FAST ADAPTATION WITH LINEARIZED NEURAL NETWORKS

WESLEY MADDOX [1], SHUAI TANG [2], PABLO GARCIA MORENO [3], ANDREW GORDON WILSON [1], ANDREAS DAMIANOU [3]

1 New York University

2 UC San Diego

3 Amazon

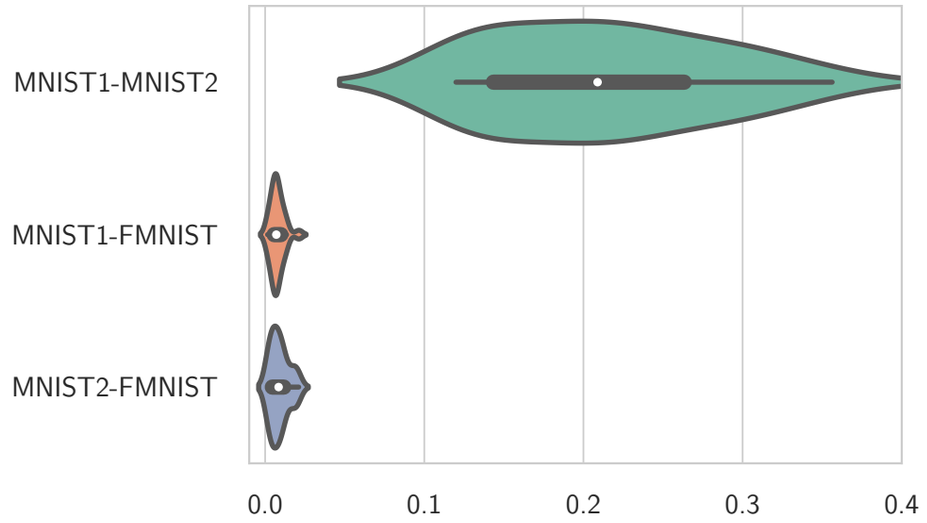
2021

MOTIVATIONS

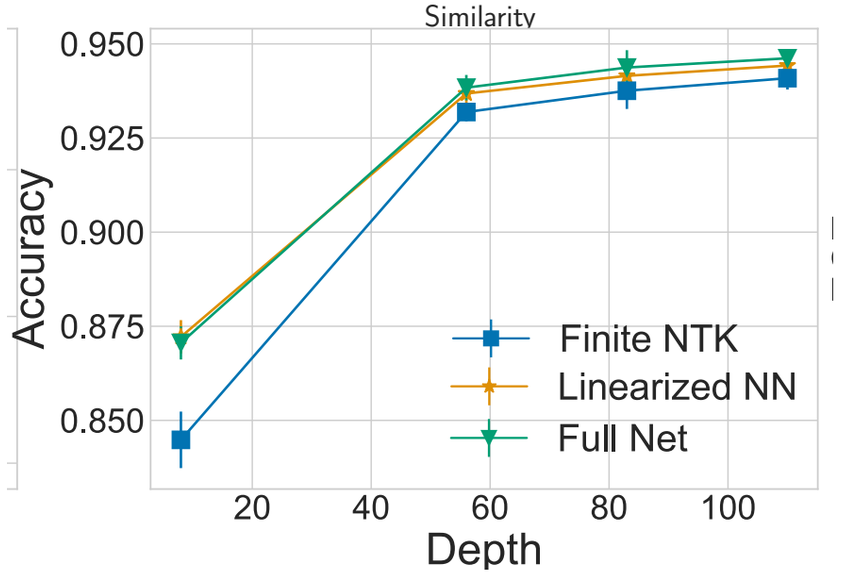
- Our **goal** is to create **efficient** methods that capture the **inductive biases** of neural networks but enable uncertainty representation and fast analytic transfer learning.

We start with two experimental observations:

- Neural network features are **similar across related tasks** (MNIST1-MNIST2) and re-training the network.



