Leveraging GANs via Non-local Features

Xuyang Peng[1], Weifeng Liu*[2], Baodi Liu[3], Kai Zhang[4], Xiaoping Lu[5], and Yicong Zhou[6]

[1] China University of Petroleum (East China), College of Oceanography and Space Informatics, China
pengxuyang19972@163.com
[2] China University of Petroleum (East China), College of Control Science and Engineering, China
liuwf@upc.edu.cn
[3] China University of Petroleum (East China), College of Control Science and Engineering, China
thu.liubaodi@gmail.com
[4] China University of Petroleum (East China), School of Petroleum Engineering, China
zhangkai@upc.edu.cn
[5] Haier Industrial Intelligence Institute Co., Ltd, China
luxiaoping@haier.com
[6] University of Macau, China
yicongzhou@um.edu.mo

Abstract. Recent years, Generative Adversarial Networks (GANs) have achieved tremendous success in image synthesis, which usually employ the convolutional operation to extract image features. However, most existing convolutional GANs only extract features in a local neighborhood at a time, which may often cause a lack of non-local information resulting in generating the wrong semantic object in the wrong position. In this paper, we propose a Graph Convolutional Architecture (GCA) for GANs to tackle this problem. GCA constructs a pixel-level graph structure between image regions through an attention mechanism and leverages Graph Convolutional Networks (GCNs) to extract non-local features. GCA extracts the connections between different regions of the image through GCNs, which is a more effective method of using relationship information than directly adding long-range dependencies to the model. We implement the GCA into Deep Convolutional Generative Adversarial Networks (DCGAN), Self-Attention Generative Adversarial Networks (SAGAN), and Concurrent-Single-Image-GAN (ConSinGAN). Extensive experiments are conducted to verify the performance of GCA. The results demonstrate that the GCA can significantly boost the quality of the generated image with more non-local features.

Keywords: Generative adversarial networks · Non-local features · Attention mechanism.

1 Introduction

Recent years, GANs attract much attention for their prodigious performance in image synthesis. And many GANs variants are reported in most of all aspects of the image generating such as single image super-resolution reconstruction [13][26][25], text-to-image synthesis [29][21][20], image-to-image translation [9][30], single image synthesis [22][8] and multi-class image synthesis [17][28], etc.. The early GANs models only design straight-forward discriminators and generators [5][19][3], which usually causes some problems such as unstable training and mode collapse. To improve the performance, many varieties of GANs are reported and can briefly divide into three categories, i.e. (1) Hierarchical Methods, (2) Iterative Methods, and (3) Loss Methods.

Hierarchical methods aim to modify the architecture of discriminators and generators with some specific modules to assist GANs for better image generating [16][17][18]. Wang et al. propose a Style and Structure Generative Adversarial Network (S2-GAN) by generating a surface normal map to encode the texture on the objects and the illumination with two GANs [24]. Karras et al. propose an Alternative Generator Architecture for Generative Adversarial Networks (Sytle-GAN) by employing a style transfer module to control the high-level attributes, such as hairstyles, freckles [11]. Odena et al. propose Auxiliary Classifier Generative Adversarial Networks (AC-GAN) which deploys an auxiliary classifier in the discriminator to exhibiting global coherence in GANs [18].

Iterative methods aim to design a skillful training process of GANs to drive generating photorealistic images [10][22][8]. Karras et al. propose a progressive growing method for GANs (ProgressiveGAN) by gradually increasing the layers of generator and discriminator to generate images from a low resolution to high resolution [10]. Shaham et al. propose a method for GANs in single image synthesis (SinGAN) by exploiting the pyramid structure to learn the whole image features from a single image [22]. Hinz et al. draw the pyramid structure of SinGAN and adopt parallel computing to reduce training time while improving the performance of the model [8].

Loss methods aim to apply suitable loss functions to stabilize the GANs training and improve generation performance [1][6][17]. Arjovsky et al. propose Wasserstein Generative Adversarial Networks (WGAN) by adopting the Wasserstein distance loss function instead of the Min-Max loss function to achieve a more stable training process [1]. Gulrajani et al. propose an improved method for WGAN by using a gradient penalty instead of a parameter clip [6]. Miyato et al. propose a regularization method for GANs (SN-GAN) by limiting the spectral norm of the parameters of the discriminator to constrain the Lipschitz constant [17].

