Self-Attention Generative Adversarial Networks

Han Zhang* Rutgers University
Ian Goodfellow Google Brain
Dimitris Metaxas Rutgers University
Augustus Odena Google Brain

Abstract

In this paper, we propose the Self-Attention Generative Adversarial Network (SAGAN) which allows attention-driven, long-range dependency modeling for image generation tasks. Traditional convolutional GANs generate high-resolution details as a function of only spatially local points in lower-resolution feature maps. In SAGAN, details can be generated using cues from all feature locations. Moreover, the discriminator can check that highly detailed features in distant portions of the image are consistent with each other. Furthermore, recent work has shown that generator conditioning affects GAN performance. Leveraging this insight, we apply spectral normalization to the GAN generator and find that this improves training dynamics. The proposed SAGAN achieves the state-of-the-art results, boosting the best published Inception score from 36.8 to 52.52 and reducing Fréchet Inception distance from 27.62 to 18.65 on the challenging ImageNet dataset. Visualization of the attention layers shows that the generator leverages neighborhoods that correspond to object shapes rather than local regions of fixed shape.

1 Introduction

Image synthesis is an important problem in computer vision. There has been remarkable progress in this direction with the emergence of Generative Adversarial Networks (GANs) [5]. GANs based on deep convolutional networks [22, 10, 38] have been especially successful. However, by carefully examining the generated samples from these models, we can observe that convolutional GANs [19, 16, 17] have much more difficulty modeling some image classes than others when trained on multi-class datasets (*e.g.*, ImageNet [25]). For example, while the state-of-the-art ImageNet GAN model [17] excels at synthesizing image classes with few structural constraints (e.g. ocean, sky and landscape classes, which are distinguished more by texture than by geometry), it fails to capture geometric or structural patterns that occur consistently in some classes (for example, dogs are often drawn with realistic fur texture but without clearly defined separate feet). One possible explanation for this is that previous models rely heavily on convolution to model the dependencies across different image regions. Since the convolution operator has a local receptive field, long range dependencies can only be processed after passing through several convolutional layers. This could prevent learning about long-term dependencies for a variety of reasons: a small model may not be able to represent them, optimization algorithms may have trouble discovering parameter values that carefully coordinate multiple layers to capture these dependencies, and these parameterizations may be statistically brittle and prone to failure when applied to previously unseen inputs. Increasing the size of the convolution kernels can increase the representational capacity of the network but doing so also loses the computational and statistical efficiency obtained by using local convolutional structure. Self-attention [4, 20, 32], on the other hand, exhibits a better balance between ability to model long-range dependencies and computational and statistical efficiency. The self-attention module calculates response at a position as a weighted sum of the features at all positions, where the weights – or attention vectors – are calculated with only a small computational cost.

*Correspondence to han.zhang@cs.rutgers.edu

Preprint. Work in progress.

Figure 1: The proposed SAGAN generates images by leveraging complementary features in distant portions of the image rather than local regions of fixed shape to generate consistent objects/scenarios. In each row, the first image shows five representative query locations with color coded dots. The other five images are attention maps for those query locations, with corresponding color coded arrows summarizing the most-attended regions.

In this work, we propose Self-Attention Generative Adversarial Networks (SAGANs), which introduce a self-attention mechanism into convolutional GANs. The self-attention module is complementary to convolutions and helps with modeling long range, multi-level dependencies across image regions. Armed with self-attention, the generator can draw images in which fine details at every location are carefully coordinated with fine details in distant portions of the image. Moreover, the discriminator can also more accurately enforce complicated geometric constraints on the global image structure.

In addition to self-attention, we also incorporate recent insights relating network conditioning to GAN performance. [18] showed that well-conditioned generators tend to perform better. We propose enforcing good conditioning of GAN generators using the spectral normalization technique that has previously been applied only to the discriminator [16].

We have conducted extensive experiments on the ImageNet dataset to validate the effectiveness of the proposed self-attention mechanism and stabilization techniques. **SAGAN significantly outperforms the state of the art in image synthesis by boosting the best reported Inception score from 36.8 to 52.52 and reducing Fréchet Inception distance from 27.62 to 18.65.** Visualization of the attention layers shows that the generator leverages neighborhoods that correspond to object shapes rather than local regions of fixed shape.