- Linearized neural networks retain many of the **strong inductive biases** of neural networks. Accuracy of the finite NTK on CIFAR10 is ~93% compared to ~94% for the full network.



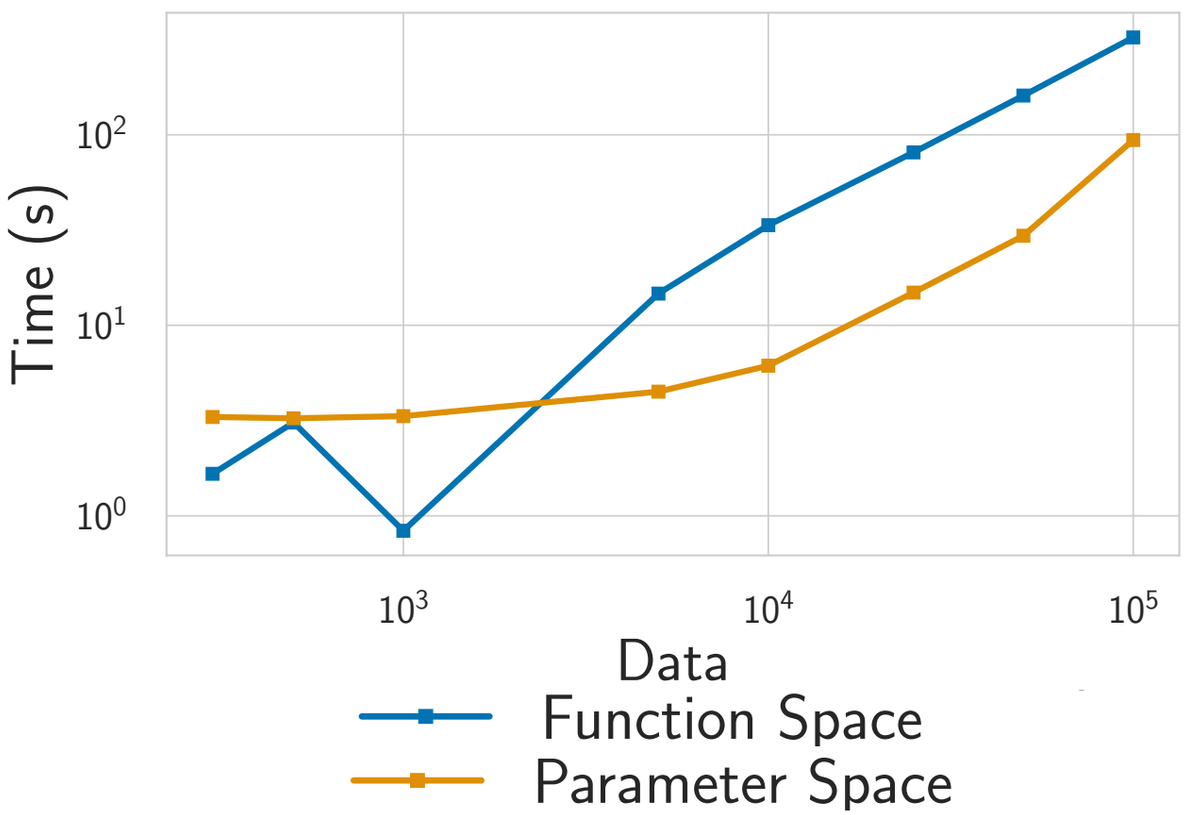
TRANSFER LEARNING APPROACH

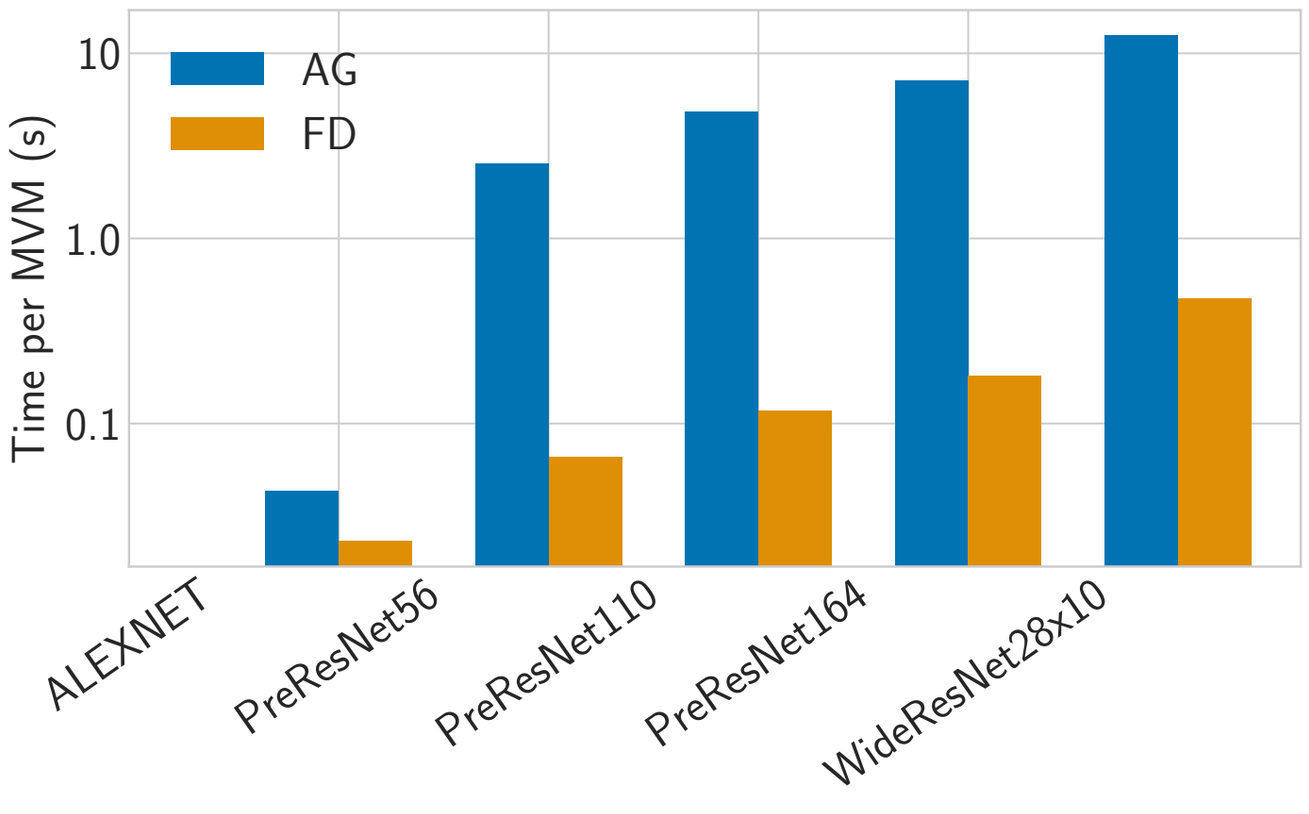
- Train a neural network on source task
- Linearize (convert to GP) and evaluate on target task
- Features are Jacobian matrix, $\boldsymbol{J}_\theta$, of the neural net
- Then use GP predictive equations w/ target data:

$$f^*|x^*, \mathcal{D} \sim \mathcal{N}\left(\boldsymbol{J}_\theta^{*T}\boldsymbol{J}_\theta(\boldsymbol{J}_\theta^\top\boldsymbol{J}_\theta + \sigma^2\boldsymbol{I}_n)^{-1}\boldsymbol{y},\right.$$
$$\left.\sigma^2\boldsymbol{J}_\theta^{*T}(\boldsymbol{I}_p - \boldsymbol{J}_\theta(\boldsymbol{J}_\theta^\top\boldsymbol{J}_\theta + \sigma^2\boldsymbol{I}_n)^{-1}\boldsymbol{J}_\theta^\top)\boldsymbol{J}_\theta^*\right)$$

