

Received January 9, 2021, accepted January 29, 2021, date of publication February 2, 2021, date of current version February 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056572

# Up and Down Residual Blocks for Convolutional Generative Adversarial Networks

YUEYUE WANG<sup>1</sup>, XINCHANG GUO<sup>2,3</sup>, PENG LIU<sup>1</sup> <sup>1</sup>, (Member, IEEE), AND BIN WEI<sup>4,5</sup>

<sup>1</sup>Computing Center, Ocean University of China, Qingdao 266100, China

<sup>2</sup>School of Marxism, Ocean University of China, Qingdao 266100, China

<sup>3</sup>The Institution for Maritime Development Studies, Ocean University of China, Qingdao 266100, China

<sup>4</sup>The Affiliated Hospital of Qingdao University, Qingdao 266000, China

<sup>5</sup>Shandong Key Laboratory of Digital Medicine and Computer Assisted Surgery, Qingdao 266000, China

Corresponding author: Peng Liu (liupengouc@ouc.edu.cn)

This work was supported by the MOE (Ministry of Education in China) Project of Humanities and Social Sciences under Grant 18YJCZH103.


**ABSTRACT** Most recent existing image generation methods have made great progress in creating high-quality images, mainly focusing on improving the generator or discriminator of convolutional generative adversarial networks (GANs). In this paper, we propose up and down residual blocks for convolutional GANs, dubbed upResBlock and downResBlock respectively. This structure is based on deconvolutions, strided convolutions, and residual blocks. With the upResBlock module for the generator of convolutional GANs, our method can further enhance the generative power of the feature extraction while synthesizing image details for the specified size. With the downResBlock module for discriminator combined with upResBlock for generator, the proposed method can speed up the back propagation of gradient and doesn't suffer from the vanishing or exploding gradients problems, generating more realistic images as well. Extensive experiments demonstrate that the proposed up and down residual blocks can help convolutional GANs in generating photo-realistic images. In addition, our method shows its universality for the improvement of existing methods.

**INDEX TERMS** Generative adversarial network, convolutional neural network, residual block, sampling.

## I. INTRODUCTION

The research on image generation is growing rapidly in the past few years. Generative Adversarial Nets (GANs) [1], an architecture of generative model, has been demonstrated the great potential in the image synthesis tasks. Although GAN is a very effective method to generate images for many application tasks, its training process is not stable enough and suffers from mode collapse problem. Besides, some tricks are required to get a better performance during producing vivid image.

Arjovsky *et al.* use Wasserstein distance to measure the similarity between true data distribution and the learned one in WGAN [2] rather than Jensen-Shannon divergence applied in origin GANs [1]. Although it avoids mode collapse, it takes a longer time for the model to converge than previous GANs. On this foundation, then WGAN-GP [3] proposes to use gradient penalty instead of weight clipping. It generally produces good images and greatly avoids mode collapse and it is easy

The associate editor coordinating the review of this manuscript and approving it for publication was Hualong Yu .

to apply this training framework to other GAN models. It is worth noting that the generator architecture of methods mentioned above can promote diversity of synthetic samples using transposed convolutions or deconvolutional networks [4].

When GAN has been applied to some applications such as text-to-image synthesis [5]–[10] and image-to-image translation tasks [11]–[14], there is an obvious observation that the quality of images can be improved by deepening the network in generator, but it may lead to the problem of gradients vanishing or exploding. In order to fix this case, residual learning framework [15] was proposed under the comprehensive empirical evidence which shows that these residual networks were easier to be optimized and could gain accuracy from considerably increased depth. Zhang *et al.* [9] designed the generator as an encoder-decoder network with residual blocks. Such a generator is able to help rectify defects in the input image and adds more details to generate images with rich details.

With the remarkable effect of ResNet [15], multi-level deconvolutional networks are used as crucial upsampling parts of architecture in recently proposed text-to-image

synthesis methods [8]–[10]. However, these methods neglect to make full use of the information between layers for upscaling. Through the analysis of upsampling process, we design the upResBlock which combines the advantages of residual learning method with deconvolutional networks, producing vivid images as well as avoiding training problems at the same time. The shortcut connections which are between convolution layers are able to contain all the necessary information. In addition, it would speed up back propagation and prevent gradients vanishing.