Most GANs mentioned above are based on Convolutional Neural Networks (CNNs). However, traditional CNNs only capture the local spatial features in the receptive field and can't cover enough non-local information. The non-local information e.g. long-range dependencies can reflect the relationship between image regions and complement the neural network. Therefore, ignoring non-local information will often make the convolutional GANs generate the wrong seman-

tic objects in the wrong positions. To alleviate the lack of non-local information in the convolutional operation, Wang et al. propose a self-attention-mechanism-based module called Non-Local (NL) block to capture long-range dependencies in CNNs [23]. Han et al. introduce the NL block into GANs, proposing Self-Attention Generative Adversarial Networks (SAGAN) to alleviate the lack of non-local information in GANs [28]. SAGAN takes the long-range dependencies captured by the NL block as the weight and performs a weighted summation with the convolution feature maps to supplement the non-local information for the convolution GAN. Although SAGAN has supplemented convolutional GANs with long-range dependencies, it has great research potential on utilizing non-local information rather than simply adding long-range dependencies into models.

In this paper, we propose a Graph Convolutional Architecture (GCA) for GANs. GCA constructs a pixel-level graph structure between image regions by the self-attention mechanism and leverages GCNs to capture non-local features. Specifically, GCA employs an attention mechanism for pixel-level graph structure construction. Compared with the NL block directly adding long-distance dependencies to models, the non-local features extracted by GCNs in GCA further refine the non-local relationship information contained in long-distance dependencies. And the non-local features also have higher generalization, because GCNs are a kind of generalized form of CNNs. Equipped with GCA, the generator and the discriminator can successfully supplement non-local information for GANs to generate more realistic images. Furthermore, GCA can be easily applied to most convolutional GANs to improve the quality of the generated images. We show the flow chart of GCA in Figure 1.

To evaluate the generalization of GCA, we implement the GCA into DCGAN, SAGAN, and ConSinGAN. And we conduct extensive experiments on these models. In addition, we also compare the NL block with GCA. The comparative results demonstrate the superiority of GCA in both quantitative and qualitative analysis. Briefly, the contribution of this paper can be summarized as the following:

(1) A Graph Convolutional Architecture (GCA) is proposed to model non-local information to GANs.

(2) The GCA is implemented into three GANs and extensive experiments are conducted to show the superiority of the proposed GCA.

The rest of this paper is arranged as follows. Sec. 2 briefly introduces the related work. Sec. 3 describes the details of GCA including the construction of pixel-level global graph structure. Sec.4, reports the experimental results and provides some analysis. And finally, Sec. 5 concludes this paper.

2 Related Work

2.1 Generative Adversarial Networks

Ian et al. first propose Generative Adversarial Networks (GANs), which can only generate gray-scale images by two fully-connected networks [5]. Inspired by

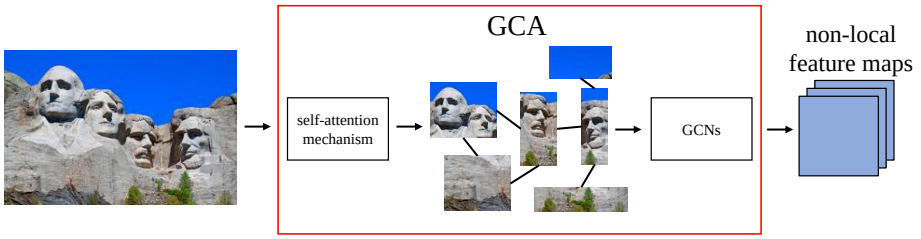

Fig. 1. GCA constructs a pixel-level graph structure among image regions through self-attention mechanism and exploits GCNs to extract non-local features.

