

Conditional Convolution Projecting Latent Vectors on Condition-Specific Space

Min-Cheol Sagong[✉], Yoon-Jae Yeo[✉], Yong-Goo Shin[✉], *Member, IEEE*, and Sung-Jea Ko[✉], *Fellow, IEEE*

Abstract—Despite rapid advancements over the past several years, the conditional generative adversarial networks (cGANs) are still far from being perfect. Although one of the major concerns of the cGANs is how to provide the conditional information to the generator, there are not only no ways considered as the optimal solution but also a lack of related research. This brief presents a novel convolution layer, called the conditional convolution (cConv) layer, which incorporates the conditional information into the generator of the generative adversarial networks (GANs). Unlike the most general framework of the cGANs using the conditional batch normalization (cBN) that transforms the normalized feature maps after convolution, the proposed method directly produces conditional features by adjusting the convolutional kernels depending on the conditions. More specifically, in each cConv layer, the weights are conditioned in a simple but effective way through filter-wise scaling and channel-wise shifting operations. In contrast to the conventional methods, the proposed method with a single generator can effectively handle condition-specific characteristics. The experimental results on CIFAR, LSUN, and ImageNet datasets show that the generator with the proposed cConv layer achieves a higher quality of conditional image generation than that with the standard convolution layer.

Index Terms—Conditional image generation, deep learning, generative adversarial networks (GANs).

I. INTRODUCTION

Generative adversarial networks (GANs) [1] have brought about remarkable improvements to image generation algorithms. In general, GANs consist of a generator and a discriminator, which are trained with competing goals. The generator is trained to mimic the target data distribution, while the discriminator is optimized to differentiate between real and generated samples [2], [3]. As an extension of the GANs, various conditional GANs (cGANs) techniques have been proposed to generate class-conditional samples [2], [4], [5]. The intuitive way to process the conditioned activation without sharing parameters is to build the same number of generators as the condition; however, this approach requires not only considerable memory but also time for training. Early studies into cGANs typically provide the conditional information to both the generator and the discriminator by naively concatenating that information to the input image or some intermediate layers [4], [6], [7], or by adopting the conditional batch normalization (cBN) [8]. van den Oord *et al.* [9] show that class conditional synthesis significantly improves the quality of generated samples. Recently, Miyato and Koyama [10] proposed a novel cGANs framework, which significantly improves the visual quality as well as the diversity of the generated image by applying a projection

discriminator. These strategies have shown promising results and have been successfully applied to various image processing techniques including style transfer [8], [11], text-to-image synthesis [3], [5], [6], and image-to-image transformation [12], [13]. Nevertheless, the method providing the condition-specific information for the generator is still a lack of study.

In this brief, we propose a novel convolution layer, called a conditional convolution (cConv), which uses different weights depending on the given class. More specifically, in each cConv, the weights are conditioned in a simple but effective way through filter-wise scaling and channel-wise shifting operations. The filter-wise scaling adjusts each filter of cConv using specialized scaling parameters to a specific condition, whereas the channel-wise shifting modulates each channel of the filters with different shifting parameters. Unlike the conventional methods, the proposed method can process the condition-specific features in the convolution layer. We conducted extensive experiments on several datasets, including CIFAR [14], LSUN [15], and ImageNet [16], [17], to demonstrate the feasibility and effectiveness of the proposed method. We show that with the help of the proposed cConv, a single generator can produce conditional images with higher quality than the state-of-the-art conditioning methods. We also performed additional experiments to prove the generalizability of cConv. We further applied the cConv to the conditional style-transfer network and succeeded in adjusting images with various styles of art paintings with the single network.

In summary, this brief makes three major contributions. First, we propose a novel cConv layer, which produces different feature maps using the specialized weights according to the particular conditions. Armed with cConv, the generator achieves higher performance in generating conditional images than the conventional methods. Second, we introduce the sequential operations of filter-wise scaling and channel-wise shifting, which adjust the weights depending on the given class with a small number of parameters. Third, we extend the application of cConv to the conditional style-transfer and successfully produce visually pleasing results.

II. RELATED WORK

The cGANs are one of the groups of GANs, and they generate images with the given class [4]. Unlike with the standard GANs, both the generator and the discriminator of the cGANs require conditional information to generate class-conditional images and differentiate them from the target class. To this end, as shown in Fig. 1, various methods have been proposed to effectively provide the conditional information for both networks [4]–[8], [10], [11], [18]–[21]. Earlier methods have provided the conditional information by concatenating the class label (one-hot vector) into the input image or intermediate layers for the discriminator [4], [6], as depicted in Fig. 1(a) and (b). Reed *et al.* [5] improved upon these frameworks by building richer conditional information, such as a target bounding box. On the other hand, some researchers [7], [18], [19] have proposed a modified discriminator with an auxiliary classification network, which is trained to predict the class label of the input image as well as differentiate between the real and fake samples, as shown in Fig. 1(c). Recently, inspired by probabilistic models, Miyato and Koyama [10] proposed the projection discriminator. As depicted in Fig. 1(d), the projection

Manuscript received 12 May 2021; revised 15 November 2021 and 24 February 2022; accepted 28 April 2022. Date of publication 18 May 2022; date of current version 5 January 2024. This work was supported by the Institute of Information and Communications Technology Promotion (IITP), MSIT, Korean Government (development of global multitarget tracking and event prediction techniques based on real-time large-scale video analysis), under Grant 2014-3-00077. (Corresponding author: Sung-Jea Ko.)

Min-Cheol Sagong, Yoon-Jae Yeo, and Sung-Jea Ko are with the Department of Electrical Engineering, Korea University, Seoul 02841, South Korea (e-mail: mcsagong@dal.korea.ac.kr; yjyeo@dal.korea.ac.kr; sjko@korea.ac.kr).

Yong-Goo Shin was with the Division of Smart Interdisciplinary Engineering, Hannam University, Daejeon 34430, South Korea. He is now with the Department of Electronics and Information Engineering, Korea University, Sejong-si 30019, Republic of Korea (e-mail: ygshin92@korea.ac.kr).

