The Orthogonality of Weight Vectors: The Key Characteristics of Normalization and Residual Connections

Zhixing Lu , **Yuanyuan Sun*** , **Zhihao Yang** , **Qin Zhou** , **Hongfei Lin**

College of Computer Science and Technology, Dalian University of Technology, Dalian, 116081, Liaoning, China

{syuan,yangzh,hflin}@dlut.edu.cn, {lzx2021, 2428074877}@mail.dlut.edu.cn

Abstract

Normalization and residual connections find extensive application within the intricate architecture of deep neural networks, contributing significantly to their heightened performance. Nevertheless, the precise factors responsible for this elevated performance have remained elusive. Our theoretical investigations have unveiled a noteworthy revelation: the utilization of normalization and residual connections results in an enhancement of the orthogonality within the weight vectors of deep neural networks. This, in turn, induces the Gram matrix of neural network weights to exhibit a pronounced tendency towards strict diagonal dominance, thereby amplifying the neural network's capacity for feature learning. Meanwhile, we have designed the parameters independence index (PII) to precisely characterize the orthogonality of parameter vectors. In tandem with our theoretical findings, we undertook empirical validations through experiments conducted on prevalent network models, including fully connected networks (FNNs), convolutional neural networks (CNNs), Transformers, pre-trained language models(PLMs) and large language models (LLMs) composed of Transformers. Finally, we have found that a fine-tuning technique (LoRA) preserves the orthogonality of parameter vectors, a revelation that carries importance within the framework of fine-tuning techniques for LLMs.

1 Introduction

Normalization techniques, such as batch normalization[Ioffe and Szegedy, 2015] and layer normalization[Ba *et al.*, 2016], belong to a class of widely used techniques for enhancing the training capabilities of deep neural networks(DNNs). In general, normalization can be attributed to several advantageous properties for DNNs, such as reducing the network's dependence on initial parameter values[De and Smith, 2020][Shao *et al.*, 2020], improving the convergence speed of the network[Karakida *et al.*, 2019], auto-tuning of learning rates[Arora *et al.*, 2018], and smoothing the loss landscape[Yong *et al.*, 2020]. The residual connection[He *et al.*, 2016] is a technique that involves adding skip connections in the middle of network layers, aiming to alleviate the gradient vanishing and exploding issues encountered during the training of DNNs. Furthermore, residual connections can significantly enhance training stability and generalization accuracy. Consequently, it has become an essential component in various domains, including biomedical imaging and generative models like U-Net[Ronneberger *et al.*, 2015]. In natural language processing, it is exemplified by the Transformer[Vaswani *et al.*, 2017], and in reinforcement learning, its effectiveness is demonstrated by AlphaGo Zero[Silver *et al.*, 2017].

In recent years, the combination of normalization and residual connections has been widely applied in DNNs. Almost all state-of-the-art models in recent years have adopted the combination of normalization and residual connections[Touvron *et al.*, 2023][Chowdhery *et al.*, 2023][Achiam *et al.*, 2023]. Particularly, the Transformer[Vaswani *et al.*, 2017] has become the fundamental building block of the most powerful large language models(LLMs) currently available. So, what is the origin of this effect ?

In addressing this question, scholars have conducted theoretical studies. De and Smith proposed that batch normalization tends to bias residual blocks towards the identity function in DNNs[De and Smith, 2020]. Balduzzi et al.[Balduzzi *et al.*, 2017] and Yang et al.[Yang *et al.*, 2019] argued that including identity skip connections and batch normalization layers on the residual branch in ResNets helps maintain correlations between various mini batches in deep networks. Liu et al.[Liu *et al.*, 2020] found that normalization can effectively address the problem of spurious gradient exploding or vanishing correlated with the depth of models resulting from residual connections. However, the origin of this effect is still poorly understood.

In DNNs that employ the combination of normalization and residual connections, such as the Transformer and pre-trained models(PLMs) composed of the Transformer, it has been observed that the parameter vectors exhibit good orthogonality, meaning the cosine similarity between any two row vectors (or column vectors) of the parameter matrix is close to zero. Based on feature learning theory[Radhakrishnan *et*

*Corresponding Author

al., 2022][Beaglehole *et al.*, 2023], the Gram matrix[Tanton, 2005] of network parameters is proportional to the average gradient outer product with respect to patches of the input to that layer. The Gram matrix and the average gradient outer product contain the same feature information[Radhakrishnan *et al.*, 2022][Beaglehole *et al.*, 2023]. In our perspective, when the column vectors of network parameters are orthogonal, the Gram matrix becomes a diagonal matrix with diagonal elements being all the eigenvalues. Consequently, at this point, the Gram matrix has the highest correlation with the average gradient outer product, indicating the best fitting capability of the trained network.

Moreover, some related studies have also demonstrated that the orthogonality of vectors can enhance the fitting capability of DNNs[Xiao *et al.*, 2018][Wang *et al.*, 2020][Li *et al.*, 2019][Huang *et al.*, 2018]. In generally, DNNs are typically initialized using random or approximately random methods[Glorot and Bengio, 2010][LeCun *et al.*, 2002], ensuring the orthogonality of initial network parameter vectors. However, little attention is paid to incorporating optimization or regularization conditions to maintain the orthogonality of parameter matrices during the training process. The factors responsible for preserving the orthogonality of parameter vectors are not well understood. To address this issue, we conducted pertinent research, and the key contributions of our study are summarized as follows.