2 Related Work

Generative Adversarial Networks. GANs have achieved great success in various image generation tasks, including image-to-image translation [9, 40, 29, 14], image super-resolution [12, 28] and text-to-image synthesis [24, 23, 37]. Despite this success, the training of GANs is known to be unstable and sensitive to the choices of hyper-parameters. Several works have attempted to stabilize the GAN training dynamics and improve the sample diversity by designing new network architectures [22, 37, 10], modifying the learning objectives and dynamics [1, 27, 15, 3, 39], adding regularization methods [7, 16] and introducing heuristic tricks [26, 19]. Recently, Miyato *et al*. [16] proposed limiting the spectral norm of the weight matrices in the discriminator in order to constrain the Lipschitz constant of the discriminator function. Combined with the projection-based discriminator [17], the spectrally normalized model greatly improves class-conditional image generation on ImageNet.

Attention Models. Recently, attention mechanisms have become an integral part of models that must capture global dependencies [2, 34, 36, 6]. In particular, self-attention [4, 20], also called intra-attention, calculates the response at a position in a sequence by attending to all positions within the same sequence. Vaswani *et al*. [32] demonstrated that machine translation models could achieve state-of-the-art results by solely using a self-attention model. Parmar *et al*. [21] proposed an Image Transformer model to add self-attention into an autoregressive model for image generation. Wang *et al*. [33] formalized self-attention as a non-local operation to model the spatial-temporal dependencies in video sequences. In spite of this progress, self-attention has not yet been explored in the context of GANs. (AttnGAN [35] uses attention over word embeddings within an *input* sequence, but

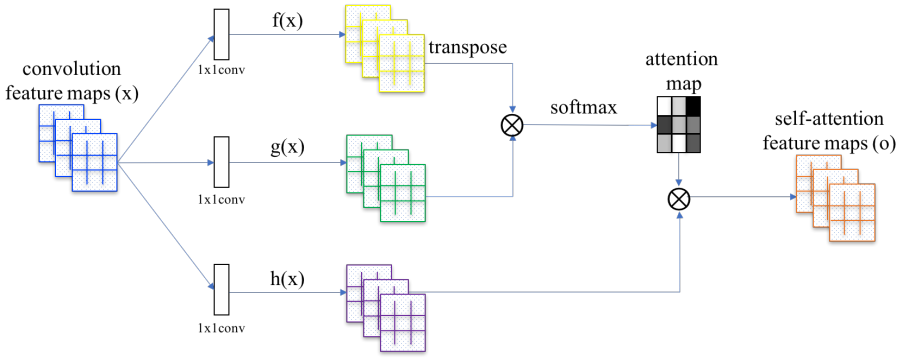


Figure 2: The proposed self-attention mechanism. The $\otimes$ denotes matrix multiplication. The softmax operation is performed on each row.

not self-attention over *internal model states*). SAGAN learns to efficiently find global, long-range dependencies within internal representations of images.

3 Self-Attention Generative Adversarial Networks

Most GAN-based models [22, 26, 10] for image generation are built using convolutional layers. Convolution processes the information in a local neighborhood, thus using convolutional layers alone is computationally inefficient for modeling long-range dependencies in images. In this section, we adapt the non-local model of [33] to introduce self-attention to the GAN framework, enabling both the generator and the discriminator to efficiently model relationships between widely separated spatial regions.

The image features from the previous hidden layer $\boldsymbol{x} \in \mathbb{R}^{C \times N}$ are first transformed into two feature spaces $\boldsymbol{f}, \boldsymbol{g}$ to calculate the attention, where $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W_f x},\ \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{W_g x}$

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp(s_{ij})}, \text{where } s_{ij} = \boldsymbol{f}(\boldsymbol{x_i})^T \boldsymbol{g}(\boldsymbol{x_j}), \tag{1}$$

and $\beta_{j,i}$ indicates the extent to which the model attends to the i^{th} location when synthesizing the j^{th} region. Then the output of the attention layer is $\boldsymbol{o} = (\boldsymbol{o_1}, \boldsymbol{o_2}, ..., \boldsymbol{o_j}, ..., \boldsymbol{o_N}) \in \mathbb{R}^{C \times N}$, where,

$$\boldsymbol{o_j} = \sum_{i=1}^{N} \beta_{j,i} \boldsymbol{h}(\boldsymbol{x_i}), \text{where } \boldsymbol{h}(\boldsymbol{x_i}) = \boldsymbol{W_h x_i}. \tag{2}$$

In the above formulation, $\boldsymbol{W_g} \in \mathbb{R}^{\bar{C} \times C}$, $\boldsymbol{W_f} \in \mathbb{R}^{\bar{C} \times C}$, $\boldsymbol{W_h} \in \mathbb{R}^{C \times C}$ are the learned weight matrices, which are implemented as 1×1 convolutions. We use $\bar{C} = C/8$ in all our experiments.