Digital Object Identifier 10.1109/TNNLS.2022.3172512

2162-237X © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

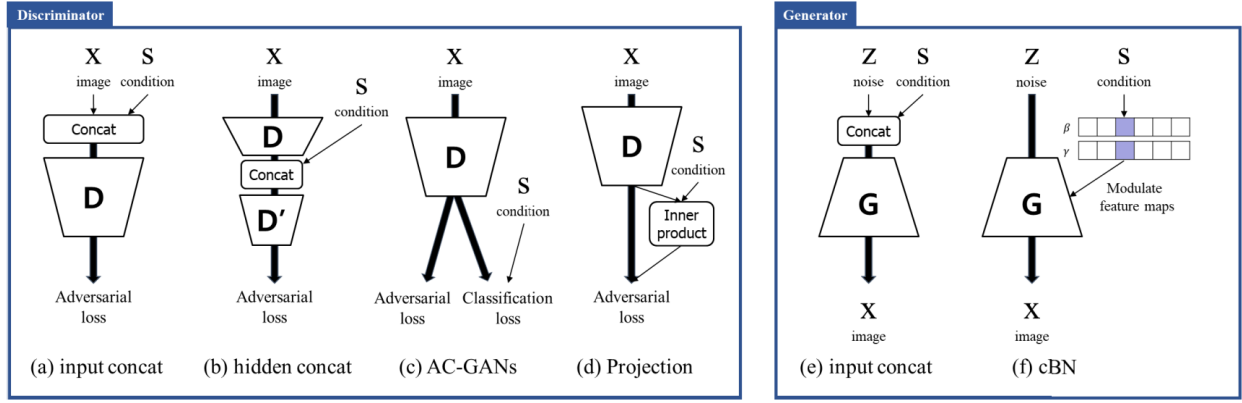


Fig. 1. Various models of the generator and discriminator for cGANs. (a) Discriminator concatenating the one-hot conditional vector to the input image. (b) Discriminator concatenating the one-hot conditional factor to the intermediate layer. (c) Discriminator with the auxiliary loss. (d) Projection discriminator [10]. (e) Generator concatenating the one-hot conditional vector to the input noise vector. (f) Generator with the conditional normalization layer.

discriminator takes an inner product between the embedded condition vector and the feature vector of the discriminator so as to impose a regularity condition. Along with significantly improving the visual quality, the projection discriminator also guarantees the diversity of the generated images.

Meanwhile, a few attempts have been made to process the conditional information for the generator. Typically, the earlier works [4], [6], [7] provided the conditional information to the generator by concatenating the class label into an input noise vector, as shown in Fig. 1(e). Some studies proposed conditional normalization layers, which supply intermediate layers with the conditional information [11], [22]. Despite the fact that these methods improve the performance of cGANs, research on conditioning methods is still lacking.

III. PRELIMINARIES

A. Conditional Generative Adversarial Network

The GANs have been improved upon since they were first introduced by Goodfellow *et al.* [1]. In the GANs, the generator G is trained to create fake images, which are indistinguishable from real ones, while the discriminator D is trained to classify between real and generated images. This competing training is defined as

$$\min_G \max_D E_{I \sim P_{\text{data}}(I)} [\log D(I)] + E_{z \sim P_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

where z and I represent a random noise vector and a real image sampled from noise distribution $P_z(z)$ and target distribution $P_{\text{data}}(I)$, respectively. More recently, cGANs, which focus on producing the class conditional images, have been actively researched [4], [7], [20], [23]. In general, cGANs techniques add the conditional information s , such as class labels or text condition, to both generator and discriminator to control the data generation process in a supervised manner. The cGANs can be formally expressed as follows:

$$\min_G \max_D E_{(I,s) \sim P_{\text{data}}(I)} [\log D(I, s)] + E_{z \sim P_z(z), s \sim P_{\text{data}}(I)} [\log(1 - D(G(z, s)))] \quad (2)$$

Using the abovementioned equation, the cGANs can select an image category to be generated, which is not possible when using the standard GANs framework.

B. Conditional Normalization Layer

The conditional normalization layer including the adaptive instance normalization [11] and the cBN [8] is considered to be an important

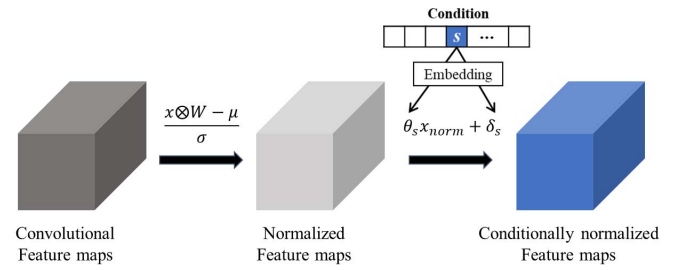


Fig. 2. Simplified diagram of the cBN. The convolutional feature maps obtained through convolution operation are first normalized with mean μ and standard deviation σ , and then, normalized feature maps are modulated using condition-specific scaling and shifting parameters.

component in the cGANs frameworks. In particular, the cBN is widely used to provide the conditional information to the generator [10], [20]. Unlike the standard normalization layers, as shown in Fig. 2, the cBN requires external conditional information, which determines the condition-specific scaling and shifting parameters. The cBN operates as follows: first, the convolutional feature maps obtained through the convolution operation are normalized with mean μ and standard deviation σ in the same way as the batch normalization (BN) process. Then, the normalized feature maps are scaled and shifted using a learned affine transformation whose parameters are specialized to the conditional information. Formally, the cBN with standard convolution layer can be expressed as follows:

$$f_{\text{cBN}}(x, s) = \theta_s \left(\frac{x \otimes W - \mu}{\sigma} \right) + \delta_s \quad (3)$$

where x , W , and $\otimes$ indicate the input feature maps, convolution weights, and convolution operation, respectively. In addition, θ_s and δ_s are the scaling and shifting parameters, respectively, and these have different values according to the class label. By allowing the conditional information to manipulate the normalized feature maps, the cBN can conduct various conditional image generation tasks [10], [21], [24], [25]. Miyato and Koyama [10] show that the generator can perform category morphism, which creates interpolated images between two classes, by applying the cBN.