Convolutional Neural Networks (CNNs), DCGAN introduces convolution into GANs and succeeds in unsupervised image synthesis [19]. The generator in DCGAN is constructed by transposed convolution, batch normalization, and ReLU activation, and the discriminator is constructed by convolution, batch normalization, and LeakyReLU activation. SAGAN introduces an NL block that models the long-range dependencies [28]. The NL block uses the weighted sum of all features to construct the relationship between image regions. SAGAN deploys the NL block in both the generator and discriminator, achieving great success in multi-classes image synthesis. ConSinGAN is currently the state-of-the-art GANs-based single image synthesis model [8]. ConSinGAN is an improvement of the Single Natural Image Generative Adversarial Network (SinGAN). Unlike SinGAN, ConSinGAN trains several stages in a sequential multi-stage manner, allowing the model to learn the whole features of a single image with fewer stages of increasing image resolution.

2.2 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) are networks that can extract information in a more general domain, especially structural information. The early GCNs are dedicated to generalizing CNNs to enable them to work on high-dimensional irregular domains (for example, social networks, brain connection groups, or reference networks) [2][16]. Bruna et al. propose two constructions, one based on a clustering of the domain, and the other based on the spectrum of the graph Laplacian [2]. Defferrard et al. propose a graph convolution method that is based on the spectrogram theory and design fast localized convolutional filters on the graph [16]. Kipf et al. propose a scalable method for semi-supervised learning of graph-structured data, which is based on the spectrum of the graph Laplacian, and the convolution kernel is approximated by shifted Chebyshev polynomials to reduce the algorithm complexity [12].

2.3 Attention Mechanism

The attention mechanism in the neural networks mainly models the relationship between neural elements based on their correlation, which is a key component of

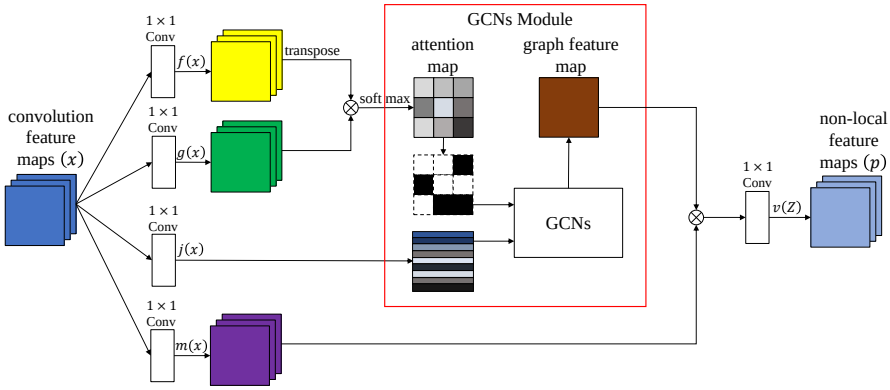

Fig. 2. The whole structure of GCA. The $\otimes$ denotes matrix multiplication.

various natural language processing and computer vision tasks. Attention mechanisms can process variable-sized inputs, focusing on the most relevant parts of the input to assist the model to make decisions. Attention mechanisms used to be adapted in many sequence-based tasks, such as machine reading [4] and learning sentence representations [14]. In image generation, the long-distance relationship modeling through the attention mechanism has proved effective for learning high-dimensional and complex image distribution. Wang et al. propose a self-attention-mechanism-based module for video processing called Non-Local (NL) block [23]. The NL block can capture long-range dependencies about image regions, and it can be inserted into many CNNs. In addition to being deployed in image processing, this non-local structure is also applicable for sequence and video problems. In addition to modeling the relationship between neural elements, the attention mechanism can also be used to construct graph structure in a graph domain. GCA employs a self-attention mechanism to construct a pixel-level global graph structure.

3 Graph Convolutional Architecture

The GCA employs an attention mechanism to construct a pixel-level graph structure and exploit GCNs to extract the non-local features. GCA is a complement to the convolutional GANs, alleviating the disadvantage of convolution that only captures local features. The whole structure of GCA is shown in Figure 2.

