \mathbf{x}_T$ from $\mathbf{x} = \mathbf{x}_0$. We fix the training set size to 300 samples, and use 30 samples for both validation and testing.

**Movie recommendation.** In this problem, the graph is a movie similarity network, the data are existing user ratings (ranging from 1 to 5), and the task is to use the existing ratings to fill in the missing ratings for movie recommendation. The movie similarity graph is a dense correlation graph built by computing pairwise correlations between the rating vectors of the movies rated in the MovieLens-100k dataset (Harper & Konstan, 2016). We only consider movies with at least 10 ratings, and the large graph has $N = 1000$ nodes sampled at random from the set of admissible movies. The learning problem is to predict the ratings of the movie “Contact”. We use a 90-10 train-test split of the data, and build the graph using only the training samples. Due to memory limitations, we only use 25% of the users (samples) to compute the GNTK.

**Citation networks.** The third problem we consider is a node classification problem on the Cora, CiteSeer and PubMed networks, where nodes represent documents and undirected edges represent citations between papers in either direction. The nodes a bag-of-words representations are grouped in $\mathbf{X} \in \{0,1\}^{n \times F}$ and each node is associated with one of $C$ classes. We consider the networks and train-test-splits from the full distribution (Bojchevski & Günnemann, 2017) available in PyTorch Geometric, but only sample $F = 1000$ features for CiteSeer and $N = 10000$ nodes for PubMed due to memory limitations. Through studying citation networks, we illustrate the empirical manifestation of our results on

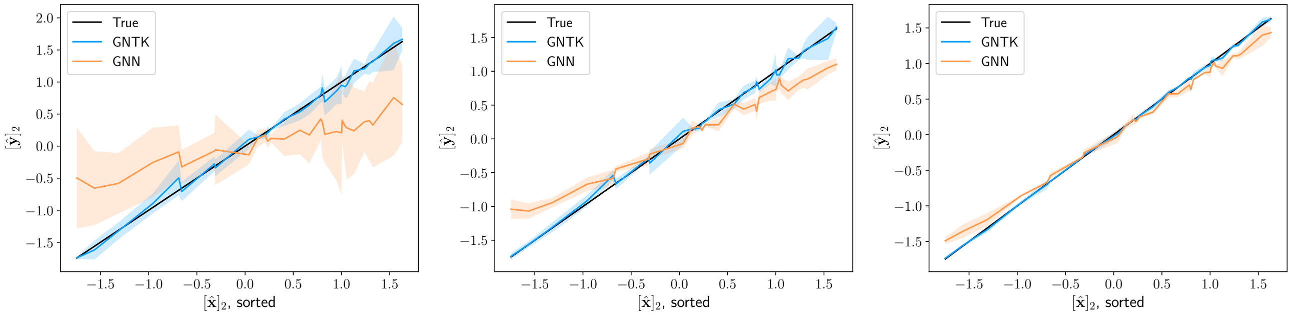

*Figure 2.* Projections of the inputs and outputs of the GNTK and GNN, and of the target or true labels onto the second eigenvector of the adjacency matrix of a 80-node SBM graph for widths $F = 10$ (left), $F = 50$ (center) and $F = 250$ (right).

real-world graphs. Although citation networks are not best modeled by graphon limits, they certainly have limits, and our results suggest that convergence is a global property of GNTKs and graph limits.

**Architectures and experiment details.** In all experiments, the GNNs have $L = 1$ layer (3) with ReLU nonlinearity followed by a perceptron layer. For opinion dynamics, $K = 2$, and for movie recommendation, $K = 5$. In both cases, we consider the MSE and fit the GNTK using linear regression. In the citation network experiment, the GNN architecture has $K = 2$ and includes a softmax layer followed by argmax; and we consider the cross-entropy (CE) and fit the GNTK using logistic regression. All reported results are averaged over 5 and 10 realizations for opinion dynamics and movie recommendation/citation networks respectively. Additional architecture and experiment details are listed in each subsection. The code can be found in this repository. All experiments were run on a NVIDIA RTX A6000 GPU.