Upsampling and downsampling are two basic and widely used image processing operations. The upBlocks within deconvolutional networks [8] is a method to generate higher resolution images. Its layers attempt to directly minimize the reconstruction error of the input image under a sparsity constraint on an over-complete set of feature maps. Besides, the upBlocks can sample images to a higher resolution and enlarge the image contour but lead to texture blur and reduce the image quality. In contrast to upBlocks, downBlocks are used for the reduction in spatial resolution, keeping the same image representation, while they cannot ensure the effective judgment of detailed features in discriminator of convolutional GANs. Therefore, the generalization performance of models might not be outstanding when upsampling and downsampling are used as two separate standalone operations.

In the original GAN [1], the discriminator does not need to consider the variety of synthetic samples, which makes it possible for the generator to spend efforts in generating only a few kinds of samples. While we propose ways to improve upsampling performance of generator, we also consider downsampling process in discriminator to solve according to similar ideas. Convolutional layers combined with ResNets could add some hierarchical details during training process in order to generate detailed images with high quality.

Most recent existing algorithms mentioned above have made great progress in image generation and synthesis. Based on those researches, we propose up and down residual blocks dubbed upResBlock and downResBlock for convolutional GANs in this paper. Specifically, we explore to combine deconvolutions, strided convolutions and ResBlock in the network structure. Besides, we quantitatively evaluate the samples generated by our model using inception score, compared with previous state-of-the-art convolutional GAN models. Beyond the quantitative evaluation, extensive experimental results also qualitatively demonstrate the effectiveness of the proposed method, showing better visual effects of generated images.

In summary, the contribution of our method is threefold. 1) The up and down residual blocks (upResBlocks and downResBlocks) for convolutional GANs are proposed for synthesizing images with high quality, which could generate more detailed images and do not suffer from the vanishing or exploding gradients problems; 2) Comprehensive experiments are carried out to evaluate the proposed method and its universality, where we explore architectures using

the proposed upResBlocks and downResBlocks to show the improvements of generator and discriminator for GAN models; 3) Extensive experiments demonstrate that the proposed upResBlock and downResBlock can be widely used for convolutional GANs with improvements of image generation and be able to enhance visualization effect for text-to-image synthesis. Therefore, we make conclusions that our method achieves better performance in image generation and synthesis task in this paper.

## II. RELATED WORK

As a framework of generative model, Generative Adversarial Net (GAN) [1] has been applied to various applications and achieves impressive performance. Research of analyzing GAN in generating better images is developing constantly [2], [3], [16]–[22]. Besides, recent works have already demonstrated the great potential of using GAN in text-to-image synthesis [5]–[10], [23], [24] and image-to-image translation tasks [11]–[14], [25]–[30]. In this section, we will mainly concentrate on the architecture of convolution nets based on GANs.

The original GAN takes fully connected layer as its generating block, while DCGAN [17] uses convolutional neural networks for better performance. Since then convolution and transposed convolution layers have been the core components in many GAN models. Transposed convolutions, or deconvolutions, work by swapping the forward and backward passes of a convolution. They can be considered as the operation that allows to recover the shape of the initial feature map and play a significant role in upsampling the output image to the higher resolution. In the context of CNNs, Zeiler *et al.* [31] showed that by using deconvolutions and filtering the maximal activation, one can find the approximate purpose of each convolutional filter in the network. Radford *et al.* proposed the DCGAN [17] architecture with a series of fractionally-strided convolutions or deconvolutional networks for Large-scale Scene Understanding (LSUN) [32] scene modeling, making generative adversarial networks more stable to train in most settings. As the first method of using GAN to generate images from text captions, GAN-INT-CLS [5] follows the same architecture of DCGAN [17], telling models not only how to generate realistic images, but also the correspondence between texts and images. In our work, similar framework of deconvolutional networks in generator with a highly diverse set of filters is able to produce sharper images.

In the previous traditional studies, features were extracted through convolutional neural network and they would be richer through deepening of the network, which leads to the problem of exploding gradient. In order to deal with it, residual learning framework [15] was proposed under the comprehensive empirical evidence which shows that these residual networks were easier to be optimized and could gain accuracy from considerably increased depth. Zhang *et al.* designed the generator as an encoder-decoder network with residual blocks. Such a generator is able to help rectify defects in the input image and adds more details to generate images

with rich details. Besides, they evaluate the effectiveness of StackGAN-v2 [9] for the unconditional image generation task by comparing with DCGAN [17], WGAN [2], LSGAN [22] and WGAN-GP [3] on the LSUN bedroom dataset, showing that generated images with more photo-realistic details. The StackGAN-v2 was able to generate  $256 \times 256$  sharper images partly due to generator improvement in attention models, adding residual blocks in two stage attentional generative network. Furthermore, Xu *et al.* proposed AttnGAN [10] with generators in its attentional generative network which has residual blocks in the hidden states to generate high quality images through a multi-stage process.

