The Surprising Effectiveness of Infinite-Width NTKs for Characterizing and Improving Model Training

Joshua DeOliveira[1], **Walter Gerych**[2], **Elke Rundensteiner**[1]
[1]Worcester Polytechnic Institute, Worcester, MA
[2]Massachusetts Institute of Technology, Cambridge, MA
jcdeoliveira@wpi.edu, wgerych@mit.edu, rundenst@wpi.edu

Abstract

Developments in deep neural nets have trended towards increasingly larger overparameterized architectures, resulting in lengthy training sessions with ever more elusive training dynamics. Thus, ensuring these models learn accurate generalizable representations of data efficiently is challenging. Previous works have developed specialized techniques from data-pruning, architecture selection, pseudo-label generation, bias identification, or label refurbishment to improve downstream training. Problematically, most methods require prohibitively expensive iterative model training. In this paper, we demonstrate that we can exploit the recent neural tangent kernel (NTK) theory for understanding and improving model training behavior before ever training a model. First, we show a powerful signal derived from the NTK theory can be computed remarkably fast. We then leverage this signal for the design of a unified suite of surprisingly effective tools for the four important tasks of architecture selection, pseudo-label verification, bias identification, and label refurbishment, all requiring zero model training.

1 Introduction

Background. Deep neural nets are a powerful tool for many data modalities, including tabular data (Gorishniy et al. 2021), computer vision (Wang et al. 2017; Girdhar et al. 2023), and natural language processing (Devlin 2018; Brown et al. 2020). In response to the availability of massive digital datasets and to aim to solve more complex learning problems, deep neural nets have become increasingly larger and heavily overparameterized. For example, models such as ResNet (He et al. 2016) and its variants have seen parameter counts rise into the tens of millions, and large language models (LLMs) such as Chat-GPT (Achiam et al. 2023) have broken into the space of trillion-parameter models.

Problem. Given these overparameterized architectures, training sessions can become remarkably expensive and lengthy (Minaee et al. 2024). Practitioners cannot afford to waste vast amounts of compute resources and risk production delays due to failed or ineffective training. Common pitfalls such as ill-suited architectures, the presence of noisy or missing labels, or underlying biases in the data can all plague model training. Thus, there is an urgent need for methods that can reduce training costs in a reliable way that doesn't sacrifice downstream performance.

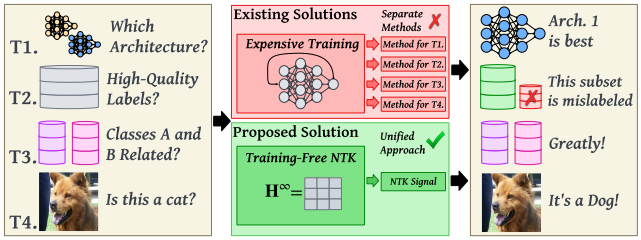

Figure 1: Training real overparameterized models is expensive. Currently, improving training requires a suite of extremely expensive tailored methods that can only be conducted during the expensive training process (top). By investigating models using their infinitely-wide analog instead (bottom), we propose a single versatile signal derived before real training is conducted that will improve model training.

State-of-the-Art. Previous works improved training by developing a wide variety of specialized methods to combat common areas of failure to make training either more accurate or cheaper. These include AutoML and neural architecture search (NAS) (Chitty-Venkata et al. 2023; Liu, Simonyan, and Yang 2018), data-pruning (Nohyun, Choi, and Chung 2022; Paul, Ganguli, and Dziugaite 2021; Feldman and Zhang 2020), pseudo-label generation (Rizve et al. 2021; Gilhuber et al. 2022; Kaul, Xie, and Zisserman 2022), inherent bias identification (Nam et al. 2020), and label refurbishment (Patel and Sastry 2023).

While each of the above tasks serve an important step in providing the best-suited scenario for model training, many modern methods can only be performed during the actual training process. This "piggybacking" exacerbates the computational costs incurred from training, and risks requiring repeated retraining upon attempted improvements.

Proposed Solution. In this paper, we demonstrate that the recent infinite-width neural tangent kernel (NTK) theory, characterizing theoretical behavior of neural nets that possess infinitely many neurons in their hidden layers (Jacot, Gabriel, and Hongler 2018; Arora et al. 2019), is surprisingly more applicable for improving training real models and conducting data valuation tasks than previously thought.

By leveraging key relationships between real architectures and their infinite-width analogs, we show that finite-width training can be characterized and improved preemptively before the training is even conducted, significantly cutting down the computational expenses of model development. We show that a single signal from the infinite-width NTKs can serve as a powerful unified solution tackling critical tasks from architecture selection, pseudo-label verification, inherent bias identification, to label refurbishment, all before model training (Figure 1).

In short, our contributions in this paper include:

- We demonstrate computing the infinite-width NTK signal is remarkably fast and widely accessible.
- For architecture selection, we show architectures better support a learning task if eigenvectors corresponding to low-magnitude eigenvalues of a Gram-Matrix formed by infinite-width NTK can project clustered classes.
- For pseudo-label verification, we show that the degree to which the Gram-Matrix formed by an infinite-width NTK is block diagonal determines the efficiency in which a task is learned on an architecture of finite-width.
- For inherent bias detection, we show that the magnitude of off-diagonal elements of the Gram-Matrix formed by an infinite-width NTK identifies the impact of inter-class patterns that will continue to be present in training.
- For label refurbishment, we show that accurate labels can be recovered from mislabeled data by optimizing training labels to increase the degree of block diagonalization of the Gram-Matrix formed by the infinite-width NTK.
- We conduct empirical studies on a variety of benchmark datasets and neural net architectures that demonstrate our proposed signal yields equal or better performance compared to a suite of methods that require training – while also being computationally cheaper.