The convolutional feature maps $x \in \mathbb{R}^{C\times H\times W}$ obtained by the previous network are mapped into four feature spaces in GCA by f, g, j, m. Here, f, g, j, m are all 1×1 convolutions and $W_f \in \mathbb{R}^{\bar{c}\times c}, W_g \in \mathbb{R}^{\bar{c}\times c}, W_j \in \mathbb{R}^{\bar{c}\times 1}, W_m \in \mathbb{R}^{c\times c}, \bar{c} = \frac{c}{k}, k = 1, 2, 4, 8$. Because $\bar{c}$ does not influence the essential characteristics of the attention maps, we choose $k = 8$ for memory efficiency. Among them, f and g are used to calculate the attention map,

$$a_{j,i} = \frac{\exp(r_{i,j})}{\sum_{i=1}^{N} \exp(r_{i,j})} \tag{1}$$

where $r_{i,j} = f(x_i)^T g(x_j), N = H \times W$, and $a_{j,i}$ indicates the attention value of the j^{th} region to the i^{th} region. The whole attention map $A \in \mathbb{R}^{N \times N}$ is assembled by $a_{j,i}$, containing the relations between all the regions in x. Based on these relations, A can directly consider as an adjacency matrix in a graph domain. Also, according to different GANs models and GCNs algorithms, some methods can be used to operate A to make GCA obtain better performance, e.g. binarization, which is marked as a dotted box in Figure 2.

$J \in \mathbb{R}^{\mathbb{N} \times \mathbb{N}}$ is the feature matrix in a graph domain according to A. A and J constitute the main input of GCNs, and the calculation formula of GCNs is

$$Y = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} J \Theta \tag{2}$$

where $\tilde{A} = A + I_N, \tilde{D}_{ii} = \sum_j \tilde{A}_{i,j}, J = W_j x$ and $\Theta \in \mathbb{R}^{N \times N}$ which is the parameter matrix. The output of GCNs $Y \in R^{N \times N}$ is the graph feature map.

To make the input and output of GCA have the same dimensions, GCA leverages convolution and matrix multiplication to adjust the size of the graph feature map, the formula is

$$Z = MY \tag{3}$$

where $M = W_j x$, and $Z \in \mathbb{R}^{C \times H \times W}$. The property that GCA doesn't change the dimensions making it a plug-and-play module. The final convolution v is deployed at the end of the model. This convolution allows GCA to map the features to non-local features, the formula is

$$p = W_v Z \tag{4}$$

where v is a 1×1 convolution and $W_v \in \mathbb{R}^{c \times c}$.

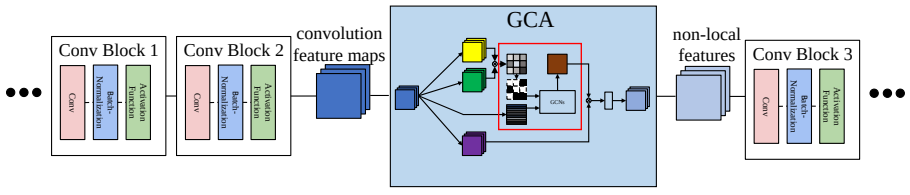

Fig. 3. The way GCA works in convolutional GANs. GCA can be easily embedded between two convolutional blocks.

To start training from easy to hard, we multiply the output of the GCNs model by a learnable scale parameter α and add back the input feature map. The α is initialized as 0. Therefore, the final output is given by

$$o = \alpha p + x \tag{5}$$

where x represents the previous convolutional feature maps.

The way GCA works in the convolutional GANs is shown in Figure 3.

4 Experiments

We implement the GCA into DCGAN, SAGAN, and ConSinGAN. Two datasets are adopted, including CelebA [15], and LSUN (church) [27]. Besides, we also conduct single image synthesis experiments. We deploy the NL block in the same position in DCGAN and SAGAN for comparison experiments. All models adopt the same hyperparameters, loss function, and training method.

The GCA and NL block deployed in DCGAN, SAGAN adopt 8×8 convolutional feature maps as input, and DCGAN and SAGAN are trained to generate 64×64 resolution images. The GCA deployed in ConSinGAN only works in the first stage of training. It should be noted that the GCA deployed in SAGAN replaces its original NL block instead of being equipped with an additional GCA module. All models are trained on NVIDIA Tesla V100 GPU. Quantitative and qualitative analyses are applied to the experimental results.