- We propose the parameters independence index (PII) to assess the orthogonality between network parameter vectors, where a lower PII indicates better overall orthogonality of parameter vectors.
- We deduce that the combination of normalization and residual connections can enhance the orthogonality of parameter vectors, improve the feature learning capability of DNNs, and consequently, enhance the performance of DNNs. Then, we validated the theoretical correctness on network models including FNNs, CNNs,Transformers,PLMs and LLMs composed of the Transformer.
- We discovered that the LLM fine-tuning technique based on LoRA[Hu *et al.*, 2021] also maintains the orthogonality of parameter vectors during the adjustment of network parameters. This discovery holds significance in the context of LLMs fine-tuning techniques and also prove the universality of theory in this study. The supplementary materials and all implementation codes are available on the Github[1].

2 Related Works

Normalization and residual connections are widely applied in DNNs, and researchers have conducted relevant theoretical research to explore the mechanisms through which they enhance network performance.

2.1 Normalization Techniques

Normalization comes in various forms, and our research primarily focuses on the widely used batch normalization and layer normalization. Therefore, we will highlight relevant studies related to these techniques. For example, batch normalization divides the optimization task into optimizing the length and direction of parameters separately[Kohler *et al.*, 2019]. Batch normalization orthogonalizes representations in deep random networks[Daneshmand *et al.*, 2021]. Batch normalization is proven to avoid rank collapse for randomly initialized deep networks[Daneshmand *et al.*, 2020]. In contrast to batch normalization, layer normalization overcomes the dependency on batch size, and empirical evidence shows that it is more suitable for recurrent neural networks(RNNs) and natural language processing tasks[Ba *et al.*, 2016].

2.2 Residual Connections

Theoretical research on residual connections is also a hot topic in the theory of the DNN. Katsman et al. extended residual connections to general Riemannian manifolds in a geometrically principled manner[Katsman *et al.*, 2023]. Orhan et al. discovered that residual connections can eliminate singularities[Orhan and Pitkow, 2017]. By employing principles from linear algebra and random matrix theory, researchers explore the reasons behind the enhanced ease of optimization and improved generalization exhibited by DNNs with residual connections[Oyedotun *et al.*, 2022].

2.3 The Combination of Normalization and Residual Connections.

Regarding the impact of the combination of normalization and residual connections on DNNs, scholars have also conducted research. For instance, studies indicate that residual connections and batch normalization can enhance data separability[Furusho and Ikeda, 2019]. Batch normalization reduces the scale of hidden activations in the residual branch by approximately the square root of the network depth[De and Smith, 2020]. Furusho and Ikeda evaluated the generalization gap and the convergence rate to demonstrate why skip connections and batch normalization improve performance[Furusho and Ikeda, 2020].

Researchers have extensively studied normalization and residual connections, offering various explanations for how they can enhance the capabilities of DNNs. However, the internal mechanisms through which they profoundly impact DNNs are far from being fully understood

3 Background and Preliminaries

3.1 Network Description

For a FNN, the forward propagation process is as follows:

$$\mathbf{z}^l = W^l \mathbf{x}^{l-1} + \mathbf{b}^l, \mathbf{x}^l = f\left(\mathbf{z}^l\right) \tag{1}$$

Where $\mathbf{x}^{l-1}$ represents the output of the preceding layer, serving as the input to the current layer. $f\left(\cdot\right)$denotes the activation function, $\mathbf{z}^l$ is the pre-activation value of the neuron, W^l and $\mathbf{b}^l$ are the weights and bias of the network, with our focus being on W^l, temporarily disregarding $\mathbf{b}^l$.Our research primarily encompasses several widely used activation functions, including ReLU[Nair and Hinton, 2010], GELU[Hendrycks and Gimpel, 2016], Sigmoid[DeMaris, 1995], and Tanh[Fan,

[1]https://github.com/kinglzx2023/orthogonality-of-weight

2000]. After incorporating residual connections and the batch normalization

$$\mathbf{x}^l = BN\left(\mathbf{x}^{l-1} + f\left(\mathbf{z}^l\right)\right) \tag{2}$$

Let $\hat{\mathbf{x}}^l = \mathbf{x}^{l-1} + f\left(\mathbf{z}^l\right)$, then $\mathbf{x}^l = BN\left(\hat{\mathbf{x}}^l\right)$. Let

$$\mu = \frac{1}{N}\sum_i \hat{x}_i^l, \sigma^2 = \frac{1}{N}\sum_i \left(\hat{x}_i^l - \mu\right)^2 \tag{3}$$

Then $x_i^l = \frac{\hat{x}_i^l - \mu}{\sigma}$.

3.2 Feature Learning

$G_{W^l} = \left(W^l\right)^T W^l$ is the Gram matrix of the weight W^l in the l-th layer of the DNN. Research shows that the structure of W^l can characterize how features are updated in a trained DNN[Radhakrishnan *et al.*, 2022][Beaglehole *et al.*, 2023]. For a trained DNN, studying the importance of a particular feature, a natural approach is to examine the amplitude of changes in that feature under perturbations. The amplitude can be calculated through the outer product of the gradients of the neural network with respect to the input, that is $\left(\nabla_x f\left(x\right)\right)\left(\nabla_x f\left(x\right)\right)^T$, where $\nabla_x f\left(x\right)$ represents the gradient of the DNN f at point x. $\left(W^l\right)^T W^l$ has the following relationship with $\left(\nabla_x f\left(x\right)\right)\left(\nabla_x f\left(x\right)\right)^T$.