### 6.1. Convergence

To visualize the convergence of the GNTK, we fit a GNTK with $F_1 = 10$ in the opinon dynamics and citation network experiments, and $F_1 = 16$ in the recommendation experiment, on the training set of a small $n$-node graph; and transfer this GNTK, without retraining, to predict the outputs on the test set of both the same $n$-node graph and a larger $N$-node graph. To visualize convergence, we plot in Fig. 1 the absolute difference between the test errors attained on the $n$-node and the $N$-node graph (normalized by the test error on the $N$-node graph) as a function of the size of the smaller graph $n$, for both the opinion dynamics (left) and node classification (right) experiments. The results of the recommendation experiment are reported in Table 2.

In opinion dynamics, $N = 300$. We consider two types of graphs, corresponding to two graphon families: (a) a symmetrized geometric $k$-nearest neighbor graph where nodes are drawn at random from a $50 \times 50$ square and $k = n/10$; and (b) a stochastic block model (SBM) graph with intra-community probability $p = 0.1$ and inter-community probability $q = 0.05$. As $n$ increases from 20 to 100, we observe that the difference between the MSEs for the $n$-node and the $N$-node graphs decrease steadily from $\sim 70\%$ to $\sim 20\%$ on the geometric graphs and from $\sim 90\%$ to $\sim 40\%$ on the SBM graphs.

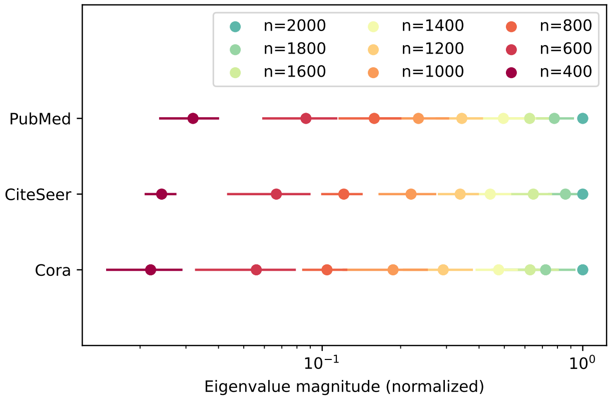

*Figure 3.* The leading eigenvalue of the GNTK with width $F = 10$ for Cora, CiteSeer and PubMed as a function of the graph size $n$.

In the recommendation experiment, $N = 1000$. As $n$ increases from 100 to 500, we see from Table 2 that the average test MSE difference on the large graph decreases from 6.8% to 0.3%, and so does the standard deviation, thus confirming our theoretical findings. In particular, $n = 300$ is enough to achieve an MSE difference of less than 1%.

In the node classification experiment, $N = 2708$ for Cora, $N = 3327$ for CiteSeer, and $N = 10000$ for PubMed (sampled from the 19717 nodes in the original due to memory limitations). For all datasets, the difference between the CE loss for the $n$-node and the $N$-node graph also decreases as $n$ increases. This can be interpreted to mean that node classification on these citation networks depends largely on local information, so beyond a critical size $n$, the transference error saturates. An interesting point to make is that although these citation networks are not dense (i.e., are not best modeled by a graphon), we still see the empirical manifestation of our theoretical results.

### 6.2. Wide Network Behavior

Next, we analyze the effect of width on both the GNN and the GNTK when they are trained/fitted on a small graph and transferred to a large graph. From the NTK analysis by Jacot et al. (2018), as the GNN width increases we expect (i) the GNTK and the GNN to exhibit less variance over multiple weight initializations and (ii) the GNN outputs to approach those of kernel regression with the GNTK.

For the opinion dynamics experiment on SBMs, we consider three GNNs with widths (number of features) $F_1^{(1)} = 10$, $F_1^{(2)} = 50$, and $F_1^{(3)} = 250$ supported on the same 80-node graph. Using their initial weights, we construct the corresponding GNTKs. We train the three GNNs by minimizing the MSE loss over 20 epochs and with batch size 32, using ADAM (Kingma & Ba, 2015) with learning rate $1e^{-3}$ and weight decay $5e^{-3}$. Simultaneously, we fit the GNTKs to the training set, and then transfer both the GNNs and the corresponding GNTKs to the $N$-node graph and compute their outputs on the test set.