IV. PROPOSED METHOD

A. cConv Layer

We propose a novel cConv layer, called cConv, which produces the conditioned feature maps by employing the separated weights

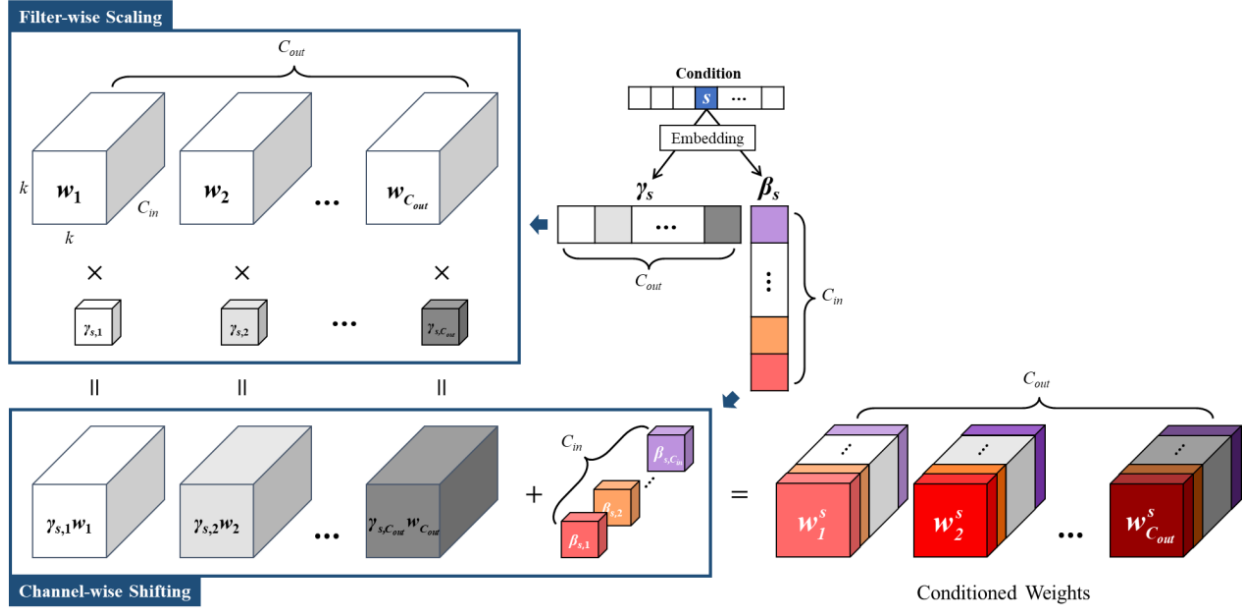


Fig. 3. Illustration of the proposed filter-wise scaling and the channel-wise shifting operations. By applying these operations, every filter and channel of the convolutional filters w are conditioned to a desired class with a small number of parameters.

depending on the given condition. In general, the weights of the convolution layer are considered as a 4-D tensor $W \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$, where k is the filter size, while C_{in} and C_{out} are the number of input and output channels, respectively. As shown in Fig. 3, each convolution layer has C_{out} filters with C_{in} channels. Each filter is denoted as w_i , $\{i = 1, \dots, C_{out}\}$. To produce conditioned weights $W^s \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$, we modulate the W through filter-wise scaling and channel-wise shifting operations as follows: as shown in Fig. 3, we initially embed each condition to specialized scaling parameters $\gamma_s \in \mathbb{R}^{C_{out}}$ and shifting parameters $\beta_s \in \mathbb{R}^{C_{in}}$. Using these parameters, we first conduct the filter-wise scaling operation on w_i with the scaling parameter $\gamma_{s,i}$ to individually provide the conditional information to each filter. Next, we apply the channel-wise shifting operation on the scaled w_i , which separately shifts each channel with shifting parameter $\beta_{s,j}$. This can be formally expressed as follows:

$$w_{i,j}^s = \gamma_{s,i} \cdot w_{i,j} + \beta_{s,j}, \quad \{i = 1, \dots, C_{out}, j = 1, \dots, C_{in}\} \quad (4)$$

where $w_{i,j}$ and $w_{i,j}^s$ indicate the j th channel of the i th filter in W and W^s , respectively. As shown in Fig. 3, with these sequential operations, all of the filters and channels of weights are specialized to a desired class with a small number of parameters. We initialize γ_s and β_s to matrices of ones and zeros, respectively. Since β_s is cloned C_{out} times, as shown in Fig. 3, we divide β_s by C_{out} to normalize the gradients of it. By utilizing W^s , the cConv can directly generate the conditioned feature maps without the need for conditional normalization process.

B. Understanding and Analysis

To elucidate how W^s can effectively process the conditioned feature maps, we analyze the full operation of the cConv. As shown in Fig. 4, the output of the cConv y can be analyzed as follows:

$$y = x \otimes (\gamma_s W + \beta_s) = x \otimes \gamma_s W + x \otimes \beta_s. \quad (5)$$

Note that (5) includes tensor broadcasting to compute matrix operation. The first term $x \otimes \gamma_s W$ describes a *conditioning process*, which produces features that are scaled differently according to the

given classes; this process helps disperse the convolutional feature maps. The second term $x \otimes \beta_s$ represents a *projection process*, which derives the condition-specific components from input feature maps x by projecting them onto condition-specific vectors. These observations reveal that the proposed sequential operations not only directly produce the conditional feature maps but can also handle the dynamic features of various conditions.