In addition, we further multiply the output of the attention layer by a scale parameter and add back the input feature map. Therefore, the final output is given by,

$$\boldsymbol{y_i} = \gamma \boldsymbol{o_i} + \boldsymbol{x_i}, \tag{3}$$

where γ is initialized as 0. This allows the network to first rely on the cues in the local neighborhood – since this is easier – and then gradually learn to assign more weight to the non-local evidence. The intuition for why we do this is straightforward: we want to learn the easy task first and then progressively increase the complexity of the task. In SAGAN, the proposed attention module has been applied to both generator and discriminator, which are trained in an alternating fashion by minimizing the hinge version of the adversarial loss [13, 30, 16],

$$\begin{aligned} L_D = &\ -\mathbb{E}_{(x,y)\sim p_{data}}[\min(0, -1 + D(x, y))]\ - \mathbb{E}_{z \sim p_z, y \sim p_{data}}[\min(0, -1 - D(G(z), y))], \\ L_G = &\ -\mathbb{E}_{z \sim p_z, y \sim p_{data}} D(G(z), y), \end{aligned} \tag{4}$$

4 Techniques to stabilize GAN training

We also investigate two techniques to stabilize the training of GANs on challenging datasets. First, we use spectral normalization [16] in the generator as well as in the discriminator. Second, we confirm that the two-timescale update rule (TTUR) [8] is effective, and we advocate using it specifically to address slow learning in regularized discriminators.

4.1 Spectral normalization for both generator and discriminator

Miyato *et al*. [16] originally proposed stabilizing the training of GANs by applying spectral normalization to the discriminator network. Doing so constrains the Lipschitz constant of the discriminator by restricting the spectral norm of each layer. Compared to other normalization techniques, spectral normalization does not require extra hyper-parameter tuning (setting the spectral norm of all weight layers to 1 consistently performs well in practice). Moreover, the computational cost is also relatively small.

We argue that the generator can also benefit from spectral normalization, based on recent evidence that the conditioning of the generator is an important causal factor in GAN performance [18]. Spectral normalization in the generator can prevent the escalation of parameter magnitudes and avoid unusual gradients. We find empirically that spectral normalization of both generator and discriminator makes it possible to use fewer discriminator updates per generator update, thus significantly reducing the computational cost of training. The approach also shows more stable training behavior.

4.2 Imbalanced learning rate for generator and discriminator updates

In previous work, regularization of the discriminator [16, 7] often slows down the GAN learning process. In practice, methods using regularized discriminators typically require multiple (*e.g*., 5) discriminator update steps per generator update step during training. Independently, Heusel *et al*. [8] have advocated using separate learning rates (TTUR) for the generator and the discriminator. We propose using TTUR specifically to compensate for the problem of slow learning in a regularized discriminator, making it possible to use fewer generator steps per discriminator step. Using this approach, we were able to produce better results given the same wall-clock time.

5 Experiments

To evaluate the proposed methods, we conducted extensive experiments on the LSVRC2012 (ImageNet) dataset [25]. First, in Section 5.1, we present experiments designed to evaluate the effectiveness of the two proposed techniques for stabilizing GAN training. Next, the proposed self-attention mechanism is investigated in Section 5.2. Finally, SAGAN is compared with state-of-the-art methods [19, 17] on image generation in Section 5.3.

Evaluation metrics. We choose the Inception score (IS) [26] and the Fréchet Inception distance (FID) [8] for quantitative evaluation. The Inception score [26] computes the KL divergence between the conditional class distribution and the marginal class distribution. Higher Inception score indicates better image quality. We include the Inception score because it is widely used and thus makes it possible to compare our results to previous work. However, it is important to understand that Inception score has serious limitations—it is intended primarily to ensure that the model generates samples that can be confidently recognized as belonging to a specific class, and that the model generates samples from many classes, not necessarily to assess realism of details or intra-class diversity. FID is a more principled and comprehensive metric, and has been shown to be more consistent with human evaluation in assessing the realism and variation of the generated samples [8]. FID calculates the Wasserstein-2 distance between the generated images and the real images in the feature space of an Inception-v3 network. Lower FID values mean closer distances between synthetic and real data distributions. In all our experiments, 50k samples are randomly generated for each model to compute the Inception score and FID.

