Revisiting Residual Networks with Nonlinear Shortcuts

Chaoning Zhang
chaoningzhang1990@gmail.com
Francois Rameau
Seokju Lee
Junsik Kim
Philipp Benz
Dawit Mureja Argaw
Jean-Charles Bazin
In So Kweon

Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, South Korea

Abstract

Residual networks (ResNets) with an identity shortcut have been widely used in various computer vision tasks due to their compelling performance and simple design. In this paper we revisit ResNet identity shortcut and propose RGSNets which are based on a new nonlinear ReLU Group Normalization (RG) shortcut, outperforming the existing ResNet by a relatively large margin. Our work is inspired by previous findings that there is a trade-off between representational power and gradient stability in deep networks and that the identity shortcut reduces the representational power. Our proposed nonlinear RG shortcut can contribute to effectively utilizing the representational power of relatively shallow networks and outperform much (3 or 4 times) deeper ResNets, which demonstrates the high efficiency of RG shortcut. Moreover, we have explored variations of RGSNets, and our experimental result shows that Res-RGSNet combining the proposed RG shortcut with the existing identity shortcut achieves the best performance and is robust to network depth. Our code and models are publicly available on our website.

1 Introduction

Visual recognition has been an active research topic in the past few decades [16, 20]. Recently, the research trend has shifted from traditional hand-crafted feature design, such as SIFT [20] and HOG [3], to feature extraction through deep networks, which has been proven to outperform hand-crafted features [9, 19, 27]. Numerous techniques have emerged to improve the performance of Deep Neural Networks (DNNs). Among them, two techniques in particular significantly contributing to the success of DNN are *ResNet (identity) shortcut* [9, 10] and *normalization* techniques [14, 34]. Both techniques improve the performance without (or with negligible) extra parameters or computation. These seminal works led to

numerous follow-ups [1, 34, 36, 37] attempting to perfect their original design. In this paper, we revisit ResNet identity shortcut and propose a nonlinear ReLU Group-Normalization (RG) shortcut to boost the performance of the widely used ResNet [9].

In the past few years, the trend in deep learning applications is that researchers deploy widely used networks, such as ResNet, as a backbone structure. Apart from network performance, there are three general concerns for the backbone network choice: computation time, memory footprint and network simplicity. In other terms, for the same performance, networks that are fast, small and simple are preferred. In practice, ResNet has almost become the standard DNN for various computer vision applications due to its compelling performance and simple design [9, 11]. DenseNets achieve better performance than ResNets under the constraint of the same amount of parameters [13]. However, DenseNets generally require large GPU memory and plenty of computation time for training. GoogleNets [28] and its variants, such as Inception-v3 [29], show favorable performance but require careful engineering design and therefore violate the key requirement of simplicity.

Despite numerous implementation differences, such as depth, width and cardinality [10, 36, 37], the ResNet family can be roughly divided into two categories: original ResNet [9] and preactivation ResNet [10], both of them adopting the identity shortcut. The original ResNet first proposed residual learning to solve the slow convergence problem in very deep networks [9]. The preactivation ResNet, then, went further to adopt the approach of a direct information propagation path in the entire network to exploit the benefit of the identity shortcut in a more direct way [10]. Despite wide popularity, it has been shown that the identity shortcut, which enables the training of very deep networks, is at the same time a weakness of ResNets [37]. The weakness is that skip-connection affects residual blocks to learn less during training and thus reduces the representational power [37]. In this paper, we propose a new ReLU and Group Normalization (RG) shortcut to get around such "weakness". Our approach is mainly inspired by findings in the previous work [22] that there is a trade-off between representational power and gradient stability.

Previous works, such as WideResNet [37] and ResNeXt [36], mainly focused on designing more efficient ResNet structures by making changes to the residual path. In contrast, our work focuses on making changes to the shortcut path while preserving the residual path.

Contributions To sum up, our contributions are as follows.

- We propose a simple yet effective nonlinear RG shortcut which alleviates the representational power reducing problem of the identity shortcut in ResNet. Our proposed relatively shallow RGSNet can outperform much (3 or 4 times) deeper ResNets.
- We explore several variants of RGSNet through extensive experiments on different datasets, which show that Res-RGSNet performs the best among all the explored models, and is robust to network depth. This empirically indicates that identity shortcut and RG shortcut are non-exclusive but complement each other to boost network performance.

