

# Norm-Preservation: Why Residual Networks Can Become Extremely Deep?

Alireza Zaeemzadeh<sup>ID</sup>, *Student Member, IEEE*,  
Nazanin Rahnavard, *Senior Member, IEEE*, and Mubarak Shah<sup>ID</sup>, *Fellow, IEEE*

**Abstract**—Augmenting neural networks with skip connections, as introduced in the so-called ResNet architecture, surprised the community by enabling the training of networks of more than 1,000 layers with significant performance gains. This paper deciphers ResNet by analyzing the effect of skip connections, and puts forward new theoretical results on the advantages of identity skip connections in neural networks. We prove that the skip connections in the residual blocks facilitate preserving the norm of the gradient, and lead to stable back-propagation, which is desirable from optimization perspective. We also show that, perhaps surprisingly, as more residual blocks are stacked, the norm-preservation of the network is enhanced. Our theoretical arguments are supported by extensive empirical evidence. Can we push for extra norm-preservation? We answer this question by proposing an efficient method to regularize the singular values of the convolution operator and making the ResNet's transition layers extra norm-preserving. Our numerical investigations demonstrate that the learning dynamics and the classification performance of ResNet can be improved by making it even more norm preserving. Our results and the introduced modification for ResNet, referred to as Procrustes ResNets, can be used as a guide for training deeper networks and can also inspire new deeper architectures.

**Index Terms**—Residual networks, convolutional neural networks, optimization stability, norm preservation, spectral regularization

## 1 INTRODUCTION

DEEP neural networks have progressed rapidly during the last few years, achieving outstanding, sometimes super human, performance [1]. It is known that the depth of the network, i.e., number of stacked layers, is of decisive significance. It is shown that as the networks become deeper, they are capable of representing more complex mappings [2]. However, deeper networks are notoriously harder to train. As the number of layers is increased, optimization issues arise and, in particular, avoiding vanishing/exploding gradients is essential to optimization stability of such networks. Batch normalization, regularization, and initialization techniques have shown to be useful remedies for this problem [3], [4].

Furthermore, it has been observed that as the networks become increasingly deep, the performance gets saturated or even deteriorates [5]. This problem has been addressed by many recent network designs [5], [6], [7], [8]. All of these approaches use the same design principle: skip connections. This simple trick makes the information flow across the layers easier, by bypassing the activations from one layer to the next using skip connections. Highway Networks [7], ResNets [5], [6], and DenseNets [8] have consistently achieved state-of-the-art performances by using skip connections in different network topologies. The main goal of skip connection is to enable

the information to flow through many layers without attenuation. In all of these efforts, it is observed empirically that it is crucial to keep the information path *clean* by using identity mapping in the skip connection. It is also observed that more complicated transformations in the skip connection lead to more difficulty in optimization, even though such transformations have more representational capabilities [6]. This observation implies that *identity* skip connection, while provides adequate representational ability, has a great feature of optimization stability, enabling deeper well-behaved networks.

Since the introduction of Residual Networks (ResNets) [5], [6], there have been some efforts on understanding how the residual blocks may help the optimization process and how they improve the representational ability of the networks. Authors in [9] showed that skip connection eliminates the singularities caused by the model non-identifiability. This makes the optimization of deeper networks feasible and faster. Similarly, to understand the optimization landscape of ResNets, authors in [10] prove that linear residual networks have no critical points other than the global minimum. This is in contrast to plain linear networks, in which other critical points may exist [11]. Furthermore, authors in [12] show that as depth increases, gradients of plain networks resemble white noise and become less correlated. This phenomenon, which is referred to as *shattered gradient* problem, makes training more difficult. Then, it is demonstrated that residual networks reduce shattering, compared to plain networks, leading to numerical stability and easier optimization.

In this paper, we present and analytically study another desirable effect of identity skip connection: *the norm preservation of error gradient*, as it propagates in the backward path. We show theoretically and empirically that each

• The authors are with the School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816 USA.  
E-mail: {zaeemzadeh, nazanin}@eeecs.ucf.edu, shah@crco.ucf.edu.

Manuscript received 2 Dec. 2019; revised 9 Mar. 2020; accepted 19 Apr. 2020.  
Date of publication 27 Apr. 2020; date of current version 1 Oct. 2021.  
(Corresponding author: Alireza Zaeemzadeh.)  
Recommended for acceptance by L. Wang.  
Digital Object Identifier no. 10.1109/TPAMI.2020.2990339

residual block in ResNets is *increasingly norm-preserving*, as the network becomes *deeper*. This interesting result is in contrast to hypothesis provided in [13], which states that residual networks avoid vanishing gradient *solely* by shortening the effective path of the gradient.

Furthermore, we show that identity skip connection enforces the norm-preservation during the training, leading to well-conditioning and easier training. This is in contrast to the initialization techniques, in which the initialization distribution is modified to make the training easier [3], [14]. This is done by keeping the variance of weights gradient the same across layers. However, as observed in [14] and verified by our experiments, using such initialization methods, although the network is initially fairly norm-preserving, the norms of the gradients diverge as training progresses.

We analyze the role of identity mapping as skip connection in the ResNet architecture from a theoretical perspective. Moreover, we use the insight gained from our theoretical analysis to propose modifications to some of the building blocks of the ResNet architecture. Two main contributions of this paper are as follows.

- *Proof of the Norm Preservation of ResNets:* We show that having identity mapping in the shortcut path leads to norm-preserving building blocks. Specifically, identity mapping shifts all the singular values of the transformations towards 1. This makes the optimization of the network much easier by preserving the magnitude of the gradient across the layers. Furthermore, we show that, perhaps surprisingly, *as the network becomes deeper, its building blocks become more norm-preserving*. Hence, the gradients can flow smoothly through very deep networks, making it possible to train such networks. Our experiments validate our theoretical findings.
- *Enhancing Norm Preservation:* Using insights from our theoretical investigation, we propose important modifications to the *transition blocks* in the ResNet architecture. The transition blocks are used to change the number of channels and feature map size of the activations. Since these blocks do not use identity mapping as the skip connection, in general, they do not preserve the norm of the gradient. We propose to change the dimension of the activations in a norm preserving manner, such that the network becomes even more norm-preserving. For that, we propose a computationally efficient method to set the nonzero singular values of the convolution operator, without using singular value decomposition. We refer to the proposed architecture as Procrustes ResNet (ProcResNet). Our experiments demonstrate that the proposed norm-preserving blocks are able to improve the optimization stability and performance of ResNets.

The rest of the paper is organized as follows. In Section 2, the theoretical results and the bounds for norm-preservation of linear and nonlinear residual networks are presented. Then, in Section 3, we show how to enhance the norm preservation of the residual networks by introducing a new computationally efficient regularization of convolutions. To verify our theoretical investigation and to demonstrate the effectiveness of the proposed regularization, we

provide our experiments in Section 4. Finally, Section 5 draws conclusions.

## 2 NORM-PRESERVATION OF RESIDUAL NETWORKS

Our following main theorem states that, under certain conditions, a deep residual network representing a nonlinear mapping is norm-preserving in the backward path. We show that, at each residual block, the norm of the gradient with respect to the input is close to the norm of gradient with respect to the output. In other words, *the residual block with identity mapping, as the skip connection, preserves the norm of the gradient in the backward path*. This results in several useful characteristics such as avoiding vanishing/exploding gradient, stable optimization, and performance gain.

Suppose we want to represent a nonlinear mapping  $\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$  with a sequence of  $L$  non-linear residual blocks of form

$$\mathbf{x}_{l+1} = \mathbf{x}_l + F_l(\mathbf{x}_l). \quad (1)$$

As illustrated in Fig. 1b,  $\mathbf{x}_l$  and  $\mathbf{x}_{l+1}$  represent respectively the input and output at  $l$ th layer.  $F_l(\mathbf{x}_l)$  is the residual transformation learned by the  $l$ th layer. Before presenting the theorem, we lay out the following assumptions on  $\mathcal{F}$ .

