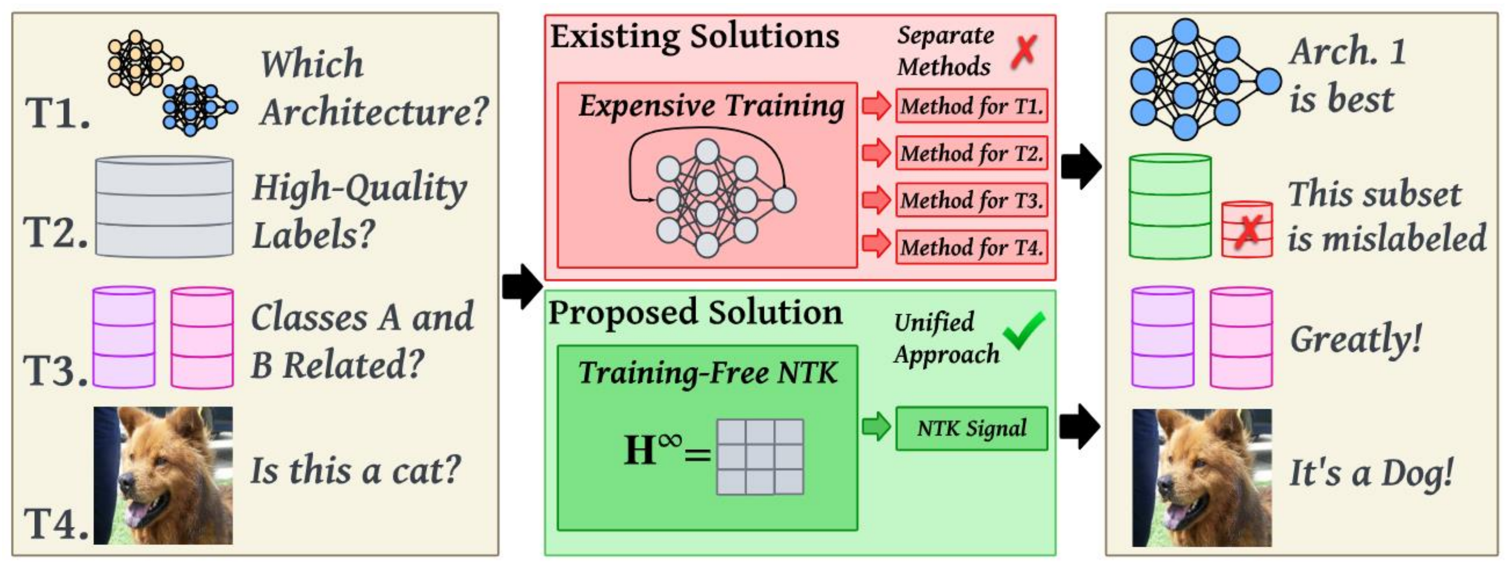
The Surprising Effectiveness of Infinite-Width NTKs for Characterizing and Improving Model Training

Joshua DeOliveira, Walter Gerych, Elke Rundensteiner

Background

Existing data-valuation methods rely on a suite of specially tailored methods, many **requiring expensive training** to be conducted [1]



We propose leveraging existing **NTK** theory to **side-step training**

Neural Tangent Kernels (NTKs)

A **similarity measure** of a neural net's parameter sensitivity [2]

$$\Theta(x', x''; \theta) = \langle \nabla_\theta f(x'; \theta), \nabla_\theta f(x''; \theta) \rangle$$

Computable numerically for real, finite-sized networks, or analytically for networks with **infinitely many neurons in its hidden layers** [3]

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.

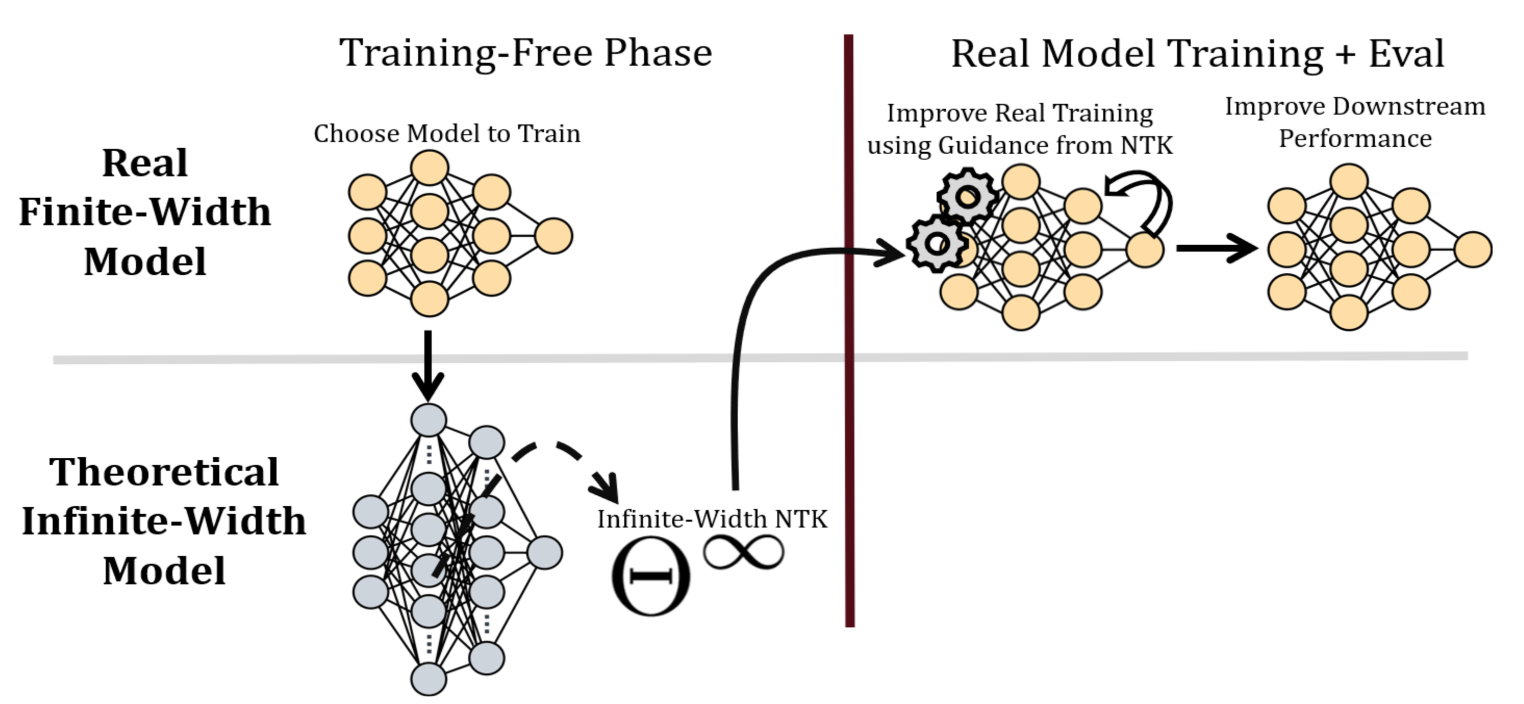
Data Set	2-L	10-L	CNN-1	CNN-2	CNN-3
D-MNIST	715	5,377	820	49,714	19,247
F-MNIST	715	5,377	820	49,714	20,036
CIFAR10	715	4,563	718	53,567	16,746
CIFAR100	731	4,617	764	54,740	17,169