We quantitatively analyze the quality of the images generated by the above model. We chose the Fréchet Inception Distance (FID) [7] and Single Image Fréchet Inception Distance (SIFID) [22] to evaluate the generated images. FID compares the distribution of a pre-trained network's activations between a set of generated and real images. Especially, SIFID is an adaptation of the FID to the single image domain. The generated on CelebA and LSUN (church) synthesized by DCGAN, DCGAN + NL, and DCGAN + GCA are shown in Figure 4. The FID scores of all models are shown in Table 1 (The real images for calculating FID are the original images in datasets sampled to 64×64. All FID scores are calculated on 50,000 generated images). We show the heatmap of convolutional operation, NL block, and GCA in Figure 5. The heatmap shows that GCA captures more non-local information than the NL block.

Visually, when models are equipped with GCA, they can generate more realistic images. In quantitative analysis, the lower FID scores indicate that GCA can bring significant enhancement to GANs. However, the NL block doesn't improve the performance, causing a side effect instead. This contrast shows that the long-range dependencies captured by NL block are not generalized information in GANs, simply adding them to the model will even bring negative effects. In contrast, the non-local features modeled by GCA are extracted by GCNs. GCNs are a generalized form of CNNs, so non-local features are more generalized than long-range dependencies.

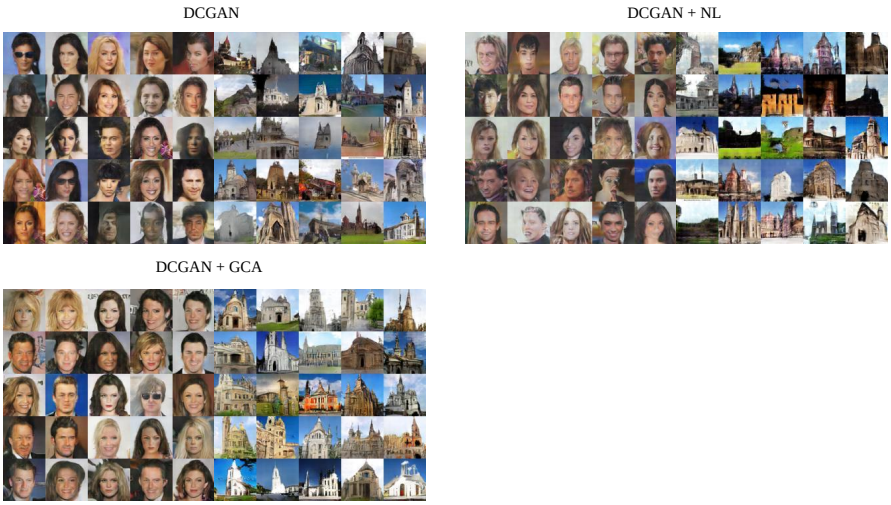

Fig. 4. Generated images of CelebA and LSUN (church) synthesized by DCGAN, DCGAN + NL, and DCGAN + GCA.

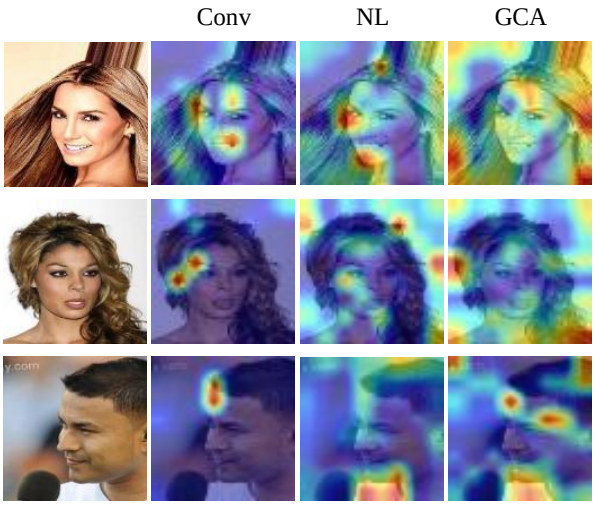

Fig. 5. Heatmaps of convolutional operation, NL block, and GCA. GCA and NL block can significantly increase the high activation regions. GCA and NL block enable CNNs to have high activation values for multiple regions at the same time instead of being limited to local regions. In addition, compared to NL block, GCA has more high activation regions. This means that GCA captures more non-local information.