**Assumption 1.** The function  $\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$  is differentiable, invertible, and satisfies the following conditions:

- (i)  $\forall \mathbf{x}, \mathbf{y}, \mathbf{z}$  with bounded norm,  $\exists \alpha > 0$ ,  $\|(\mathcal{F}'(\mathbf{x}) - \mathcal{F}'(\mathbf{y}))\mathbf{z}\| \leq \alpha \|\mathbf{x} - \mathbf{y}\| \|\mathbf{z}\|$ ,
- (ii)  $\forall \mathbf{x}, \mathbf{y}$  with bounded norm,  $\exists \beta > 0$ ,  $\|\mathcal{F}^{-1}(\mathbf{x}) - \mathcal{F}^{-1}(\mathbf{y})\| \leq \beta \|\mathbf{x} - \mathbf{y}\|$ , and
- (iii)  $\exists \mathbf{x}$  with bounded norm such that  $\text{Det}(\mathcal{F}'(\mathbf{x})) > 0$ ,

$\alpha$  and  $\beta$  are constants, independent of network size and architecture. Also, we assume that the domain of inputs is bounded. By rescaling inputs, we can assume, without loss of generality, that  $\|\mathbf{x}_1\|_2 \leq 1$  for any input  $\mathbf{x}_1$ .

We would like to emphasize the point that these assumptions are on the mapping that we are trying to represent by the network, not the network itself. Thus, assumptions are independent of architecture. Assumptions (i) and (ii) mean that the function  $\mathcal{F}$  is smooth, Lipschitz continuous, and its inverse is differentiable. The practical relevance of invertibility assumption is justified by the success of reversible networks [15], [16], [17]. Reversible architectures look for the true mapping  $\mathcal{F}$  *only* in the space of invertible functions and it is shown that imposing such strict assumption on the architecture does not hurt its representation ability [16]. Thus, the mapping  $\mathcal{F}$  is either invertible or can be well approximated by an invertible function, in many scenarios. However, unlike the reversible architectures, we do not assume residual blocks or the residual transformations,  $F_l(\cdot)$ , are invertible, which makes the assumption less strict. Furthermore, our extensive experiments in Section 4 show that our theoretical analysis, which is based on these assumptions, hold. This is further empirical justification that these assumptions are relevant in practice. Finally, Assumption (iii) is without loss of generality [10], [18].

**Theorem 1.** Suppose we want to represent a nonlinear mapping  $\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ , satisfying Assumption 1, with a sequence of  $L$

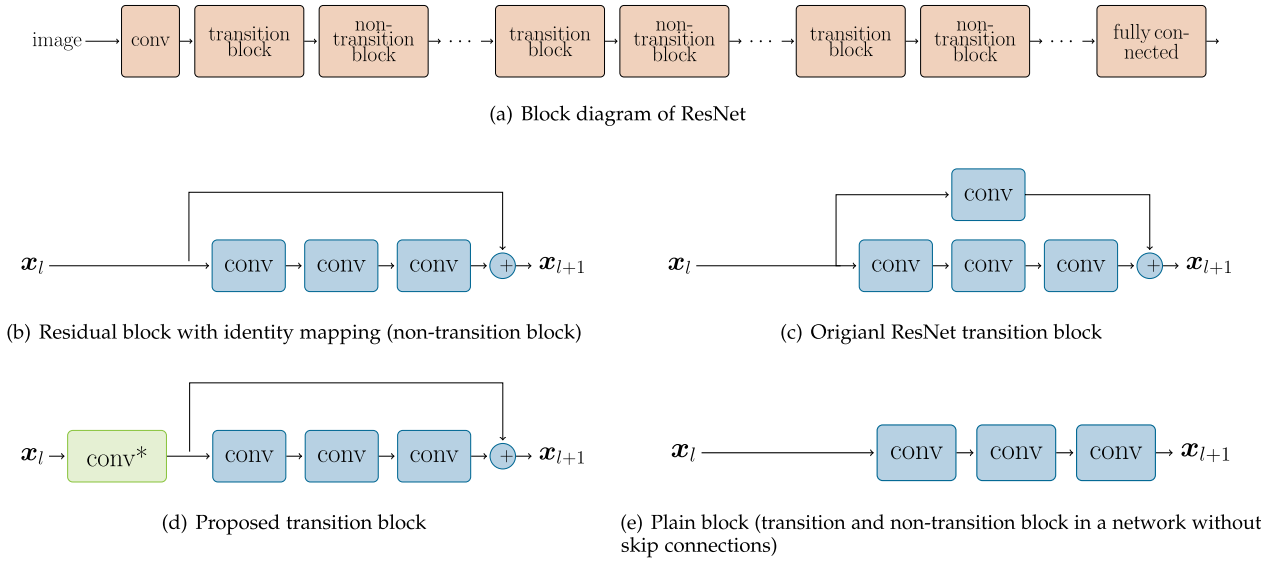


Fig. 1. ResNet architecture and its building blocks. Each conv block represents a sequence of batch normalization, ReLU, and convolution layers. conv\* block represents the regularized convolution layer.

non-linear residual blocks of form  $x_{l+1} = x_l + F_l(x_l)$ . There exists a solution such that for all residual blocks we have

$$(1 - \delta) \left\| \frac{\partial \mathcal{E}}{\partial x_{l+1}} \right\|_2 \leq \left\| \frac{\partial \mathcal{E}}{\partial x_l} \right\|_2 \leq (1 + \delta) \left\| \frac{\partial \mathcal{E}}{\partial x_{l+1}} \right\|_2,$$

where  $\delta = c \frac{\log(2L)}{L}$ ,  $\mathcal{E}(\cdot)$  is the cost function, and  $c = c_1 \max\{\alpha\beta(1 + \beta), \beta(2 + \alpha) + \alpha\}$  for some  $c_1 > 0$ .  $\alpha$  and  $\beta$  are constants defined in Assumption 1.

**Proof.** See Section A.1, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.2990339/>.  $\square$

This theorem shows that the mapping  $\mathcal{F}$  can be represented by a sequence of  $L$  non-linear residual blocks, such that the norm of the gradient does not change significantly, as it is backpropagated through the layers. One interesting implication of Theorem 1 is that as  $L$ , the number of layers, increases,  $\delta$  becomes smaller and the solution becomes more norm-preserving. This is a very desirable feature because vanishing or exploding gradient often occurs in deeper network architectures. However, by utilizing residual blocks, as more blocks are stacked, the solution becomes extra norm-preserving.

Now that we proved such a solution exists, we show why residual networks can remain norm preserving throughout the training. For that, we consider the case where  $F_l(x_l)$  consists of two layers of convolution and non-linearity. The following corollary shows the bound on norm preservation of the residual block depends on the norm of the weights. Therefore, if we bound the optimizer to search only in the space of functions with small norms, we can ensure that the network will remain norm preserving throughout the training. Therefore, any critical point in this space is also norm-preserving. On the other hand, based on Theorem 1, we know that at least one norm preserving solution exists. It is also known that, under certain conditions, any critical point achieved during optimization of ResNets is a global minimizer, meaning that it achieves

the same loss function value as the global minimum [10], [18], [19]. Thus, this result implies that ResNets are able to maintain norm-preservation throughout the training and if they converge, the solution is a norm-preserving global minimizer. The conclusions of the corollary can be easily generalized for residual block with more than two layers.