In Fig. 2, we plot the projection of the outputs $\mathbf{y}$ onto the second graph eigenvector $\mathbf{v}_2$, $[\hat{\mathbf{y}}]_2 = \mathbf{v}_2^T \mathbf{y}$ against the projections of the inputs onto the same vector $[\hat{\mathbf{x}}]_2 = \mathbf{v}_2^T \mathbf{x}$ (sorted in ascending order) for both the GNNs and the GNTKs for $F_1^{(1)} = 10$ (left), $F_1^{(2)} = 50$ (center), and $F_1^{(3)} = 250$ (right). The second graph eigenvector was chosen as it captures the community structure, however, note that the same behavior was observed upon projecting onto other eigenvectors, as shown in Appendix C. The results are averaged over 5 GNN initializations, with the solid lines representing the mean and the shaded areas representing the standard deviation. The behavior is as expected: as the width increases, the GNN and the GNTK have smaller variance in behavior for different weight initializations, and the GNN and GNTK curves align. In Appendix C, we include additional plots displaying the mean and variance over different initializations as we vary the graph size.

### 6.3. Application: Eigenvalue Convergence

In this section, we elucidate an important application of the convergence of the GNTK. Jacot et al. (2018) shows that the convergence of kernel gradient descent follows the kernel principal components. Therefore, if the GNN is sufficiently wide, and $f(\mathbf{X}_n; \mathbf{A}_n, \mathcal{H}_0) - f(\mathbf{X}_n; \mathbf{S}_n, \mathcal{H}^\star)$ is aligned with the $i$th GNTK principal component, we can expect gradient descent to converge with rate proportional to $\lambda_i$, the $i$th eigenvalue of the GNTK[2].

Recall that we prove, in Thm. 2, that the spectrum of the GNTKs converge in the graphon limit. In this context, a simple application of the convergence of the GNTK spectrum is as follows: say that we know that $f(\mathbf{X}_N; \mathbf{A}_N, \mathcal{H}_0) - f(\mathbf{X}_N; \mathbf{A}_N, \mathcal{H}^\star)$ is aligned with the $p$th GNTK eigenvector, and we want to estimate its speed of convergence, but the kernel $\Theta(\mathbf{X}_N, \mathbf{X}_N'; \mathbf{A}_N, \mathcal{H})$ is too expensive to compute. We could in this case sample a graph $\mathbf{G}_n$, $n \ll N$ from the induced graphon $\mathbf{W}_N$ and calculate $\Theta(\mathbf{X}_n, \mathbf{X}_n'; \mathbf{A}_n, \mathcal{H})$. From Thm. 5.2, $\lambda_{n,p}$ converges to $\lambda_{N,p}$. Hence, for large enough $n$, $\lambda_{n,p}$ should provide a good approximation of $\lambda_{N,p}$.

To illustrate this empirically, in Fig. 3 we plot the dominant eigenvalues of the GNTK associated with a GNN with $F_1 = 10$ features supported on graphs of increasing size $n$ sampled from the Cora, CiteSeer, and PubMed networks (averaged over 10 initializations). As $n$ grows, we see that the difference between the dominant eigenvalues of consecutive GNTKs reduces. I.e., for each citation dataset, the value of the dominant eigenvalue converges as $n$ increases, indicating that the speed of convergence of the GNN along the dominant GNTK eigenvector converges as the graph grows. For all but the PubMed dataset, the 1600-node graph gives an approximation of $\lambda_{N,1}$ that is over 70% good.

## 7. Conclusions

In this paper, we define WNTKs as the limiting objects of GNTKs, and study how the GNTK evolves as the underlying graph of the corresponding GNN grows. We show that GNTKs converge to the WNTK and that their spectra also converges, thus providing theoretical insight into how the learning dynamics of a GNN evolves as an important dimension—the graph size—grows. In practice, these convergences imply that one can transfer the GNTK of a smaller graph to solve the same task on a larger graph without any further optimization, with theoretical guarantees of performance and insight on the rates of learning along eigendirections that converge as the graph grows. These results were demonstrated through simulations on synthetic and real-world graphs. To conclude, a limitation of this work is that graphons are only good models for dense graphs. In future work, we plan to extend our results to more general graph models, and to better understand the relationship between the spectra of the graph and of the GNTK.

