

---

# ON THE INFINITE WIDTH LIMIT OF NEURAL NETWORKS WITH A STANDARD PARAMETERIZATION

---

Jascha Sohl-Dickstein, Roman Novak, Samuel S. Schoenholz, Jaehoon Lee

{jaschasd, romann, schsam, jaehlee}@google.com

January 28, 2020

## ABSTRACT

There are currently two parameterizations used to derive fixed kernels corresponding to infinite width neural networks, the NTK (Neural Tangent Kernel) parameterization and the naive standard parameterization. However, the extrapolation of both of these parameterizations to infinite width is problematic. The standard parameterization leads to a divergent neural tangent kernel while the NTK parameterization fails to capture crucial aspects of finite width networks such as: the dependence of training dynamics on relative layer widths, the relative training dynamics of weights and biases, and a nonstandard learning rate scale. Here we propose an improved extrapolation of the standard parameterization that preserves all of these properties as width is taken to infinity and yields a well-defined neural tangent kernel. We show experimentally that the resulting kernels typically achieve similar accuracy to those resulting from an NTK parameterization, but with better correspondence to the parameterization of typical finite width networks. Additionally, with careful tuning of width parameters, the improved standard parameterization kernels can outperform those stemming from an NTK parameterization. We release code implementing this improved standard parameterization as part of the Neural Tangents library [24] at <https://github.com/google/neural-tangents>.

## 1 Introduction

Infinite width Bayesian [21, 15, 17, 18, 23, 9, 7, 28, 29, 6] and gradient descent trained [12, 16, 5, 28, 13, 8, 3, 1, 2, 26] neural networks are an area of active and extremely promising work. There are currently two parameterizations used to derive fixed kernels corresponding to infinite width neural networks<sup>1</sup>: the NTK parameterization [12, §2]; and the naive standard parameterization [25, §2.1]; [10, 11]. However, the extrapolations of both of these parameterizations to infinite width fail to capture crucial aspects of finite width networks:

- In finite width networks, differences in relative layer widths can have a profound effect on training dynamics. Under the NTK parameterization, as layer width goes to infinity, relative layer width has no effect on training dynamics or predictions.
- As the naive standard parameterization is extended to large widths, the largest stable learning rate scales like  $\frac{1}{\text{width}}$  [14, Theorem 7]; [25, §H]. A learning rate that goes to zero as width goes to infinity poses a variety of practical and theoretical challenges, including a neural tangent kernel with entries that diverge to infinity.
- At finite width, convolutional networks with an NTK parameterization have been reported to generalize more poorly than those with a standard parameterization [25, §I] (though we do not consistently reproduce this relationship in our own experiments, see Figure 3).
- For neither NTK nor naive standard parameterization do infinite width learning rates agree closely with those typically used to train finite width standard parameterization networks.
- The relative learning dynamics of bias and weight parameters are different in the NTK parameterization than they are for a standard parameterization finite-width network.

In this note we propose an improved extrapolation of the standard parameterization to infinite width that resolves these inconsistencies while simultaneously leading to a well-defined neural tangent kernel. Namely, in this parameterization

---

<sup>1</sup>Another line of work applies a different scaling, and derives non-fixed infinite width kernels [20, 19, 4, 22].

Parameterization	Standard (naive)	NTK	Standard (improved)
Layer equation, $x^{l+1} =$	$W^l x^l + b^l$	$\frac{\sigma_w}{\sqrt{sN^l}} W^l x^l + \sigma_b b^l$	$\frac{1}{\sqrt{s}} W^l x^l + b$
Weight shape, $W^l \in$	$\mathcal{R}^{sN^{l+1} \times sN^l}$		
$W$ initialization, $W_{ij}^l \sim$	$\mathcal{N}\left(0, \frac{\sigma_w^2}{sN^l}\right)$	$\mathcal{N}(0, 1)$	$\mathcal{N}\left(0, \frac{\sigma_w^2}{N^l}\right)$
$b$ initialization, $b_i^l \sim$	$\mathcal{N}(0, \sigma_b^2)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, \sigma_b^2)$
NNGP, $s \rightarrow \infty$ , $K^{l+1} =$	$\sigma_w^2 K^l + \sigma_b^2$		
NTK, $s \rightarrow \infty$ , $\Theta^{l+1} =$	diverges	$\sigma_w^2 K^l + \sigma_b^2 + \sigma_w^2 \Theta^l$	$N^l K^l + 1 + \sigma_w^2 \Theta^l$

