Signal Processing: Image Communication 115 (2023) 116959

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

CT-GAN: A conditional Generative Adversarial Network of transformer architecture for text-to-image

Xin Zhang [a], Wentao Jiao [a], Bing Wang [b,c,*], Xuedong Tian [d]

[a] College of Electronic Information Engineering, Hebei University, Baoding, 071000, Hebei, China
[b] Hebei Key Laboratory of Machine Learning and Computational Intelligence, Hebei University, Baoding, 071000, Hebei, China
[c] College of Mathematics and Information Science, Hebei University, Baoding, 071000, Hebei, China
[d] College of Cyber Security and Computer, Hebei University, Baoding, 071000, Hebei, China

ARTICLE INFO

Keywords:
Generative Adversarial Networks (GANs)
Text-to-image
Transformer
Filter module
Shift self-attention

ABSTRACT

How to generate an image from a text description is an imaginative and challenging task. This study proposes a conditional generative adversarial network (GAN) of transformer architecture for text-to-image tasks called CT-GAN by employing the GAN generator based on transformer architecture. We also propose a filtering module suitable for non-end-to-end multi-stage models. This module can screen out the good images generated in the previous stage and allows only the good images to participate in the generation of the later stage. This method significantly improves the quality of the generated images. Furthermore, we designed a generator and discriminator based on symmetry. In the generator, we propose a shift self-attention technology to establish information communication between grids, reduce boundary loss, and improve image quality. We established two modes of local and global discriminations based on the grid, which can balance the performance of the generator and discriminator, improve the training stability, and accelerate the model convergence. We conducted several experiments on the widely used conditional datasets (CUB and COCO) and unconditional datasets (CelebA and LSUN church). The experimental results show that the proposed CT-GAN is superior to the most advanced convolution model in generating diversity and semantic consistency. Codes are available at: https://github.com/Jwtcode/CT-GAN.

1. Introduction

Goodfellow et al. [1] first proposed the idea of generative adversarial networks (GANs) [2] and generated images through a game between a generator and a discriminator. With the continuous advancement of GAN technology, people are no longer satisfied with the aimless generation of GAN; thus, some authors proposed conditional GAN (cGAN) [3] to limit the generation of GAN by introducing some conditional variables based on various information, such as category labels and image descriptions. Compared to directly synthetic images, synthetic images from text faces several challenges, such as ensuring semantic consistency between text and image, generating high-resolution images with multiple objects and developing suitable and reliable evaluation metrics relevant to human judgments [4]. Currently, almost all text-to-image GANs are based on convolution architecture. They are committed to improving the image resolution and effectively using text information. StackGAN [5] first used multi-stage method to generate high-resolution images. Although this multi-stage method can synthesize high-resolution images, it still has some drawbacks. For example, the generation of the later stage depends heavily on the input of the previous stage. If the quality of the images generated in the previous stage is poor, it is not easy to correct and refine them in the later stage. To solve this problem, StackGAN++ [6] changed the training mode from non-end-to-end to end-to-end. This improvement achieved fine-tuning effects by increasing the information communication between different stages. The subsequent multi-stage models [7,8] extended this method; however, they still had great drawbacks. The end-to-end training mode greatly increased the memory overhead and calculation scale, resulting in slow training and difficult convergence. Inspired by a proposal in faster recurrent convolutional neural network (R-CNN) [9], we seek a network that can distinguish the quality of the generated images to filter out the bad images generated in the first stage and complete the cleaning of the generated images without increasing the computational cost. We call it a filter.

The processing of text information is essential for text-to-image tasks. It is mainly divided into text information coding and utilization. Text information coding ensures a one-to-one mapping between coding and images. Currently, more convolution encoders are used, including long short-term memory (LSTM) [10] and RNN [11], and some models [12,13] use transformer-based encoders. Compared with text

[*] Corresponding author at: Hebei Key Laboratory of Machine Learning and Computational Intelligence, Hebei University, Baoding, 071000, Hebei, China.
E-mail address: wangbing@hbu.edu.cn (B. Wang).