[2] Theoretically, constancy of the NTK in the infinite-width limit and equivalence between GNN training and kernel regression only holds for architectures with linear output layer (Liu et al., 2020). Although this is not the case of the GNNs used here, GNTKs associated with GNNs with nonlinear output layers seem to exhibit constancy empirically (Sabanayagam et al., 2021)

## References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *Int. Conf. Mach. Learning*, pp. 242–252. PMLR, 2019.

Anselone, P. and Palmer, T. Spectral analysis of collectively compact, strongly convergent operator sequences. *Pacific Journal of Mathematics*, 25(3):423–431, 1968.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. In *Int. Conf. Mach. Learning*, pp. 244–253. PMLR, 2018.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. *Neural Inform. Process. Syst.*, 32, 2019a.

Arora, S., Du, S. S., Li, Z., Salakhutdinov, R., Wang, R., and Yu, D. Harnessing the power of infinitely wide deep nets on small-data tasks. *arXiv preprint arXiv:1910.01663*, 2019b.

Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv:1707.03815 [cs.LG]*, 2017.

Borgs, C., Chayes, J. T., Lovász, L., Sós, V. T., and Vesztergombi, K. Convergent sequences of dense graphs I: Subgraph frequencies, metric properties and testing. *Adv. Math.*, 219(6):1801–1851, 2008.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Neural Inform. Process. Syst.*, 32, 2019.

Cerviño, J., Ruiz, L., and Ribeiro, A. Learning by transference: Training graph neural networks on growing graphs. *IEEE Trans. Signal Process.*, 71:233–247, 2023. doi: 10.1109/TSP.2023.3242374.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Neural Inform. Process. Syst.*, Barcelona, Spain, 5-10 Dec. 2016. NIPS Foundation.

DeGroot, M. H. Reaching a consensus. *Journal of the American Statistical association*, 69(345):118–121, 1974.

Du, S. S., Hou, K., Salakhutdinov, R. R., Poczos, B., Wang, R., and Xu, K. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In *Neural Inform. Process. Syst.*, volume 32, 2019.

Gama, F., G. Marques, A., Leus, G., and Ribeiro, A. Convolutional neural network architectures for signals supported on graphs. *IEEE Trans. Signal Process.*, 67:1034–1049, 2018.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *Int. Conf. Mach. Learning*, pp. 1263–1272. PMLR, 2017.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Neural Inform. Process. Syst.*, 30, 2017.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *ACM Trans. Interactive Intell. Syst.*, 5:1–19, Jan. 2016.

Hegselmann, R., Krause, U., et al. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3), 2002.

Huang, K., Wang, Y., Tao, M., and Zhao, T. Why do deep residual networks generalize better than deep feedforward networks?—a neural tangent kernel perspective. *Neural Inform. Process. Syst.*, 33:2698–2709, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Neural Inform. Process. Syst.*, 31, 2018.

Keriven, N., Bietti, A., and Vaiter, S. Convergence and stability of graph convolutional networks on large random graphs. In *34th Neural Inform. Process. Syst.*, volume 33, pp. 21512–21523. NeurIPS Foundation, 2020.

Kingma, D. P. and Ba, J. L. ADAM: A method for stochastic optimization. In *3rd Int. Conf. Learning Representations*, San Diego, CA, 7-9 May 2015. Assoc. Comput. Linguistics.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th Int. Conf. Learning Representations*, Toulon, France, 24-26 Apr. 2017. Assoc. Comput. Linguistics.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *Neural Inform. Process. Syst.*, 32, 2019.

Levie, R., Huang, W., Bucci, L., Bronstein, M., and Kutyniok, G. Transferability of spectral graph convolutional neural networks. *J. Mach. Learning Res.*, 22(272):1–59, 2021.

Li, Z., Wang, R., Yu, D., Du, S. S., Hu, W., Salakhutdinov, R., and Arora, S. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019.

Liu, C., Zhu, L., and Belkin, M. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Neural Inform. Process. Syst.*, 33:15954–15964, 2020.

Lorenz, J. Continuous opinion dynamics under bounded confidence: A survey. *International Journal of Modern Physics C*, 18(12):1819–1838, 2007.