**Corollary 1.** Suppose a network contains non-linear residual blocks of form  $x_{l+1} = x_l + W_l^{(2)} \rho(W_l^{(1)} \rho(x_l))$ , where  $\rho(\cdot)$  is an element-wise non-linear operator with bounded derivative, i.e.,  $0 \leq \frac{\partial \rho_n(x_l)}{\partial x_{l,n}} \leq c_\rho, \forall n = 1, \dots, N$ . Then, we have

$$(1 - \delta) \left\| \frac{\partial \mathcal{E}}{\partial x_{l+1}} \right\|_2 \leq \left\| \frac{\partial \mathcal{E}}{\partial x_l} \right\|_2 \leq (1 + \delta) \left\| \frac{\partial \mathcal{E}}{\partial x_{l+1}} \right\|_2$$

$$\text{and } \delta = c_\rho^2 \left\| W_l^{(1)} \right\|_2 \left\| W_l^{(2)} \right\|_2.$$

**Proof.** See Section A.3, available in the online supplemental material  $\square$

Here,  $\|\cdot\|_2$  is the induced matrix norm and is the largest singular value of the matrix, which is known to be upper bounded by the entry-wise  $\ell_2$  norm. This means that norm-preservation is enforced throughout the training process, as long as the norm of the weights are small, not just at the beginning of the training by good initialization. This is the case in practice, since the weights of the network are regularized either explicitly using  $\ell_2$  regularization, also known as weight decay, or implicitly by the optimization algorithm [20], [21]. Thus, the gradients will have very similar magnitudes at different layers, and this leads to well-conditioning and faster convergence [14].

Although Theorem 1 holds for linear blocks as well, we can derive tighter bounds for linear residual blocks by taking a slightly different approach. For that, we model each linear residual block as

$$x_{l+1} = x_l + W_l x_l, \quad (2)$$

where,  $x_l, x_{l+1} \in \mathbb{R}^N$  are respectively the input and output of the  $l$ th residual block, with dimension  $N$ . The weight matrix  $W_l \in \mathbb{R}^{N \times N}$  is the tunable linear transformation. The goal of learning is to compute a function  $y = \mathcal{M}(x, \mathcal{W})$ , where  $x = x_1$  is the input,  $y = x_{L+1}$  is its corresponding output, and  $\mathcal{W}$  is the collection of all adjustable linear transformations, i.e.,  $W_1, W_2, \dots, W_L$ . In the case of simplified linear residual networks, function  $\mathcal{M}(x, \mathcal{W})$  is a stack of  $L$  residual blocks, as formulated in (2). Mathematically speaking, we have

$$y = \mathcal{M}(x, \mathcal{W}) = \prod_{l=1}^L (I + W_l)x, \quad (3)$$

where  $I$  is an  $N \times N$  identity matrix.  $\mathcal{M}(x, \mathcal{W})$  is used to learn a linear mapping  $R \in \mathbb{R}^{N \times N}$  from its inputs and outputs. Furthermore, assume that  $y$  is contaminated with independent identically distributed (i.i.d) Gaussian noise, i.e.,  $\hat{y} = Rx + \epsilon$ , where  $\epsilon$  is a zero mean noise vector with covariance matrix  $I$ . Hence, our objective is to minimize the expected error of the maximum likelihood estimator as

$$\min_{\mathcal{W}} \mathcal{E}(\mathcal{W}) = \mathbb{E} \left\{ \frac{1}{2} \|\hat{y} - \mathcal{M}(x, \mathcal{W})\|_2^2 \right\}, \quad (4)$$

where the expectation  $\mathbb{E}$  is with respect to the population  $(x, y)$ . The following theorem states the bound on the norm preservation of the linear residual blocks.

**Theorem 2.** For learning a linear map,  $R \in \mathbb{R}^{N \times N}$ , between its input  $x$  and output  $y$  contaminated with i.i.d Gaussian noise, using a network consisting of  $L$  linear residual blocks of form  $x_{l+1} = x_l + W_l x_l$ , there exists a global optimum for  $\mathcal{E}(\cdot)$ , as defined in (4), such that for all residual blocks we have

$$(1 - \delta) \left\| \frac{\partial \mathcal{E}}{\partial x_{l+1}} \right\|_2 \leq \left\| \frac{\partial \mathcal{E}}{\partial x_l} \right\|_2 \leq (1 + \delta) \left\| \frac{\partial \mathcal{E}}{\partial x_{l+1}} \right\|_2,$$

for  $L \geq 3\gamma$ , where  $\delta = \frac{c}{L}$ ,  $c = 2(\sqrt{\pi} + \sqrt{3}\gamma)^2$ , and  $\gamma = \max(|\log \sigma_{\max}(R)|, |\log \sigma_{\min}(R)|)$ , where  $\sigma_{\max}(R)$  and  $\sigma_{\min}(R)$ , respectively, are maximum and minimum singular values of  $R$ .

**Proof.** See Section A.2, available in the online supplemental material  $\square$

Similar to the nonlinear residual blocks, the linear blocks become more norm-preserving as we increase the depth. However, the linear blocks become norm-preserving at a faster rate. The gradient norm ratio for the linear blocks approaches 1 with a rate of  $\mathcal{O}(\frac{1}{L})$ , while this ratio for nonlinear blocks approaches 1 with a rate of  $\mathcal{O}(\frac{\log(L)}{L})$ .

### 3 PROCRUSTES RESIDUAL NETWORK

As depicted in Fig. 1a, residual networks contain four different types of blocks: (i) convolution layer (first layer), (ii) fully connected layer (last layer), (iii) transition blocks (which change the dimension) as depicted in Fig. 1c, and (iv) residual blocks with identity skip connection, as illustrated in Fig. 1b, which we also refer to as non-transition blocks. Theoretical investigation presented in Section 2 holds only for residual blocks with identity mapping as the skip connection. Such

identity skip connection cannot be used in the transition blocks, since the size of the input is not the same as the size of output. If the benefits of residual networks can be explained, at least partly, by norm-preservation, then one can improve them by alternative methods for preserving the norm. In this section, we propose to modify the transition blocks of ResNet architecture, to make them norm-preserving. Due to multiplicative effect through the layers, making these layers norm-preserving may be important, although they make up a small portion of the network. In the following, we discuss how to preserve the norm of the back-propagated gradients across all the blocks of the network.

As depicted in Fig. 1c, in the original ResNet architecture, the dimension changing blocks, also known as transition blocks, use  $1 \times 1$  convolution with stride of 2 in their skip connections to match the dimension of input and output activations. Such transition blocks are not norm-preserving in general.

Fig. 1d shows the block diagram of the proposed norm-preserving transition block. To change the dimension in a norm-preserving manner, we utilize a norm preserving convolution layer, conv\*. For that, we project the convolution kernel onto the set of norm preserving kernels by setting its singular values. Here, we show how we can make the convolution layer norm preserving by regularizing the singular values, without using singular value decomposition. Specifically, the gradient of a convolution layer with kernel of size  $k$ , with  $c$  input channels, and  $d$  output channels can be formulated as

$$\Delta_x = \hat{W} \Delta_y, \quad (5)$$

where  $\Delta_x$  and  $\Delta_y$  respectively are the gradients with respect to the input and output of the convolution.  $\Delta_y$  is an  $n^2 d$  dimensional vector, representing  $n \times n$  pixels in  $d$  output channels, and  $\Delta_x$  is an  $n^2 c$  dimensional vector, representing the gradient at the input. Furthermore,  $\hat{W}$  is an  $n^2 c \times n^2 d$  dimensional matrix embedding the back-propagation operation for the convolution layer. We can represent this linear transformation as

$$\Delta_x = \sum_{j=1}^{n^2 c} \sigma_j u_j < \Delta_y, v_j >, \quad (6)$$

where  $\{\sigma_j, u_j, v_j\}$  is the set of singular values and singular vectors of  $\hat{W}$ . Furthermore, since the set of the right singular vectors, i.e.,  $\{v_j\}$ , is an orthonormal basis set for  $\Delta_y$ , we can write the gradient at the output as