Theorem 1. *[Radhakrishnan* et al.*, 2022] Let f denote an L-hidden layer network with ReLU activation, suppose we sample weight $a_{k'}$,W^l for $l > 1$ in an i.i.d manner so that $E\left[a_{k'}^2\right] = 1, E\left[W_{l,k''}^2\right] = 1, E\left[a_{k'}\right] = 0$ and $E\left[W_{l,k''}\right] = 0$. Suppose W^1 is fixed and arbitrary. Let $\{(x_i, y_i)\}_{i=1}^n \subset R^d \times R$. If $x \sim N\left(0, I_d\right)$, then*

$$\begin{aligned}\frac{1}{k_1}\left(W^1\right)^T W^1 \ &= \\ E_x&\left[\lim_{k_2,\cdots,k_L\to\infty} E_a\left[\nabla_x f\left(x\right)\nabla_x f\left(x\right)^T\right]\right]\end{aligned} \tag{4}$$

$N\left(0, I_d\right)$ is the standard normal distribution, k_1 is the number of neurons in the first layer, and W^1 is the network parameter of the first layer. The results of Theorem 1 can be naturally extended to other network layers. Furthermore, for DNNs with finite width

$$\left(W^i\right)^T W^i \ \propto \ \frac{1}{n}\sum_{p=1}^n \nabla f_i\left(h_i\left(x_p\right)\right)\nabla f_i\left(h_i\left(x_p\right)\right)^T \tag{5}$$

where $h_i\left(x\right)$ is the input into layer i, $\nabla f_i\left(h_i\left(x_p\right)\right)$ denotes the gradient of f_i with respect to $h_i\left(x_p\right)$. Therefore, the outer product of gradients of the DNN with respect to the input data $h_i\left(x_p\right)$ is proportional to the Gram matrix of the weights.

In our study, We found that if $G_{W^i} = \left(W^i\right)^T W^i$ is a diagonal matrix or diagonally dominant matrix, the average outer product of gradients is also a diagonal or diagonally dominant matrix. The proportional relationship between them is mainly concentrated on the diagonal elements. In this case, the network can more fully learn the feature information of the input data. Therefore, if the Gram matrix of the network tends to be a diagonally dominant matrix after training, it will enhance the DNN's ability to fit the training data

3.3 Gram Matrix and Parameters Independence Index

For the weight $W^l = \left\{w_1^l, w_2^l, \cdots, w_n^l\right\} \in R^{m\times n}$, $w_i^l = \left\{w_{1i}^l, w_{2i}^l, \cdots, w_{mi}^l\right\}^T, i \in \{1, 2, \cdots, n\}$, the Gram matrix[Tanton, 2005] of W^l is $G_{W^l} = \left(W^l\right)^T\left(W^l\right)$ with elements $G_{ij} = \left\langle w_i^l, w_j^l\right\rangle$, where $\langle\cdot\rangle$ is the inner product. The diagonal elements of G_{W^l} are $\left\langle w_i^l, w_i^l\right\rangle, i \in 1, 2, \ldots, n$, and the off-diagonal elements are $\left\langle w_i^l, w_j^l\right\rangle, i \neq j$. Therefore, when G_{W^l} is a diagonal matrix, for any i and j, it is necessary to satisfy $\left\langle w_i^l, w_j^l\right\rangle = 0$ for $i \neq j$. In practical applications, cosine similarity is commonly used to characterize the angle between vectors. When the cosine similarity is 0, the vectors are orthogonal to each other. When the cosine similarity is 1 or -1, the angle between vectors is 0 or 180 degrees, respectively. We focus on the orthogonality between vectors, where a cosine similarity closer to 0 indicates a closer approach to orthogonality. Therefore, we assess the orthogonality between vectors by computing the absolute value of the cosine similarity. In most cases, the parameter size of the DNN is large. To evaluate the overall orthogonality between parameter vectors, we introduce the PII.

Definition 1. *For the weight matrix $W^l = \left\{w_1^l, w_2^l, \cdots, w_n^l\right\} \in R^{m\times n}$,*

$$PII = average\left(abs\left(\frac{w_i^l \cdot w_j^l}{\left\|w_i^l\right\|\left\|w_j^l\right\|}\right)\right), \tag{6}$$

for all $i, j \in n, i \neq j$, where $abs()$ denotes the absolute value, and $average()$ calculates the average of all data. Specifically, the PII is the average absolute value of cosine similarities between any two row vectors in the parameter matrix. The PII ranges from $[0, 1]$, where PII=0 indicates that any two row vectors in the parameter matrix are independent, and evidently, G_{W^l} is a diagonal matrix in this case. PII=1 signifies that the vectors are linearly dependent.

We conducted a study on the orthogonality of vectors by examining vector angles. Negative cosine similarity values lack practical significance. Therefore, we used the absolute value of cosine similarity to represent vector orthogonality. Concurrently, we calculated the cosine similarity between all column vector pairs in a parameter matrix. For an $M\times N$ matrix, there are N^2 total results, which is a substantial quantity. We need an indicator that can holistically reflect the orthogonality of the parameter vectors. The mean, or expected value, can represent the global orthogonality of the matrix vectors. Moreover, using the absolute value function avoids canceling out between positive and negative cosine similarities. This ensures the mean accurately reflects the overall orthogonality of the parameter vectors.