2 Related works

We propose RGSNet and its variants based on the new RG shortcut. In this section, we review the findings and understandings of recent works related to our proposed RG shortcut.

Normalization techniques Normalization techniques have been used in numerous applications to facilitate the convergence and improve the performance of DNNs [1, 14, 34]. Batch normalization [14], one milestone technique in DNN, is the most widely used normalization technique in various computer vision tasks. However, it introduces the depen-

dencies between examples in a minibatch, making it less applicable to noise-sensitive applications [6, 7, 25]. Alternative normalization techniques have been proposed in follow-up works, such as weight normalization [25], instance normalization [31] and layer normalization [1]. The recently proposed group normalization [34], has achieved comparable performance as batch normalization without normalizing along the batch direction. It has the advantage of being used for applications with large memory demand [6, 30]. Empirically, we have found that group normalization is more efficient in the shortcut than batch normalization (see Section 4.1).

ResNet understandings Due to the compelling performance and simple design of ResNets, many researchers have put efforts to understand its mechanisms [22, 32]. It has been shown that ResNet behaves as an ensemble of exponentially many shallow networks [32]. Wu et al. Wu et al. [35] have confirmed this ensemble characteristic while arguing that the number of shallow networks grows linearly instead of exponentially. Philipp et al. [22] recently have shown that the success of ResNet lies in solving the exploding gradient problem. Balduzzi et al. [2] found that ResNet shortcut improves the performance by handling the shattering gradient problem, demonstrating that the gradient of DNN behaves as noise which can be suppressed by the ResNet shortcut. Both exploding gradient [22] and shattering gradient [2] have shown that identity shortcut improves the performance because it improves gradient stability. On the other hand, Zagoruyko and Komodakis [37] have identified that skip-connection affects residual blocks to learn less during training, which is the weakness of ResNets (reducing the representational power) [37]. As a result, in DNN there is a trade-off between preserving representational power[1] and improving gradient stability [2, 22, 26, 37].

Trade-off between representational power and gradient stability This trade-off philosophy traces back to one core problem of biological neural networks, widely known as the plasticity-stability dilemma [21]. More recently, it has been highlighted by [22] that there is an inherent tension between preserving representational power and avoiding exploding gradients (gradient stability). The decrease in representational power is related to pseudo-linearity [22]. The representational power of DNN has been defined as the non-linearity that the computed function (DNN) captures [23]. Interestingly, the trade-off is also applied to the normalization technique [22]. Normalization technique helps optimization through improving gradient stability [26] and meanwhile limits the network representational power by discarding the absolute scale of activations [14]. In practice, the downside of less representational power caused by the identity shortcut and normalization does not get much attention, since the improved stability contributes to superior performance, especially when the network is very deep. However, it does not mean that gradient stability is more important than representational power. It has been claimed in [22] that degrading either of them can severely hamper training.

3 Proposed RG Shortcut

Various shortcut methods, including constant scaling, exclusive gating, shortcut-only gating, 1×1 conv shortcut, dropout shortcut, have been explored in [10]. However, these variations systematically demonstrate inferior performance compared to the identity shortcut [9, 10]. The reason for the failure of these shortcut methods is that they significantly disrupt information propagation, resulting in low stability. The identity shortcut in ResNet imposes stability

[1] The term "expressive power" instead of "representational power" is used in some works [15, 23]

on the network through its inherent linearity, while limiting representational power, which leads to a dilemma situation for the shortcut design. Therefore, we conjecture that a reasonable shortcut design should meet three criteria: (1) the shortcut has non-linear characteristics but avoids unnecessary information transformation; (2) it should contribute to extra stability to compensate gradient stability decrease due to the non-linear characteristics of (1). (3) it should be differentiable for network training and involve minimum engineering effort. Designing a shortcut fulfilling the above criteria is not a trivial task. We come up with a simple yet intuitive shortcut design: **(nonlinear) ReLU activation function + group normalization**. The nonlinear activation function ReLU is included to contain nonlinear characteristics and the normalization serves the purpose to add stability. We employ group normalization as our choice of normalization after the ReLU to add extra stability. The extra stability might be due to the fact that group normalization performs normalization along the channel direction, which is orthogonal to batch direction in batch normalization [34]. This design choice has been validated by our empirical result as shown in Table 1. For simplicity, we set the hyper-parameter G (number of groups) to 32 in our experiments by default as in [34]. When the width is set to $0.25\times$, we proportionally scale G to 8. We term a network adopting our RG shortcut RGSNet. The understanding of the RGSNet will be analyzed based on the comparison with the original ResNet and preactivation ResNet, as shown in Figure 1. To facilitate the discussion, a general form for the residual unit blocks is applied [10]:

$$y_l = h(x_l) + F(x_l, W_l) \quad \text{and} \quad x_{l+1} = f(y_l), \tag{1}$$

where F is the residual function and W_l denotes the learnable weights. x_l and x_{l+1} are the inputs of the l th and $(l+1)$ th residual units respectively. h and f are the two mapping functions. Both functions h and f in Eq. 1 are identity mappings in preactivation ResNet [10]. The core idea of preactivation ResNet is to create a direct path for propagating information through the entire network, which benefits when the network is extremely deep (*i.e.*, more than 200 layers) [10]. The term "direct" indicates the information has been added without any modification, which makes the information from the beginning have a direct representation (path) until the end as shown in Figure 2. For the original ResNet, since the ReLU after summation becomes less active after some training, it can be roughly seen as an inactive ReLU, which makes the original ResNet have a pseudo-direct path, which has been analyzed by [10]. The group normalization in our design can help avoid the problem of inactive ReLU because group normalization can shift the channel average to around zero by subtracting the average of its group. Another purpose of this normalization term is to add extra stability to the network because pure average shifting can make the network unstable. Recognizing the trade-off between representational power and gradient stability, our RG shortcut intentionally avoids such a direct path propagation to introduce non-linearity. This RG shortcut still uses the previous layer information as an identity shortcut but in a fundamentally different way.

Besides the base RGSNet shown in Figure 3 (a), we propose two improved variants of RGSNets, as shown in Figure 3 (b) and (c). The RGSNet with only one RG shortcut is called base RGSNet, or simply *b*-RGS-Net, to differentiate from the two improved variants. The improved *d*-RGSNet is proposed by inserting one additional inner RG shortcut. This inner RG shortcut is added on the Conv (3×3). It can not be added on the Conv (1×1) because the channel dimension does not match. Since each convolutional block has two shortcuts, we name it dual-RGSNet, or simply *d*-RGSNet. Furthermore, by combining the *d*-RGSNet with the identity shortcut, we obtain the residual RGSNet, or simply Res-RGSNet.

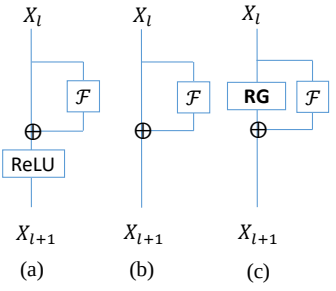

Figure 1: Shortcut schematics: (a) original ResNet [9], (b) preactivation ResNet [10], (c) proposed RGSNet.

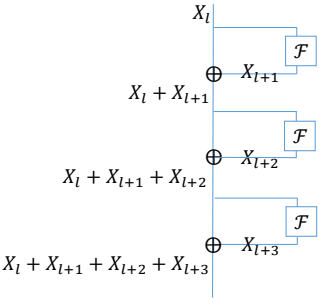

Figure 2: Direct path of a preactivation ResNet.

4 Experiments

We first conduct the ablation studies of the proposed RG shortcut on ImageNet-1K, which is the benchmark dataset for classification [4]. To show the generalization of the proposed idea on different datasets, we also perform experiments on CIFAR-100 as well as two common object detection datasets: PASCAL-VOC 2007 and MS COCO 2014. We then perform extensive ablation studies on CIFAR-100 and show that the proposed RG shortcut outperforms the identity shortcut for various widths (number of channels) and depths (number of layers), as well as on the ResNeXt structure. Since depth is one of the main factors that influences the stability of the network, we performed an additional series of experiments by changing the (network) depth. All the results in this paper are produced with the PyTorch framework.