The normalized output of the cConv can be expressed as follows:

$$f_{cConv}(x, s) = \theta \left(\frac{\gamma_s \cdot x \otimes W + x \otimes \beta_s - \mu}{\sigma} \right) + \delta \quad (6)$$

where θ and δ are nonspecialized scaling and shifting parameters, respectively. Therefore, both (3) and (6) can be rewritten as follows:

$$\begin{aligned} f_{cBN}(x, s) &= \frac{\theta_s}{\sigma} (x \otimes W) + \left(\delta_s - \frac{\mu \theta_s}{\sigma} \right) \\ &= \hat{\theta}_s (x \otimes W) + \hat{\delta}_s \end{aligned} \quad (7)$$

$$\begin{aligned} f_{cConv}(x, s) &= \frac{\theta \gamma_s}{\sigma} (x \otimes W) + \frac{\theta}{\sigma} (x \otimes \beta_s) + \left(\delta - \frac{\mu \theta}{\sigma} \right) \\ &= \hat{\gamma}_s (x \otimes W) + \hat{\theta} (x \otimes \beta_s) + \hat{\delta}. \end{aligned} \quad (8)$$

The first terms of (7) and (8), that are $\hat{\theta}_s (x \otimes W)$ and $\hat{\gamma}_s (x \otimes W)$, play the same role of conditioning the feature maps. The second term of (7), $\hat{\delta}_s$, is a condition-specialized bias, which is independent of the input feature maps. On the other hand, the second term of (8), $\hat{\theta} (x \otimes \beta_s)$, can be considered as a conditioned residual path, which derives the conditional information from the previous layer. Since cConv projects the latent feature vectors on condition-specific vectors, it distills the intrinsic conditional features in the latent vectors. In other words, unlike the generator using the cBN, which embeds the conditional information using the external shifting vectors, the one using the cConv handles conditional information internally with the projection of the adjacent layers on the condition-specific space. Indeed, some studies [26], [27] proved that utilizing latent vectors for the conditioning task improves the performance of the generative model by making the latent space disentangled. However, they exploit extra layers to produce the affine matrices from the latent vectors. In [27], for instance, multiple fully connected

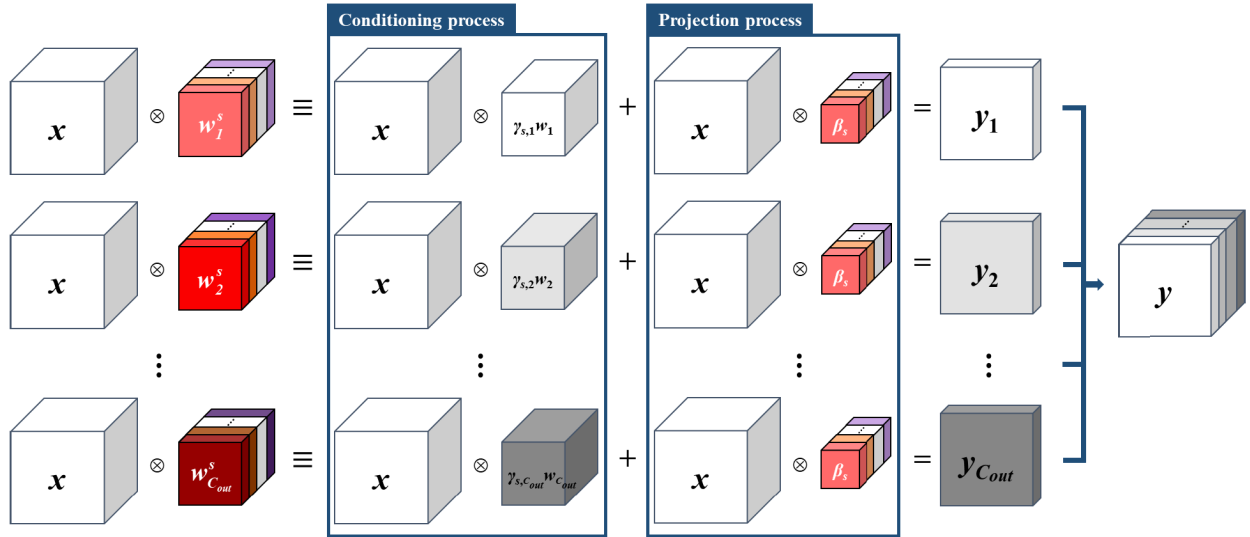


Fig. 4. Analysis of the cConv layer. The conditioned weights are divided into two terms, the conditioning and the projection processes. The conditioning process scales output feature maps with the specialized scale parameters corresponding to the condition. The projection process derives the condition-specific characteristics from the input feature maps and adds them back to the output feature maps.

layers, called a mapping network, are used to utilize the latent vectors. Although the mapping network gives meaningful results, additional layers increase the complexity of the generator. In the proposed method, by the conditioned residual path, the generator can use the latent vectors without additional layers.

In fact, more diverse W^s can be obtained by performing scaling and shifting operations with different values on each channel of every filter. However, this approach requires $2 \times N \times C_{in} \times C_{out}$ operating parameters in N conditions, which means the number of parameters rapidly increases with an increase in the dimension of the input and output feature maps. In contrast with this approach, although the proposed sequential modulation only requires $N \times (C_{in} + C_{out})$ parameters, it can specialize the weights to each condition. Therefore, with the advantage of simplicity and effectiveness, we adopt the proposed sequential operations to the cConv.