$$\Delta_y = \sum_{j=1}^{n^2 d} v_j < \Delta_y, v_j >.$$

Thus, we can compute the expected value of the norm of the gradients as

$$\begin{aligned} \mathbb{E}[\|\Delta_x\|_2^2] &= \sum_{j=1}^{n^2 c} \sigma_j^2 \mathbb{E}[|< \Delta_y, v_j >|^2], \\ \mathbb{E}[\|\Delta_y\|_2^2] &= \sum_{j=1}^{n^2 d} \mathbb{E}[|< \Delta_y, v_j >|^2], \end{aligned}$$



where we use the fact that  $\mathbf{u}_i^T \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = 0$  for  $i \neq j$  and  $\mathbf{u}_j^T \mathbf{u}_j = \mathbf{v}_j^T \mathbf{v}_j = 1$  and the expectation is over the data population. We propose to preserve the norm of the gradient, i.e.,  $\mathbb{E}[\|\Delta_x\|_2^2] = \mathbb{E}[\|\Delta_y\|_2^2]$ , by setting all the non-zero singular values to  $\sigma$ . It is easy to show that we can achieve this by setting

$$\sigma^2 = \frac{\sum_{j=1}^{n^2 d} \mathbb{E}[\|\Delta_y, \mathbf{v}_j\|_2^2]}{\sum_{j, \sigma_j \neq 0} \mathbb{E}[\|\Delta_y, \mathbf{v}_j\|_2^2]}, \quad (7)$$

where the summation in the denominator is over the singular vectors  $\mathbf{v}_j$  corresponding to the nonzero singular values, i.e.,  $\sigma_j \neq 0$ . The ratio in (7) is the ratio of expected energy of  $\Delta_y$ , i.e.,  $\mathbb{E}[\|\Delta_y\|_2^2]$ , divided by the portion of energy that does not lie in the null space of  $\hat{W}$ . We make the assumption that this ratio can be approximated by  $\frac{n^2 d}{n^2 \min(d, c)}$ . This assumption implies that about  $\frac{n^2 d}{n^2 \min(d, c)}$  of the total energy of  $\Delta_y$  will lie in the  $n^2 \min(d, c)$ -dimensional subspace, corresponding to orthonormal basis set  $\{\mathbf{v}_j | \sigma_j \neq 0\}$ , of our  $n^2 d$ -dimensional space. It is easy to notice that the assumption holds if the energy of  $\Delta_y$  is distributed uniformly among the directions in the basis set  $\{\mathbf{v}_j\}$ . But, since we are taking the sum over a large number of bases, it can also hold with high probability in cases where there is some variation in the distribution of energies along different directions. This is not a strict assumption in high dimensional spaces and we will investigate the practical relevance of this assumption in a real-world setting shortly. Thus, we can achieve norm preservation by setting all the nonzero singular values to  $\sqrt{\frac{d}{\min(d, c)}}$ . We can enforce this equality without using singular value decomposition. For that, we use the following theorem from [22]. This theorem states that the singular values of the convolution operator can be calculated by finding the singular values of the Fourier transform of the slices of the convolution kernels.

**Theorem 3.** (Theorem 6 from [22]) For any convolution kernel  $\mathbf{K} \in \mathbb{R}^{k \times k \times d \times c}$  acting on an  $n \times n \times d$  input, let  $\hat{W}$  be the matrix encoding the linear transformation computed by a convolutional layer parameterized by  $\mathbf{K}$ . Also, for each  $u, v \in [n] \times [n]$ , let  $\mathbf{P}^{(u, v)} \in \mathbb{C}^{d \times c}$  be the matrix given by  $\mathbf{P}_{i, j}^{(u, v)} = (\mathcal{F}_n(\mathbf{K}_{::, i, j}))_{u, v}$ , where  $\mathcal{F}_n(\cdot)$  is the operator describing an  $n \times n$  2D Fourier transform. Then, the set of singular values of  $\hat{W}$  is the union (allowing repetitions) of all the singular values of  $\mathbf{P}^{(u, v)}$  matrices  $\forall u, v$ .

**Proof.** See [22].  $\square$

Hence, to satisfy the condition (7), we can set all the nonzero singular values of  $\mathbf{P}^{(u, v)}$  to  $\sqrt{\frac{d}{\min(d, c)}}$  for all  $u$  and  $v$ . This can be done by finding the matrix  $\hat{\mathbf{P}}^{(u, v)}$  that minimizes  $\|\mathbf{P}^{(u, v)} - \hat{\mathbf{P}}^{(u, v)}\|_F^2$ , such that  $\hat{\mathbf{P}}^{(u, v)T} \hat{\mathbf{P}}^{(u, v)} = \frac{d}{\min(d, c)} \mathbf{I}$ , where  $\|\cdot\|_F$  denotes the Frobenius norm and  $\mathbf{I}$  is a  $c \times c$  identity matrix. It can be shown that the solution to this problem is given by

$$\hat{\mathbf{P}}^{(u, v)} = \sqrt{\frac{d}{\min(d, c)}} \mathbf{P}^{(u, v)} \left( \mathbf{P}^{(u, v)T} \mathbf{P}^{(u, v)} \right)^{-\frac{1}{2}}. \quad (8)$$

This is closely related to *Procrustes* problems, in which the goal is to find the closest orthogonal matrix to a given

matrix [23]. Finding the inverse of the square root of product  $\mathbf{P}^{(u, v)T} \mathbf{P}^{(u, v)}$  can be computationally expensive, specifically for large number of channels  $c$ . Thus, we exploit an iterative algorithm that computes the inverse of the square root using only matrix multiplications. Specifically, one can use the following iterations to compute  $(\mathbf{P}^{(u, v)T} \mathbf{P}^{(u, v)})^{-\frac{1}{2}}$  [24]:

$$\begin{aligned} T_k &= 3\mathbf{I} - \mathbf{Z}_k \mathbf{Y}_k, \\ \mathbf{Y}_{k+1} &= \frac{1}{2} \mathbf{Y}_k T_k, \\ \mathbf{Z}_{k+1} &= \frac{1}{2} T_k \mathbf{Z}_k, \end{aligned} \quad (9)$$

for  $k = 0, 1, \dots$  and the iterators are initialized as

$$\mathbf{Y}_0 = \mathbf{P}^{(u, v)T} \mathbf{P}^{(u, v)}, \mathbf{Z}_0 = \mathbf{I}.$$

It has been shown that  $\mathbf{Z}_k$  converges to  $(\mathbf{P}^{(u, v)T} \mathbf{P}^{(u, v)})^{-\frac{1}{2}}$  quadratically [24]. Since the iterations only involve matrix multiplication, they can be implemented efficiently on GPUs.