Lovász, L. *Large Networks and Graph Limits*, volume 60. American Mathematical Society, 2012.

Lovász, L. and Szegedy, B. Limits of dense graph sequences. *J. Comb. Theory, Series B*, 96(6):933–957, 2006.

Maskey, S., Levie, R., Lee, Y., and Kutyniok, G. Generalization analysis of message passing neural networks on large random graphs. In *Advances in Neural Information Processing Systems*, 2022.

Maskey, S., Levie, R., and Kutyniok, G. Transferability of graph neural networks: an extended graphon approach. *Applied and Computational Harmonic Analysis*, 63:48–83, 2023.

Neal, R. M. and Neal, R. M. Priors for infinite networks. *Bayesian learning for neural networks*, pp. 29–53, 1996.

Ruiz, L., Chamon, L. F. O., and Ribeiro, A. Graphon neural networks and the transferability of graph neural networks. In *34th Neural Inform. Process. Syst.*, Vancouver, BC (Virtual), 6-12 Dec. 2020. NeurIPS Foundation.

Ruiz, L., Chamon, L. F. O., and Ribeiro, A. Graphon signal processing. *IEEE Trans. Signal Process.*, 69:4961–4976, 2021a.

Ruiz, L., Chamon, L. F. O., and Ribeiro, A. Transferability properties of graph neural networks. *arXiv:2112.04629 [eess.SP]. Submitted to IEEE TSP*, 2021b.

Sabanayagam, M., Esser, P., and Ghoshdastidar, D. New insights into graph convolutional networks using neural tangent kernels. *arXiv preprint arXiv:2110.04060*, 2021.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, May 2013.

Williams, C. Computing with infinite networks. *Neural Inform. Process. Syst.*, 9, 1996.

Xia, X., Mishne, G., and Wang, Y. Implicit graphon neural representation. *arXiv preprint arXiv:2211.03329*, 2022.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *7th Int. Conf. Learning Representations*, pp. 1–17, New Orleans, LA, 6-9 May 2019. Assoc. Comput. Linguistics.

Yehudai, G., Fetaya, E., Meirom, E., Chechik, G., and Maron, H. From local structures to size generalization in graph neural networks. In *Int. Conf. Mach. Learning*, pp. 11975–11986. PMLR, 2021.

# A. Proof of GNTK Convergence

*Proof of Thm. 5.1.* To simplify notation, we will write $\Theta(X, X'; \mathbf{W}, \mathcal{H}) = \Theta(X, X')$ and $\Theta(X_n, X_n'; \mathbf{W}_n, \mathcal{H}) = \Theta(X_n, X_n')$. Let $Y$ be an arbitrary $L^2$ signal. Using the triangle inequality,

$$\begin{aligned} ||\Theta(X,X')Y - \Theta(X_n,X_n')Y|| \leq \sum_k ||&\sigma'(U_L)T_W^{(k)}X_{L-1}\langle\sigma'(U_L')T_W^{(k)}X_{L-1}',Y\rangle \\ &- \sigma'(U_{L,n})T_{W_n}^{(k)}X_{n,L-1}\langle\sigma'(U_{L,n}')T_{W_n}^{(k)}X_{L-1,n}'Y\rangle|| + \dots \end{aligned} \tag{15}$$

where there are $L$ terms for each $k$.

Since $\sigma$ is Lipschitz, from the formula we see that it suffices to prove the following convergence statements for all $k \in 1, ..., K-1$ as $n \to \infty$:

$$\|U_{l,n} - U_l\| \to 0 \tag{16}$$

$$\|U_{l,n}' - U_l'\| \to 0 \tag{17}$$

as well as

$$\left|\left|T_{W_n}^{(k)} - T_W^{(k)}\right|\right| \to 0 \tag{18}$$

$$\left|\left|T_{h_{l,n}} - T_{h_l}\right|\right| \to 0. \tag{19}$$

Here, we prove these statements for each finite $K$ using mathematical induction. For fixed, finite $K$, (18) implies (19). The convergence in (16) and (17) are essentially the same, so let us focus on the first without loss of generality. Expanding the first layer, we have

$$\begin{aligned} U_{1,n} - U_1 &= \sum_k h_{1,k}(T_{W_n}^{(k)}X_n - T_W^{(k)}X) = \sum_k h_{1,k}(T_{W_n}^{(k)}X_n - T_W^{(k)}X + T_W^{(k)}X_n - T_W^{(k)}X_n) \\ &= \sum_k h_{1,k}((T_{W_n}^{(k)} - T_W^k)X_n + T_W^{(k)}(X_n - X)). \end{aligned}$$