ResNet [15] or deconvolutional network is used as a part of architecture in recently proposed text-to-image synthesis methods [8]–[10]. However, these methods neglect to make full use of the information between layers for upscaling. Huang *et al.* proposed DenseNet, which allows connections between two layers within the same dense block [33]. With the local dense connections, each layer reads information from all the preceding layers within the same dense block. Inspired by this method, we design upResBlock in generator, which contains local residual learning and fractionally-strided convolutions. This structure can avoid losing hierarchical details during upsampling process.

While paying attention to performance improvement of generator, we also notice the discriminator in original GAN [1]. By effective combination with shortcut in ResNet [15] and convolution layers, downResBlock is introduced to add useful hierarchical features from origin images. To deal with the deficiency in the method about GANs in several researchs, we propose two universal structures named upResBlock and downResBlock. And our experiments show that these two blocks have been improved in popular GANs and inception scores of text-to-image tasks.

### III. METHOD

As is known to all, Generative Adversarial Networks (GANs) are powerful to generate high-quality images [1]–[3], [17], [22], [34]. Obviously, the structure of generator and discriminator has played an important role in the task of generating images. Generally, in previous works, generator and discriminator are composed of several deconvolutional layers and strided convolutional layers. In order to resolve this problem, a lot of attempts have been made. Inspired by ResBlock, we propose upResBlock and downResBlock by combining deconvolution, strided convolution and ResBlock.

#### A. REVIEW ON UPBLOCK

Starting from [4], the deconvolution operation has been widely used for visualization of neural networks [31], semantic segmentation [35], and generative models [1]–[3], [17], [22], [34]. In particular, generative adversarial networks proposed by Goodfellow [1] is a brilliant method to generate various images. Furthermore, Reed *et al.* [5] applied it to the text to image task [8]–[10].

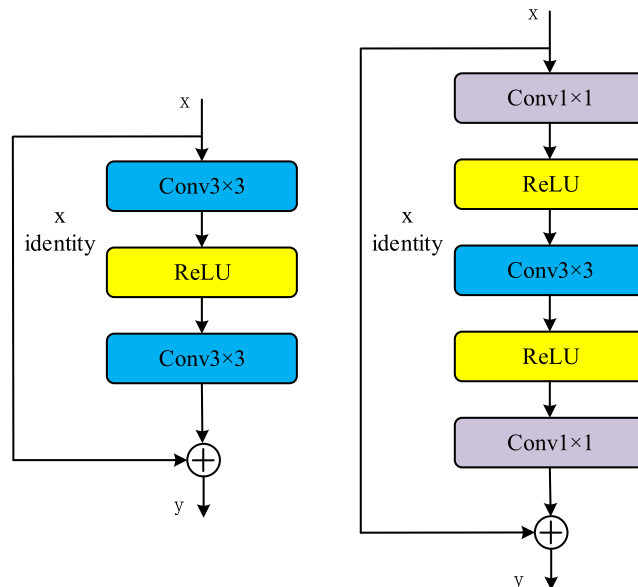


FIGURE 1. Residual block [15].

For instance, in [9], three generators made extensive use of upBlocks (deconvolutional layers) to generate bigger images. Similar to classification network, they also use lots of strided convolutional layers in discriminators. However, simply stacking deconvolutional layers and strided convolutional layers couldn't generate high-quality images. To cope with this problem, we propose a novel block by combining deconvolution, strided convolution and ResBlock.

#### B. REVIEW ON RESIDUAL BLOCK

Residual block [15] was proposed to resolve the problems of vanishing gradient and degradation. The biggest difference between residual block and upBlock is "shortcut connection" which skips one or more layers as illustrated in Figure 1. Generally, the residual learning could be formulated as:

$$y = \mathcal{F}(x, W_i) + x \tag{1}$$

where  $x$  and  $y$  are the input and output of the layer considered. And  $\mathcal{F}(x, W_i)$  represents the residual mapping. Through residual block, networks are easier to optimize and could gain higher performance in a variety of tasks. But, the ResBlock couldn't change the size of feature map. Therefore, we combine it with upBlock and propose a novel block dubbed upResBlock.

#### C. THE PROPOSED UP AND DOWN RESIDUAL BLOCKS

**Motivation:** Though upBlock has been applied extensively in many tasks such as generative adversarial networks, text to image and object detection, there is a margin for improvement because of the simple structure which couldn't master the detailed discriminative information of images. Meanwhile, the classification task has gained breakthrough with the help of ResBlock. Inspired by these two blocks, we propose two universal blocks for image generation, named up and down residual blocks.