4 Main Results

4.1 Conformal Capability of The Optimizer

Theorem 1 stipulates that the network parameters must form a matrix with zero mean and unit variance. Our investigation reveals that even when the variance approximates 1, it does not significantly alter the proportional relationship between

G_{W^l} and the the average gradient outer product. Currently prevalent network initialization methods, such as orthogonal initialization, Xavier initialization[Glorot and Bengio, 2010], and Lecun initialization[LeCun *et al.*, 2002], all adhere to this condition. The incorporation of normalization techniques in network layers, such as batch normalization and layer normalization, can satisfy the condition for each layer's input $x \sim N(0, I_d)$. Consequently, it is advantageous for the network's training method to possess a certain conformal capability, ensuring that the angles between parameter vectors do not undergo significant changes. Subsequent research suggests that the SGD optimizer exhibits superior conformal capabilities when compared to the Adam optimizer.

Theorem 2. *The SGD optimizer outperforms the Adam optimizer in terms of conformal capability among parameter vectors.*

The proof is provided in the appendix. Here, we offer an intuitive understanding. DNNs typically employ mini-batch sample data for gradient computation and parameter updates. Let the training set be $\mathcal{D} = \left\{\left(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\right)\right\}_{n=1}^{N}$, where N is the batch size. Each sample $\mathbf{x}^{(n)}$ is the input of the network, resulting in the network output $\hat{\mathbf{y}}^{(n)}$. The loss on the dataset D is given by:

$$R(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\left(\mathbf{y}^{(n)}, \left(\hat{\mathbf{y}}^{(n)}\right)\right) \tag{7}$$

When using the SGD to update parameters,

$$\mathbf{w}_{i,t+1}^{l} = \mathbf{w}_{i,t}^{l} - \eta_t \nabla \mathbf{w}_{i,t}^{l} \tag{8}$$

η_t is the learning rate, $\mathbf{w}_{i,t}$ is the parameter vector and $\nabla \mathbf{w}_{i,t}^{l}$ the gradient with respect to $\mathbf{w}_{i,t}$. When updating parameters using the Adam,

$$\mathbf{w}_{i,t+1}^{l} = \mathbf{w}_{i,t}^{l} - \alpha_t \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v_{t+1}}} + \epsilon} b_{t+1} \tag{9}$$

$$\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1 - \beta_1) \nabla \mathbf{w}_{i,t}^{l} \tag{10}$$

$$\mathbf{v}_{t+1} = \beta_2 \mathbf{v}_t + (1 - \beta_2) \left(\nabla \mathbf{w}_{i,t}^{l}\right)^2 \tag{11}$$

$$b_{t+1} = \frac{\sqrt{1 - \beta_2^{t+1}}}{1 - \beta_1^{t+1}} \tag{12}$$

$\mathbf{m}_{t+1}$ and $\mathbf{v}_{t+1}$ are momentums obtainied by the $\nabla \mathbf{w}_{i,t}^{l}$.

Therefore, when the gradients of the network are the same, the SGD optimizer is closer to a translation transformation of the parameter vector $\mathbf{w}_{i,t}^{l}$. Translation of a vector does not change the angle between vectors before and after the transformation, making it an conformal mapping. Hence, SGD possesses better conformal capabilities than the Adam optimizer.

To validate Theorem 2, we conducted the following experiments about the FNN and CNN. The datasets are MNIST, CIFAR-10 and CIFAR-100. Results of CIFAR-100 are presented in the appendix. We computed the PIIs of the parameter matrix under both SGD and Adam optimizer. A PII closer to the initial value indicates better conformal capability of the optimizer.

Optimizers	PII Layer_1↑	PII Layer_2↓	PII Layer_3↓	ACC %
SGD	1.0/0.996	0.035/ 0.049	0.035/ 0.048	93.59
Adam	1.0/0.432	0.035/ 0.153	0.035/ 0.109	98.22
AdamW	1.0/ 0.411	0.035/ 0.153	0.035/ 0.114	97.92
RMSprop	1.0/ 0.766	0.035/ 0.234	0.035/ 0.149	97.94

Table 1: Conformal capabilities of optimizers in FNNs.

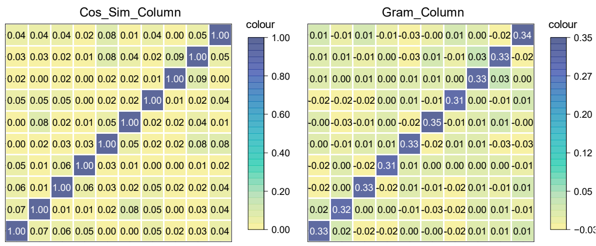
Optimizers	PII Layer_1↑	PII Layer_2↓	PII Layer_3↓	ACC %
SGD	1.0/0.795	0.070/0.195	0.050/0.149	84.04
Adam	1.0/0.328	0.070/0.217	0.050/0.242	85.94
AdamW	1.0/0.337	0.070/0.208	0.050/0.237	85.96
RMSprop	1.0/0.37	0.070/0.230	0.050/0.207	84.80

Table 2: Conformal capabilities of optimizers in CNNs.

Tables 1 and 2 present the conformal capabilities of optimizer in FNNs and CNNs. We selected four common optimizers, adjusted the learning rates to achieve similar test accuracy, and kept all other network parameters consistent. Table 1 shows the results for a FNN with three network layers. To assess the optimizer conformal capabilities, different initialization methods were used for the network layer. The parameters in the "Layer_1" were set to the same values, resulting in PII=1. For "Layer_2" and "Layer_3", the default initialization was used, i.e., the values are initialized from $\mathcal{U}\left(-\sqrt{k}, \sqrt{k}\right)$, where k is the input width of the network. Hence, PIIs are close to zero. In Table 1 and Table 2, an upward arrow indicates that a higher value is better, while conversely, a downward arrow signifies that a smaller value is better.