Therefore, by the triangle inequality and the definition of the operator norm, we have

$$\|U_{1,n} - U_1\| \leq \sum_k |h_{1,k}| \|T_{W_n}^{(k)} - T_W^k\| \|X_n\| + \sum_k |h_{1,k}| \left|\left|T_W^{(k)}\right|\right| \|X_n - X\| \tag{20}$$

$$\leq \sum_k \max(|h_{1,k}|)\|T_{W_n}^{(k)} - T_W^k\|\|X_n\| + \sum_k \max(|h_{1,k}|)\|T_W^{(k)}\|\|X_n - X\|. \tag{21}$$

Looking at layer $l$, we have

$$U_{l,n} - U_l = \sum_k h_{l,k}(T_{W_n}^{(k)}\sigma(U_{l-1,n}) - T_W^{(k)}\sigma(U_{l-1})).$$

From 21, using mathematical induction, we have, for arbitrary $l$,

$$\|U_{l,n} - U_l\| \leq (\|X_n\|\|X_n - X\| + \|T_{W_n}^{(k)} - T_W^k\|) \cdot (\max(|h|)k)^l \tag{22}$$

$$+ \sum_{j=1}^{l-1} (\max(|h|)k)^{l-1-j} \max(|h|) \|T_{W_n}^{(k)} - T_W^k\| \tag{23}$$

where $h$ is the max of $h_{l,k}$ over all $l$ and $k$, and so convergence of $T_{W_n}^{(k)} \to T_W^{(k)}$ as in (18) and $X_n \to X$ implies convergence of $U_{1,n} \to U_1$ and $U_{l,n} \to U_l$. Since the nonlinearity $\sigma$ is Lipschitz, convergence of (17) follows from convergence of (16).

We now focus on proving (18). By Cauchy-Schwarz,

$$\begin{aligned} ||(T_{W_n} - T_W)Y||^2 &= \int_0^1 \left(\int_0^1 (\mathbf{W}_n(u,v) - \mathbf{W}(u,v))Y(v)dv\right)^2 du \\ &\leq \left(\int_0^1 \int_0^1 (\mathbf{W}_n(u,v) - \mathbf{W}(u,v))^2 dvdu\right) ||Y||^2. \end{aligned}$$

Therefore, taking the square root, dividing by $||Y||$, and using (12), we get

$$||T_{W_n} - T_W|| \leq ||\mathbf{W}_n - \mathbf{W}||_{L^2} \, . \tag{24}$$

It follows that convergence of the induced graphons $\mathbf{W}_n$ to the limiting graphon $\mathbf{W}$ implies (18) for $k = 1$. For higher iterates we note

$$\begin{aligned} \left|\left|(T^{(k)}_{W_n} - T^{(k)}_{W})Y\right|\right|^2 &\leq \int_0^1 \left(\int_0^1 (\mathbf{W}_n T^{(k-1)}_{W_n} Y(v) - \mathbf{W} T^{k-1}_{W} Y(v))dv\right)^2 du \\ &\leq \int_0^1 \left(\int_0^1 \mathbf{W}_n (T^{(k-1)}_{W_n} - T^{(k-1)}_{W}) Y(v) dv\right)^2 du + \int_0^1 \left(\int_0^1 (\mathbf{W}_n - \mathbf{W}) T^{(k-1)}_{W} Y(v) dv\right)^2 du. \end{aligned}$$

Therefore, by the same Cauchy-Schwarz argument above, $\left|\left|T^{(k)}_{W_n} - T^{(k)}_{W}\right|\right|$ is upper bounded by

$$||\mathbf{W}_n|| \left|\left|T^{(k-1)}_{W_n} - T^{(k-1)}_{W}\right|\right| + ||\mathbf{W}_n - W|| \left|\left|T^{(k-1)}_{W}\right|\right|$$

and so convergence $\mathbf{W}_n \to \mathbf{W}$ implies the convergence of all the iterates $T^{(k)}_{W_n} \to T^{(k)}_{W}$ by induction. In fact, one can write

$$\left|\left|T^{(k)}_{W_n} - T^{(k)}_{W}\right|\right| \leq k \, ||\mathbf{W}_n - \mathbf{W}||_{L^2} \tag{25}$$

since $\left|\left|T^{(k-1)}_{W}\right|\right| \leq 1$ and $||\mathbf{W}_n|| \leq 1$.