V. EXPERIMENTS

A. Implementation Details

To evaluate the superiority of the cConv, we conducted extensive experiments using the CIFAR-10 [14], CIFAR-100 [14], LSUN [15], and ImageNet [16], [17] datasets. The CIFAR-10 and CIFAR-100 datasets consist of 32×32 resolution images with 10 and 100 classes, respectively. The ImageNet and LSUN datasets contain 1000 and 10 classes of images, respectively, which have more detail and complex scenes. We compressed the images from the LSUN and ImageNet datasets as 128×128 pixels. For the objective function, we adopted the *hinge* version of adversarial loss, which is defined as follows:

$$L_D = E_{(I,s) \sim P_{data}(I)} [\max(0, 1 - D(I, s))] + E_{z \sim P_{z(z)}, s \sim P_{data}} [\max(0, 1 + D(G(z, s)))] \quad (9)$$

$$L_G = -E_{z \sim P_{z(z)}, s \sim P_{data}} D(G(z, s)). \quad (10)$$

For all of the parameters in the generator and discriminator, we follow the implementation in [10] and [28]. For the CIFAR-10 and CIFAR-100 datasets, we used a batch size of 64 and trained the generator for 100k iterations, whereas the generator for the LSUN and ImageNet datasets was trained with a batch size of 32 for 500k iterations. Our experiments were conducted on CPU

Intel(R) Xeon(R) CPU E3-1245 v5 and GPU TITAN X (Pascal), and implemented in *PyTorch* v1.4.

B. Base Lines

To evaluate the effectiveness of the proposed method, we employed the same generator and discriminator architectures as those used in the leading cGANs scheme [10]. The model utilizes the multiple residual blocks (ResBlock) [29]. Note that the discriminator does not contain the BN layer, since the spectral normalization [20] is applied to all of the weights in the discriminator. For the discriminator, we performed downsampling (average pooling) following the second convolutional layer in the ResBlock, whereas the generator upsampled the feature maps using a nearest neighbor interpolation prior to the first convolution layer in the ResBlock.

C. Performance Metrics

To evaluate how “realistic” each generated image is, we employed the most popular assessments, inception score (IS) [18] and Frechet inception distance (FID) [30], which measure the visual appearance and diversity of the generated images. The IS [18] computes the KL divergence between the conditional class distribution and the marginal class distribution to measure the diversity and quality of the generated images. It is demonstrated in [18] that this metric is strongly correlated with the subjective human judgment of image quality. With a generated image X , the IS can be defined as follows:

$$I = \exp(E[D_{KL}(p(n|X)||p(n))]) \quad (11)$$

where n is the label predicted by the inception model [31], and $p(n|X)$ and $p(n)$ represent the conditional class distributions and marginal class distributions, respectively. With higher IS, the quality of the image that is generated from the model is higher. On the other hand, the FID is another assessment, which is more principled and comprehensive. The FID can be obtained by calculating the Wasserstein-2 distance between the distribution of the generated samples, P_G , and the distribution of the real samples, P_R , in the feature space of the inception model [31],

TABLE I

RESULTS OF THE QUANTITATIVE ASSESSMENTS INCLUDING IS AND FID ON BOTH CIFAR-10 AND CIFAR-100 DATASETS

Method	Dataset			
	CIFAR-10		CIFAR-100	
	IS	FID	IS	FID
<i>concat</i>	8.21	14.51	8.67	16.06
cBN [10]	8.45	11.12	8.99	15.58
ccBN [26]	8.58	11.29	8.47	15.83
cConv (proposed)	8.60	10.82	9.17	14.23

TABLE II

ABLATION STUDY DEMONSTRATING THE EFFECTS OF FILTER-WISE SCALING AND CHANNEL-WISE SHIFTING ON CIFAR-10 (TOP) AND CIFAR-100 (BOTTOM) DATASETS

filter-wise scaling	channel-wise shifting	IS	FID
✓		8.45	11.42
	✓	8.37	12.10
✓	✓	8.60	10.82
filter-wise scaling	channel-wise shifting	IS	FID
✓		8.19	15.44
	✓	8.31	18.67
✓	✓	9.17	14.23

which is expressed as follows:

$$F(p, q) = \|\mu_p - \mu_q\|_2^2 + \text{trace}(C_p + C_q - 2(C_p C_q)^{1/2}) \quad (12)$$

where $\{\mu_p, C_p\}$ and $\{\mu_q, C_q\}$ are the mean and covariance of the samples with distribution P_G and P_R , respectively. In contrast with the IS, the generated samples with high quality have lower FID. In our experiments, we randomly generated 50000 samples and computed the IS and FID using the same number of real images.

D. Quantitative Comparison

To highlight the novelty and effectiveness of the proposed conditioning operation, we conduct experiments, which replaced the widely used conventional conditioning operation with ours: 1) input concat (*concat*), which provides the conditional information by concatenating the one-hot vector into the input noise, and conditional normalization, which employs 2) cBN [10]; or 3) ccBN [26] for the generator. In the ResBlock, the BN is replaced by conditional normalization, and the standard convolution is replaced by the cConv for the experiments of the proposed method. To ensure fair comparison, both the proposed and conventional methods use the same discriminator, which is projection discriminator [10]. Table I presents the comprehensive benchmarks between the proposed and conventional methods. The bold numbers in Table I indicate the best performance among the results. It is shown that the proposed method outperforms the conventional conditioning methods by a considerable margin in terms of IS and FID. As compared with the cBN-based methods, in particular, which have been known as the strong solution to supply the conditional information to the generator [10], [20], [26], [28], the proposed method shows better performance. Note that the cConv significantly improves the FID, which mainly evaluates the quality of the images, in the case of the CIFAR-100 dataset; this indicates that the cConv can successfully generate higher-quality images on many conditions than the conventional conditioning techniques.

To demonstrate the validity of the filter-wise scaling and the channel-wise shifting operations, we conducted ablation experiments by eliminating each operation from the cConv. Table II also shows

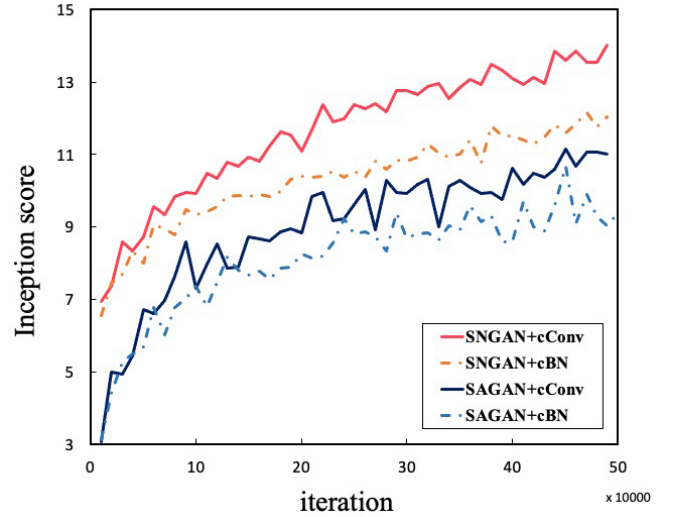


Fig. 5. Learning curve showing the performance growth of the IS over the training iteration. The solid and dashed lines indicate the model employing the cConv and the cBN, respectively.