Table 1. The FID scores of DCGAN, DCGAN + NL, DCGAN + GCA, SAGAN and SAGAN + GCA.

Model	Dataset	
	CelebA	LSUN (church)
DCGAN [19]	33.39	33.59
DCGAN + NL	34.56	54.36
DCGAN + GCA	**25.75**	**22.15**
SAGAN [28]	54.75	36.56
SAGAN + GCA	**37.39**	**28.74**

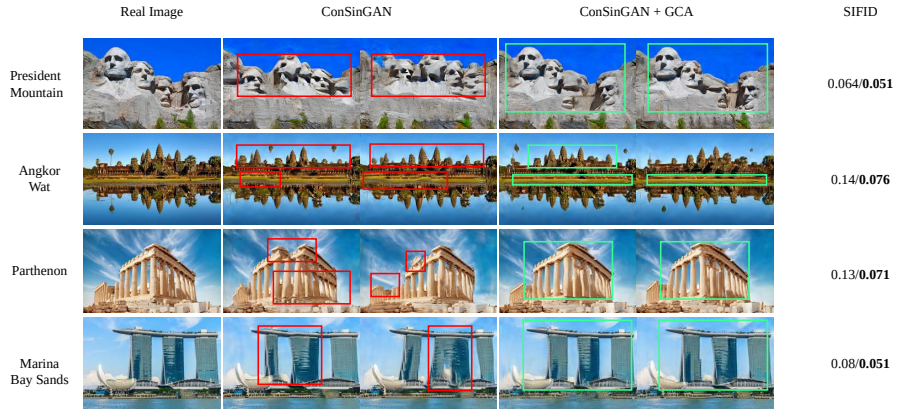

Fig. 6. Results of single image synthesis. The semantics that ConSinGAN fails to model are marked with red boxes and the improvements of GCA are marked with green boxes. The left side of the fourth column is the SIFID scores of ConSinGAN, and the right side is the SIFID scores of ConSinGAN deployed with GCA.

Single image synthesis can intuitively reflect the improvement of GCA to the GANs. However, SIFID itself has a large variance, qualitative analysis is more intuitive in single image synthesis. The results of the experiments are shown in Figure 6 (All SIFID scores are the average of the scores of 10 generated samples).

Intuitively, after GCA is equipped, the semantics of the generated images are significantly improved in space, structure, and texture. For example, in the President Mountain image, GCA can correctly model the positional relation between semantics. The generative model can correctly generate the position of each semantic object, avoiding the defect that the semantic objects are mixed. The improvements confirm that the pixel-level graph structure constructed by GCA can indeed successfully model the relationship information between the various features of the image. ConSinGAN uses a phased training method similar to ProgressiveGAN. In Figure 6, we show the results of different training