From the proof one can, with some simple algebra, compute a quantitative estimate of $||\Theta(X_n, X'_n) - \Theta(X, X')||$ in terms of $||\mathbf{W}_n - \mathbf{W}||$, $||X_n - X||$, $||X||$ and $||X'||$. Precisely, by adding and subtracting cross-terms for each term in (15) and using the triangle inequality, we get

$$||\Theta(X_n, X'_n) - \Theta(X, X')|| \leq C(K^{4+L} \, ||\mathbf{W}_n - \mathbf{W}|| + K^{2+L} \, ||X_n - X||), \tag{26}$$

where $C$ is a constant in terms of $\|X\|, \|X'\|, max(|h|), l$. For different choices of graphs, one can then compute these quantities for an explicit estimate of the convergence bound. (Ruiz et al., 2021b)[Appendix B] provides quantitative estimates on $||\mathbf{W}_n - \mathbf{W}||$ and $||X_n - X||$ for random graph models such as so-called template graphs, weighted graphs and stochastic graphs.

□

## B. Proof of Spectrum Convergence

*Proof of Theorem 5.2.* The proof relies on (Anselone & Palmer, 1968)[Prop. 7.1] which proves that if a sequence of compact operators $T_n$ converges to a compact operator $T$ in the appropriate operator norm, the spectrum of $T_n$ converges to the spectrum of $T$. Additionally, from (Anselone & Palmer, 1968)[Thm. 6.3 and Prop 7.1], for every $p$, the eigenspace associated with the $p^{th}$ eigenvalue of $T_n$ also converge to the corresponding eigenspace of $T$. From Thm. 5.1, we know that $\mathbf{\Theta}_n \to \mathbf{\Theta}$. It remains to show that $\mathbf{\Theta}_n$ and $\mathbf{\Theta}$ are compact.

For any graphon $\mathbf{W}$, the operators $T^{(k)}_{W}$ and $T_H$ [cf. (5)] are compact as they are linear compositions of compact operators with bounded linear weights. From (11), we thus conclude $\|\Theta(X, X'; \mathbf{W}, \mathcal{H})\| < \infty$. Therefore, for $Y \in L^2([0, 1])$, $\|\Theta(X, X'; \mathbf{W}, \mathcal{H})Y\| < \infty$, by which we conclude that $\mathbf{\Theta}$ is compact. □

## C. Wide Network Behavior: Additional Experiments

Under the same expeirmental setting of Sec. 6.2, in Fig. 4, we plot the projection of the outputs $\mathbf{y}$ onto the first graph eigenvector $\mathbf{v}_1$, $[\hat{\mathbf{y}}]_1 = \mathbf{v}_1^T \mathbf{y}$ against the projections of the inputs onto the same vector $[\hat{\mathbf{x}}]_1 = \mathbf{v}_1^T \mathbf{x}$ (sorted in ascending order) for both the GNNs and the GNTKs for $F_1^{(1)} = 10$ (left), $F_1^{(2)} = 50$ (center), and $F_1^{(3)} = 250$ (right), and $n = 20, 40, 60, 80, 100$ (top to bottom). The results are averaged over 5 GNN initializations, with the solid lines representing the mean and the shaded areas representing the standard deviation. The behavior is as expected: as the width increases, the

GNN and the GNTK have smaller variance in behavior for different weight initializations, and the GNN and GNTK curves align. As $n$ increases, we also observe a slight improvement in the variance of the GNN curves. This is expected as, when GNNs are trained on larger graphs, they have better transferability (Ruiz et al., 2020). No significant trends are observed for the GNTK curves for increasing $n$.