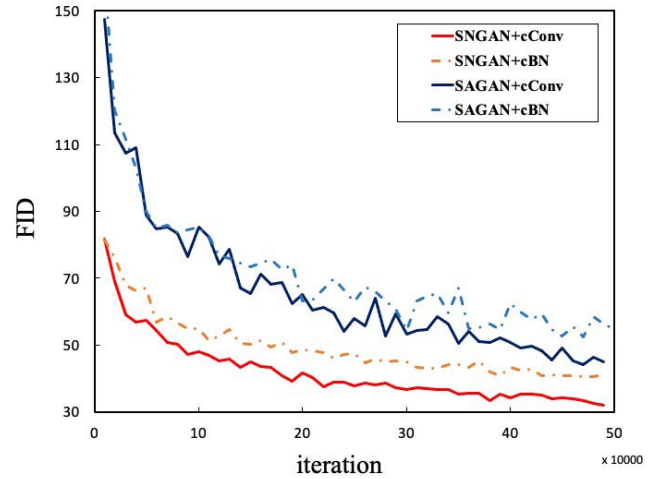


Fig. 6. Learning curve showing the performance growth of the FID over the training iteration. The solid and dashed lines indicate the model employing the cConv and the cBN, respectively.

that the cConv without the filter-wise scaling operation exhibits inferior performance on both datasets compared with the cConv. These results show that the filter-wise scaling operation is very effective in producing the conditioned feature maps, resulting in the fact that the filter-wise scaling is an essential part of cConv to generate the conditioned feature maps. On the other hand, we observed that the cConv without the channel-wise shifting operation performs worse especially on the CIFAR-100 datasets. These results also indicate that the channel-wise shifting acting as conditioned residual path supports the derivation of conditional components from the adjacent layer to generate high-quality images in numerous conditions. Consequently, with a small number of parameters, both operations should be combined to maximize the performance of the cGANs scheme.

To make quantitative results more reliable, we also compared the cBN and the cConv on ImageNet with 128×128 resolution. Widely used to this day, most existing conditioning strategies, including ccBN [26], adaIN [27], and adaAttN [32], are based on the cBN. Thus, we utilize, as the baselines, SNGAN [10] and SAGAN [28], which are the leading models employing the cBN. Figs. 5 and 6 show

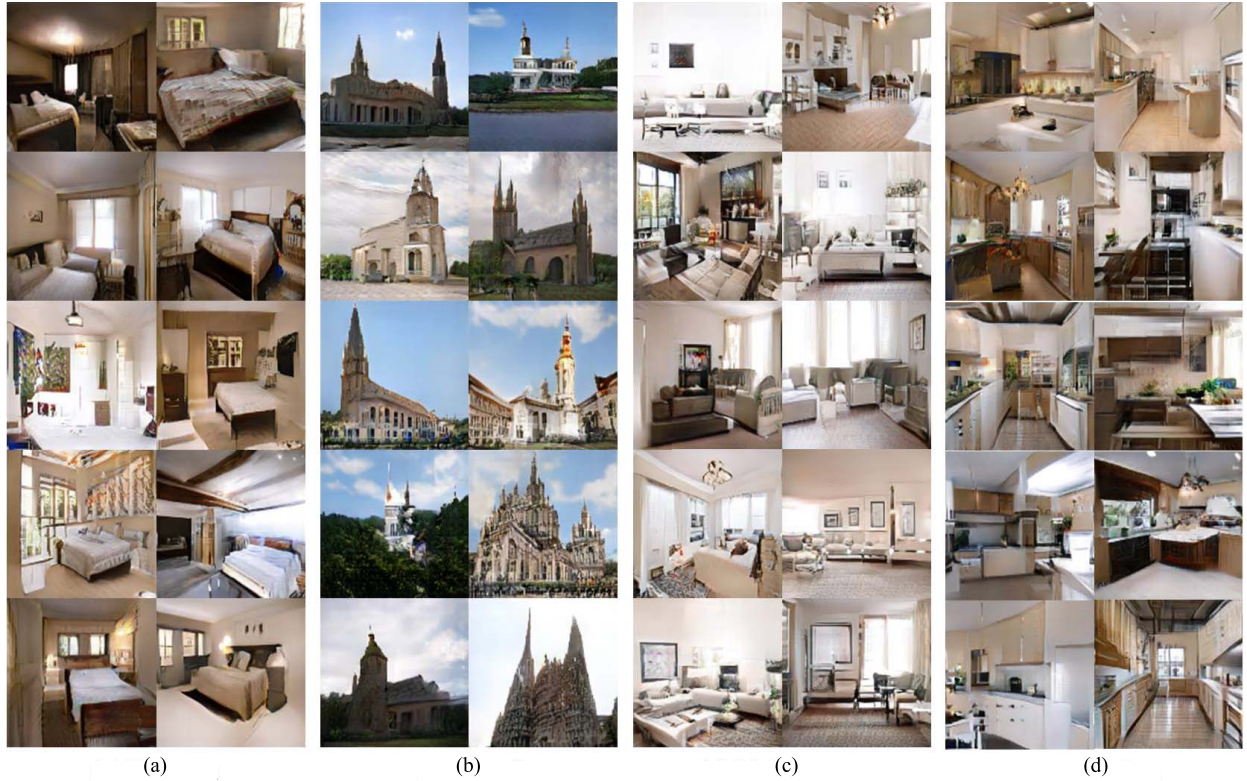


Fig. 7. High-resolution images of 128×128 pixel generated with the proposed method using LSUN datasets. (a) Bedroom. (b) Church. (c) Living room. (d) Kitchen.