https://doi.org/10.1016/j.image.2023.116959
Received 7 September 2022; Received in revised form 25 January 2023; Accepted 21 February 2023
Available online 17 March 2023

information coding, the use of text information is more critical. DM-GAN [7] designed a series of gates that can dynamically store words to obtain the information most relevant to image generation. Its advantage is that it can dynamically pay attention to text information while introducing additional and complex networks. DF-GAN [8] introduced text information into each subblock to deeply integrate text and image features. However, frequent introduction of text information may lead to dimension disaster and increase the difficulty of model learning. Fundamentally speaking, their improvement is to make up for the inability of convolution to obtain long-distance dependence, which is proved by the advantages of adopting self-attention [14] and nonlocal [15] operations in computer vision (CV). In contrast, a transformer can obtain long-distance dependence without an additional network. Owing to this advantage, we use a transformer to process image and text information simultaneously and completely rely on the self-attention mechanism in the transformer to complete the information communication between text and image without introducing an additional network.

Based on the excellent performance of a transformer in natural language processing (NLP), several authors [16,17] have introduced it into the field of CV. Vision transformer (ViT) [18] is the first visual classification network built with a pure transformer structure. ViT [18] regards pixels in images as words and builds pixel-level association through a self-attention mechanism. However, unlike NLP tasks, the number of pixels in images is far more than the number of words in sentences. Simply using a self-attention mechanism to calculate the entire image causes huge computational overhead and memory consumption, which is also the challenge we encounter when generating high-resolution images. TransGAN [19] proposed a grid self-attention, which divides the entire feature image into multiple grids and applies self-attention to each grid separately. Although this is a good solution to the memory bottleneck, it is accompanied by the loss of boundaries between divided grids due to the lack of information communication and the easy occurrence of training crashes. Inspired by the Swin-Transformer [20], we propose shift self-attention (SSA) using a sliding grid instead of the previous fixed grid to establish the communication between grids. It is worth noting that using a sliding grid cannot cause each grid to gain global attention, meaning that this generator does not start from the global perspective in the grid generation stage. Facing this situation, the design of the discriminator is particularly important. We know that the improvement of generator performance cannot be separated from the guidance of the discriminator. Since the generator does not start from the global perspective, if the traditional discrimination mode is used to discriminate from the global perspective, it will easily lead to a performance imbalance between the two and crash the training. Thus, we design a discriminator symmetrical to the generator and add a local discrimination mode. These two improvements solve the problems of boundary loss [21] and training imbalance.

The main contributions of this study are summarized as follows.

- We proposes a conditional GAN of transformer architecture for text-to-image tasks by employing the GAN generator based on transformer architecture.
- We propose a filtering module suitable for non-end-to-end multi-stage models. This module can screen out the good images generated in the previous stage and allows only the good images to participate in the generation of the later stage.
- We designed a generator and discriminator based on symmetry. In the generator, we propose an SSA technology to establish information communication between grids.
- We also designed two training modes in the discriminator to balance the performance of the generator and discriminator.
- We conducted several experiments on the widely used conditional datasets (CUB [22] and COCO [23]) and unconditional datasets (CelebA [24] and LSUN church [25]).

2. Related work

Currently, almost all text-to-image GANs are convolution-based stack structures, and they focus on how to make better use of text information. StackGAN [5] pioneered the stacked GAN architecture using multiple generators and discriminators to synthesize images. We also inherited the stacked architecture. Unlike it, we added a filtering module between different stages to filter out the bad images generated in the previous stage. AttnGAN [26] introduced a cross-pattern attention mechanism to help the generator synthesize more detailed images. DM-GAN [7] proposed a dynamic memory module to record the importance of words to continuously refine images. They added an attention mechanism in the model to record the importance of each word for image synthesis. Unlike them, we did not introduce an additional network to record the importance of words. We completely relied on the self-attention mechanism in the transformer generator to complete the relationship between text and images.