2 Related Work

Investigations of the NTK. Previous works experimentally explored the inductive bias of empirically computed NTK of different finite-width neural net architectures (Bietti and Mairal 2019; Chen, Gong, and Wang 2021). One recent work looks into training-free data pruning (Nohyun, Choi, and Chung 2022) using NTK specifically related to 2-layer networks with ReLU activation functions (Arora et al. 2019). There has been research into empirically exploring infinite-width NTKs for NLP-type networks, in particular transformer-based networks with infinitely many attention (Hron et al. 2020). In contrast, our work deals with concrete applications of infinite-width NTKs for infinite analogs of popular architectures containing linear layers, convolutional layers, or some combination of the two in general.

Training-Free Analysis. Some works explore the analysis of training regimes before training. For example, it has been shown that training time can be predicted (Zancato et al. 2020) using empirical finite-width NTKs computed after a finite model's parameters have been initialized. Similarly, work in neural architecture search (NAS) using non-NTK kernel methods (Mellor et al. 2021; Park et al. 2020) has been investigated. Our paper demonstrates that infinite-width NTKs can surprisingly single-handedly perform many such training-related tasks before training is conducted.

3 Neural Tangent Kernels

As our proposed data valuation methods all depend on a signal computed from NTKs, it is important to briefly introduce NTKs. First studied by (Jacot, Gabriel, and Hongler 2018), the NTK Θ is a bi-linear *kernel function* that explains the similarity between the way any two instances x' and x'' in the training set would be learned by a neural network f parameterized by θ. Namely,

$$\Theta(x', x''; \theta) = \langle \nabla_\theta f(x'; \theta), \nabla_\theta f(x''; \theta) \rangle \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product, and $f : \mathbb{R}^n \to \mathbb{R}^m$ is a neural network using the definition provided by (Jacot, Gabriel, and Hongler 2018).

Equation 2, derived by (Jacot, Gabriel, and Hongler 2018), shows that Θ directly weights the rate at which f's inference would continuously change during training under an infinitesimal learning rate, where $\mathcal{X} = \{x_i \in \mathbb{R}^n\}_{i=1}^N$ and $\mathcal{Y} = \{y_i \in \mathbb{R}^m\}_{i=1}^N$ are the training data and label sets, respectively. Interestingly, Θ is agnostic of the task being learned and agnostic to the objective function $\mathcal{C}$ employed.

$$\delta_t f(x; \theta(t)) = - \sum_{x_i \in \mathcal{X}} \Theta(x, x_i; \theta(t)) \delta_{\hat{y}} \mathcal{C}(\hat{y}, y_i) \Big|_{\hat{y}=f(x_i;\theta(t))} \tag{2}$$

As the number of neurons in the hidden layers, or the number of filters in convolutional layers, of a given architecture f approaches infinitely many, (Jacot, Gabriel, and Hongler 2018) showed theoretically that the infinitely-wide NTK (Θ^∞) remains constant during infinite-width training. Empirically, it has been shown by (Jacot, Gabriel, and Hongler 2018) and others (Seleznova and Kutyniok 2022) that for a finite-width architecture, the Θ asymptotically converges late into training, and as the finite-width increases, Θ converges towards Θ^∞ late into training.

4 Training-Free Signal: The Infinite-Width Gram-Matrix

In this section, we propose our infinite-width signal and illustrate that its computation is remarkably cheap and easily available to practitioners. In a nutshell, from the infinite-width NTK theory, we leverage the Gram-Matrix formed by Θ^∞. This matrix $\mathbf{H}^\infty$, as shown in Equation 3, is a positive-definite symmetric matrix that encodes all pairs of instances in $\mathcal{X}$ as input to Θ^∞. In other words, $\mathbf{H}^\infty$ encodes similarity in the context of infinite-width training.

$$\mathbf{H}^\infty = \begin{bmatrix} \Theta^\infty(x_1, x_1) & \Theta^\infty(x_1, x_2) & \dots & \Theta^\infty(x_1, x_N) \\ \Theta^\infty(x_2, x_1) & \Theta^\infty(x_2, x_2) & \dots & \Theta^\infty(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \Theta^\infty(x_N, x_1) & \Theta^\infty(x_N, x_2) & \dots & \Theta^\infty(x_N, x_N) \end{bmatrix} \tag{3}$$

Through extensive theoretical analysis (Jacot, Gabriel, and Hongler 2018; Arora et al. 2019; Sohl-Dickstein et al. 2020; Han et al. 2022), the infinite-width NTK (Θ^{∞}) is computable analytically and remarkably quickly with GPUs. For example, the `neural-tangents` software library (Novak et al. 2020) was developed to compute infinite-width NTKs for infinite-width versions of architectures in the deep learning framework JAX (Bradbury et al. 2018).

In Tables 1 and 2, we show that not only is Θ^{∞} cheap to compute, but $\mathbf{H}^{\infty}$ can be computed faster than training a large, deep neural net. Table 1 shows the time in seconds for us to compute the entire Gram-Matrix for common benchmark datasets (LeCun 1998; Xiao, Rasul, and Vollgraf 2017; Krizhevsky, Hinton et al. 2009).

We perform this analysis on five different architectures, all using ReLU activation functions: a feed forward network with 2 hidden layers (**2-L**), a feed forward network with 10 hidden layers (**10-L**), a CNN with 2 convolutional layers each with 64 3x3 kernels and 2 hidden layers (**CNN-1**), a CNN with 4 convolutional layers each with 256 3x3 kernels and 4 hidden layers (**CNN-2**), and VGG-19 (Simonyan and Zisserman 2023) (**CNN-3**).