---

#### Algorithm 1. Update Rules for Transition Kernels at Each Iteration

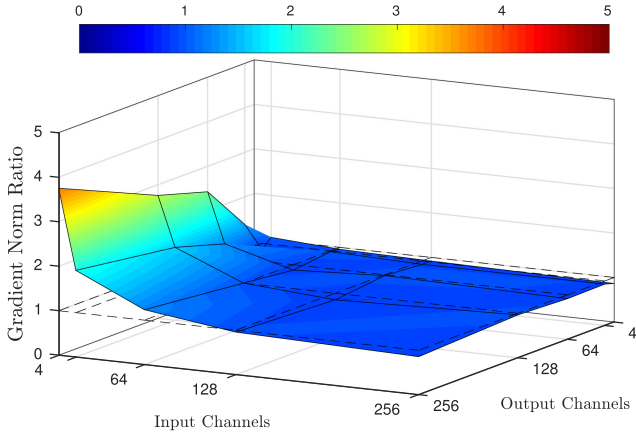
---

**Input:** Convolution kernel  $\mathbf{K}$  at the current iteration

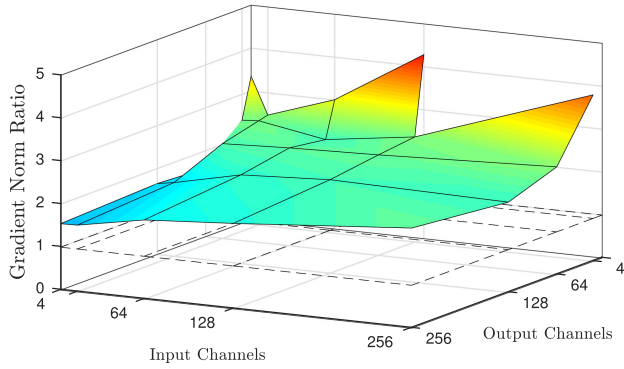
- 1: Perform the gradient descent step on the kernel  $\mathbf{K}$ .
  - 2: Calculate  $\mathbf{P}^{(u, v)}$  for each  $u, v \in [n] \times [n]$  as  $\mathbf{P}_{i, j}^{(u, v)} = (\mathcal{F}_n(\mathbf{K}_{::, i, j}))_{u, v}$ .
  - 3: Compute  $(\mathbf{P}^{(u, v)T} \mathbf{P}^{(u, v)})^{-\frac{1}{2}}$  using (9).
  - 4: Calculate  $\hat{\mathbf{P}}^{(u, v)}$  using (8).
  - 5: Update  $\mathbf{K}$  using the inverse 2D Fourier transform of  $\hat{\mathbf{P}}^{(u, v)}$ .
- 

Thus, to keep the convolution kernels norm preserving throughout the training, at each iteration, we compute the matrices  $\mathbf{P}^{(u, v)}$  and set the nonzero singular values using (8). Algorithm 1 summarizes the operations performed at each iteration on the kernels of the regularized convolution layers. To keep the desired norm-preservation property after performing the gradient descent step, such as SGD, Adam, etc, the proposed scheme is used to re-enforce norm-preservation on the updated kernel. In this manner, we can maintain norm-preservation, while updating the kernel during the training. Our experiments in Section 4 show that performing the proposed projection on the transition block of deep ResNets increases the training time by less than 8 percent. Also, since the number of transition blocks are independent of depth, the deeper the network gets, the computational overhead of the proposed modification becomes less significant. Fig. 1d shows the diagram of the proposed transition block, where a regularized convolution layer,  $\text{conv}^*$ , is used to change the dimension. Hence, we are able to exploit a regular residual block with identity mapping, which is norm preserving.

Similar to [3], to take into the account the effect of a ReLU nonlinearity and to make a Conv-Relu layer norm-preserving, we just need to add a factor of  $\sqrt{2}$  to the singular values and set them to  $\sqrt{\frac{2d}{\min(d, c)}}$ . Intuitively, the element-wise ReLU sets half of the units to zero on average, making the expected value of the energy of the gradient equal to  $\mathbb{E}[\|\Delta_x\|_2^2] = \frac{1}{2} \sum_{j=1}^{n^2 c} \sigma_j^2 \mathbb{E}[\|\Delta_y, \mathbf{v}_j\|_2^2]$ . Therefore, to compensate this, we need to satisfy this condition



(a) With the Proposed Regularization



(b) Without Regularization

Fig. 2. The ratio of gradient norm at output to gradient norm at input, i.e.,  $\|\frac{\partial \mathcal{E}}{\partial x_{l+1}}\|_2$  to  $\|\frac{\partial \mathcal{E}}{\partial x_l}\|_2$ , of a convolution layer for different number of input and output channels at 10th training epoch (a) with, and (b) without the proposed regularization on the singular values of the convolution.

$$\frac{1}{2} \sum_{j=1}^{n^2 c} \sigma_j^2 \mathbb{E}[\langle \Delta y, v_j \rangle^2] = \sum_{j=1}^{n^2 d} \mathbb{E}[\langle \Delta y, v_j \rangle^2].$$

It is also worthwhile to mention that since we are trying to preserve the norm of the backward signal, the variable  $n$  in Theorem 3 represents the size of feature map size at the output of the convolution.

To evaluate the effectiveness of the proposed projection, we design the following experiment. We perform the projection on the convolution layers of a small 3-layer network. The network consists of 3 convolutional layers, followed by ReLU non-linearity. To examine the gradient norm ratio for different number of input and output channels, the second layer is a  $3 \times 3$  convolution with  $c$  input channels and  $d$  output channels. The first and third layers are  $1 \times 1$  convolutions to change the number of channels and to match the size of the input and output layers. Fig. 2 shows the gradient norm ratio, i.e.,  $\|\frac{\partial \mathcal{E}}{\partial x_{l+1}}\|_2$  to  $\|\frac{\partial \mathcal{E}}{\partial x_l}\|_2$ , for different values of  $c$  and  $d$  at 10th training epoch on CIFAR-10, with and without the proposed projection. The values are averaged over 10 different runs.

It is evident that the proposed projection enhances the norm preservation of the Conv-ReLU layer, as it moves the gradient norm ratios toward 1. The only failure case is for networks with very small  $c$  and  $c \ll d$ . This is because,

due to the smaller size of the space, our assumption that the energy of the signal in the  $n^2 c$  dimensional subspace, corresponding to the  $n^2 c$  non-zero singular values, is approximately  $\frac{n^2 c}{n^2 d}$  of the total energy of the signal, is violated with higher probability. However, in more practical settings, where the number of channels is large and the assumption is held, the proposed projection performs as expected. This experiment illustrates the validity of our analysis as well as the effectiveness of the proposed projection for such practical scenarios. In the next section, we demonstrate the advantages of the proposed method for image classification task.

## 4 EXPERIMENTS

To validate our theoretical investigation, presented in Section 2, and to empirically demonstrate the behavior and effectiveness of the proposed modifications, we experimented with Residual Network (ResNet) and the proposed Procrustes Residual Network architectures on CIFAR10 and CIFAR100 datasets. Training and testing datasets contain 50,000 and 10,000 images of visual classes, respectively [25]. Standard data augmentation (flipping and shifting), same as [5], [6], [8], is adopted. Furthermore, channel means and standard deviations are used to normalize the images. The network is trained using stochastic gradient descent. The weights are initialized using the method proposed in [3] and the initial learning rate is 0.1. Batch size of 128 is used for all the networks. The weight decay is  $10^{-4}$  and momentum is 0.9. The results are based on the top-1 classification accuracy.

Experiments are performed on three different network architectures: 1) *ResNet* contains one convolution layer,  $L$  residual blocks, three of which are transition blocks, and one fully connected layer. Each residual block consists of three convolution layers, as depicted in Figs. 1b and 1c, resulting in a network of depth  $3L + 2$ . This is the same architecture as in [6]. 2) *ProcResNet* has the same architecture as ResNet, except the transition layers are modified, as explained in Section 3. In this design, 3 extra convolution layers are added to the network. However, we can use the first convolution layer of the original ResNet design to match the dimensions and only add two extra layers. This leads to a network of depth  $3L + 4$ . 3) *Plain* network is also same as ResNet without the skip connection in all the  $L$  residual blocks, as shown in Fig. 1e.