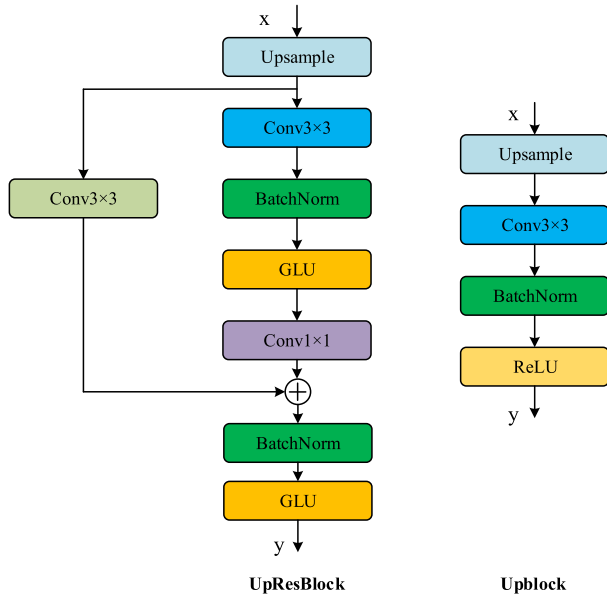


FIGURE 2. upResBlock and upBlock.

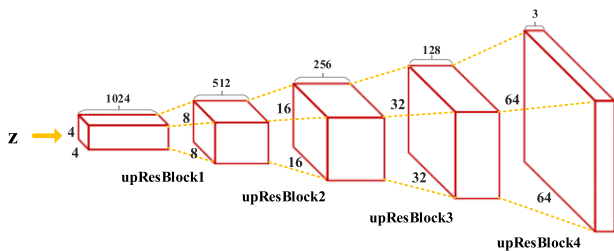


FIGURE 3. A typical structure of convolutional GANs' generator. We improve it with upResBlock.

**Definitions:** The purpose of up and down residual blocks is to make use of the advantages of upBlock and ResBlock, *i.e.*, generating images quickly from noise and more efficient backward propagation of gradients.

1. upResBlock

As illustrated in Figure 2, by using upBlock, data from low-dimensional input space can be mapped to high-dimensional feature space. And, the upBlock could be formulated as:

$$y = R(B(D(x))) \tag{2}$$

where y indicates the output of the layer considered; D refers to deconvolution; B means the batch normalization and R indicates the activation function ReLU. For instance, as shown in Figure 3, generator could generate images with size of 64 × 64 by simply stacking four upBlocks. However, simply stacking four upBlock couldn't make full use of the information in former layers.

On the contrary, our upResBlock could overcome these problems:

$$y = G(B(G(B(C(U(x))) + C(U(x)))))) \tag{3}$$

where B,D are same as Equation 2. G refers to activation function GLU; U indicates the upsample; C means the convolutional layers.

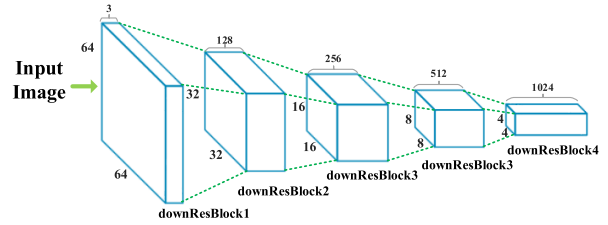


FIGURE 4. A typical structure of convolutional GANs' discriminator. We improve it with downResBlock.

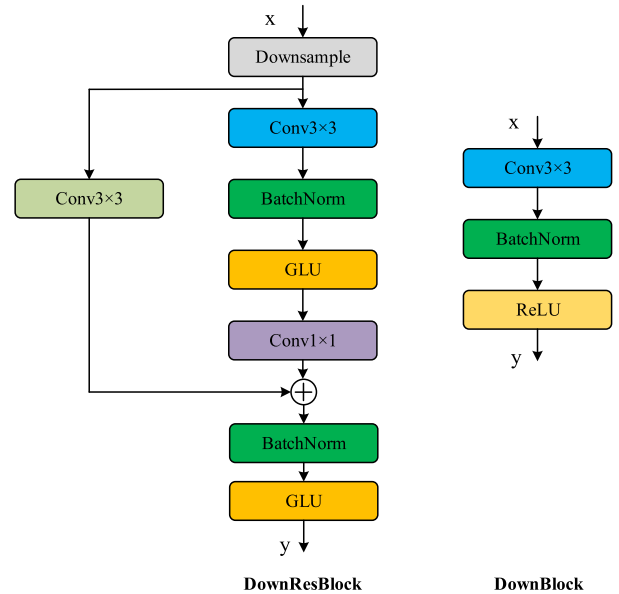


FIGURE 5. DownResBlock and downBlock.