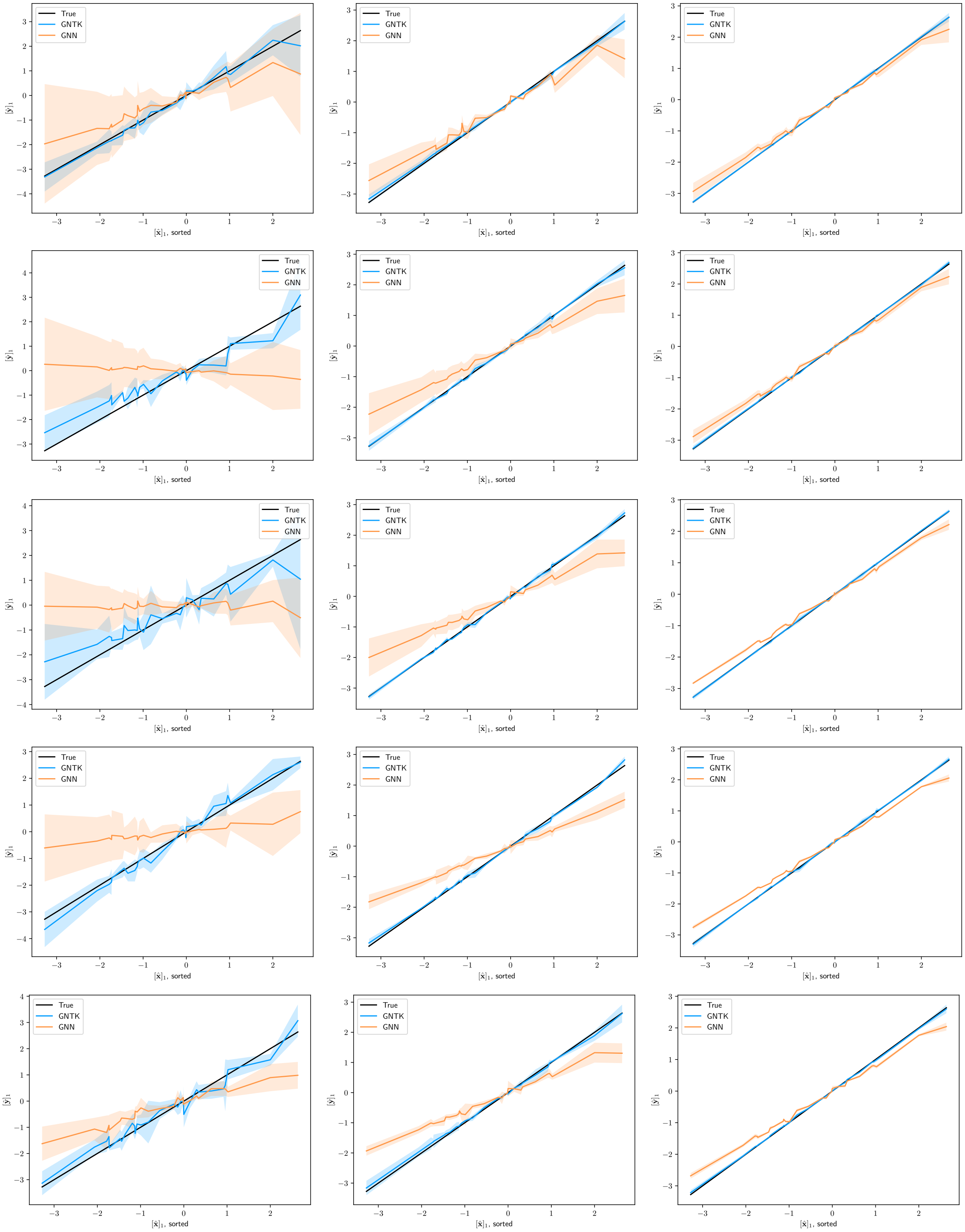

*Figure 4.* Projections of the inputs and outputs of the GNTK and GNN, and of the target or true labels onto the first eigenvector of the adjacency matrix of a $n$-node SBM graph for widths $F = 10$ (left), $F = 50$ (center) and $F = 250$ (right). From top to bottom $n = 20, 40, 60, 90, 100$.