4.1 Ablation studies for RG shortcut design

ResNet-50 is one of the most widely used ResNet models and has also become the benchmark ResNet model [11, 33, 34] to evaluate new approaches. In this section, we use it to conduct ablation studies on ImageNet-1K with the single-crop results on validation dataset shown in Table 1. The common setup for training on ImageNet is to use 8 GPUs with batch size of 256, with an initial learning rate of 0.1 which is divided by 10 at every 30 epochs [9, 12, 33]. We train our network on 4 GPUs with a batch size of 128 and accordingly the initial learning is set to 0.05 based on the linear scaling rule [8]. The ablation study result is shown in Table 1, in which "original" refers to the original ResNet (which is the same for the following tables), "Preact" refers to the preactivation Resnet, "RBS" refers to adopting ReLU and batch normalization for the shortcut, "RGS" refers to the base RGSNet, "GS" refers to that the shortcut is only group normalization without ReLU, "RGB" refers to RG module put in the bottleneck, or f of Eq. 1 instead of the shortcut.

First, the result shows that the original ResNet achieves comparable (even slightly better) performance with the preactivation ResNet, which is not surprising as it has been demonstrated in [10] that the preactivation ResNet outperforms the original ResNet only when the network becomes extremely deep (more than 200 layers). Thus in the remainder of this paper, our result is only compared with the original ResNet. Table 1 shows that RGSNet outperforms the original ResNet by a relatively large margin, while RBS performance is inferior to RGS, indicating that group normalization is a more preferable choice for the nor-

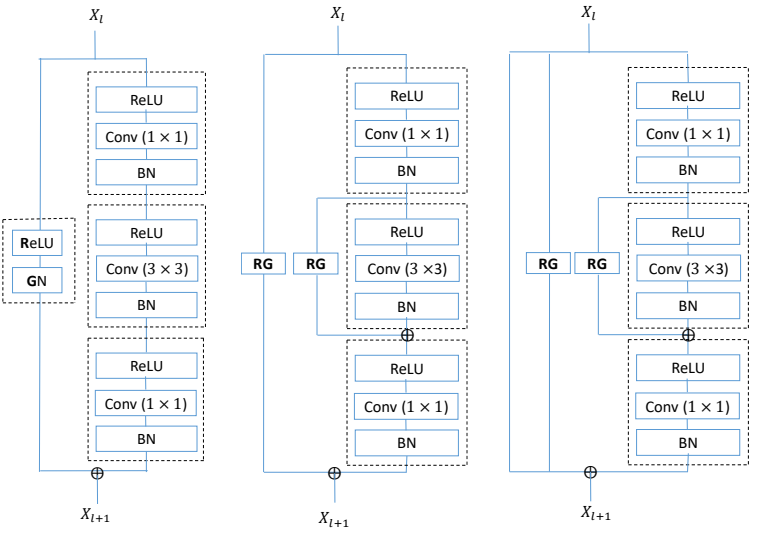

Figure 3: Proposed RGSNets: (a) *b*-RGSNet, (b) *d*-RGSNet, (c) Res-RGSNet.

malization. The comparison between RGS and GS indicates the importance of ReLU in the shortcut path. Moreover, to validate the importance of group normalization in the position of the shortcut, we insert group normalization after each ReLU (totally two in F) in the residual path. The comparison in Table 1 (see 2GN) shows that such design leads to marginal improvement over the original ResNet, implying that group normalization complements batch normalization originally placed in the residual modules. However still, adding group normalization in the residual path shows less favorable performance compared to inserting it in the shortcut path. Overall, the above analysis demonstrates that applying the RG shortcut is an appropriate design for boosting the performance.

4.2 Evaluation of the proposed RGSNets

In this section, we aim to evaluate the proposed base RGSNet and its two variants. We first perform the evaluation on ImageNet-1K and the results are available in Table 2. It shows that all the proposed RGSNets outperform the original ResNet, and the proposed Res-RGSNet, performing the best among all the proposed RGSNets, outperforms original ResNet by a large margin of 1.60%. To evaluate the robustness of the proposed RGSNets, different widths and depths are further tested on CIFAR-100 and the results are available in Table 3, where $1\times$ indicates the same width as the classical ResNet50 [9] and $0.25\times$ indicates 0.25 times of that width. We train for 164 epochs with the learning rate starting from 0.1 and divided by 10 at 82 and 124 epochs respectively. We set the batch size to 128 and weight decay to 5e-4. By default, two GPUs are used for the CIFAR-100 experiment unless specified. Similar to the result on ImageNet, all the proposed RGSNets consistently outperform the original ResNet, and Res-RGSNet performs the best, followed by the *d*-RGSNet and the *b*-RGSNet. To further confirm that the proposed RGSNets are robust to different widths, extra experiments are conducted on ResNet26 and the results are in Table 4. To further prove that