Network structures and implementation details. All the SAGAN models we train are designed to generate 128×128 images. By default, spectral normalization [16] is used for the layers in both generator and discriminator. Similar to [17], SAGAN uses conditional batch normalization in the generator and projection in the discriminator. For all models, we use the Adam optimizer [11] with $\beta_1 = 0$ and $\beta_2 = 0.9$ for training. By default, the learning rate for the discriminator is 0.0004 and the learning rate for the generator is 0.0001.

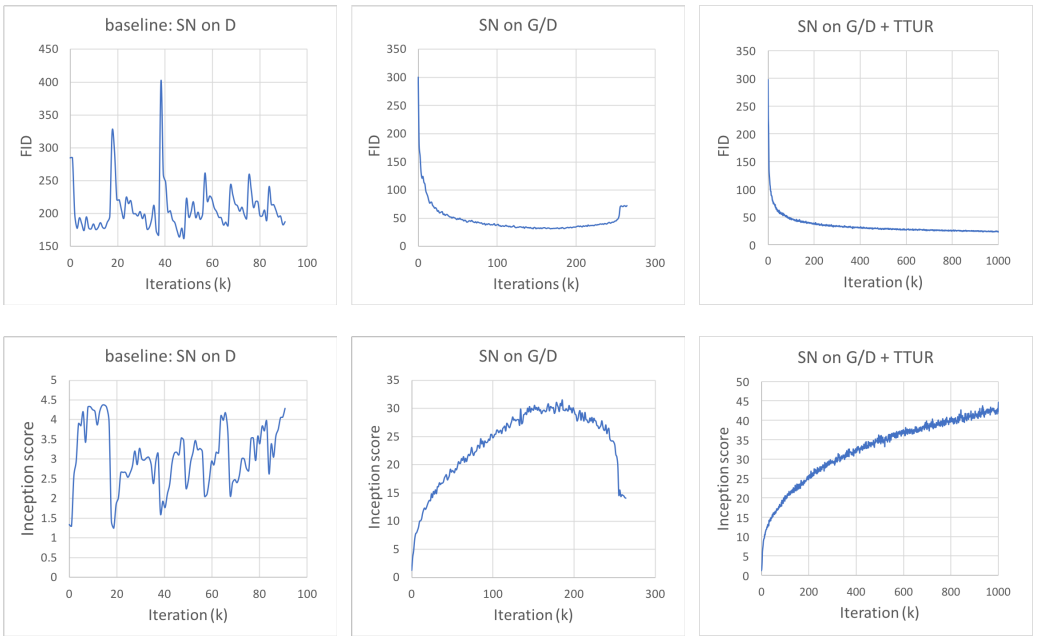


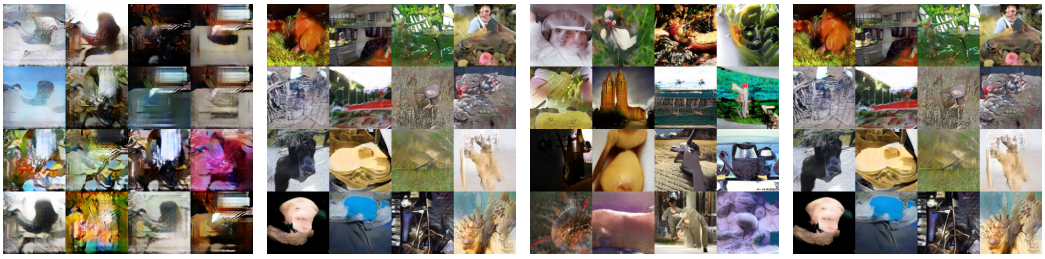
Figure 3: Training curves for the baseline model and our models with the proposed stabilization techniques, "SN on G/D" and two-timescale learning rates (TTUR). All models are trained with 1:1 balanced updates for G and D.