Each pair of values in the Table 1 and the Table 2 consists of the initial and post-trained PII, such as 1.0/0.996, it indicates that the initial PII is 1.0, and the post-trained is 0.996. After training, PIIs of each layer of SGD are closest to the initial values,the conformal capability of the SGD is the best. It is noteworthy that PIIs between Adam and AdamW are the closest. Table 2 presents the experimental results of a CNN network, with each convolutional layer containing a Max Pooling layer. Compared to the FNN, the CNN has a more complex structure, but these results consistent with Table 1. In summary, the experimental findings suggest that SGD exhibits a notably superior conformal capability compared to other optimization algorithms with momentum, however, the momentum leads to a higher accuracy than SGD.

Figure 1 illustrates the distribution of cosine similarity and Gram matrix of parameter vectors following initialization using $\mathcal{U}\left(-\sqrt{k}, \sqrt{k}\right)$. The cosine similarity matrix's diagonal signifies the similarity of a vector with itself, consistently equal to 1, while other entries tend to be close to zero. And the PII is 0.035, also close to zero. The Gram matrix exhibits strict diagonal dominance, ensuring that data features in the matrix are predominantly concentrated in the diagonal elements.



(a) PII=0.035 (b) PII=0.035

Figure 1: The cosine similarity and Gram matrix in the initialization of the DNN.

4.2 The Conformal Capability of The Optimizer with Normalization and Residual Connections

The earlier results indicate that, under the same conditions, the conformal capability of the SGD is the best. However, the SGD optimizer tends to trap DNNs in local optimum, preventing the network from achieving a better testing capabilities. Optimizers like Adam and AdamW overcome this issue by introducing momentum mechanisms. However, previous research suggests that these optimizers have poor conformal capability. The subsequent research suggests that adding normalization and residual connections in the middle of network layers can enhance the conformal capability of all the optimizers.

Additionally, our research has found that only normalization or residual connections cannot guarantee a definite improvement in the conformal capability of optimizers. Due to space constraints, relevant studies are presented in the appendix.

Theorem 3. *The combination of normalization and residual connections can enhance the conformal capability of the optimizer.*

The proof is provided in the appendix. Here, we also offer an intuitive understanding. We take batch normalization as an example, the result are the same for layer normalization. When DNNs do not incorporate batch normalization and residual connections, the gradient of the parameter W^l is

$$\nabla W^l = f'\left(\mathbf{z}^l\right) \odot \left(W^{l+1}\left(\mathbf{z}^{l+1}\right)'\right)\left(\mathbf{x}^{l-1}\right)^T = \delta^l\left(\mathbf{x}^{l-1}\right)^T \tag{13}$$

where $\odot$ represents element-wise multiplication, and $\delta^l = f'\left(\mathbf{z}^l\right) \odot \left(W^{l+1}\left(\mathbf{z}^{l+1}\right)'\right)$ has the same form across different network layers, we refer to δ^l as the error term of the neuron. After incorporating batch normalization and residual connections, the gradient of the parameter W^l is

$$\nabla W^l = \frac{1}{\sigma^l} f'\left(\mathbf{z}^l\right) \odot P_{\mathbf{1}^\perp} P_{\mathbf{x}^{l\perp}} \tag{14}$$

$$\left[\frac{1}{\sigma^{l+1}}\left(\mathbf{1} + f'\left(\mathbf{z}^{l+1}\right)\right) \odot W^{l+1} P_{\mathbf{1}^\perp} P_{\mathbf{x}^{l+1\perp}}\left(\mathbf{x}^{l+1}\right)'\right] \tag{15}$$

$$\cdot\left(\mathbf{x}^{l-1}\right)^T = \delta^l_{NR}\left(\mathbf{x}^{l-1}\right)^T \tag{16}$$

δ^l_{NR} represents the error term of the l-th layer neuron after incorporating normalization and residual connections, $\left(\mathbf{x}^{l+1}\right)' = \frac{\partial R}{\partial \mathbf{x}^{l+1}}$, $P_{\mathbf{1}^\perp}(\cdot)$ and $P_{\mathbf{x}^{l\perp}}(\cdot)$ indicate the projection onto the directions of vectors $\mathbf{1}$ and $\mathbf{x}^l$ respectively. σ^l and σ^{l+1} represent the variances of neurons in the l-th and $l+1$-th layers, respectively. It can be observed that normalization and residual connections modify the error term of neurons. Compared to Equation (13), the addition of the vector $\mathbf{1}$ with the activation function prevents the problem of gradient vanishing in ∇W^l. In the computation of the error term δ^l_{NR}, division by the variance of each layer's neurons is required. Therefore, for neurons with distributions having variances greater than 1 and sparser gradients, this approach reduces the sparsity of the gradients. In summary, the parameter gradient $\delta^l_{NR}\left(\mathbf{x}^{l-1}\right)^T$ in Equation (14) is a product of a unit-norm vector and a more uniformly distributed vector. During optimizer updates, this tends to map towards translations along the gradient direction, displaying better isotropy characteristics.