Arch	Top-1 (%)	Top-5 (%)
Original [9]	23.81	7.14
Preact [10]	23.93	7.13
RBS	23.72	7.02
RGS	**22.68**	**6.42**
GS	23.53	6.67
RGB	23.07	6.44
Original (2 GN)*	23.41	6.80

Table 1: Classification error on ImageNet-1K for ablation study.

Arch	Top-1 (%)	Top-5 (%)
Original	23.81	7.14
b-RGS	22.68	6.42
d-RGS	22.40	6.25
Res-RGS	22.21	5.99

Table 2: Classification error on ImageNet-1K with different structures.

Arch (width)	Layers	Top-1 (%)
Original (1×)	ResNet26	23.17
b-RGS (1×)	ResNet26	21.32
d-RGS (1×)	ResNet26	20.71
Res-RGS (1×)	ResNet26	20.49
Original (1×)	ResNet50	21.43
b-RGS (1×)	ResNet50	20.42
d-RGS (1×)	ResNet50	20.00
Res-RGS (1×)	ResNet50	19.59
Original (0.25×)	ResNet50	26.13
b-RGS (0.25×)	ResNet50	25.40
d-RGS (0.25×)	ResNet50	24.80
Res-RGS (0.25×)	ResNet50	23.32

Table 3: Classification error on CIFAR-100 with different widths and depths.

the proposed RGSNets are robust to different structures, we also conduct experiments on the popular ResNeXT29 (64×8d) [36] and the results are shown in Table 5. All the above results consistently show that the proposed RGSNets outperform the original ResNet, and the *d*-RGSNet consistently perform in between the *b*-RGSNet and the Res-RGSNet. Therefore, in this paper we formally propose only two RGSNets: ***b*-RGSNet** and **Res-RGSNet**. In the next section, we will test their generalization capability on two detection datasets.

Width	ResNet	*b*-RGSNet	*d*-RGSNet	Res-RGS
0.25×	27.74	26.76	26.43	26.00
0.5×	24.41	23.34	22.91	22.46
1×	23.17	21.32	20.71	20.49
2×	22.40	20.79	20.16	19.76

Table 4: Classification error (in %) on CIFAR-100 based on ResNet26.

Arch	Layers	Top-1 (%)
Original	ResNeXt29	18.34
b-RGS	ResNeXt29	17.67
d-RGS	ResNeXt29	17.48
Res-RGS	ResNeXt29	17.37

Table 5: Classification error on CIFAR-100 based on ResNeXt. 4GPUs are used.

4.3 Test on the detection datasets

We evaluate the formally proposed *b*-RGSNet and Res-RGSNet on two benchmark detection datasets. First, we conduct experiments on the MS COCO 2014 dataset [17], and the average mAP over different IoU thresholds is used for evaluation. Similar to [18, 33], our model, with Faster-RCNN [24] as the detection method, is trained with all the training images as well as a subset of the validation images, holding out 5,000 examples for validation. The network is trained for 5 epochs for fast performance validation on one single GPU and the results are available in Table 6. We further perform experiments on the PASCAL VOC 2007 dataset for 10 epochs for fast convergence and the results are available in Table 7. The experiment result shows that both the proposed *b*-RGSNet and Res-RGSNet outperform the original ResNet, indicating satisfying dataset generalization of the proposed RGSNets.

4.4 Exploring the effect of depth

Our proposed *b*-RGSNet outperforms the original ResNet as analyzed in the above sections, which seems to suggest that the RG shortcut is superior to the existing identity shortcut. However, this is not always true. As summarized in related works, a trade-off between representational power and gradient stability always exists [22]. Our proposed RG shortcut is superior to the identity shortcut in the above results because the identity shortcut imposes strong linearity on the model, reducing the representational power. It can be predicted that the RG shortcut without strong linearity cannot perform well when the network becomes very deep. To verify this, we further compare the *b*-RGSNet and the original ResNet over a wide range of depths. The result in Table 8 indeed shows that after 38 (or 50) layers, the performance of *b*-RGSNet begins to decrease. This is not surprising because when the network goes deep, the stability becomes a more serious concern. Nevertheless, the *b*-RGSNet with 38 layers achieves comparable performance as that of the original ResNet with 152 layers. Combining with an identity shortcut is an intuitive way to overcome the limitation of the RG shortcut. In fact, our proposed Res-RGSNet is inspired by this intuition.