Table 1: Equations describing a **fully connected layer** for each parameterization, both for a finite width network and for the corresponding infinite width NNGP and NT kernels. Here  $N^l$  is the baseline (finite network) width of layer  $l$ , and  $s$  is a width-scaling factor that is taken to  $\infty$  for infinite width networks.

Parameterization	Standard (naive)	NTK	Standard (improved)
Layer equation, $x_{i,p}^{l+1} =$	$W_{i,j,m}^l x_{j,p+m}^l + b_i^l$	$\frac{\sigma_w}{\sqrt{sN^l M}} W_{i,j,m}^l x_{j,p+m}^l + \sigma_b b_i^l$	$\frac{1}{\sqrt{s}} W_{i,j,m}^l x_{j,p+m}^l + b_i^l$
Weight shape, $W^l \in$	$\mathcal{R}^{sN^{l+1} \times sN^l \times M}$		
$W$ initialization, $W_{ijm}^l \sim$	$\mathcal{N}\left(0, \frac{\sigma_w^2}{sN^l M}\right)$	$\mathcal{N}(0, 1)$	$\mathcal{N}\left(0, \frac{\sigma_w^2}{N^l M}\right)$
$b$ initialization, $b_i^l \sim$	$\mathcal{N}(0, \sigma_b^2)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, \sigma_b^2)$
NNGP, $s \rightarrow \infty$ , $K^{l+1} =$	$\sigma_w^2 \mathcal{A}(K^l) + \sigma_b^2$		
NTK, $s \rightarrow \infty$ , $\Theta^{l+1} =$	diverges	$\sigma_w^2 \mathcal{A}(K^l) + \sigma_b^2 + \sigma_w^2 \mathcal{A}(\Theta^l)$	$N^l M \mathcal{A}(K^l) + 1 + \sigma_w^2 \mathcal{A}(\Theta^l)$

Table 2: Equations describing a **convolutional layer** for each parameterization, both for a finite width network and for the corresponding infinite width NNGP and NT kernels. We use Einstein notation for summation – indices that appear only in a single term are implicitly summed over.  $M$  is the number of spatial positions in the convolution kernel,  $m$  indexes over spatial locations within the kernel,  $p + m$  corresponds to input spatial location  $p$  offset by  $m$ ,  $N^l$  is the baseline (finite network) channel count of layer  $l$ ,  $\mathcal{A}(\cdot)$  is the diagonal averaging operator defined in Xiao et al. [27] and Novak et al. [23, §2.2.1], and  $s$  is a width-scaling factor that is taken to  $\infty$  for infinite channel count networks.

the resulting infinite width network maintains a learning rate scale that agrees with that used to train the original network, preserves the impact of relative layer widths on training dynamics for finite width networks, and similarly preserves the relative training dynamics of weights and biases.

## 2 Improved standard parameterization

Affine layers in neural networks are typically written as,

$$z^{l+1} = W^l y^l + b^l \quad (1)$$

where  $z^l$  are pre-activations,  $y^l = \phi(z^l)$  are activations,  $W^l$  are weights, and  $b^l$  are biases. To preserve the scale of the pre-activations as the width of the network,  $N^l$ , is varied one typically initializes the weights as  $W^l \sim \mathcal{N}(0, \sigma_w^2/N^l)$  and biases as  $b^l \sim \mathcal{N}(0, \sigma_b^2)$ . However, as was noted in [12], this leads to divergent gradient flow dynamics as  $N^l \rightarrow \infty$ . In [12], the authors resolve this situation by using an alternative parameterization where affine layers are written as,