Furthermore, to decrease the computational burden of the proposed regularization, we perform the projection, as described in Section 3, every 2 iterations. This reduces the computation time significantly without hurting the performance much. In this setting, performing the proposed regularization increases the training time for ResNet164 about 7.6 percent. However, since we perform the regularization only on three blocks, regardless of the depth, as the network becomes deeper the computational overhead becomes less significant. For example, implementing the same projections on ResNet1001 increases the training time by only 3.5 percent. This is significantly less computation compared to regularization using SVD, which leads to 53 and 23 percent training time overhead for ResNet164 and ResNet1001, respectively.<sup>1</sup>

1. An implementation of ProcResNet is provided here: <https://github.com/zaeemzadeh/ProcResNet>

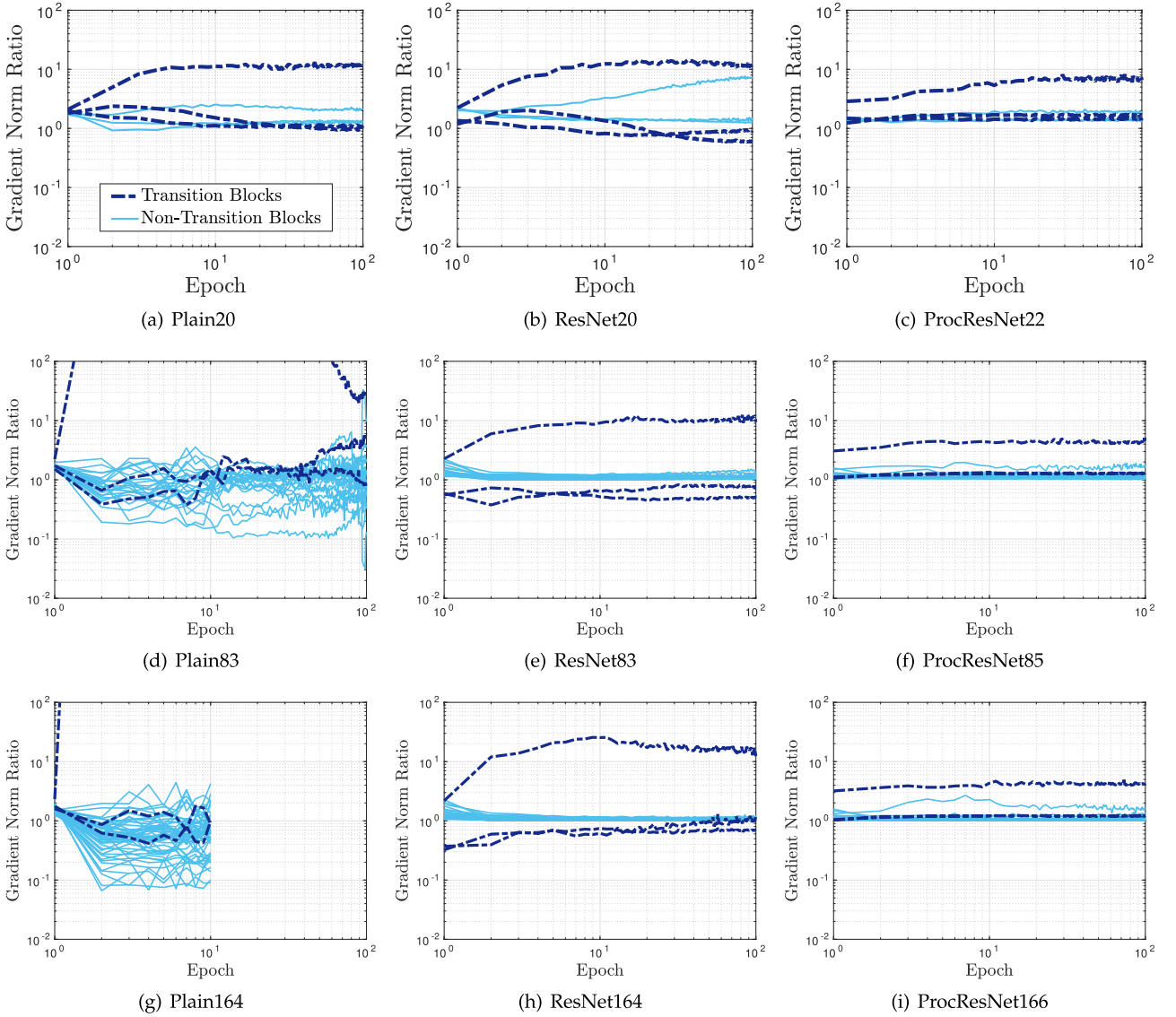


Fig. 3. Training on CIFAR10. Gradient norm ratio over the first 100 epochs for transition blocks (blocks that change the dimension) and non-transition blocks (blocks that do not change the dimension). The darker color lines represent the transition blocks and the lighter color lines represent the non-transition blocks. The proposed regularization enhances the norm-preservation of the transition blocks effectively.

#### 4.1 Norm-Preservation

In the first set of experiments, the behavior of different architectures is studied as the function of network depth. To this end, the ratio of gradient norm at output to gradient norm at input, i.e.,  $\|\frac{\partial \mathcal{E}}{\partial x_{l+1}}\|_2$  to  $\|\frac{\partial \mathcal{E}}{\partial x_l}\|_2$ , is captured for all the residual blocks,<sup>2</sup> both transition and non-transition. Fig. 3 shows the ratios for different blocks over training epochs. We ran the training for 100 epochs, without decaying the learning rate. Plain network (Fig. 3g) with 164 layers became numerically unstable and the training procedure stopped after 10 epochs.

Several interesting observations can be made from this experiment:

- This experiment emphasizes the fact that one needs more than careful initialization to make the network

norm-preserving. Although the plain network is initially norm-preserving, the range of the gradient norm ratios becomes very large and diverges from 1, as the parameters are updated. However, ResNet and ProcResNet are able to enforce the norm-preservation during training procedure by using identity skip connection.

- As the networks become deeper, the plain network becomes less norm preserving, which leads to numerical instability, optimization difficulty, and performance degradation. On the contrary, the non-transition blocks, the blocks with identity mapping as skip connection, of ResNet and ProcResNet become extra norm preserving. This is in line with our theoretical investigation for residual networks, which states that as we stack more residual blocks the network becomes extra norm-preserving.
- Comparing Plain83 (Fig. 3d) and Plain164 (Fig. 3h) networks, it can be observed that most of the blocks behave fairly similar, except one transition block.

2. In Plain architecture, which does not have skip connections, the gradient norm ratio is obtained at the input and output of its building blocks as depicted in Fig. 1e.