To examine how Res-RGSNet performs for a deep network, we further test it on a wide range of depths and the results are available in Figure 4. Note that these results are for the width of 0.25 $\times$ to reduce computation cost, different from the depth of 1$\times$ in Table 8. The result indicates that Res-RGSNet behaves similar to original ResNet but with better performance over the whole range. Furthermore, to show that this merit of Res-RGSNet comes from the identity shortcut instead of the dual RG shortcut as in the *d*-RGSNet, the result of *d*-RGSNet is also presented in Figure 4. It is not surprising that *d*-RGSNet also suffers from gradient stability problem despite relatively superior performance compared with *b*-RGSNet. We further increase the depth for both ResNet and Res-RGSNet to 200 layers, and the top-1 error of them are 23.74% and 21.30% respectively, which further confirms that Res-RGSNet is robust to different depths. The result supports that when combined together, nonlinear RG shortcut and identity shortcut complement each other to improve the performance. The mechanism how they interact to complement each other will be studied in future work.

4.5 Comparison with SE-ResNet

The above analysis shows that the RG shortcut is superior to the identity shortcut for relatively shallow networks. Furthermore, shallow networks with the RG shortcut achieve comparable performance to much deeper ResNets. However, the RG shortcut features less gradient stability compared with the identity shortcut. The Res-RGSNet adopting both RG shortcut and identity shortcut combines their advantages and achieves a better trade-off of representational power and gradient stability. Thus Res-RGSNet is the optimal design among all explored networks. The difference between Res-RGSNet and the original ResNet lies in that the added RG shortcut module, which improves the representational power. In the re-

Arch	Layers	mAP.5	mAP.75	mAP [.5, .95]
Original	ResNet50	51.5	33.7	31.5
b-RGS	ResNet50	53.2	34.8	32.8
Res-RGS	ResNet50	54.4	35.3	33.4

Table 6: mAP (%) on MS COCO validation dataset.

Arch	Layers	mAP.5
Original	ResNet50	73.89
b-RGS	ResNet50	74.20
Res-RGS	ResNet50	74.59

Table 7: mAP (%) on PASCAL-VOC-2007 validation dataset.

Layer	Layer design	original	*b*-RGSNet
26	[2, 2, 2, 2]	22.56	20.96
32	[2, 3, 3, 2]	22.19	20.45
38	[3, 3, 3, 3]	21.72	**20.27**
50	[3, 4, 6, 3]	21.43	20.46
62	[3, 4, 10, 3]	21.20	22.13
77	[3, 4, 15, 3]	20.91	24.15
101	[3, 4, 23, 3]	20.42	25.17
152	[3, 8, 36, 3]	**20.08**	31.59

Table 8: Classification error (%) on CIFAR-100 with respect to different number of layers for the width of 1×. 4 GPUs are used.

Arch(width)	44 layers	50 layers
original (0.25×)	26.58	26.13
b-RGS (0.25×)	25.31	25.40
Res-RGS (0.25×)	23.80	**23.34**
SE-ResNet (0.25×)	25.05	24.76
Res-RGS+SE (0.25×)	**23.55**	23.36

Table 9: Classification error (%) on CIFAR-100 with different structures (0.25× width).

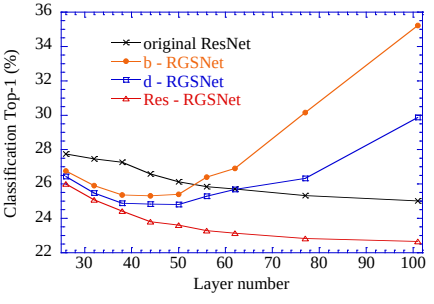

Figure 4: Classification error on CIFAR-100 over a wide range of depths.