In perspective, Table 2 gives the time to train the same architectures but with a large overparameterized finite-width architecture. All our results are run on a single A100 GPU.

We observe that not only is the infinite-with Gram-Matrix nearly 100x faster than the training time of the same large-but-finite version of the architecture, and nearly 400x faster for larger CNNs, but also the infinite-width Gram-Matrix only needs to be computed a single time regardless of the length of finite-width training. This shows infinite-width NTKs are wildly efficient to use with common hardware, providing a promising avenue for training-free "tricks".

Data Set	2-L	10-L	CNN-1	CNN-2	CNN-3
D-MNIST	25	56	50	89	6,067
F-MNIST	25	55	60	90	5,931
CIFAR10	25	46	90	153	5,390
CIFAR100	25	46	86	152	5,407

Table 1: Time in seconds to compute the Gram-Matrix of common benchmark image datasets using the infinite-width NTK (Θ^{∞}) across four architectures with infinite-widths.

Data Set	2-L	10-L	CNN-1	CNN-2	CNN-3
D-MNIST	715	5,377	820	49,714	19,247
F-MNIST	715	5,377	820	49,714	20,036
CIFAR10	715	4,563	718	53,567	16,746
CIFAR100	731	4,617	764	54,740	17,169

Table 2: Time in seconds to compute 100 epochs for finite-width architectures with hidden layers of 10,000 neurons.

5 Infinite-Width NTK for Data Valuation

In this section, we show how $\mathbf{H}^{\infty}$ provides a rich widely-applicable signal for improving downstream training before ever training a model. Namely, we will demonstrate its ability to (i) find well suited architectures for training, (ii) verify whether one set of pseudo-labels generated by one technique are better than those by other techniques, (iii) identify inherent bias and entangle inter-class relationships, and (iv) refurbish mislabeled instances to their correct class. In each subsection, we propose how $\mathbf{H}^{\infty}$ can be leveraged to conduct such a task, and then provide empirical results of its ability compared to actual model training.

5.1 Solving the Architecture Selection Task

Methods for selecting the best suited architecture falls into a category of AutoML methods called neural architecture search (NAS) (Chitty-Venkata et al. 2023), which seek to automate the selection of the neural nets for a given learning task. Extensive work has been conducted in this subdomain (Chitty-Venkata et al. 2023), where methods are either naively training models to search (Li and Talwalkar 2020), or training models with iterative adjustments (Liu, Simonyan, and Yang 2018).

For a given neural net architecture f with a corresponding infinite-width Gram-Matrix $\mathbf{H}_f^{\infty}$, the eigenvectors $\mathbf{V}$ in its eigen-decomposition ($\mathbf{V}\Lambda\mathbf{V}^{\top}$) explain the principal components in which convergence occurs in the case of infinite-width training, with data lying on eigenvectors corresponding to high-magnitude eigenvalues being learned first, and eigenvectors corresponding to low-magnitude eigenvalues being learned last (Jacot, Gabriel, and Hongler 2018). We propose that the degree in which low-magnitude eigenvectors of $\mathbf{H}_f^{\infty}$ clusters data based on the learning task demonstrates the ability for f to learn with greater generalization.

For example, the infinite-width NTK derived by (Arora et al. 2019) for a 2-layer network with ReLU activation is defined in Equation 4. Under the constraint[1] where $||\mathbf{x}||= 1$, we can trivially see that the way this 2-layer network encodes learning similarity is directly proportional to the cosine similarity between instances, as seen in Equation 5.

$$\Theta^{\infty}_{\text{2-L ReLU}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^{\top}\mathbf{x}_j(\pi - \arccos(\mathbf{x}_i^{\top}\mathbf{x}_j))}{2\pi} \quad (4)$$

$$\cos(\theta) = \frac{2\pi\Theta^{\infty}_{\text{2-L ReLU}}(\mathbf{x}_i, \mathbf{x}_j)}{\pi - \theta} \quad (5)$$

This relationship, however, is not universally true for all architectures of infinite width. Despite (Jacot, Gabriel, and Hongler 2018) proving that all infinite-width neural nets can perfectly learn any given training task on a dataset, we show that the behavior in which that learning occurs, even in the infinite-width, is **not** identical across architectures. This implies that infinite-width architectures may provide different NTK signals that are more closely related to their finite-width analogs than to other infinite-width architectures.

To demonstrate this, we devise 2 synthetic 32x32 image datasets: *Corners* and *Shapes*. The *Corners* dataset contains 4 classes where all images contain a single bright pixel. The

[1]Enforcing $||\mathbf{x}||= 1$ projects all data onto a hypersphere, ensuring $\mathbf{H}^{\infty}$ is positive definite: a critical assumption in NTK theory.

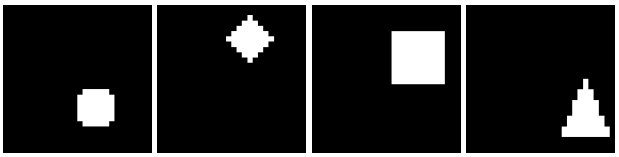

Figure 2: Example instances of each class in the *Shapes* dataset before augmenting with random pixel-level noise.

defining characteristic of each class is whether the single bright pixel exists in the top-right, top-left, bottom-right, or bottom-left of the image. The *Shapes* dataset also contains 4 classes. Instead, the defining characteristic depends on which of 4 shapes (circle, diamond, square, or triangle) is randomly placed somewhere in the image. An example of each class of *Shapes* is shown in Figure 2.