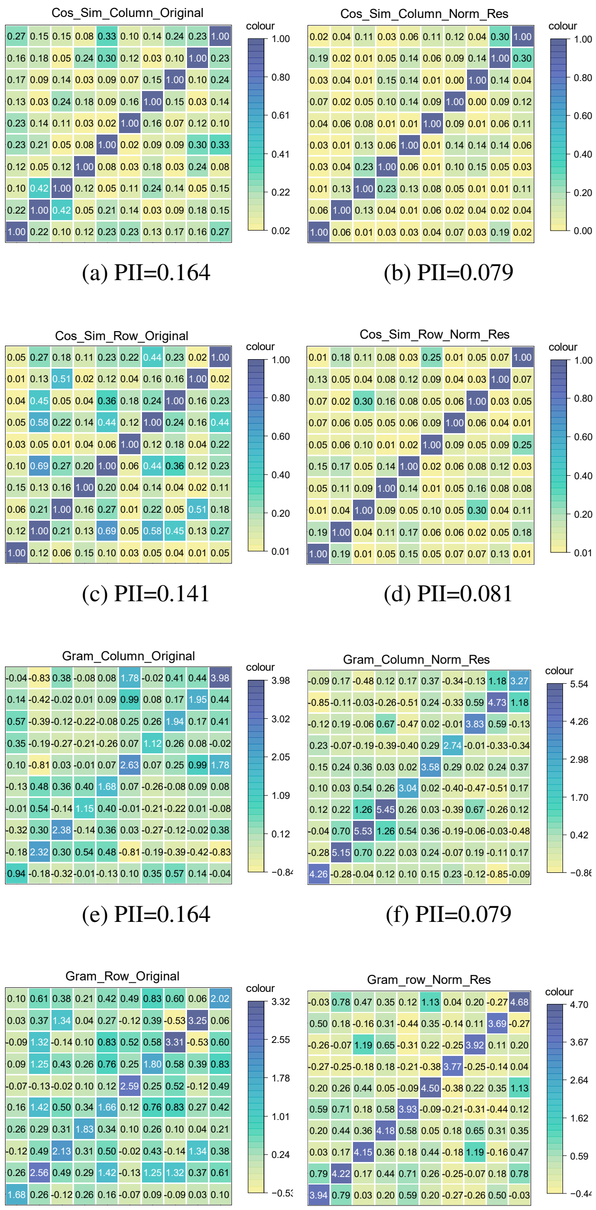
4.3 Experiments

To validate the correctness of the theory, we conducted extensive experiments. The neural network comprise FNNs, CNNs, Transformers, PLMs, and LLMs. The datasets consist of MNIST, CIFAR-10, CIFAR-100, WikiText-2. For PLMs and LLMs, calculating the PII of their parameters is sufficient, and there is no need to retrain the models. Moreover, to confirm whether the fine-tuning process of LLMs preserves the orthogonality of parameter vectors, we conducted LoRA[Hu *et al.*, 2021] fine-tuning on the Qwen model[Bai *et al.*, 2023] and assessed changes of the PII before and after the fine-tuning.

Figure 2 illustrates the impact of normalization and residual connections on the orthogonality of neural network parameter vectors. The left side presents results without normalization and residual connections, while the right side showcases results after their inclusion. Both result sets indicate a significant enhancement in the orthogonality of network parameter row and column vectors through the incorporation of normalization and residual connections.

Figures 2 (a)-(d) depict the distribution of cosine similarity matrices, where a closer proximity of non-diagonal elements to yellow signifies improved vector orthogonality. The results suggest that, upon integrating normalization and residual connections, the values of non-diagonal elements in the matrix notably decrease, indicating a pronounced tendency toward orthogonality for parameter vectors. Figures 2 (e)-(h) portray the distribution of Gram matrices, with larger values in the diagonal elements indicating superior vector orthogonality. The results indicate that, following the introduction of normalization and residual connections, the Gram matrices of parameters distinctly trend towards being diagonally dominant. Particularly noteworthy are the outcomes in Figure (e) and Figure (f), where, despite minimal changes in the color of non-diagonal elements, the values of diagonal elements significantly increase. This suggests that, with the addition of normalization and residual connections, more data features converge in the diagonal elements, thereby enhancing the neural network's capability for feature learning.

Moreover, Figure 2 demonstrates that the combination of



(a) PII=0.164 (b) PII=0.079

(c) PII=0.141 (d) PII=0.081

(e) PII=0.164 (f) PII=0.079

(g) PII=0.141 (h) PII=0.081

Figure 2: The cosine similarity and Gram matrix with and without normalization and residual connections in FNNs.

normalization and residual connections leads to a significant decrease in PII. PII accurately captures the orthogonality of parameter vectors. The combination of normalization and residual connections can simultaneously enhance the orthogonality of row vectors and column vectors, while the Gram matrix focuses primarily on column vectors. Therefore, in the subsequent experiments, we only present the results for column vectors.

In Table 3, the term 'Original' denotes the scenario where normalization and residual connections are excluded, while 'Norm_Res' signifies the inclusion of normalization and

Functions/ Optimizers	PII Original	PII ↓ Norm_Res	ACC % Original/Norm_Res
Sigmoid/Adam	0.278	0.139	97.88/98.13
Tanh/Adam	0.136	0.118	98.01/98.14
ReLU/Adam	0.138	0.09	98.23/98.45
GELU/Adam	0.141	0.081	98.16/98.24
GELU/SGD	0.053	0.036	97.54/98.13

Table 3: The influence of the combination of and residual connections on the orthogonality of parameter vectors and testing accuracy in FNNs.

Functions/ Optimizers	PII Original	PII ↓ Norm_Res	ACC % Original/Norm_Res
GELU/Adam	0.196	0.176	95.34/98.66
GELU/SGD	0.206	0.155	85.48/90.22

Table 4: The influence of the combination of normalization and residual connections on the orthogonality of parameter vectors and testing accuracy in CNNs.

residual connections. The initial four rows of data in the Table present PIIs for different activation functions under the Adam optimizer. The curves of ReLU and GELU functions display similar distributions, and their PIIs before and after incorporating normalization and residual connections are also comparable. The last two rows of data showcase results for various optimizers under the same activation function. The SGD inherently exhibits a good conformal capability, evident in the already relatively low PIIs even without normalization and residual connections. Nevertheless, the combination of normalization and residual connections still succeeds in further reducing its PII.