INFERENCE

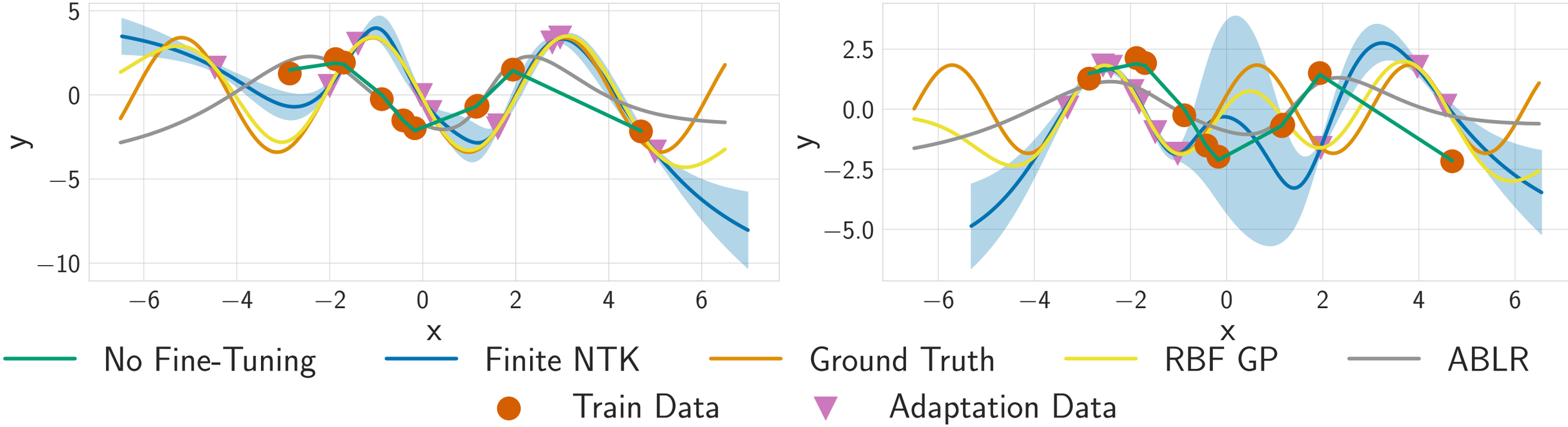
- Use fast Jacobian-vector and vector-Jacobian products alongside conjugate gradient methods for GP inference
  - Can also exploit Fisher vector products for even faster inference on some problems.