Model	no attention	SAGAN				Residual			
		$feat_8$	$feat_{16}$	$feat_{32}$	$feat_{64}$	$feat_8$	$feat_{16}$	$feat_{32}$	$feat_{64}$
FID	22.96	22.98	22.14	**18.28**	18.65	42.13	22.40	27.33	28.82
IS	42.87	43.15	45.94	51.43	**52.52**	23.17	44.49	38.50	38.96

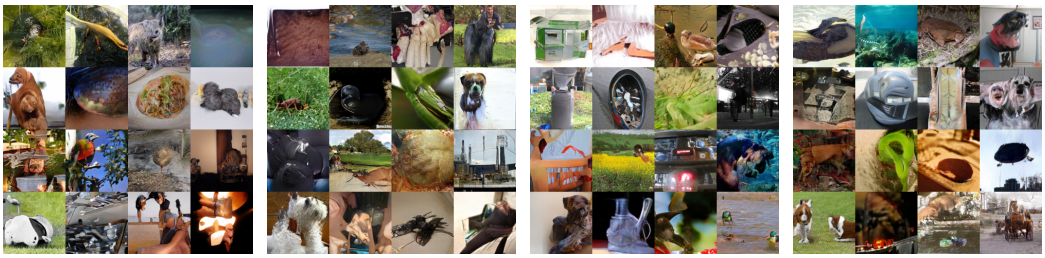
Table 1: Comparison of Self-Attention and Residual block on GANs. These blocks are added into different layers of the network. All models have been trained for one million iterations, and the best Inception scores (IS) and Fréchet Inception distance (FID) are reported.

5.1 Evaluating the proposed stabilization techniques.

In this section, experiments are conducted to evaluate the effectiveness of the proposed stabilization techniques, *i.e.*, applying spectral normalization (SN) to the generator and utilizing imbalanced learning rates (TTUR). In Figure 3, our models "SN on G/D" and "SN on G/D+TTUR" are compared with a baseline model, which is implemented based on the state-of-the-art image generation method [16]. In this baseline model, SN is only utilized in the discriminator. When we train it with 1:1 balanced updates for the discriminator (D) and the generator (G), the training becomes very unstable, as shown in the leftmost sub-figures of Figure 3. It exhibits mode collapse very early in training. For example, the top-left sub-figure of Figure 4 illustrates some images randomly generated by the baseline model at the 10k-th iteration. Although in the the original paper [16] this unstable training behavior is greatly mitigated by using 5:1 imbalanced updates for D and G, the ability to be stably trained with 1:1 balanced updates is desirable for improving the convergence speed of the model. Thus, using our proposed techniques means that the model can produce better results given the same wall-clock time. Given this, there is no need to search for a suitable update ratio for the generator and discriminator. As shown in the middle sub-figures of Figure 3, adding SN to both the generator and the discriminator greatly stabilized our model "SN on G/D", even when it was trained with 1:1 balanced updates. However, the quality of samples does not improve monotonically during training. For example, the image quality as measured by FID and IS is starting to drop at the 260k-th iteration. Example images randomly generated by this model at different iterations can be found in Figure 4. When we also apply the imbalanced learning rates to train the discriminator and the generator, the quality of images generated by our model "SN on G/D+TTUR" improves monotonically during the whole training process. As shown in Figure 3 and Figure 4, we do not



Baseline: SN on D (10k, FID=181.84) | SN on G/D (10k, FID=93.52) | SN on G/D (160k, FID=33.39) | SN on G/D (260k, FID=72.41)



SN on G/D+TTUR (10k, FID=99.04) | SN on G/D+TTUR (160k, FID=40.96) | SN on G/D+TTUR (260k, FID=34.62) | SN on G/D+TTUR (1M, FID=22.96)

Figure 4: 128×128 examples randomly generated by the baseline model and our models "SN on G/D" and "SN on G/D+TTUR".

observe any significant decrease in sample quality or in the FID or the Inception score during one million training iterations. Thus, both quantitative results and qualitative results demonstrate the effectiveness of the proposed stabilization techniques for GAN training. They also demonstrate that the effect of the two techniques is at least partly additive. In the rest of experiments, all models use spectral normalization for both the generator and discriminator and use the imbalanced learning rates to train the generator and the discriminator with 1:1 updates.