As illustrated in Figure 2, our upResBlock could make all information pass through and make backward propagation of gradients more efficiently with the help of shortcut. In particular, the GLU activation [36] is used in our up and down residual blocks. And GLU could be computed as:

$$f(x) = (X * W + b) \otimes \sigma(X * V + c) \tag{4}$$

where X and f are input and output. And W, b, V, c are learned parameters. σ refers to sigmoid function and ⊗ indicates the element-wise product between matrices. The reason we adopt GLU is that networks using GLU could converge to a lower perplexity which has been proved in [36].

2. downResBlock

As shown in Figure 4, in most generative adversarial networks, the discriminator always stacks a lot of strided convolutions to reduce the size of input images and extract features. However, such an approach may not preserve the full features, especially the usefull features, which increases the difficulty of the convergence of loss.

In contrast, our downResblock (see Figure 5) could solve this problem by combining it with residual block. The structure of this block can be formulated as:

$$y = G(B(G(B(C(S(x))) + C(S(x)))))) \tag{5}$$

where S indicates strided convolution which halves the size of input and C, B, G are same as Equation 2. Through such a structure, our downResBlock could preserve the feature from former layers which help the convergence of loss.

#### D. DISCUSSION

**Compared with upBlock** Although both of them could map the data from low-dimensional to high-dimensional feature space, upsample and strided deconvolution used in upBlock couldn't preserve the effective features from former layers. Different from upBlock, our upResBlock could make most of useful informations pass through and make backward propagation of gradients more efficiently by combining with ResBlock.

**Compared with ResBlock** ResBlock could preserve the features with the help of shortcut. But, it couldn't change the size of input. And downBlock, composed of strided convolution and activation function has been widely used in discriminator in most of GANs. Inspired by those two blocks, we propose a novel block named downResBlock which combines the downBlock and ResBlock to master the detailed discriminative information on images and accelerate the backward propagation of gradient.

#### IV. EXPERIMENTS

In this section, we experimentally compare two kinds of image-generating tasks to verify the effectiveness of our method. The former is generating images from noise using several typical GANs. And the latter is text-to-image task which means generating images according to natural language descriptions.

##### A. GANS FOR IMAGE GENERATION

We conduct extensive experiments to evaluate the proposed methods. The popular GANs [1]–[3], [22] on image generation are compared with the methods which are improved with our the proposed network model. We first evaluate the effectiveness of our method with previous popular GAN models for generating images. Then samples generated for these architectures are compared in the same iteration to show the visual efforts.

##### 1) DATASET AND EVALUATION PROTOCOL

In this section, we introduce the dataset to illustrate the effectiveness of the upResBlock and downResBlock.

**Datasets.** Referring to previous image generating methods [3], [22], we use the CIFAR-10 dataset [37], which are commonly used in computer vision and image generation, to prove the effectiveness of the upResBlock and downResBlock. The CIFAR-10 dataset consists of 50,000 training images with 32 pixel  $\times$  32 pixel and 10,000 test images in 10 classes.

**Evaluation.** As the quantitative evaluation, we use inception score [19] as [3] in order to measure the effect of image generation before and after the method improvement. We train models using four typical GANs:

**TABLE 1.** Inception score results of the popular GAN models without (blank) or with (✓) upResBlock/downResBlock for image generation on CIFAR10 dataset. The values in bold font indicate the best results.

Network	Up and Down Residual Block		Inception Score CIFAR 10
	upResBlock	downResBlock	
GAN			2.55 $\pm$ .18
	✓		2.69 $\pm$ .17
		✓	2.65 $\pm$ .14
	✓	✓	<b>2.73 <math>\pm</math> .18</b>
LSGAN			2.39 $\pm$ .15
	✓		2.66 $\pm$ .15
		✓	2.59 $\pm$ .15
	✓	✓	<b>2.94 <math>\pm</math> .23</b>
DCGAN			6.57 $\pm$ .06
	✓		6.72 $\pm$ .09
		✓	6.84 $\pm$ .09
	✓	✓	<b>7.50 <math>\pm</math> .10</b>
WGAN-GP			7.86 $\pm$ .07
	✓		8.31 $\pm$ .05
		✓	7.93 $\pm$ .07
	✓	✓	<b>8.40 <math>\pm</math> .06</b>

GAN [1], LSGAN [22], DCGAN [17] and WGAN-GP [3] on CIFAR-10 dataset. Each of these GANs and their improved ones are trained for 100000 iterations. Figure 6 compares the visual quality of samples generated by the four existing popular GANs and their improved models without or with upResBlock/downResBlock. Furthermore, we compute inception scores with generated images to evaluate our method.