$$z^{l+1} = \frac{\sigma}{\sqrt{N^l}} \omega^l y^l + b^l \quad (2)$$

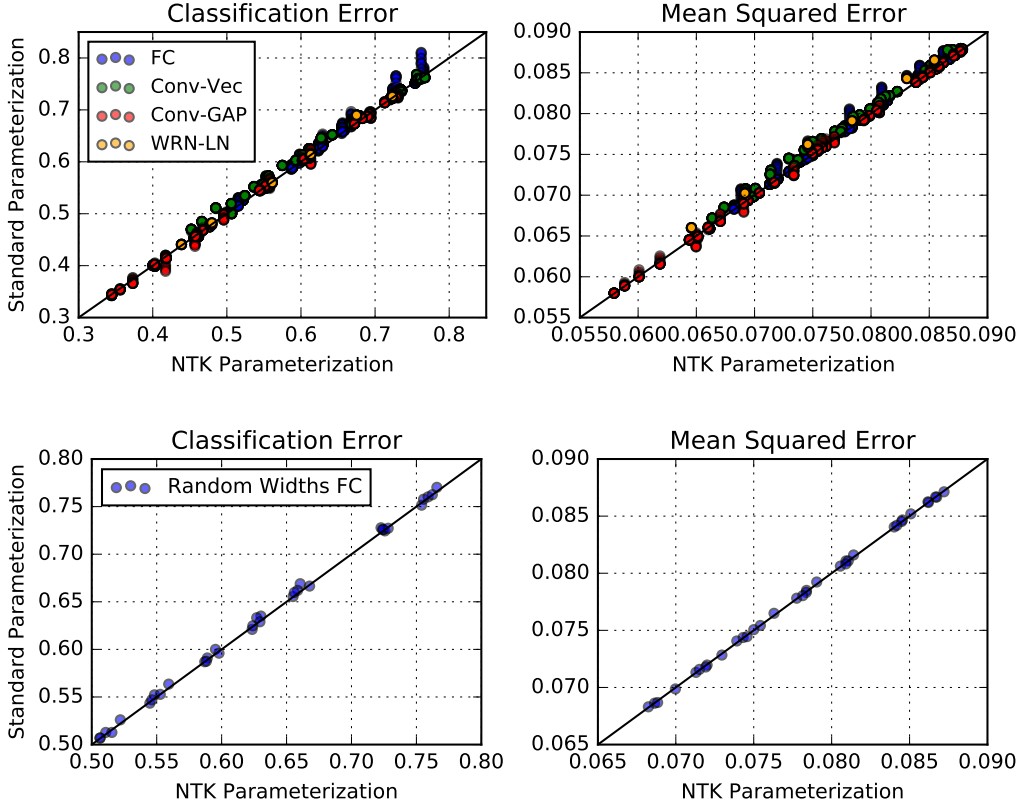


Figure 1: Infinite width networks with various architectures achieve similar error when using the improved standard parameterization or the NTK parameterization, while the improved standard parameterization better matches properties of typical finite width networks. Each point compares the neural tangent kernel prediction error for the same architecture on CIFAR-10, but using NTK (x-axis) or improved standard (y-axis) parameterization. (*Upper*) Each point corresponds to varying training set size ( $\{80, 160, 400, 800, 2000, 4000, 8000\}$ ), depth ( $\{1, 2, 4, 8, 16\}$  for FC / Conv, fixed number of block of 4 for WRN) and widths ( $\{2^k | k = 0, \dots, 13\}$  for FC / Conv and widening factor  $\{2^k | k = -4, \dots, 2\} \cup \{10, 16, 64, 256\}$  for WRN). **FC** is fully connected network with constant hidden width and **Conv-Vec** / **GAP** correspond to constant channel convolutional neural networks without / with global average pooling. **WRN-LN** is Wide Residual Network with four residual blocks and Batch Normalization layer replaced with Layer Normalization. (*Lower*) Each layer width of fully connected architecture are randomly sampled from  $2^k$  with  $k \in \{3, \dots, 13\}$ .

where  $\omega^l \sim \mathcal{N}(0, 1)$ . This leads to a well-behaved infinite-width limit, but involves a number of inconsistencies relative to standard neural networks.