5.2 Self-attention mechanism.

To explore the effect of the proposed self-attention mechanism, we build several SAGAN models by adding the self-attention mechanism to different stages of the generator and discriminator. As shown in Table 1, the SAGAN models with the self-attention mechanism at the middle-to-high level feature maps (*e.g.*, $feat_{32}$ and $feat_{64}$) achieve better performance than the models with the self-attention mechanism at the low level feature maps (*e.g.*, $feat_8$ and $feat_{16}$). For example, the FID of the model "SAGAN, $feat_8$" is improved from 22.98 to 18.28 by "SAGAN, $feat_{32}$". The reason could be that the network receives more evidence with larger feature maps and enjoys more freedom to choose the conditions. The attention mechanism gives more power to both generator and discriminator to directly model the long-range dependencies in the feature maps. Thus, it is complementary to the convolutions, whose advantage lies in modeling local dependencies. In addition, the comparison of our SAGAN and the baseline model without attention (2nd column of Table 1) demonstrate the effectiveness of the proposed self-attention mechanism.

Compared with residual blocks with the same number of parameters, the self-attention blocks also achieve better results. For example, the training is not stable when we replace the self-attention block with the residual block in 8×8 feature maps, which leads to a significant decrease in performance (*e.g.*, FID increases from 22.98 to 42.13). Even for the cases when the training goes smoothly, replacing the self-attention block with the residual block still leads to worse results in terms of FID and Inception score. (*e.g.*, FID 18.28 vs 27.33 in feature map 32 × 32). This comparison demonstrates that the performance improvement given by using SAGAN is not simply due to an increase in model depth and capacity.

To better understand what has been learned during the generation process, we visualize the attention weights of the generator in SAGAN for different images. Some sample images with attention are shown in Figure 5 and Figure 1. See the caption of Figure 5 for descriptions of some of the properties of learned attention maps.