Table 3 illustrates that, under various activation functions and optimizer conditions, the integration of normalization and residual connections enhances the orthogonality of parameter vectors. Furthermore, as PII decreases, there is a general improvement in the overall test accuracy of the network. This implies that, for FNNs, improving the orthogonality of parameter vectors indeed enhances the network's fitting capability.

Table 4 delineates the influence of normalization and residual connections on the PII and test accuracy of CNNs on CIFAR-10, results of CIFAR-100 are presented in the appendix . The network adheres to a ResNet, incorporating normalization and residual connections between each convolutional layer. The 'Original' outcomes represent the scenario where normalization and residual connections are omitted, consequently transforming the network into a fully connected convolutional network. We have extracted experimental outcomes from a single convolutional layer, with other network layers exhibiting comparable characteristics, a comprehensive set of experimental results is available in the appendix. The findings in Table 4 showcase that normalization and residual connections have the potential to improve both the orthogonality of parameter vectors and the test accuracy of CNNs.

The Transformer[Vaswani *et al.*, 2017] stands out as one of the most prevalent neural network architectures in natural lan-

Models	Query↓	Keys↓	Values↓	Atten_Output↓	Linear_1.weight↓	Linear_2.weight↓
Transformer[Vaswani *et al.*, 2017]	0.085	0.066	0.026	0.025	0.091	**0.162**
Bert[Kenton and Toutanova, 2019]	0.046	0.045	0.037	0.039	0.055	0.021
Roberta[Liu *et al.*, 2019]	0.035	0.044	0.036	0.040	0.059	0.023
GPT-2[Radford *et al.*, 2019]	0.043	0.033	0.045	0.045	0.030	0.036
T5/Encoder[Raffel *et al.*, 2020]	0.032	0.036	0.034	0.033	0.043	0.019
T5/Decoder[Raffel *et al.*, 2020]	0.032	0.033	0.034	0.033	_	_
LLama[Touvron *et al.*, 2023]	0.023	0.024	0.013	0.013	0.013	0.011
LLama 2[Touvron *et al.*, 2023]	0.025	0.026	0.013	0.013	0.012	0.013
Qwen[Bai *et al.*, 2023]	0.022	0.021	0.015	0.015	0.009	0.014
LoRA[Hu *et al.*, 2021]	0.022	0.021	0.017	0.024	0.012	0.019

Table 5: The influence of the combination of normalization and residual connections on the orthogonality of parameter vectors in the Transformer, PLMs and LLMs.

guage processing. Within the Transformer framework, both Pre-trained Language Models (PLMs) and Language Models (LLMs), which are composed of the Transformer, utilize normalization and residual connections to link attention and fully connected layers. In order to assess their influence on the orthogonality of parameter vectors in PLMs and LLMs, we conducted the following experiments.

The content of Table 5 consists of three sections. The PII for each section includes the attention's queries, keys, and values matrices, the output layer parameter matrix of the attention, and the two parameter matrices of the fully connected layer. The first section presents the experimental results for the Transformer, which is composed of six Transformers and trained on the WikiText-2 dataset. The initial five PIIs are all close to zero, suggesting a high level of orthogonality among parameter vectors. However, the last PII is notably greater than zero. This discrepancy arises from the fact that, in our designed Transformer, the final fully connected layer in the last segment did not integrate the combination of normalization and residual connections. Consequently, this experiment confirms that it is the combination of normalization and residual connections that improves the orthogonality of parameter vectors in the Transformer.

The second section of Table 5 showcases the experimental outcomes for well-known PLMs. All of their PIIs are close to zero, signifying exceptional orthogonality among their parameter vectors. It's noteworthy that the T5/Decoder omits a FFN layer, leading to empty data entries for Linear1.weight and Linear2.weight, denoted by '-'.

The third section of Table 5 showcases the experimental outcomes for LLMs. Their PIIs are smaller than those of PLMs, suggesting superior orthogonality in the parameter vectors of LLMs. Moreover, to investigate whether fine-tuning LLMs preserves vector orthogonality, we calculated the PIIs before and after LoRA fine-tuning. The fine-tuning experiment was conducted using the Qwen model, with the PII for Qwen in the table representing the result before LoRA fine-tuning, and the corresponding PII for LoRA indicating the result after fine-tuning. The findings indicate that LoRA adjusts network parameters while upholding the orthogonality of parameter vectors. Our study contributes to an enhanced understanding of LLM fine-tuning techniques.

5 Conclusion

The combination of normalization and residual connections is commonly employed in DNNs, and it often enhances the network's stability and fitting capability. However, there has been a lack of theoretical explanations for this phenomenon. We have identified that the combination of normalization and residual connections maintains robust orthogonality between parameter vectors, leading to a Gram matrix that tends to be diagonal or diagonally dominant. The Gram matrix of parameters is proportionate to the outer product of the network's gradient vector concerning input data. Diagonal or diagonally dominant matrices concentrate effective data features in the diagonal elements, thereby improving the neural network's ability to learn data features. To validate the correctness of the theory, we conducted experiments on various network models suitable for the combination of normalization and residual connections. These models encompass FNNs, CNNs, Transformers, PLMs, LLMs, and networks fine-tuning based on LLMs. All the results confirm that the combination of normalization and residual connections significantly improves the orthogonality between parameter vectors after training.