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

Surprisingly, computing the infinite-width NTK is significantly **faster** than training a large, real neural net of the exact same architecture

Proposed Solution

Infinite-Width NTKs allows for an elegant, unified way of improving model training across 4 tasks **before any training is conducted**

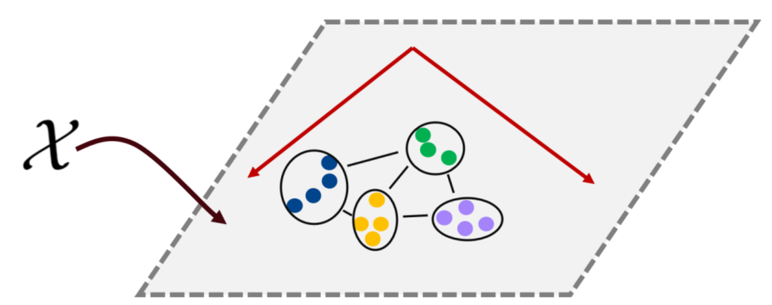


Architecture Selection

The **Gram Matrix** formed by the Infinite-Width NTK describes the similarities between all pairs of points present in the training set

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

The clustering of classes by **KPCA** using this Gram can inform us what architectures can naturally learn certain tasks better than others



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.

Pseudo-Label Verification

The matrix **Z** describes how **orthogonal each class's learning dynamics** are to each other within the context of their ground-truth labels

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

We propose a novel metric *Infinite-Width Block Diagonalization Error* using **Z** can **accurately identify which datasets** may contain noisy or incorrect training labels

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

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.

Inherent Bias Detection

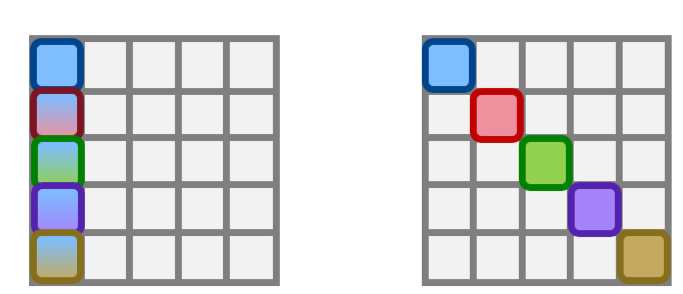
We can leverage **Z** to find what classes will be **highly entangled** during training

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

When there is no class entanglement **Z=Z***

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

The **columns** and **diagonal** encode inter-class and intra-class relationships



Class 1 Inter-Class Relationships — Intra-Class Relationships

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.

Our training-free rankings strongly predict the real class entanglement present in training

Label Refurbishment

By performing discrete alterations on labels to minimize *Infinite-Width Block Diagonalization Error*, incorrect labels can be **identified and refurbished**

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 −∇_Y 𝓛(Y) according to Eq. 12
    b ← −∇_Y 𝓛(Y)_{i,a_i} |_{i∈{1,...,N}}
    c ← max_{j∈{1,...,K}} −∇_Y 𝓛(Y)_{i,j} |_{i∈{1,...,N}}
    d ← c − b
    I = arg max d
    J = arg max −∇_Y 𝓛(Y)_{I,:}
    Y_{I,:} ← e_J
end for
return 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.

This simple, training-free approach yields competitive results to existing label refurbishment methods [4] that require training

Takeaways

Infinite-width NTKs provide a rich signal that expands the predictability of model training behavior for a given neural net architecture

Infinite-width NTKs are applicable as a low-cost yet powerful signal that can single-handedly realize various data valuation tasks

References

[1] Nam, J.; Cha, H.; Ahn, S.; Lee, J.; and Shin, J. 2020. Learn-ing from failure: De-biasing classifier from biased classifier. Advances in Neural Information Processing Systems,33: 20673–20684.
[2] Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, *31*.
[3] 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 Inter-national Conference on Learning Representations.
[4] 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.

Acknowledgements

DAISY Research Lab & NSF Grant DMS-1337943