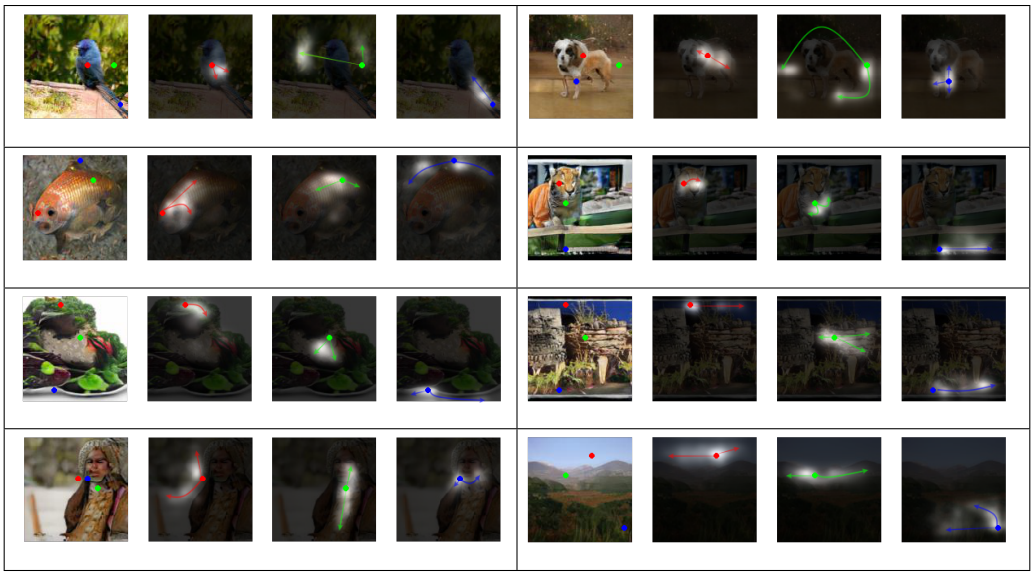


Figure 5: Visualization of attention maps. These images were generated by SAGAN. We visualize the attention maps of the last generator layer that used attention, since this layer is the closest to the output pixels and is the most straightforward to project into pixel space and interpret. In each cell, the first image shows three representative query locations with color coded dots. The other three images are attention maps for those query locations, with corresponding color coded arrows summarizing the most-attended regions. We observe that the network learns to allocate attention according to similarity of color and texture, rather than just spatial adjacency. For example, in the top-left cell, the red point attends mostly to the body of the bird around it, however, the green point learns to attend to other side of the image. In this way, the image has a consistent background (*i.e.*, trees from the left to the right though they are separated by the bird). Similarly, the blue point allocates the attention to the whole tail of the bird to make the generated part coherent. Those long-range dependencies could not be captured by convolutions with local receptive fields. We also find that although some query points are quite close in spatial location, their attention maps can be very different, as shown in the bottom-left cell. The red point attends mostly to the background regions, whereas the blue point, though adjacent to red point, puts most of the attention on the foreground object. This also reduces the chance for the local errors to propagate, since the adjacent position has the freedom to choose to attend to other distant locations. These observations further demonstrate that self-attention is complementary to convolutions for image generation in GANs. As shown in the top-right cell, SAGAN is able to draw dogs with clearly separated legs. The blue query point shows that attention helps to get the structure of the joint area correct.

Model	Inception Score	FID
AC-GAN [31]	28.5	/
SNGAN-projection [17]	36.8	27.62*
SAGAN	**52.52**	**18.65**

Table 2: Comparison of the proposed SAGAN with state-of-the-art GAN models [19, 17] for class conditional image generation on ImageNet. FID of SNGAN-projection is calculated from officially released weights.

5.3 Comparison with the state-of-the-art

SAGAN is also compared with state-of-the-art GAN models [19, 17] for class conditional image generation on ImageNet. As shown in Table 2, our proposed SAGAN achieves the best Inception score and FID. SAGAN significantly improves the best published Inception score from 36.8 to 52.52. The lower FID (18.65) achieved by SAGAN also indicates that SAGAN can better approximate the original image distribution by using the self-attention module to model the global dependencies between image regions. Figure 6 shows some sample images generated by SAGAN.

Figure 6: 128×128 example images generated by SAGAN for different classes. Each row shows samples from one class.

6 Conclusion

In this paper, we proposed Self-Attention Generative Adversarial Networks (SAGANs), which incorporate a self-attention mechanism into the GAN framework. The self-attention module is effective in modeling long-range dependencies. In addition, we show that spectral normalization applied to the generator stabilizes GAN training and that TTUR speeds up training of regularized discriminators. SAGAN achieves the state-of-the-art performance on class-conditional image generation on ImageNet.

Acknowledgments

We thank Surya Bhupatiraju for feedback on drafts of this article. We also thank David Berthelot and Tom B. Brown for help with implementation details. Finally, we thank Jakob Uszkoreit, Tao Xu, and Ashish Vaswani for helpful discussions.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv:1701.07875*, 2017.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014.

[3] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. In *ICLR*, 2017.

[4] J. Cheng, L. Dong, and M. Lapata. Long short-term memory-networks for machine reading. In *EMNLP*, 2016.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[6] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. DRAW: A recurrent neural network for image generation. In *ICML*, 2015.

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[8] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, pages 6629–6640, 2017.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[10] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[12] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[13] J. H. Lim and J. C. Ye. Geometric gan. *arXiv:1705.02894*, 2017.

[14] M. Liu and O. Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016.

[15] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017.

[16] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[17] T. Miyato and M. Koyama. cgans with projection discriminator. In *ICLR*, 2018.

[18] A. Odena, J. Buckman, C. Olsson, T. B. Brown, C. Olah, C. Raffel, and I. Goodfellow. Is generator conditioning causally related to gan performance? In *ICML*, 2018.

[19] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICLR*, 2017.

[20] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. In *EMNLP*, 2016.

[21] N. Parmar, A. Vaswani, J. Uszkoreit, Łukasz Kaiser, N. Shazeer, and A. Ku. Image transformer. *arXiv:1802.05751*, 2018.

[22] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[23] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *NIPS*, 2016.

[24] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text-to-image synthesis. In *ICML*, 2016.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[26] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016.

[27] T. Salimans, H. Zhang, A. Radford, and D. N. Metaxas. Improving gans using optimal transport. In *ICLR*, 2018.

[28] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. In *ICLR*, 2017.

[29] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *ICLR*, 2017.

[30] D. Tran, R. Ranganath, and D. M. Blei. Deep and hierarchical implicit models. *arXiv:1702.08896*, 2017.

[31] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *NIPS*, 2016.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv:1706.03762*, 2017.

[33] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018.

[34] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[35] 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 *CVPR*, 2018.

[36] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016.

[37] 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 *ICCV*, 2017.

[38] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv: 1710.10916*, 2017.

[39] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In *ICLR*, 2017.

[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.