TABLE III
RESULTS OF THE QUANTITATIVE ASSESSMENTS INCLUDING
IS AND FID ON ImageNet

Method	Inception Score	FID
SNGAN [10]+cBN	12.09 ± 0.186	40.99
SNGAN [10]+cConv	13.99 ± 0.175	32.26
SAGAN [28]+cBN	9.40 ± 0.125	54.66
SAGAN [28]+cConv	10.84 ± 0.185	44.91

the IS and FID results of the models employing the cBN and the cConv on two different cGANs methods. As shown in Figs. 5 and 6, the performance of the model employing the cBN is saturated earlier than the one adopting cConv. As a result, the cConv improves the performance of the generator significantly. Table III shows the final results of the cBN and the cConv. As shown in Table III, the model adopting the proposed cConv shows better performance than the cBN model on both cGANs methods. Unlike the condition-specialized shifting in the cBN, the *projection process* of the cConv utilizes the residual bias, which is affected by the previous layer. Therefore, the cConv needs adequate training iterations to attain the maximum performance.

E. Qualitative Results

1) *Conditional Image Generation*: To meaningfully demonstrate visual results at high resolution, we conducted extra experiments on both LSUN [15] and tiny-ImageNet [17] datasets, which is a subset of ImageNet, which are considered to be challenging ones consisting of detail and complex conditions. As shown in Fig. 7, the cConv can generate high-quality and diverse images including complex scenes with high resolution. Fig. 8(a) shows the samples for the classes

with low FID, i.e., classes on which the generative distribution is similar to the target conditional distribution, and Fig. 8(b) shows the reverse. As shown in Fig. 8(a), the images in the classes with low FID exhibit visually pleasing results with high diversity. However, as illustrated in Fig. 8(b), the images in the classes with high FID show slightly weak performance, since they contain complex objects, such as humans; that is, the diversity of the target data distribution is too wide. Nevertheless, the proposed model tends to produce features somewhat close to the original images even for complex classes. Note that this brief does not intend to design an optimal network structure for a generator with the cConv; there could be another architecture that leads to better performance. On the contrary, we care more about whether it is possible to learn the conditioned feature maps by simply replacing the standard convolution layer with the proposed cConv.

2) *Category Morphing*: Using the proposed cConv, we can also successfully conduct category morphism. When there are two different conditions s_1 and s_2 , we simply inter-divide both filter-wise scaling and channel-wise shifting parameters of the cConv, corresponding to these conditions. Fig. 9 shows the interpolated images with the same input noise vector. Note that cConv can produce morphing images when s_1 and s_2 are significantly different.

F. Conditional Image Style Transfer

To demonstrate the generalization ability of the cConv, we applied it to the image style transfer. In our experiments, we chose ten artistic paintings, including Ukiyo-e, Van gogh, and Jung-seob lee, and classed them into ten conditions. Similar to the cGANs scheme, we trained a style transfer model to generate artistic style-transferred images with given conditions. As a result, a single model can produce the images with ten different artistic styles depending on