The core idea here is to write the width of the neural network in each layer in terms of an auxiliary parameter,  $s$ ,  $n^l = sN^l$ . We then write an affine layer as,

$$z^{l+1} = \frac{1}{\sqrt{s}} W^l y^l + b^l \quad (3)$$

The infinite width limit can be taken by letting  $s \rightarrow \infty$ . The parameter variances  $\sigma_w^2, \sigma_b^2$  and original layer widths  $N^l$  instead appear in the variance of the initializer (as is typically done for finite width networks). A complete set of equations describing an affine layer, and corresponding infinite width kernels, for this parameterization are given in Tables 1 and 2, for fully connected and convolutional architectures respectively.

A formal proof of convergence of the improved standard parameterization to the specified kernels is beyond the scope of this short note. However, we observe that the proof technique in Lee et al. [16, §Apps. F, G] applies with minimal modification. Additionally, Monte Carlo validation of the correctness of the introduced kernels is performed as part of the Neural Tangents [24] unit test suite.

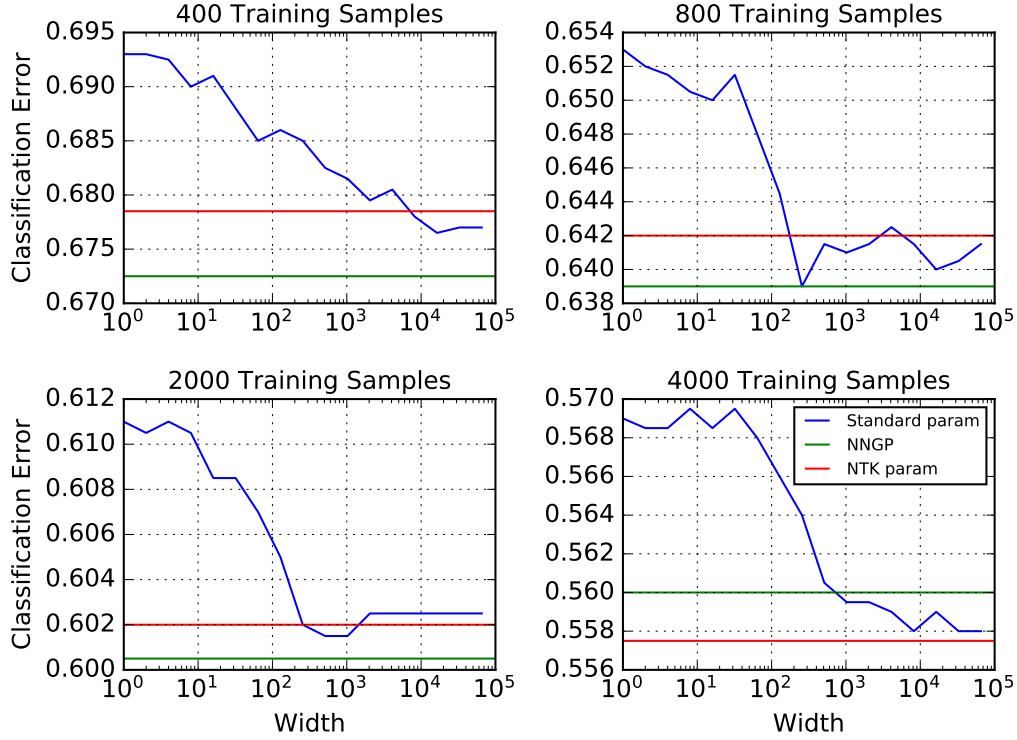


Figure 2: For fully connected networks, the neural tangent kernel prediction for the improved standard parameterization can outperform the NTK parameterization, especially when the layer widths  $N^l$  used in the standard parameterization are tuned. Experiments are performed on the CIFAR-10 dataset with networks corresponding to 5 hidden layers.