To add some additional variation, we add a small amount of random pixel-level noise to the images in each dataset. Since the classes of *Corners* are solely defined by pixel locations, it is purposefully designed for simple linear network to perform equal-to or better than a CNN. However, in the case of the *Shapes* where classes are pattern-based, the learning task is designed such that a finite-width CNN will perform better than a simple linear network at the task.

In Table 3, we show that by conducting Kernel PCA using the low-magnitude eigenvectors of $\mathbf{H}^{\infty}$, where $\mathbf{H}^{\infty}$ we were able to predict that both a simple 2-linear layer would be better suited for learning *Corners* than a CNN, whereas a CNN would be better suited for learning *Shapes* than a 2-linear layer network **without ever training a model.** Moreover, we show that using KPCA with an infinite-width NTK could accurately predict that a finite-width CNN would be better suited for learning a variety of benchmark image-based classification tasks than using a finite-width 2-linear layer with the image's flattened representations.

Furthermore, we demonstrate that using a suitable infinite-width architecture boosts the quality of information Θ^{∞} provides for NTK-driven signals. For example, a prior work (Nohyun, Choi, and Chung 2022) applied infinite-width NTK theory of infinitely-wide 2-layer networks, and developed a state-of-the-art signal, *Complexity-Gap*, for data pruning and identifying mislabeled data without training a model. By generalizing their method to work with Θ^{∞} beyond infinitely-wide 2-layer networks, we show in Table 4 that their method sees a significant boost in performance.

We also show that utilizing Kernel PCA using the infinite-width NTKs is on par with state-of-the-art NAS methods. We select 8 different neural net architectures, each with different numbers of hidden layers, convolutional layers, and kernel sizes. For the models with convolutional layers, we also have different architectures with different kernel sizes and stride lengths. We then *actually trained* the finite-width version of these models, and ranked their performances on a protected test set at the end. We then use our training-free method, to predict the ranking produced by actual training.

In Table 5, we see that our proposed method yielded similar rankings, and predicted the best model to choose correctly. To quantify this similarity, we use Ranked Biased Overlap (RBO) (Webber, Moffat, and Zobel 2010), a similarity measure between rankings ranging between 0 (worst) and 1 (best). Our method had an RBO score of 0.829 with respect to the ground-truth labels found via brute force.

Thus, we showed the powerful capabilities of the infinite-width Gram-Matrix for selecting well-suited architectures using principled theory related to the behavior of infinite-

Data Set	2-L	CNN-2
Digit MNIST	2.728 (25.274)	**3.785 (33.755)**
Fashion MNIST	3.488 (23.475)	**5.022 (31.799)**
CIFAR10	1.179 (2.234)	**1.186 (2.097)**
CIFAR100	1.893 (13.058)	**2.550 (19.421)**
Shapes	1.022 (1.400)	**1.057 (1.921)**
Corners	**1.051 (1.259)**	1.015 (1.132)

Table 3: Ratio of mean (standard deviation) between inter-class distances and intra-class distances when projecting datasets into the last 4 principal components of Θ^{∞}-KPCA using different infinite-wide architectures. Larger values correspond to classes being strongly clustered by KPCA.

Data Set	Noise Found After Checking 20% of Data			
	2-L	10-L	CNN-1	CNN-2
D-MNIST	93.70%	94.88%	94.70%	**95.50%**
F-MNIST	87.23%	**91.35%**	89.23%	91.25%
CIFAR10	54.05%	58.28%	58.85%	**59.80%**
CIFAR100	53.20%	57.60%	57.86%	**59.35%**
	AUC After Checking 100% of Data			
	2-L	10-L	CNN-1	CNN-2
D-MNIST	.9955	.9973	.9970	**.9978**
F-MNIST	.9823	**.9902**	.9865	.9899
CIFAR10	.8397	.8663	.8695	**.8765**
CIFAR100	.8387	.8636	.8663	**.8760**

Table 4: After infecting datasets with 20% random label noise, the percentage of noise found (top) and AUC (bottom) according to *Complexity GAP*: a signal designed specifically for 2-layer infinite-width NTKs (2-L). When modifying their method to use Θ^{∞} from architectures predicted by Θ^{∞}-KPCA to generalize better, the signal always improved.

Architecture	Θ^{∞}-KPCA (Ours)	Exhaustive Training
Arch 6.	(1)	(1)
Arch 2.	(3)	(2)
Arch 1.	(4)	(3)
Arch 4.	(2)	(4)
Arch 5.	(6)	(5)
Arch 7.	(7)	(6)
Arch 3.	(5)	(7)

Table 5: Predicted rankings of how well each finite-width architecture learns compared to the true ranking (brute force) found by training each model for 100 epochs on CIFAR-10.

width training. Additionally, the infinite-width Gram-Matrix yielded from an infinite-width architecture is more similarly related to its finite-width analogs than they are to other infinite-width architectures. This allows for future practitioners to be better equipped to quickly and cheaply conduct architecture selection with deeper insights.

5.2 Solving the Pseudo-Label Verification Task

Pseudo-label generation is a widely utilized field of methods for conducting semi-supervised learning by automatically generating labels for a given learning task. When employing an ensemble of pseudo-label techniques to generate labels (Rizve et al. 2021; Gilhuber et al. 2022; Kaul, Xie, and Zisserman 2022), practitioners want to identify the best labeling scheme generated. We show that such a need can be fulfilled by the infinite-width NTK.