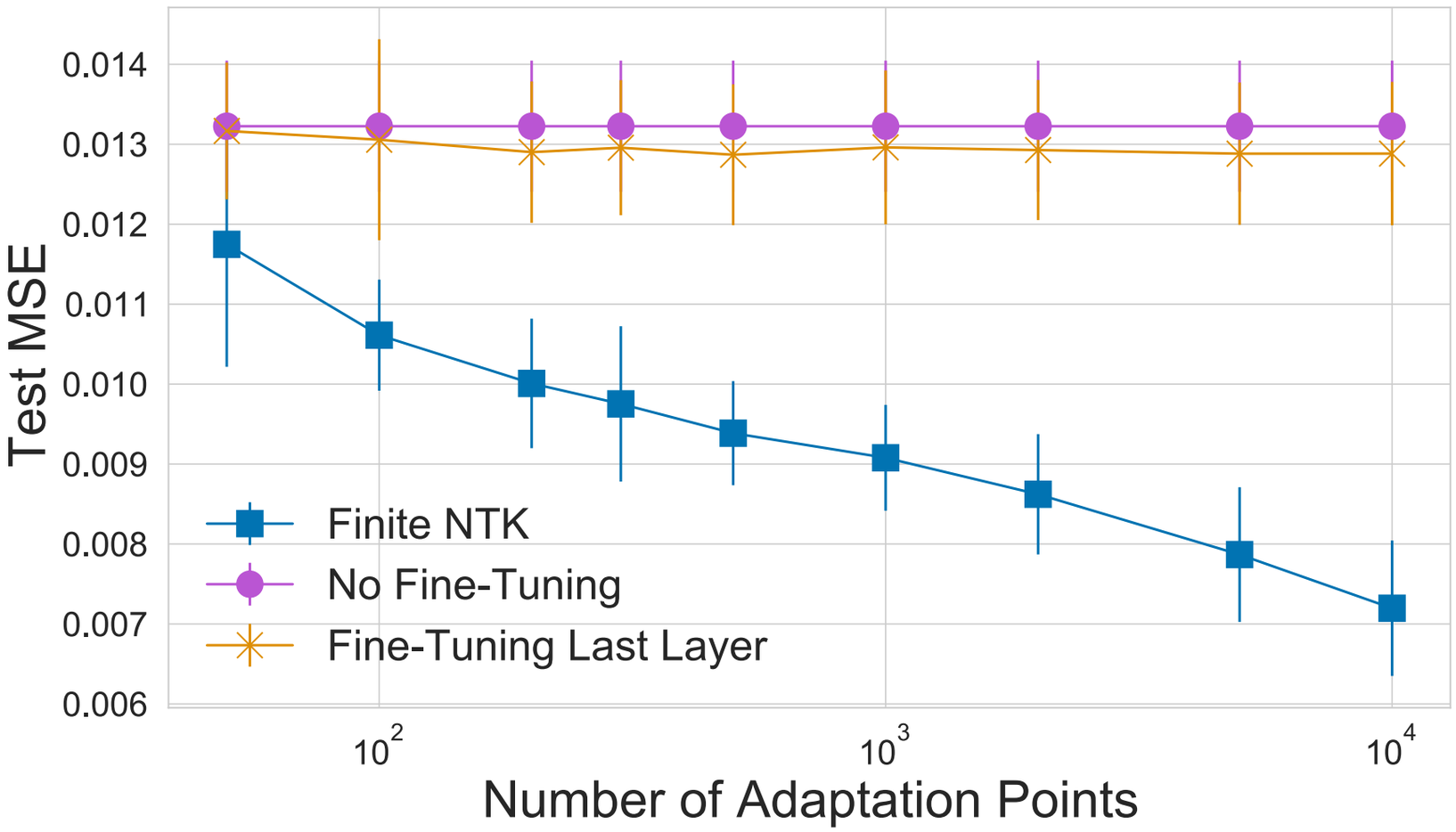
EXPERIMENTS

- Take a 3 layer NN trained on one sine curve, then transfer it to another
  - Finite NTK matches performance of an RBF GP



EXPERIMENTS

- Finite NTK outperforms last layer fine-tuning in the small data setting and when the dataset is very noisy (malaria incidence dataset over several years)