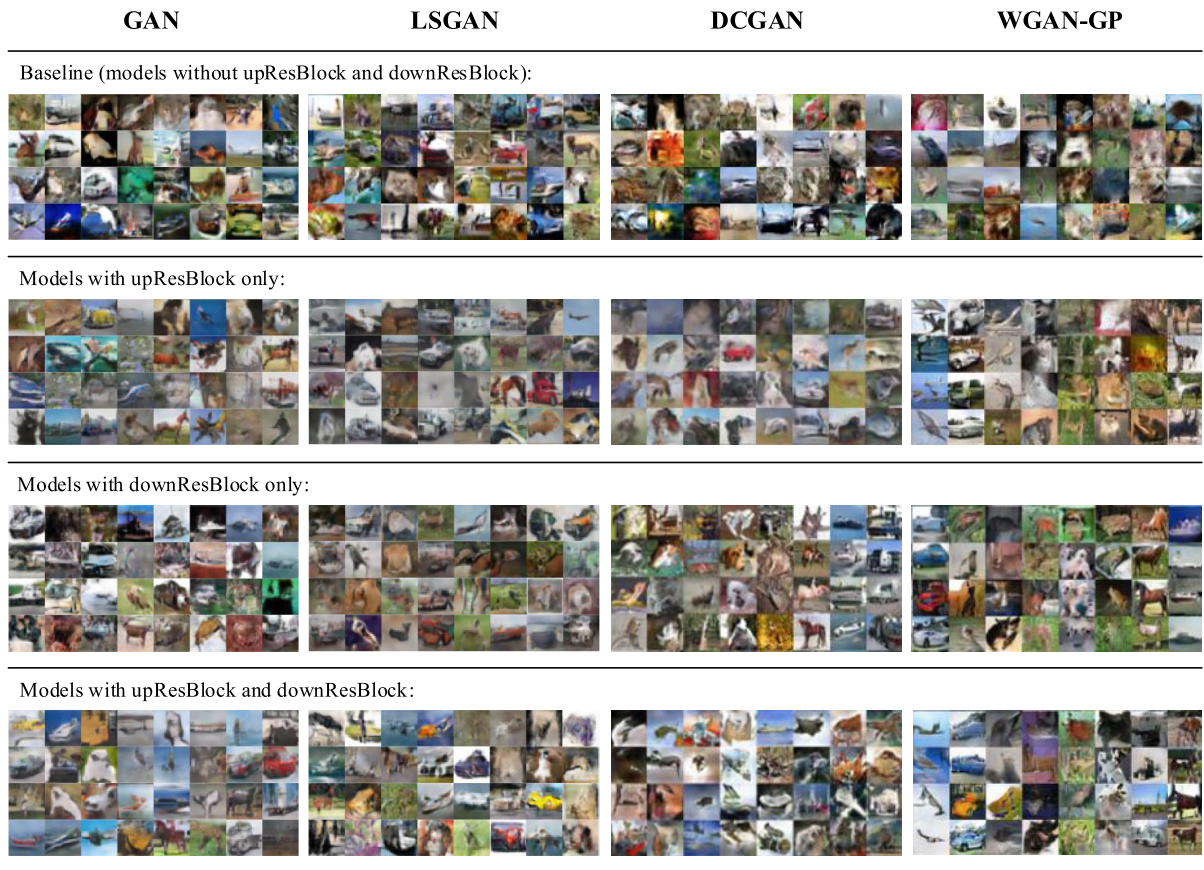
##### 2) EVALUATION OF THE UPRESBLOCK AND DOWNRESBLOCK

In order to demonstrate the effectiveness of our upResBlock and downResBlock, we train four typical GANs on CIFAR-10 dataset.

As shown in Figure 6, the comparison experiments of baseline model, baseline model with upResBlock only, baseline model with downResBlock only, and baseline model with upResBlock and downResBlock are performed. Specifically, we train each model for 100000 iterations for each experiment. It's not hard to see that models using our upResBlock and downResBlock could generate images with the most details. The same GAN architecture with our upResBlock or downResBlock shows better visual effect than the original one. In addition, the generated samples without upResBlock or downResBlock are not as photo-realistic as that of model with both them. Thus, from Figure 6, the different models are compared directly, using their generated images for qualitative measures, which shows our method's better performance in image details and diversity.