cently proposed SE-Net [12] which won the first place of ILSVRC 2017, an attention module was added to improve network performance through enhancing the representational power. Thus we further compare our proposed Res-RGSNet with SE-ResNet on both ImageNet and CIFAR-100. On ImageNet, the top-1 error of SE-ResNet50 we reproduced is 22.92%, and our proposed Res-RGSNet outperforms it by 0.71%. The result comparison of CIFAR-100 is shown in Table 9. The Res-RGSNet achieves noticeably superior performance (more than 1%) than SE-Net. Moreover, compared with the SE attention module, our proposed RG module adds almost zero parameters (except those few in the group affine transformation), making our proposed RG module more favorable. Besides, it is interesting to note that by adding the SE attention module to our Res-RGSNet, the performance is further improved by a small margin (0.25%) when the depth is 44 layers. However, such improvement is absent when the depth increased to 50 layers. The reason could be attributed to the high representational power that Res-RGSNets already capture.

4.6 Results summary

In the previous sections, we demonstrated the power of the proposed RG shortcut through extensive experimental studies. In this section, we compare our results with those reported by previous works. From Table 10, we can observe that Res-RGSNet-50 achieves comparable or better performance than previous works. Compared with the original ResNet, our proposed RG shortcut can boost the performance with a negligible amount of additional parameters and computation burden. Our results on CIFAR-100 are summarized in Table 11. Overall,

Architecture	#params	Top-1 (%)	Top-5 (%)
ResNet50 [9]	25.6M	24.01	7.02
ResNet101 [9]	44.6M	22.44	6.21
ResNet152 [9]	60.2M	22.16	6.16
WRN-50-2-bottleneck [37]	68.9M	21.9	6.03
ResNeXt50, 32×4d [37]	25M	22.2	-
SE-ResNet50 [12]	28.1M	23.29	6.62
DenseNet201 [13]	20M	22.58	6.34
DenseNet264 [13]	33.3M	22.15	6.12
b-RGSNet-50	25.6M	22.68	6.42
Res-RGSNet-50	25.6M	22.21	5.99

Table 10: Performance comparison on ImageNet-1K. The results of all the networks (except our RGSNets) are as reported in the corresponding works.

Architecture	Top-1 (%)
ResNet152 (1×)	20.08
Res-RGSNet38 (1×)	19.98
ResNet200 (0.25×)	23.74
Res-RGSNet50 (0.25×)	23.34

Table 11: Results comparison on CIFAR-100. Res-RGSNets can achieve comparable or better performance than corresponding 4 times deeper ResNets.

the results show that Res-RGSNets can achieve comparable or better performance than much deeper ResNets on both ImageNet (152 vs. 50, 3 times deeper) and CIFAR-100 (200 vs. 50, 4 times deeper), which demonstrates the efficiency of the proposed RG shortcut.

5 Conclusion

Motivated by the trade-off between representational power and gradient stability, we propose RGSNet based on a novel RG shortcut which adds negligible extra parameters and computation time. The experimental results show that relatively shallow RGSNet and its variants can outperform much deeper ResNets. Especially, Res-RGSNet achieves the most favorable performance and is robust to depth, which empirically indicates that RG shortcut and identity shortcut are non-exclusive but complement each other to boost network performance. Our proposed Res-RGSNet also outperforms SE-ResNet by a relatively large margin, further verifying the efficiency of the proposed RG shortcut. The in-depth analysis of how nonlinear RG shortcut and identity shortcut interact to complement each other can be an interesting direction to explore in future research.

Acknowledgements This work was funded by Naver Labs. JC Bazin gratefully acknowledges the support of NVIDIA Corporation with the GPU donation used for this research. Francois Rameau was supported by Korean Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2015H1D3A1066564).

References

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

[2] 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 *ICML*, 2017.

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[5] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.

[6] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[8] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[15] Valentin Khrulkov, Alexander Novikov, and Ivan Oseledets. Expressive power of recurrent neural networks. In *ICLR*, 2018.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[20] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[21] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Front Psychol*, 4:504, 2013.

[22] George Philipp, Dawn Song, and Jaime G. Carbonell. Gradients explode-deep networks are shallow-resnet explained. In *ICLR Workshop*, 2018.

[23] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. *ICML*, 2017.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[25] Tim Salimans and Durk P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NIPS*, 2016.

[26] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *NIPS*, 2018.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[30] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.

[31] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[32] Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *NIPS*, 2016.

[33] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *ECCV*, 2018.

[34] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[35] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.