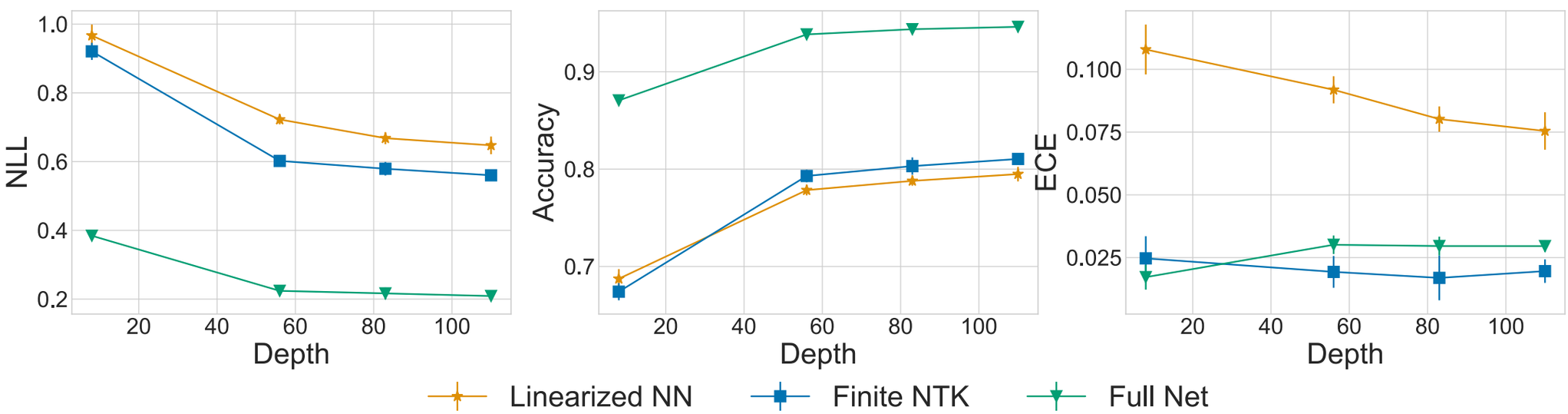
EXPERIMENTS

- Can expand to deep nets on classification tasks too with approximate inference!

Table 1: Test Accuracy on CIFAR10 for linearizing different layers of the BiGAN encoder. $*$ denotes numbers reproduced from Table 1 of Mu et al. (2020), which are included as baselines. Using all of the layers produces slightly better results than using only the top layers. All linearization settings are best compared to the activation baseline of 62.87% which consists of training a classifier on top of the feature extractor. Fine-tuning all of the layers is also better overall.

Layers	Top Layer	Top 2 Layers	Top 3 Layers	All Layers	All Layers (VI)
Fine-tuning	71.78*	73.18*	74.30*	*77.25*	-
Finite NTK	70.28*	71.08*	70.64*	***72.65***	*71.89*

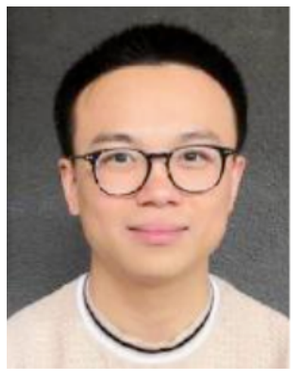
PreResNets: CIFAR-10 to STL-10



CONCLUSION

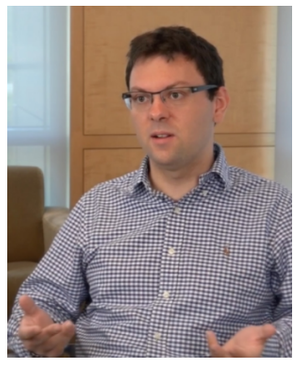
- Finite NTK for probabilistic transfer learning
- Can take any trained neural network and easily perform probabilistic transfer learning with it
- Code: https://github.com/amzn/xfer/linear_ntk
- Paper: https://arxiv.org/abs/2103.01439

Collaborators:

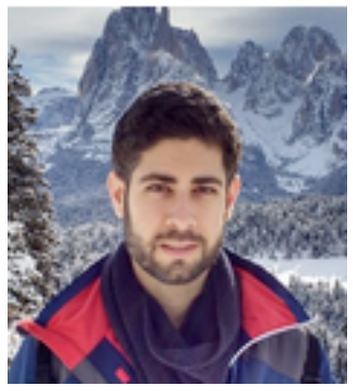


Shuai Tang,
UCSD -> AWS AI

Pablo Garcia Moreno
Amazon



Andrew Gordon Wilson,
NYU



Andreas Damianou,
Amazon