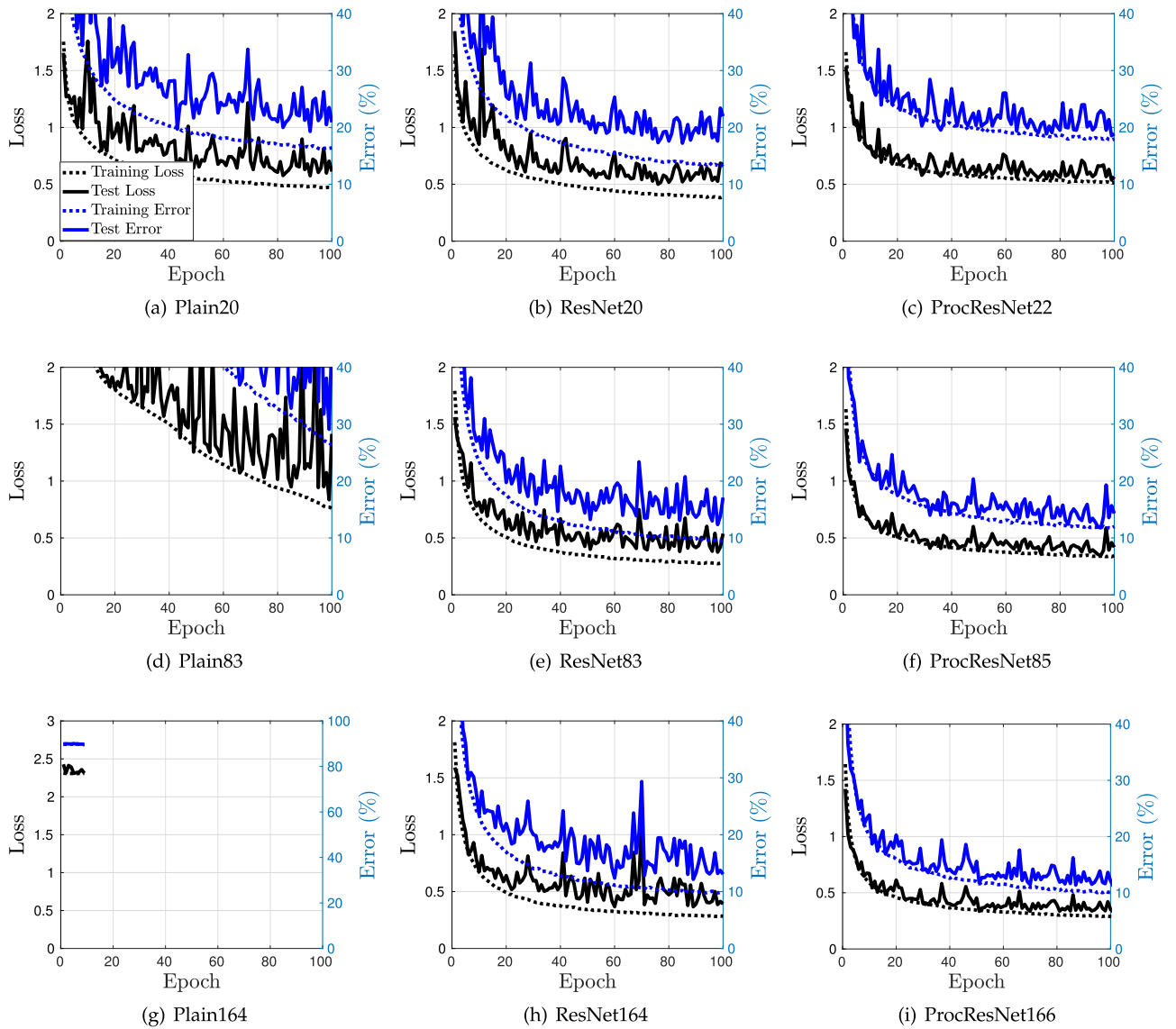


Fig. 4. Loss (black lines) and error (blue lines) during training procedure on CIFAR10. Solid lines represent the test values and dotted lines represent the training values. This experiments shows how the residual connections enhance the stability of the optimization and how the proposed regularization enhances the stability even further.

Specifically, in Plain83, the gradient norm ratio of the first transition block goes up to 100 in the first few epochs. But it eventually decreases and the network is able to converge. On the other hand, in Plain164, the gradient norm ratio of the same block becomes too large, which makes the network unable to converge. Hence, a single block is enough to make the optimization difficult and numerically unstable. This highlights the fact that it is necessary to enforce norm-preservation on all the blocks.

- In ResNet83 (Fig. 3e) and ResNet164 (Fig. 3h), it is easy to notice that only 3 transition blocks are not norm preserving. As mentioned earlier, due to multiplicative effect, the magnitude of the gradient will not be preserved because of these few blocks.
- The behaviors of ResNet and Plain architectures are fairly similar for depth of 20. This was somehow expected, since it is known that the performance gain achieved by ResNet is more significant in deeper

architectures [5]. However, even for depth of 20, ProcResNet architecture is more norm preserving.

- In ProcResNet, the only block that is less norm preserving is the first transition block, where the 3 RGB channels are transformed into 64 channels. This is because, as we have shown in Fig. 2, under such condition, where the number of input channels is very small, the assumption that energy of the gradient signal in the low-dimensional subspace, corresponding to the few non-zero singular values, is approximately proportional to the size of the subspace is violated with higher probability.
- The ratios of the gradients for all networks, even the Plain network, are roughly concentrated around 1, while training is stable. This shows that some degree of norm preservation exists in any stable network. However, as clear in the Plain network, such biases of the optimizer is not enough and we need skip connections to enforce norm preservation throughout



TABLE 1  
Mean and Maximum Generalization Gap (%) During the  
First 100 Epochs of Training on CIFAR10 for Different  
Network Architectures, Averaged over 10 Runs

Depth	Plain		ResNet		ProcResNet	
	mean	max	mean	max	mean	max
20	6.7	20.0	5.5	23.1	2.3	8.3
83	7.5	30.1	5.1	12.5	2.0	7.7
164	-	-	5.2	18.7	3.3	8.7

training and to enjoy its desirable properties. Furthermore, although the transition blocks of ResNet tend to converge to be more norm preserving, our proposed modification enforces this property for all the epochs, which leads to stability and performance gain, as will be discussed shortly.

This experiment both validates our theoretical arguments and clarifies some of the inner workings of ResNet architecture, and also shows the effectiveness of the proposed modifications in ProcResNet. It is evident that, as stated in Theorem 1, addition of identity skip connection makes the blocks increasingly extra norm-preserving, as the network becomes deeper. Furthermore, we have been able to enhance norm-preserving property by applying the changes proposed in Section 3.

## 4.2 Optimization Stability and Learning Dynamics

In the next set of experiments, numerical stability and learning dynamics of different architectures is examined. For that, loss and classification error, in both training and testing phases, are depicted in Fig. 4. This experiment illustrates that how optimization stability of deep networks is improved significantly, and how it can be further improved by having norm preservation in mind during the design procedure.

As depicted in Fig. 4, unlike the plain network, training error and loss curves corresponding to ResNet and ProcResNet architectures are consistently decreasing as the number of layers increases, which was the main motivation behind proposing residual blocks [5]. Moreover, Figs. 4a and 4d show that the plain networks have a poor generalization performance. The fluctuations in testing error shows that the points along the optimization path of the plain networks do not generalize well. This issue is also present, to a

lesser extent, in ResNet architecture. Comparing Figs. 4h and 4b, we can see that the fluctuations are more apparent in deeper ResNet networks. However, in proposed ProcResNet architecture, the amplitude of the fluctuations is smaller and does not change as the depth of the network is increased. This indicates that ProcResNet architecture is taking a better path toward the optimum and has better generalization performance.

To quantify this, we repeated the training 10 times with different random seeds and measured the generalization gap, which is the difference between training and testing classification error, for the first 100 epochs. Table 1 shows the mean and max generalization gap, averaged over 10 different runs. This results indicate that generalization gap of ProcResNet is smaller. Furthermore, the generalization gap fluctuates far less significantly for ProcResNet, as quantified by the difference between mean generalization gap and maximum generalization gap.

The implication of this is that by modifying only a few blocks in an extremely deep network, it is possible to make the network more stable and improve the learning dynamics. This emphasizes the utmost importance of norm-preservation of all blocks in avoiding optimization difficulties of very deep networks. Moreover, this sheds light on the reasons why architectures using residual blocks, or identity skip connection in general, perform so well and are easier to optimize.

## 4.3 Classification Performance

In this section, we show the impact of the proposed norm-preserving transition blocks on the classification performance of ResNet. Table 2 compares the performance of ResNet and its EraseReLU version, as proposed in [26], with and without the proposed transition blocks. The results for standard ResNet are the best results reported by [6] and [26] and the results of ProcResNet are obtained by making the proposed changes to standard ResNet implementation.

Table 2 shows that the proposed network performs better than the standard ResNet. This performance gain comes with a slight increase the number of parameters (under 1 percent) and by changing only 3 blocks. The total number of residual blocks for ResNet164 and ResNet1001 are 54 and 333, respectively. Furthermore, Fig. 5 compares the parameter efficiency of ResNet and ProcResNet architectures. The

TABLE 2  
Performance of Different Methods on CIFAR-10 and CIFAR-100 Using Moderate Data  
Augmentation (flip/translation). The modified transition blocks in ProcResNet can  
improve the accuracy of ResNet significantly.