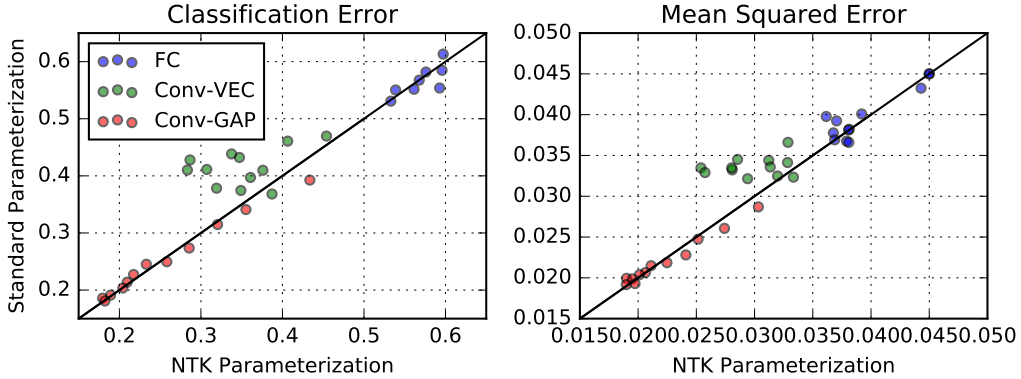


Figure 3: SGD trained finite width neural networks perform similarly when using the standard parameterization or the NTK parameterization. For all experiments, the network was trained with an MSE loss on the full CIFAR-10 dataset (45k/5k/10k split). Each point in **FC** corresponds to varying width  $\{2^k | k = 0, \dots, 12\}$ , and each point in **Conv-VEC** and **Conv-GAP** corresponds to varying number of channels  $\{8, 11, 16, 23, 32, 45, 64, 90, 128, 181, 256, 362, 512\}$ . All networks are ReLU networks with  $\sigma_w^2 = 2.0$ ,  $\sigma_b^2 = 0.0$ . They were trained with vanilla SGD without L2 regularization and data augmentation. Constant learning rate was grid searched over 20 log spaced values within  $[0.01, 100]$ . For standard parameterization learning rate is divided by  $\max(N^l)$ . **FC** networks were trained with batch size 1024 for 3,000 epochs whereas **Conv** networks were trained with batch size 256 for 10,000 epochs.

### 3 Experiments

In this section, we study empirical properties of infinite and finite width networks stemming from both the NTK and improved standard parameterization. All of the experiments in this section were done using Neural Tangents library [24]. Here we focus on kernels corresponding to ReLU networks with  $\sigma_w^2 = 2.0$ ,  $\sigma_b^2 = 0.1$ .

In Figure 1 we compare the predictions of kernels for pairs of identical networks, but using the improved standard or NTK parameterization. We find that the performance of the kernels resulting from the two parameterizations are extremely similar, while the training dynamics of the improved standard parameterization network are expected to better match those of typical finite width networks. In Figure 2 we show that if the width parameter  $N^l$  is carefully tuned, then the neural tangent kernel for a fully connected network using the improved standard parameterization can outperform the kernel for an NTK parameterized network. In Figure 3, we show that random finite width networks using the standard and NTK parameterization perform similarly.

### 4 Discussion

The analytic forms for the various kernels inspire some additional interesting observations:

- For the NTK parameterization, the kernel resulting from a Bayesian neural network and from gradient descent training of the readout layer of an infinite width network are the same. For the both the naive and improved standard parameterization however, the two differ.
- For neural networks with a standard parameterization, the magnitude of the contribution of the bias to the neural tangent kernel (and thus to learning dynamics) remains constant with increasing width. However, the contribution of the weights to the learning dynamics grows like  $N^l$ . We should thus expect that as networks become wide, the role played by the bias in training becomes less important.