For GANs based on transformer architecture, TransGAN [19] proposed a pure transformer GAN to generate images without convolution. This model generator used grid generation when generating high-resolution images. However, due to the lack of information interaction between grids, each grid only generated what it thought was right and ignored the rationality of the whole. Unlike them, we used a sliding grid instead of the previous fixed grid to increase information communication between grids. CombinGAN [27] combined convolution with a transformer using the transformer structure in the generator and convolution structure in the discriminator. This model has no module generated by the grid and can only generate low-resolution images. Different from this, we fully considered the performance gap between the transformer generator and convolution discriminator, referred to the structure of the generator, and designed a discriminator based on symmetry. We also designed local and global discriminations on the discriminator to balance their performance.

In text information, we do not use an extra attention module to record the importance of each word for image generation. We rely entirely on the transformer's self-attention mechanism to complete the information interaction between text and image and ensure the semantic consistency between text and image. In the generator part, we designed a filtering module located at the junction of the first and second stages. It filters out the bad images generated in the first stage and only allows the better quality images to participate in the second stage. We also propose an SSA using a sliding grid instead of the previous fixed grid to increase the information interaction between grids. In the discriminator part, we refer to the generator and design a discriminator based on symmetry. Additionally, we design two modes: local and global discriminations. Compared with the traditional convolution discriminator, our designed discriminator can adjust its performance according to the generator's performance to better guide the generator, improve the training stability, and speed up the convergence of the model. We designed the generator and discriminator around the grid to ensure that the semantics of each grid in the generator is as correct as possible and that the semantics of all grids combined are as correct as possible.

3. CT-GAN

The proposed CT-GAN employs the most commonly used multi-stage method (Fig. 1), which consists of two pairs of generators and discriminators and a filter using the Stage-I discriminator. CT-GAN takes text as input and extracts sentence features through a LSTM [28] text encoder. The sentence feature combined with noise samples by the enhancement method is the generator input, through a two-stage generator and filter, the final generated image serves as the output result.

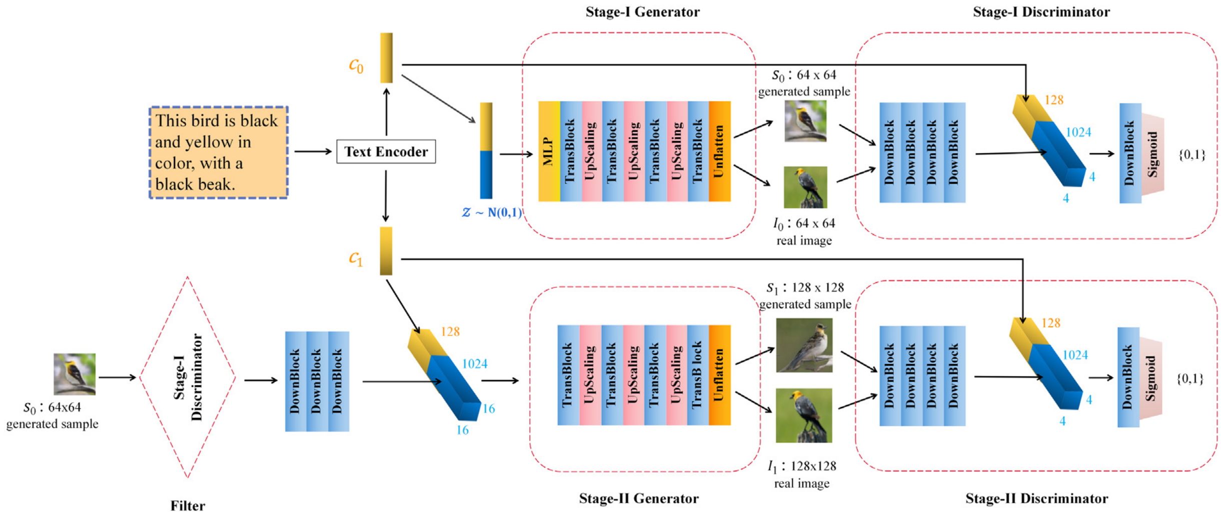


Fig. 1. Architecture of the proposed CT-GAN for text-to-image synthesis. The proposed CT-GAN consists of two pairs of generators and discriminators. The generator uses a pure transformer architecture, and the discriminator uses a convolution architecture. Here, c_0 and c_1 represent the conditional inputs of the first and second stages, respectively; s_0 and I_0 represent the generated images and real samples of the first stage, respectively; s_1 and I_1 represent the generated images and real samples of the second stage, respectively. The first stage generates 64 × 64 resolution images; the second stage generates 128 × 128 resolution images; the two stages have independent training.