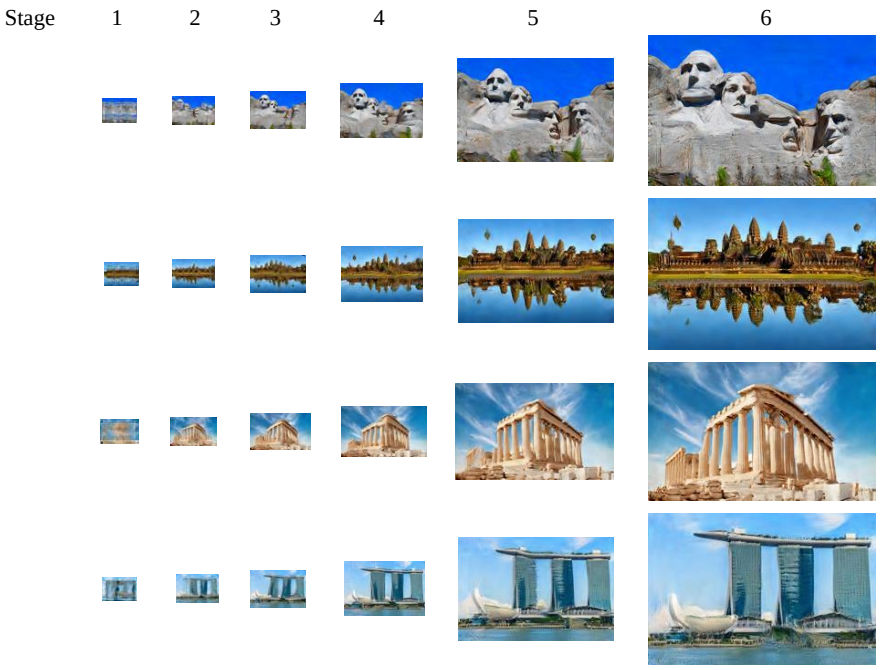

Fig. 7. The results of ConSinGAN + GCA in different training stages. The GCA helps ConSinGAN to model the image semantics early in the training.

phases of ConSinGAN after deploying GCA. The results of each stage of the ConSinGAN deployed with GCA are shown in Figure 7.

5 Conclusion

In this paper, we propose the Graph Convolutional Architecture (GCA) for GANs. The GCA employs the self-attention mechanism to construct a pixel-level graph structure and then incorporates the GCNs into the GANs. With the captured graph structure, GCA successfully supplements non-local feature extraction of GANs. Finally, we embed it into three representative GANs i.e. DCGAN, SAGAN, and ConSinGAN for evaluation. Experimental results verify the superiority of GCNs and show that GCA can significantly improve the performance of the convolutional GANs.

6 Acknowledgment

The paper was supported by the National Natural Science Foundation of China (Grant No.61671480), the Major Scientific and Technological Projects of CNPC

under Grant ZD2019-183-008,the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (Grant No.20200009) .

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223 (2017)
2. Bruna, J., Zaremba, W., Szlam, A., Lecun, Y.: Spectral networks and locally connected networks on graphs. In: International Conference on Learning Representations (2014)
3. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2180–2188 (2016)
4. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading. In: Conference on Empirical Methods in Natural Language Processing, pp. 551–561 (2016)
5. Goodfellow, I., Pougetabadie, J., Mirza, M., Xu, B., Warde-farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
6. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems, pp. 5769–5779 (2017)
7. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems, pp. 6626–6637 (2017)
8. Hinz, T., Fisher, M., Wang, O., Wermter, S.: Improved techniques for training single-image gans. In: IEEE Winter Conference on Applications of Computer Vision, pp. 1300–1309 (2021)
9. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5967–5976 (2017)
10. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (2018)
11. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4401–4410 (2019)
12. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
13. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A.: Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 105–114 (2017)
14. Lin, Z., Feng, M., Santos, C.N.D., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. In: International Conference on Learning Representations (2017)
15. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: IEEE International Conference on Computer Vision, pp. 3730–3738 (2015)

16. Michaël Defferrard, Xavier Bresson, P.V.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, pp. 3844–3852 (2016)
17. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)
18. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: International Conference on Machine Learning, pp. 2642–2651 (2017)
19. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (2016)
20. Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. In: Advances in Neural Information Processing Systems, pp. 217–225 (2016)
21. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: International Conference on Machine Learning, pp. 1060–1069 (2016)
22. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: IEEE International Conference on Computer Vision, pp. 4570–4580 (2019)
23. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)
24. Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks. In: European Conference on Computer Vision, pp. 318–335 (2016)
25. Wang, X., Yu, K., Dong, C., Loy, C.C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 606–615 (2018)
26. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Chen Change Loy, Y.Q.: Esrgan: Enhanced super-resolution generative adversarial networks. In: European Conference on Computer Vision, pp. 63–79 (2018)
27. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv: 1411.7766 (2014)
28. Zhang, H., Goodfellow, I., Metaxas, D.N., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning, pp. 7354–7363 (2019)
29. Zhang, H., Xu, T., Li, H.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: IEEE International Conference on Computer Vision, pp. 1060–1069 (2016)
30. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision, pp. 2242–2251 (2017)