Furthermore, we compute inception score with generated samples to quantitatively evaluate our method. As shown in Table 1, we implement four groups of comparison experiments on various popular GANs by using up and down





**FIGURE 6.** Example of generated images reconstructed by various popular GANs without or with upResBlock/downResBlock on CIFAR-10 dataset.

residual blocks or not. Table 1 shows the inception score results on CIFAR-10 dataset. The upResBlock, or downResBlock could improve inception scores on those typical GANs. In particular, the combination of upResBlock and downResBlock effectively improves the inception score on CIFAR-10 dataset by mastering the detailed discriminative information of images. Therefore, these two blocks can be helpful for most GANs to generate images with more details and higher quality.

### B. GANS FOR TEXT-TO-IMAGE SYNTHESIS

We have already validate the effectiveness of our up and down residual blocks on those popular GANs in Section IV-A. To prove the generality of our blocks further, we have also done several experiments on text-to-image task.

#### 1) DATASET AND EVALUATION PROTOCOL

We implement a series of experiments by using two commonly used datasets to validate the effectiveness of our upResBlock and downResBlock in text-to-image task.

**Datasets.** First, we validate the effectiveness of our upResBlock and downResBlock on StackGAN-v2 and AttnGAN using CUB dataset [38]. CUB contains 200 bird species with

11,788 images and the data preprocess of this dataset is same as [10]. And, in order to validate the performance of our method in text-to-image generation for complex scenes, we also use the COCO dataset [39]. Each image in COCO dataset provide 5 descriptions and each image has multiple objects which is more difficult than CUB dataset. Same as [10], we also split COCO dataset into training and validation sets provided by COCO.

**Evaluation metrics** As before, we use inception score (IS) [19] for our quantitative evaluation. The inception score is calculated as follows:

$$IS = \exp(E_x D_{KL}(p(y|x)||p(y))) \quad (6)$$

where  $x$  indicates one generated sample,  $y$  refers to the label predicted by the pretrained inception model. Generally speaking, the inception score could reflect the diversity of images generated by our generative models.

#### 2) EVALUATION OF THE UPRESBLOCK AND DOWNRESBLOCK

We provide a detailed explanation related to the motivation and structure of our upResBlock and downResBlock in Section III-C. We implement several pairs of comparison experiments on two commonly used datasets, namely, CUB

**TABLE 2.** Inception score results of the AttnGAN and StackGAN-v2 models without (blank) or with (✓) upResBlock/downResBlock for text-to-image synthesis on CUB and COCO dataset. The values in bold font indicate the best results.

Network	Up and Down Residual Block		Inception Score	
	upResBlock	downResBlock	CUB	COCO
AttnGAN			4.36 ± .03	25.89 ± .47
	✓		4.49 ± .02	26.28 ± .36
		✓	4.42 ± .02	26.25 ± .34
	✓	✓	<b>4.68 ± .03</b>	<b>26.56 ± .43</b>
StackGAN-v2			4.04 ± .06	/
	✓		<b>4.33 ± .04</b>	/
		✓	4.08 ± .04	/
	✓	✓	4.05 ± .04	/

and COCO datasets, of text-to-image in a computer vision system to validate the generalization of the upResBlock and downResBlock.

As shown in Table 2, we implement three groups of comparison experiments on various types of AttnGAN by using different blocks.

Table 2 shows the compared image synthesis results of AttnGAN, AttnGAN using upResBlock, and AttnGAN using downResBlock, AttnGAN using both up and down Resblocks on CUB and COCO datasets. Using either two blocks has a limited role in the inception score. However, when they are combined, the improvements are remarkable, thereby indicating that two kinds of blocks are complementary. Furthermore, we also validate our upResBlock in StackGAN-v2, which demonstrates the generalization capability of our proposed blocks.

## V. CONCLUSION

In this study, we propose up and down residual blocks for convolutional generative adversarial networks. This method enables the network to learn the features of the target dataset accurately and comprehensively. Thus, it achieves good performance in image generation. Moreover, the quality of images will be improved if better network architecture is used. In future study, we will combine our method with the newly updated GANs for image-generation to generate images with higher quality in actual applications.

## ACKNOWLEDGMENT

(Y. Wang and X. Guo contributed equally to this work.)