Prior work studying the effects of finite-width NTKs (Seleznova and Kutyniok 2022) empirically showed that late into learning – specifically under multi-class classification – the finite-width NTKs consistently evolved towards producing a block-diagonal Gram-Matrix concerning class labels. Namely, they showed for a dataset $\mathcal{X}$ broken into K classes,

$$\mathcal{X} = \bigcup_{k=1}^{K} \mathcal{X}_k$$

that Equation 6, a measure of how block diagonal the Gram-Matrix is, increased during finite-width training.

$$\frac{1}{K}\sum_{k=1}^{K}\left[\frac{\sum\limits_{\substack{x_i,x_j\in\mathcal{X}_k\\ x_i\neq x_j}}\Theta(x_i,x_j)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} - \frac{\sum\limits_{\substack{x_i\in\mathcal{X}_k\\ x_j\notin\mathcal{X}_k}}\Theta(x_i,x_j)}{|\mathcal{X}_k|(|\mathcal{X}|-|\mathcal{X}_k|)}\right] \tag{6}$$

We propose utilizing the inverse of the infinite-width Gram matrix, $(\mathbf{H}^{\infty})^{-1}$, which encodes at each element $(\mathbf{H}^{\infty})^{-1}_{ij}$ the degree of orthogonality between $\mathbf{H}^{\infty}_{:,i}$ and $\mathbf{H}^{\infty}_{:,j}$ conditioned on the subspace formed by the remaining column vectors of $\mathbf{H}^{\infty}$. By constructing an $N \times K$ matrix $\mathbf{Y}$ where the i-th row is the 1-hot vector representation of the label assigned to the instance in the i-th row of $\mathbf{H}^{\infty}$ as described in Equations 7 and 8, the resulting product $\mathbf{Y}^{\top}(\mathbf{H}^{\infty})^{-1}\mathbf{Y} = \mathbf{Z} \in \mathbb{R}^{K\times K}$ provides a valuable signal.

$$\mathbf{y} = [y_1 \quad y_2 \quad \dots \quad y_K] \tag{7}$$

$$\mathbf{Y} = \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \vdots\\ \mathbf{y}_N\end{bmatrix} \tag{8}$$

Recalling Equation 2, we see that to increase the rate of change of the inference of an architecture f for all training instances of class $\mathcal{X}_k$ converge towards a desired label vector $\mathbf{y}_k$, that Equation 9 summarizes the cumulative correlations of the rates of change at a class-level resolution.

$$\mathbf{Z} = \mathbf{Y}^{\top}(\mathbf{H}^{\infty})^{-1}\mathbf{Y} \tag{9}$$

By modifying Equation 6 based off this insight under infinite-width regime, in Equation 10 we define a novel metric called *Infinite-Width Block Diagonalization Error*.

$$\mathcal{L}(\mathbf{Z}) = \frac{1}{K}\sum_{k=1}^{K}\frac{\mathbf{Z}_{kk}}{(\mathbf{Y}^{\top}\mathbf{1})_k\left[(\mathbf{Y}^{\top}\mathbf{1})_k - 1\right]} - \frac{\beta}{K^2-K}\sum_{k=1}^{K}\sum_{k\neq d}\left[\frac{\mathbf{Z}_{k,d}}{(\mathbf{Y}^{\top}\mathbf{1})_k(\mathbf{Y}^{\top}\mathbf{1})_d}\right] \tag{10}$$

Infinite-Width Block Diagonalization Error describes how lacking a labeling scheme is from exhibiting block diagonalization before training. In ideal training, $(\mathbf{H}^{\infty})^{-1}$ is truly block-diagonal, resulting in $\mathbf{Z}$ be a diagonalized matrix with strictly positive diagonal elements, as shown in Equation 11.

$$\mathbf{Z}^{*} = \begin{bmatrix}C_1 & 0 & \dots & 0\\ 0 & C_2 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & C_K\end{bmatrix} \tag{11}$$

Thus, assuming a suitable architecture is previously known for a given learning task, we can assert that noisier labeling schemes are labelings that alter $\mathbf{Y}$ such that $\mathcal{L}(\mathbf{Z})$ yields greater error. Table 6 shows 4 benchmark datasets, where a deep CNN network is trained from scratch, each time under 1 of 5 possible labeling schemes: (1) **1 Class**

Dataset	Label Scheme	$\mathcal{L}(\mathbf{Z})$	Trained Rank
Digit MNIST	70% Noise	29662.4	5 ✓
	30% Noise	20401.4	4 ✓
	10% Noise	11784.6	3 ✓
	Clean	6702.7	2 ✓
	1 Class	1.3	1 ✓
Fashion MNIST	70% Noise	30889.0	5 ✓
	30% Noise	21433.0	4 ✓
	10% Noise	13248.3	3 ✓
	Clean	8558.3	2 ✓
	1 Class	1.8	1 ✓
CIFAR-10	70% Noise	1387.0	5 ✓
	30% Noise	1269.4	4 ✓
	10% Noise	1159.7	3 ✓
	Clean	1109.8	2 ✓
	1 Class	0.2	1 ✓
CIFAR-100	70% Noise	19907.0	5 ✓
	30% Noise	19166.1	4 ✓
	10% Noise	18732.8	3 ✓
	Clean	18456.4	2 ✓
	1 Class	0.2	1 ✓

Table 6: The infinite-width block-diagonalization error $\mathcal{L}(\mathbf{Z})$ computed without training, and the ranking of lowest training loss after a large but finite-width deep CNN trained for 250 epochs (Trained Rank) using different labeling schemes. $\mathcal{L}(\mathbf{Z})$ perfectly predicts the real ranking.

– all training data is assigned to a single class, (2) **Clean** – all training data is assigned to its ground-truth labeling, (3-5) **Noise** – a percentage of instances are selected to be purposely mislabeled to a different class than its ground-truth labeling. When training under each labeling scheme, we track the loss throughout the 250 epochs of training.