Our research explains from the perspective of parameter vector orthogonality why normalization and residual connections can enhance the performance of DNNs, contributing to improve the interpretability of DNNs. Additionally, we have uncovered a novel insight that LoRA does not compromise the orthogonality of parameter vectors. This discovery holds significance in the context of LLMs fine-tuning techniques. Moving forward, our next phase of research will focus on investigating the influence of orthogonality on LLMs fine-tuning methods and exploring more efficient, resource-saving LLMs fine-tuning techniques.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (No. 2022YFC3301801), Natural Science Foundation of China (No. 62276043).

References

[Achiam *et al.*, 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni

Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[Arora *et al.*, 2018] Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. *arXiv preprint arXiv:1812.03981*, 2018.

[Ba *et al.*, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[Bai *et al.*, 2023] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[Balduzzi *et al.*, 2017] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *International Conference on Machine Learning*, pages 342–350. PMLR, 2017.

[Beaglehole *et al.*, 2023] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in convolutional neural networks. *arXiv preprint arXiv:2309.00570*, 2023.

[Chowdhery *et al.*, 2023] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[Daneshmand *et al.*, 2020] Hadi Daneshmand, Jonas Kohler, Francis Bach, Thomas Hofmann, and Aurelien Lucchi. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. *Advances in Neural Information Processing Systems*, 33:18387–18398, 2020.

[Daneshmand *et al.*, 2021] Hadi Daneshmand, Amir Joudaki, and Francis Bach. Batch normalization orthogonalizes representations in deep random networks. *Advances in Neural Information Processing Systems*, 34:4896–4906, 2021.

[De and Smith, 2020] Soham De and Sam Smith. Batch normalization biases residual blocks towards the identity function in deep networks. *Advances in Neural Information Processing Systems*, 33:19964–19975, 2020.

[DeMaris, 1995] Alfred DeMaris. A tutorial in logistic regression. *Journal of Marriage and the Family*, pages 956–968, 1995.

[Fan, 2000] Engui Fan. Extended tanh-function method and its applications to nonlinear equations. *Physics Letters A*, 277(4-5):212–218, 2000.

[Furusho and Ikeda, 2019] Yasutaka Furusho and Kazushi Ikeda. Resnet and batch-normalization improve data separability. In *Asian Conference on Machine Learning*, pages 94–108. PMLR, 2019.

[Furusho and Ikeda, 2020] Yasutaka Furusho and Kazushi Ikeda. Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives. *APSIPA Transactions on Signal and Information Processing*, 9:e9, 2020.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[Hu *et al.*, 2021] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[Huang *et al.*, 2018] Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[Karakida *et al.*, 2019] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. The normalization method for alleviating pathological sharpness in wide neural networks. *Advances in neural information processing systems*, 32, 2019.

[Katsman *et al.*, 2023] Isay Katsman, Eric Ming Chen, Sidhanth Holalkere, Anna Asch, Aaron Lou, Ser-Nam Lim, and Christopher De Sa. Riemannian residual neural networks. *arXiv preprint arXiv:2310.10013*, 2023.

[Kenton and Toutanova, 2019] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2, 2019.

[Kohler *et al.*, 2019] Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, Thomas Hofmann, Ming Zhou, and Klaus Neymeyr. Exponential convergence rates for batch normalization: The power of length-direction decoupling in non-convex optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 806–815. PMLR, 2019.

[LeCun *et al.*, 2002] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002.

[Li *et al.*, 2019] Shuai Li, Kui Jia, Yuxin Wen, Tongliang Liu, and Dacheng Tao. Orthogonal deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 43(4):1352–1368, 2019.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[Liu *et al.*, 2020] Fenglin Liu, Xuancheng Ren, Zhiyuan Zhang, Xu Sun, and Yuexian Zou. Rethinking skip connection with layer normalization. In *Proceedings of the 28th international conference on computational linguistics*, pages 3586–3598, 2020.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[Orhan and Pitkow, 2017] A Emin Orhan and Xaq Pitkow. Skip connections eliminate singularities. *arXiv preprint arXiv:1701.09175*, 2017.

[Oyedotun *et al.*, 2022] Oyebade K Oyedotun, Kassem Al Ismaeil, and Djamila Aouada. Why is everyone training very deep neural network with skip connections? *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Radhakrishnan *et al.*, 2022] Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Feature learning in neural networks and kernel machines that recursively learn features. *arXiv preprint arXiv:2212.13881*, 2022.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[Shao *et al.*, 2020] Jie Shao, Kai Hu, Changhu Wang, Xiangyang Xue, and Bhiksha Raj. Is normalization indispensable for training deep neural network? *Advances in Neural Information Processing Systems*, 33:13434–13444, 2020.

[Silver *et al.*, 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[Tanton, 2005] James S Tanton. *Encyclopedia of mathematics*. Infobase Publishing, 2005.

[Touvron *et al.*, 2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2020] Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X Yu. Orthogonal convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11505–11515, 2020.

[Xiao *et al.*, 2018] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. 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*, pages 5393–5402. PMLR, 2018.

[Yang *et al.*, 2019] Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S Schoenholz. A mean field theory of batch normalization. *arXiv preprint arXiv:1902.08129*, 2019.

[Yong *et al.*, 2020] Hongwei Yong, Jianqiang Huang, Xiansheng Hua, and Lei Zhang. Gradient centralization: A new optimization technique for deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 635–652. Springer, 2020.