Architecture	Setting	# Params	Depth	Error (%)	
				CIFAR10	CIFAR100
ResNet [6]	pre-activation	1.71M	164	5.46	24.33
		10.32M	1001	4.62	22.71
	ErasedReLU [26]	1.70M	164	4.65	22.41
		10.32M	1001	4.10	20.63
ProcResNet	pre-activation	1.72M	166	4.75	22.61
		10.33M	1003	3.72	19.99
	ErasedReLU [26]	1.72M	166	4.53	21.91
		10.33M	1003	<b>3.42</b>	<b>18.12</b>

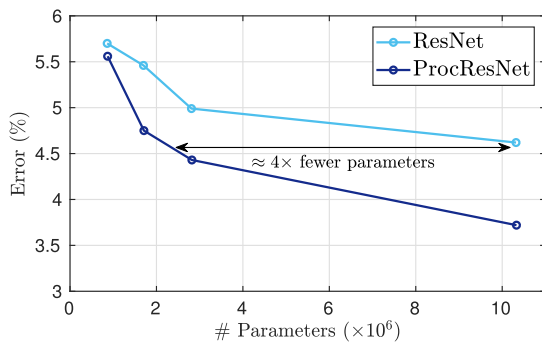


Fig. 5. Comparison of the parameter efficiency on CIFAR10 between ResNet and ProcResNet.

TABLE 3  
Ablation Study on ResNet With 164 Layers on CIFAR100

Transition Block	Projection	Error (%)
Original	No	24.33
Modified	No	23.06
Modified	Yes	22.61

results indicate that the proposed modification can improve the parameter efficiency significantly. For example, ProcResNet274 (with 2.82M parameter) slightly outperforms ResNet1001 (with 10.32M parameters). This translates into about 4x reduction in the number of parameters to achieve the same classification accuracy. This illustrates that we are able to improve the performance by changing a tiny portion of the network and emphasizes the importance of norm-preservation in the performance of neural networks.

Finally, Table 3 investigates the impact of changing the architecture, i.e., moving the convolution layer from the skip connection to before the skip connection, and performing the proposed regularization, separately. Each of these design components have positive impact on the performance of the network, as both of them enhance the norm preservation of the transition block, which further highlights the impact of norm preservation on the performance of the network.

## 5 CONCLUSION

This paper theoretically analyzes building blocks of residual networks and demonstrates that adding identity skip connection makes the residual blocks norm-preserving. Furthermore, the norm-preservation is enforced during the training procedure, which makes the optimization stable and improves the performance. This is in contrast to initialization techniques, such as [14], which ensure norm-preservation only at the beginning of the training. Our experiments validate our theoretical investigation by showing that (i) identity skip connection results in norm preservation, (ii) residual blocks become extra norm-preserving as the network becomes deeper, and (iii) the training can become more stable through enhancing the norm preservation of the network. Our proposed modification of ResNet, Procrustes ResNet, enforces norm-preservation on the transition blocks of the network and is able to achieve better

optimization stability and performance. For that we propose an efficient regularization technique to set the nonzero singular values of the convolution operator, without performing singular value decomposition. Our findings can be seen as design guidelines for very deep architectures. By having norm-preservation in mind, we will be able to train extremely deep networks and alleviate the optimization difficulties of such networks.

## ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grants 1741431 and CCF-1718195 and the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. D17PC00345. The views, findings, opinions, and conclusions or recommendations contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

## REFERENCES

- [1] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, Oct. 2017.
- [2] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2924–2932.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.
- [4] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.
- [7] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2377–2385.
- [8] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
- [9] A. E. Orhan and X. Pitkow, "Skip connections eliminate singularities," in *Proc. 6th Int. Conf. Learn. Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=HkwBEMWCZ>
- [10] M. Hardt and T. Ma, "Identity matters in deep learning," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.
- [11] K. Kawaguchi, "Deep learning without poor local minima," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 586–594.
- [12] D. Balduzzi, M. Frean, L. Leary, J. P. Lewis, K. W.-D. Ma, and B. McWilliams, "The shattered gradients problem: If resnets are the answer, then what is the question?" in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, pp. 342–350, 2017.
- [13] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Proc. 29th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 550–558.
- [14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
- [15] L. Dinh, J. Sohl-Dickstein, Google, B. Samy, and B. Google Brain, "Density estimation using Real NVP," in *Proc. Int. Conf. Learn. Representations*, 2017.

- [16] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The reversible residual network: Backpropagation without storing activations," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 2211–2221.
- [17] J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J. Jacobsen, "Invertible residual networks," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, pp. 573–582, 2019.
- [18] P. L. Bartlett, S. N. Evans, and P. M. Long, "Representing smooth functions as compositions of near-identity functions with implications for deep network optimization," 2018, *arXiv: 1804.05012*. [Online]. Available: <https://arxiv.org/abs/1804.05012>
- [19] K. Kawaguchi and Y. Bengio, "Depth with nonlinearity creates no bad local minima in ResNets," *Neural Netw.*, vol. 118, pp. 167–174, 2019.
- [20] S. Gunasekar, J. D. Lee, N. Srebro, and D. Soudry, "Implicit bias of gradient descent on linear convolutional networks," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9482–9491.
- [21] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: Closing the generalization gap in large batch training of neural networks," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1729–1739.
- [22] H. Sedghi, V. Gupta, and P. M. Long, "The singular values of convolutional layers," in *Proc. Int. Conf. Learn. Representations*, 2019.
- [23] J. C. Gower, *et al.*, *Procrustes Problems*. vol. 30. Oxford, U.K.: Oxford Univ. Press, 2004.
- [24] N. J. Higham, "Stable iterations for the matrix square root," *Numer. Algorithms*, vol. 15, no. 2, pp. 227–242, 1997.
- [25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images.(2009)," Univ. Toronto, Toronto, ON, *Tech. Rep. TR-2009*, 2009.
- [26] X. Dong, G. Kang, K. Zhan, and Y. Yang, "EraseReLU: A simple way to ease the training of deep convolution neural networks," 2017, *arXiv: 1709.07634*. [Online]. Available: <https://arxiv.org/abs/1709.07634>



**Alireza Zaeemzadeh** (Student Member, IEEE) received the BS degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2014. He is currently working toward the PhD degree in electrical engineering at the University of Central Florida, Orlando, Florida. His current research interests include the areas of machine learning, linear algebra, and optimization. His awards and honors include the University of Central Florida multidisciplinary doctoral fellowship and graduate dean's fellowship.



**Nazanin Rahnavard** (Senior Member, IEEE) received the PhD degree in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, in 2007. She is currently an associate professor with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida. She is the recipient of NSF CAREER award, in 2011 and 2020 UCF's College of Engineering and Computer Science Excellence in Research Award. She has interest and expertise in a variety of research topics in the communications, networking, signal processing, and machine learning areas. She serves on the editorial board of the *Elsevier Journal on Computer Networks (COMNET)* and on the Technical Program Committee of several prestigious international conferences.



**Mubarak Shah** (Fellow, IEEE) is currently the UCF Trustee chair professor and the founding director of the Center for Research in Computer Vision at the University of Central Florida (UCF), Orlando, Florida. He is a fellow of the NAI, AAAS, IAPR, and SPIE. He is an editor of an international book series on video computing, was the editor-in-chief of the *Machine Vision and Applications Journal*, and an associate editor of the *ACM Computing Surveys Journal*. He was the program cochair of CVPR 2008, an associate

editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and a guest editor of the special issue of the *International Journal of Computer Vision on Video Computing*. His research interests include video surveillance, visual tracking, human activity recognition, visual analysis of crowded scenes, video registration, UAV video analysis, and so on. He has served as an ACM distinguished speaker and IEEE distinguished visitor speaker. He is a recipient of ACM SIGMM Technical Achievement award; IEEE Outstanding Engineering Educator Award; Harris Corporation Engineering Achievement Award; an honorable mention for the ICCV 2005 Where Am I? Challenge Problem; 2013 NGA Best Research Poster Presentation; 2nd place in Grand Challenge at the ACM Multimedia 2013 Conference; and Runner-up for the Best Paper Award in ACM Multimedia Conference, in 2005 and 2010. At UCF, he has received Pegasus Professor Award, University Distinguished Research Award, Faculty Excellence in Mentoring Doctoral Students; Scholarship of Teaching and Learning Award, Teaching Incentive Program Award, and Research Incentive Award.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/csdl](http://www.computer.org/csdl).