Without knowing about the real training results nor knowing explicitly knowledge about ground truth labels or ground-truth class ratios, *Infinite-Width Block Diagonalization Error* correctly predicted the order of training losses during actual training. Moreover, *Infinite-Width Block Diagonalization Error* shows that learning under the "1 Class" labeling scheme is not only a trivial task, but orders of magnitude simpler than even learning the cleanly labeled task, a phenomenon that is simply explained by a model soley optimizing its bias terms to achieve perfect performance.

Therefore, for non-trivial pseudo-labels, *Infinite-Width Block Diagonalization Error* provides a meaningful signal in a simple-to-compute loss function for ranking the labeling quality produced by different labeling schemes.

5.3 Solving the Bias Identification Task

When conducting training based on a specific learning task, there may be latent patterns patterns of biases present that unsuspectingly impacts training. Tools to identify such biases and inter-class relationships concerning a learning task lets practitioners identify potential problems that may have occurred during data acquisition, as well as as what subpopulations may be underrepresented.

We show that inspecting the diagonal and non-diagonal elements of $\mathbf{Y}\left(\mathbf{H}^{\infty}\right)^{-1}\mathbf{Y}$ from Equation 9 provides deeper insight into the intra-class and inter-class relationships present during finite-width learning.

We show that the absolute magnitude of off-diagonal elements in the k-th column of $\mathbf{Y}\left(\mathbf{H}^{\infty}\right)^{-1}\mathbf{Y}$ well approximate the inter-class training dynamics with respect to the k-th class's dynamics. Similarly, the kth diagonal elements approximates the intra-class training dynamics of class k. More specifically, let f be a given architecture, and $\mathbf{Z} = \mathbf{Y}\left(\mathbf{H}_f^{\infty}\right)^{-1}\mathbf{Y}$ be the Gram-Label matrix product, then $|\mathbf{Z}_{ij}|$ demonstrates the non-orthogonal correlation between elements of class $\mathcal{X}_i$ and elements of class $\mathcal{X}_j$ when conditioned on the remaining training elements from other classes. When $|\mathbf{Z}_{ij}|$ is large, the column vectors of $\mathbf{H}_f^{\infty}$ that represent instances that are members of different classes are strongly correlated. When positive, this correlation means the column vectors are nearly parallel, whereas negative values correspond to anti-parallel column vectors.

We train a deep CNN model on a variety of benchmark datasets using their correct, ground-truth labeling, and track the average class accuracy per class during the course of training. We consider the model to infer the correct class during a given epoch if the class with the maximum inferred probability is the assigned class to learn. Additionally, for each class, we track the average probabilities yielded from the K-hot vector produced by the network. Thus, for each dataset with K classes, we gather 1 "intra-class learning speed" ranking, and K "inter-class" entanglement rankings at different checkpoints during training.

Dataset	Ranking	RBO Score
Digit MNIST	Intra-Class	0.962
	Inter-Class	0.904
Fashion MNIST	Intra-Class	0.963
	Inter-Class	0.915
CIFAR-10	Intra-Class	0.734
	Inter-Class	0.916
CIFAR-100	Intra-Class	0.557
	Inter-Class	0.688

Table 7: The predicted rankings computed without training using the magnitudes of off-diagonal elements of the infinite-width Gram-Label product, and the rankings produced after training a large but finite-width deep CNN trained for 250 epochs. Our proposed technique strongly predicts the ranking.

Without knowledge of training, we then predict these ground-truth orderings solely by inspecting the elements of $\mathbf{Z}$. Table 7 shows the similarity between the actual and predicted orderings by the Ranked Biased Overlap (RBO) (Webber, Moffat, and Zobel 2010), a measure for quantifying the similarity between rankings by taking in account both concordant pairs and relative ordering. RBO produces a similarity score ranging between 0 and 1, where 1 is completely identical. Our proposed method yields strong predictions of the rankings produced from actually training the finite-width version of model.

Predicting the ranking in which classes are entangled allows for investigations into the underlying biases present. For example, these rankings give us insights into what classes a model is biased towards learning confidently earlier in training. Conversely, for applications in critical domains such as in healthcare, these rankings can inform practitioners of what classes or subgroups may be subjected unfairly to poorer downstream inference if training is ended too soon.

5.4 Solving the Label Refurbishment Task

Methods to refurbish mislabeled data are vital when datasets may be corrupted by incorrect labelings, whether as a product of erroneous pseudo-labels automatically generated or inaccuracies by human labelers.

By treating $\mathbf{Y}$ as a matrix containing continuous values representing the probability of class membership, we can treat relabeling as a continuous optimization problem.

$$\mathcal{L}(\mathbf{Y}) = \frac{1}{K}\sum_{k=1}^{K}\frac{\left(\mathbf{Y}\left(\mathbf{H}_f^{\infty}\right)^{-1}\mathbf{Y}\right)_{kk}}{(\mathbf{Y}^{\top}\mathbf{1})_k\left[(\mathbf{Y}^{\top}\mathbf{1})_k - 1\right]} - \frac{\beta}{K^2 - K}\sum_{k=1}^{K}\sum_{k\neq d}\frac{\left(\mathbf{Y}\left(\mathbf{H}_f^{\infty}\right)^{-1}\mathbf{Y}\right)_{k,d}}{(\mathbf{Y}^{\top}\mathbf{1})_k(\mathbf{Y}^{\top}\mathbf{1})_d} \tag{13}$$