In this note, we introduced an improved extrapolation of finite width networks to infinite width that better matches the parameterization and learning dynamics of typical finite width networks. It is our hope that this will enable theory and experiments with infinite width networks to better explain the behavior of practical finite width networks.

### References

- [1] Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. (2019a). On exact computation with an infinitely wide neural net. In *Advances In Neural Information Processing Systems*.
- [2] Arora, S., Du, S. S., Li, Z., Salakhutdinov, R., Wang, R., and Yu, D. (2019b). Harnessing the power of infinitely wide deep nets on small-data tasks.
- [3] Bietti, A. and Mairal, J. (2019). On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*.
- [4] Chizat, L. and Bach, F. (2018). On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046.
- [5] Chizat, L., Oyallon, E., and Bach, F. (2019). On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*.
- [6] De Palma, G., Kiani, B., and Lloyd, S. (2019). Random deep neural networks are biased towards simple functions. In *Advances in Neural Information Processing Systems*, pages 1962–1974.
- [7] Du, S. S., Hou, K., Salakhutdinov, R. R., Póczos, B., Wang, R., and Xu, K. (2019). Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 5724–5734. Curran Associates, Inc.
- [8] Dyer, E. and Gur-Ari, G. (2019). Asymptotics of wide networks from feynman diagrams. *arXiv preprint arXiv:1909.11304*.
- [9] Garriga-Alonso, A., Rasmussen, C. E., and Aitchison, L. (2019). Deep convolutional networks as shallow gaussian processes. In *International Conference on Learning Representations*.
- [10] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.
- [11] 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*, pages 770–778.

- [12] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*.
- [13] Jacot, A., Gabriel, F., and Hongler, C. (2019). Freeze and chaos for dnns: an NTK view of batch normalization, checkerboard and boundary effects. *arXiv preprint arXiv:1907.05715*.
- [14] Karakida, R., Akaho, S., and Amari, S.-i. (2018). Universal statistics of fisher information in deep neural networks: mean field approach. *International Conference on Artificial Intelligence and Statistics*.
- [15] Lee, J., Bahri, Y., Novak, R., Schoenholz, S., Pennington, J., and Sohl-dickstein, J. (2018). Deep neural networks as gaussian processes. In *International Conference on Learning Representations*.
- [16] Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*.
- [17] Matthews, A., Hron, J., Rowland, M., Turner, R. E., and Ghahramani, Z. (2018a). Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*.
- [18] Matthews, A. G. d. G., Rowland, M., Hron, J., Turner, R. E., and Ghahramani, Z. (2018b). Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*.
- [19] Mei, S., Misiakiewicz, T., and Montanari, A. (2019). Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. *Annual Conference on Learning Theory*.
- [20] Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671.
- [21] Neal, R. M. (1994). Priors for infinite networks (tech. rep. no. crg-tr-94-1). *University of Toronto*.
- [22] Nguyen, P.-M. (2019). Mean field limit of the learning dynamics of multilayer neural networks. *ArXiv*, abs/1902.02880.
- [23] Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. (2018). Bayesian deep convolutional networks with many channels are gaussian processes.
- [24] Novak, R., Xiao, L., Hron, J., Lee, J., Sohl-Dickstein, J., and Schoenholz, S. S. (2020). Neural tangents: Fast and easy infinite neural networks in python. <https://github.com/google/neural-tangents>.
- [25] Park, D. S., Sohl-Dickstein, J., Le, Q. V., and Smith, S. L. (2019). The effect of network width on stochastic gradient descent and generalization: an empirical study. In *International Conference on Machine Learning*.
- [26] Schwartz-Ziv, R. and Alemi, A. A. (2019). Information in infinite ensembles of infinitely-wide neural networks. *arXiv preprint arXiv:1911.09189*.
- [27] Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., and Pennington, J. (2018). Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*.
- [28] Yang, G. (2019a). Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*.
- [29] Yang, G. (2019b). Wide feedforward or recurrent neural networks of any architecture are gaussian processes. In *Advances in Neural Information Processing Systems*, pages 9947–9960.