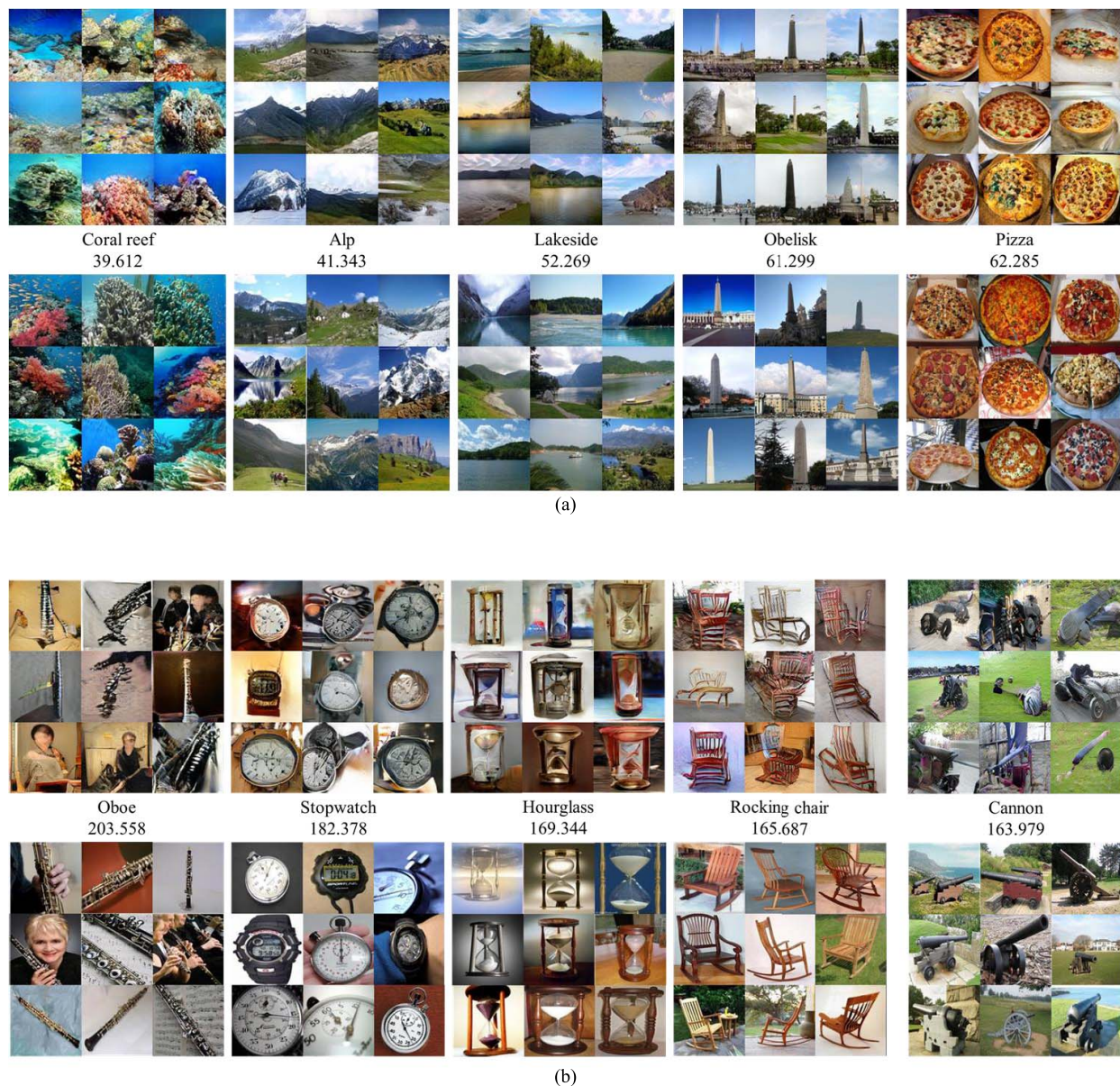


Fig. 8. 128 × 128 pixel images generated with the cConv for the classes with (a) five low FID scores and (b) five high FID scores. The strings and values between each panel, respectively, indicate the name of the corresponding class and the FID score. The second row in each panel corresponds to the target dataset.



Fig. 9. Results of category morphing images obtained using tiny-ImageNet datasets. The first row shows morphism from Alp to Seashore. The second row shows morphism from Beacon to Cliff.

given conditions. We adopted the network architecture as well as the loss function presented in [22]. Note that we replaced the standard convolutional layer of the generator with the cConv. Fig. 10 illustrates

the results for a subset of the style. As shown in Fig. 10, the networks with the cConv can provide the input images with different color palettes and textures depending on the given condition.

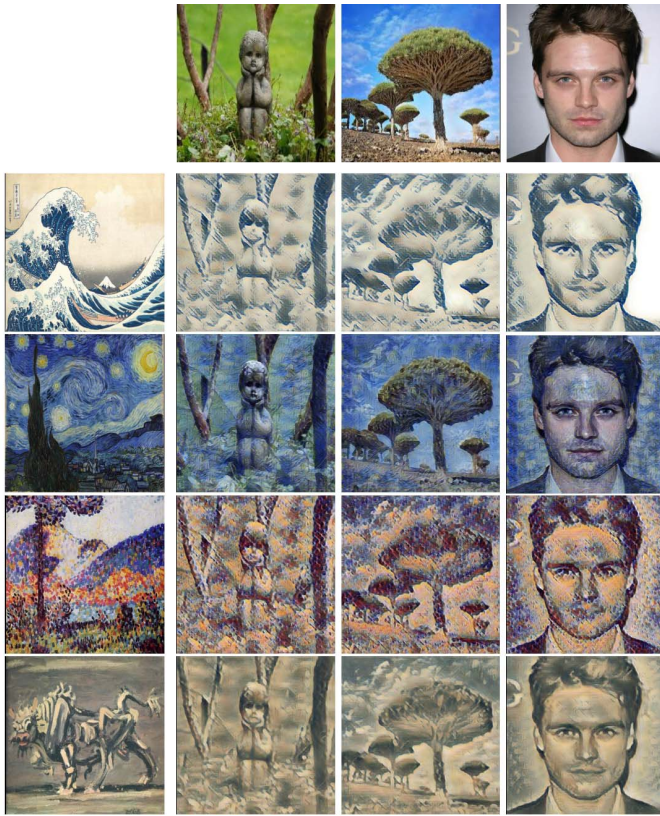


Fig. 10. Selected results of the conditional style transfer. Each column illustrates the resultant images with different artistic styles, which are Ukiyo-e, Van gogh, Henri edmond, and Jung-seob lee, from the same input image.

VI. CONCLUSION

This brief presented a novel convolution layer, called a cConv layer, for cGANs. We demonstrated that the cConv can effectively handle the conditioned feature maps by applying the proposed filter-wise scaling and channel-wise shifting operations. One of the main advantages of cConv is that it can be incorporated into existing architectures, and our experiments reveal that the generator with the cConv improves the performance of the baseline models. In addition, we proved the generalization ability of cConv by applying it to image style transfer. It is expected that the proposed method, with the advantage of simplicity and effectiveness, will be applicable to other cGAN frameworks, such as image-to-image translation and text-to-image synthesis.

REFERENCES

- [1] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [2] 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.
- [3] H. Zhang *et al.*, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," 2017, *arXiv:1710.10916*.
- [4] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
- [5] 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.*, 2016, pp. 217–225.
- [6] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," 2016, *arXiv:1605.05396*.
- [7] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2642–2651.
- [8] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *Proc. ICLR*, vol. 2, 2017.
- [9] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," 2016, *arXiv:1606.05328*.
- [10] T. Miyato and M. Koyama, "cGANs with projection discriminator," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–21.
- [11] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.
- [12] 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. 1125–1134.
- [13] 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.
- [14] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.
- [15] 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*.
- [16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [17] L. Yao and J. Miller, "Tiny ImageNet classification with convolutional neural networks," CS 231N, Tech. Rep., 2015.
- [18] 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.*, 2016, pp. 2234–2242.
- [19] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016, *arXiv:1606.01583*.
- [20] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.
- [21] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 172–189.
- [22] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 694–711.
- [23] H. Zhang *et al.*, "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.
- [24] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," 2019, *arXiv:1903.07291*.
- [25] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6594–6604.
- [26] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*.
- [27] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.
- [28] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*.
- [29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
- [30] 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 *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.
- [31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
- [32] S. Liu *et al.*, "AdaAttN: Revisit attention mechanism in arbitrary neural style transfer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6649–6658.