The gradient $-\nabla_{\mathbf{Y}}\mathcal{L}(\mathbf{Y})$ computes the local changes in class membership for each label vector $\mathbf{y}_i$ that minimizes

$$\nabla_{\mathbf{Y}}\mathcal{L}(\mathbf{Y}) =$$
$$\frac{1}{K}\sum_{k=1}^{K}\frac{\left(2\left(\mathbf{H}_f^{\infty}\right)^{-1}_{p,q}\mathbf{Y}_{q,k}\right)\left(\left((\mathbf{Y}^\top\mathbf{1})_k\right)^2-(\mathbf{Y}^\top\mathbf{1})_k\right)-\left(\mathbf{Y}\left(\mathbf{H}_f^{\infty}\right)^{-1}\mathbf{Y}\right)_{kk}\left(e_k^\top\left[(\mathbf{Y}^\top\mathbf{1})_k-1\right]+e_k^\top(\mathbf{Y}^\top\mathbf{1})_k\right)}{((\mathbf{Y}^\top\mathbf{1})_k\left[(\mathbf{Y}^\top\mathbf{1})_k-1\right])^2} \tag{12}$$
$$-\frac{\beta}{K^2-K}\sum_{k=1}^{K}\sum_{k\neq d}\frac{\left(\left(\mathbf{H}_f^{\infty}\right)^{-1}\mathbf{Y}e_d^\top+\left(\mathbf{H}_f^{\infty}\right)^{-1}\mathbf{Y}e_k^\top\right)\left((\mathbf{Y}^\top\mathbf{1})_k(\mathbf{Y}^\top\mathbf{1})_d\right)-\left(\mathbf{Y}\left(\mathbf{H}_f^{\infty}\right)^{-1}\mathbf{Y}\right)_{k,d}\left(e_k^\top(\mathbf{Y}^\top\mathbf{1})_d+e_d^\top(\mathbf{Y}^\top\mathbf{1})_k\right)}{((\mathbf{Y}^\top\mathbf{1})_k(\mathbf{Y}^\top\mathbf{1})_d)^2}$$

block-diagonalization error of $\mathbf{Y}\left(\mathbf{H}^{\infty}\right)^{-1}\mathbf{Y}$. Equation 12 describes the gradient $\nabla_{\mathbf{Y}}\mathcal{L}(\mathbf{Y})$ explicitly, where e_k is the k-th basis vector of $\mathbb{R}^K$, such that the k-th element of e_k is a 1, and all other elements are 0.

However, this optimization is rather delicate with the existence of spurious stationary points in the vector space of $\mathbb{R}^{N\times K}$, nor is the argument of the local minimum $\mathbf{Y}^*$ guaranteed to produce rows of $\mathbf{Y}^*$ that are discrete 1-hot vectors. Instead of allowing the optimization of $\mathbf{Y}$ to evolve completely continuously, we instead propose that the gradient information informs how to discretely modify $\mathbf{Y}$ such that each row maintains a 1-hot vector. Our proposed algorithm (Algorithm 1) takes a conservative approach in only relabeling 1 instance at a time. For each iteration, we select the instance whose maximum gradient has the greatest difference from the gradient corresponding to the element in the 1-hot vector containing the 1 (its current label).

To demonstrate the performance of our proposed algorithm, we test its ability to conduct label refurbishment on benchmark datasets with varying degrees of synthetically added random label noise: 10%, 20%, 30%, and 70% noise. Table 8 shows the percentage of mislabeled data that was correctly relabeled to the correct class. Thus, for 10-class datasets like Digit-MNIST, Fashion-MNIST, and CIFAR-10, a method that randomly re-assigns labels would perform at a 10%. This expectation would drop down to 1% for 100-class CIFAR-100. Additionally, we compare Algorithm 1's training-free relabeling capabilities against a recent noisy labeling learning method BARE (Patel and Sastry 2023). By allowing BARE to learn the robust intra-class patterns, we can ascertain the new relabeling by the final inference on each training instance at the end of training. Without requiring any model training, our proposed algorithm not only yields an equivalent performance, but also has an embarrassingly simple implementation and is relatively cheaper to compute. Thus, we have shown that by utilizing the label-gradients that minimize *Infinite-Width Block Diagonalization Error*, true labels can be recovered.

Algorithm 1: Label Refurbishment Using $(\mathbf{H}^{\infty})^{-1}$

Require: Total iterations L; initial one-hot label matrix $\mathbf{Y}$; vector initial positions of 1's in each row of $\mathbf{Y}$: $\mathbf{a}$

for L Iterations **do**
  Compute $-\nabla_{\mathbf{Y}}\mathcal{L}(\mathbf{Y})$ according to Eq. 12
  $\mathbf{b} \leftarrow -\nabla_{\mathbf{Y}}\mathcal{L}(\mathbf{Y})_{i,\mathbf{a}_i}\Big|_{i\in\{1,\ldots,N\}}$
  $\mathbf{c} \leftarrow \max_{j\in\{1,\ldots,K\}} -\nabla_{\mathbf{Y}}\mathcal{L}(\mathbf{Y})_{i,j}\Big|_{i\in\{1,\ldots,N\}}$
  $\mathbf{d} \leftarrow \mathbf{c}-\mathbf{b}$
  $I = \arg\max \mathbf{d}$
  $J = \arg\max -\nabla_{\mathbf{Y}}\mathcal{L}(\mathbf{Y})_{I,:}$
  $\mathbf{Y}_{I,:} \leftarrow \mathbf{e}_J$
end for
return $\mathbf{Y}$

Dataset	Noise Added	Ours	BARE
Digit MNIST	70%	25.21%	79.61%
	30%	85.66%	95.73%
	20%	85.00%	93.65%
	10%	83.50%	87.50%
Fashion MNIST	70%	13.36%	-
	30%	65.67%	-
	20%	64.25%	-
	10%	57.00%	-
CIFAR-10	70%	11.50%	-
	30%	16.16%	-
	20%	13.00%	-
	10%	9.00%	-

Table 8: After infecting datasets with different amounts of random label noise, the percentage of noise correctly refurbished (top) according to our method (Algorithm 1), compared to BARE, a noisy label learning algorithm, after 200 epochs of model learning. Entries with "-" indicate that after 400 epochs, BARE's model performance still did not achieve better training than naively training with label noise.