## REFERENCES

- [1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2672–2680.
- [2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: <http://arxiv.org/abs/1701.07875>
- [3] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. NIPS*, 2017, pp. 5767–5777.
- [4] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.
- [5] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1060–1069.
- [6] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 217–225.
- [7] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3510–3520.
- [8] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5907–5915.
- [9] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," 2017, *arXiv:1710.10916*. [Online]. Available: <http://arxiv.org/abs/1710.10916>
- [10] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1316–1324.
- [11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.
- [12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.
- [13] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2868–2876.
- [14] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1857–1865.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [16] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: <http://arxiv.org/abs/1411.1784>
- [17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: <http://arxiv.org/abs/1511.06434>
- [18] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 1486–1494.
- [19] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2234–2242.
- [20] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2172–2180.

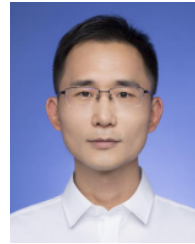
- [21] S. Nowozin, B. Cseke, and R. Tomioka, "F-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 271–279.
- [22] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2794–2802.
- [23] S. Ma, J. Fu, C. W. Chen, and T. Mei, "DA-GAN: Instance-level image translation by deep attention generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5657–5666.
- [24] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical Text-to-Image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7986–7994.
- [25] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain Image-to-Image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.
- [26] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. ECCV*, 2018, pp. 35–51.
- [27] M. Li, H. Huang, L. Ma, W. Liu, T. Zhang, and Y. Jiang, "Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks," in *Proc. ECCV*, 2018, pp. 184–199.
- [28] S. Benaim, T. Galanti, and L. Wolf, "Estimating the success of unsupervised image to image translation," in *Proc. ECCV*, 2018, pp. 218–233.
- [29] A. Gokaslan, V. Ramanujan, D. Ritchie, K. In Kim, and J. Tompkin, "Improving shape deformation in unsupervised image-to-image translation," in *Proc. ECCV*, 2018, pp. 649–665.
- [30] C. Wang, H. Zheng, Z. Yu, Z. Zheng, Z. Gu, and B. Zheng, "Discriminative region proposal adversarial networks for high-quality image-to-image translation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 770–785.
- [31] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*. Springer, 2014, pp. 818–833.
- [32] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," 2015, *arXiv:1506.03365*. [Online]. Available: <http://arxiv.org/abs/1506.03365>
- [33] G. Huang, Z. Liu, L. V. Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017, pp. 2261–2269.
- [34] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*. [Online]. Available: <http://arxiv.org/abs/1703.10717>
- [35] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
- [36] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 933–941.
- [37] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Handbook of Systemic Autoimmune Diseases*. 2009.
- [38] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 Dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
- [39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*. Springer, 2014, pp. 740–755.



**YUEYUE WANG** received the B.E. degree in electronic information engineering and the M.S. degree in signal and information processing from The Ocean University of China, in 2004 and 2009, respectively, where she is currently pursuing the Ph.D. degree in intelligent information and communication system.

From 2004 to 2006, she was a Teacher of electronic technology with the SHANDONG Electric Power College of China. In 2007, she joined the

Computing Center, Ocean University of China, where she is currently a Lecturer. Her research interest includes deep learning, computer vision, and marine biological image detection.



**XINCHANG GUO** received the M.S. degree from East China Normal University, Shanghai, China, in 2005. He is currently pursuing the Ph.D. degree in college of oceanic and atmospheric sciences with the Ocean University of China.

Since 2014, he was an Associate Professor with the School of Marxism, Ocean University of China, Qingdao. In 2005, he taught at the Ocean University of China. Meanwhile, he was a Researcher in the institution for maritime development studies with the Ocean University of China, the Humanities and Social Science Key Research Base of Ministry of Education, Qingdao. His research interests include deep learning, fuzzy evaluation model, and the game theory.



**PENG LIU** (Member, IEEE) received the B.S. degree in electronic information engineering, the M.S. degree in telecommunication and information system, and the Ph.D. degree in computer application technology from the Ocean University of China, Qingdao, China, in 2004, 2007, and 2020, respectively.

In 2007, he joined the Computing Center, Ocean University of China, where he is currently a Lecturer. His current research interests include image processing, computer vision, and deep learning.



**BIN WEI** is currently a Senior Engineer, with many academic titles including Deputy Director of the Shandong Provincial Key Laboratory of Digital Medicine and Computer-assisted Surgery in affiliated hospital of Qingdao university, Office Director of the Institute for Digital Medicine and Computer-assisted Surgery in Qingdao University, a Secretary of digital medicine branch of Shandong medical association, standing committee of Chinese medical equipment association

Cross and integration of medical equipment information branch. He has a strong working experience in hospital informatization, the clinical application of 3D reconstruction diagnosis and treatment technology, digital medicine assisted clinical diagnosis, virtual surgical simulation, 3D printing technology, and medical image processing.

• • •