6 Conclusion

We showed that infinite-width NTKs provide a rich signal that expands the predictability of model training behavior for a given neural net architecture. Specifically, infinite-width NTK theory allows for investigating a given real neural net architecture by analyzing properties of an identical architecture in the infinite-width limit: a theoretical variant of the real architecture with hidden layers each containing infinitely many neurons. We have demonstrated the applicability of infinite-width NTKs as a low-cost powerful signal that can single-handedly realize various data valuation tasks such as architecture selection, pseudo-label verification, bias identification, and label refurbishment, before any real model has been trained.

Acknowledgements

Results in this paper were obtained in part using a high-performance computing system acquired through NSF MRI grant DMS-1337943 to WPI.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Arora, S.; Du, S.; Hu, W.; Li, Z.; and Wang, R. 2019. Fine-grained analysis of optimization and generalization for over-parameterized two-layer neural networks. In *International Conference on Machine Learning*, 322–332. PMLR.

Bietti, A.; and Mairal, J. 2019. On the inductive bias of neural tangent kernels. *Advances in Neural Information Processing Systems*, 32.

Bradbury, J.; Frostig, R.; Hawkins, P.; Johnson, M. J.; Leary, C.; Maclaurin, D.; Necula, G.; Paszke, A.; VanderPlas, J.; Wanderman-Milne, S.; and Zhang, Q. 2018. JAX: composable transformations of Python+NumPy programs.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, W.; Gong, X.; and Wang, Z. 2021. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. *arXiv preprint arXiv:2102.11535*.

Chitty-Venkata, K. T.; Emani, M.; Vishwanath, V.; and Somani, A. K. 2023. Neural Architecture Search Benchmarks: Insights and Survey. *IEEE Access*, 11: 25217–25236.

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Feldman, V.; and Zhang, C. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33: 2881–2891.

Gilhuber, S.; Jahn, P.; Ma, Y.; and Seidl, T. 2022. VERIPS: Verified Pseudo-label Selection for Deep Active Learning. In *2022 IEEE International Conference on Data Mining (ICDM)*, 951–956.

Girdhar, R.; El-Nouby, A.; Liu, Z.; Singh, M.; Alwala, K. V.; Joulin, A.; and Misra, I. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15180–15190.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943.

Han, I.; Zandieh, A.; Lee, J.; Novak, R.; Xiao, L.; and Karbasi, A. 2022. Fast Neural Kernel Embeddings for General Activations. In *Advances in Neural Information Processing Systems*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hron, J.; Bahri, Y.; Sohl-Dickstein, J.; and Novak, R. 2020. Infinite attention: NNGP and NTK for deep attention networks. In *International Conference on Machine Learning*, 4376–4386. PMLR.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Kaul, P.; Xie, W.; and Zisserman, A. 2022. Label, verify, correct: A simple few shot object detection method. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14237–14247.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Li, L.; and Talwalkar, A. 2020. Random search and reproducibility for neural architecture search. In *Uncertainty in artificial intelligence*, 367–377. PMLR.

Liu, H.; Simonyan, K.; and Yang, Y. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Mellor, J.; Turner, J.; Storkey, A.; and Crowley, E. J. 2021. Neural architecture search without training. In *International conference on machine learning*, 7588–7598. PMLR.

Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; and Gao, J. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Nam, J.; Cha, H.; Ahn, S.; Lee, J.; and Shin, J. 2020. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33: 20673–20684.

Nohyun, K.; Choi, H.; and Chung, H. W. 2022. Data valuation without training of a model. In *The Eleventh International Conference on Learning Representations*.

Novak, R.; Xiao, L.; Hron, J.; Lee, J.; Alemi, A. A.; Sohl-Dickstein, J.; and Schoenholz, S. S. 2020. Neural Tangents: Fast and Easy Infinite Neural Networks in Python. In *International Conference on Learning Representations*.

Park, D. S.; Lee, J.; Peng, D.; Cao, Y.; and Sohl-Dickstein, J. 2020. Towards nngp-guided neural architecture search. *arXiv preprint arXiv:2011.06006*.

Patel, D.; and Sastry, P. 2023. Adaptive sample selection for robust learning under label noise. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3932–3942.

Paul, M.; Ganguli, S.; and Dziugaite, G. K. 2021. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34: 20596–20607.

Rizve, M. N.; Duarte, K.; Rawat, Y. S.; and Shah, M. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*.

Seleznova, M.; and Kutyniok, G. 2022. Neural tangent kernel beyond the infinite-width limit: Effects of depth and initialization. In *International Conference on Machine Learning*, 19522–19560. PMLR.

Simonyan, K.; and Zisserman, A. 2023. Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015. doi: 10.48550. *arXiv preprint arXiv.1409.1556*.

Sohl-Dickstein, J.; Novak, R.; Schoenholz, S. S.; and Lee, J. 2020. On the infinite width limit of neural networks with a standard parameterization.

Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164.

Webber, W.; Moffat, A.; and Zobel, J. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4): 1–38.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zancato, L.; Achille, A.; Ravichandran, A.; Bhotika, R.; and Soatto, S. 2020. Predicting training time without training. *Advances in Neural Information Processing Systems*, 33: 6136–6146.