3.1. Preliminaries

GAN consists of a generator G and a discriminator D. The generator G and the discriminator D are trained alternately. The generator G tries to fool the discriminator D, and the discriminator D tries not to be fooled by the generator G. The training process is similar to two-person arm wrestling, and the objective function is as follows:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}}[\log\ D(x)] + \mathbb{E}_{z\sim p_z}[\log\ (1-D(G(x)))] \tag{1}$$

3.2. Filter module

For the stacked GAN model, there are two modes to choose from: the non-end-to-end mode, where each stage is trained separately, and the end-to-end mode, where multiple stages are trained together. The most advanced text-to-image GANs [7,8] are based on the latter. The advantages of the former are memory-friendly and faster convergence; however, it has the disadvantage that there is no interaction between the stages and cannot make some minor adjustments. The advantage of the latter is that it can solve the interaction problem among different stages and make some minor adjustments; however, it has the disadvantages of slow convergence speed, high memory occupation, and long training period.

CT-GAN is trained independently at each stage because of its memory-friendly non-end-to-end mode. This training mode highly depends on the quality of the initial image. Although the second stage can correct the defects of the image generated in the first stage, it cannot be corrected when faced with extremely low-quality images. To address this situation, we propose using a filter to filter out the low-quality images generated in the first stage as much as possible. After the first stage of training, we obtain a pretrained generator and discriminator. We use the discriminator in the first stage as our filter (Fig. 2). For the same text description, the generator generates n images simultaneously. Our filter scores these n images, selects the image with the highest score, and sends it to the second stage for training. From the shallow level, it can be understood that the filter can filter out some bad images with a high probability. However, from the deep level, it can be understood that we artificially narrow the feature distribution space of the generator; thus, increasing the probability of intersecting with the real image feature distribution space.

The selection of a filter should follow the following principles. We hope that the filter can distinguish the low-quality and real images generated in the first stage and cannot distinguish the high-quality and real images generated by the first-stage generator, which ensures that only the generated high-quality images can enter the second stage. If the filter cannot distinguish the generated image from the real image, or if it can completely distinguish the generated image from the real image, then this filter is not available.

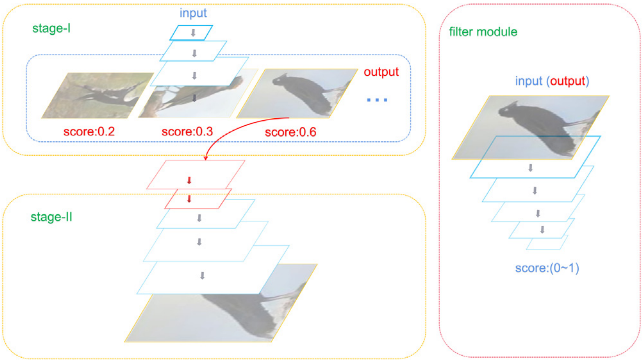


Fig. 2. The input of the first stage is random noise and text description encoding, and the output is n images corresponding to the text description, where n is a hyperparameter, and the default setting is 10. The input of the filter module is the n images generated in the first stage, corresponding to the text description of the n images; the output is a value from 0 to 1, which is used to evaluate the quality of the generated images.

Our filter is the discriminator trained in the first stage. Theoretically, the final state of the GAN model training is that the discriminator cannot distinguish the real image from the generated image. However, in the final test results, it was found that the score distribution of the real image does not overlap with that of the generated image. We use the model trained in the first stage to obtain the score distribution map of the COCO dataset (Fig. 3). It can be seen that the scores of the generated images are mostly distributed between 0 and 0.1, whereas those of the real images are mostly distributed after 0.1. Thus, it can be considered that the generated images with scores after 0.1 are closer to the real images, which also satisfies the above principle that the filter should be able to distinguish the generated low-quality images from the real images, but not the high-quality generated images from the real images. We also verified the effectiveness of the method in subsequent ablation experiments.

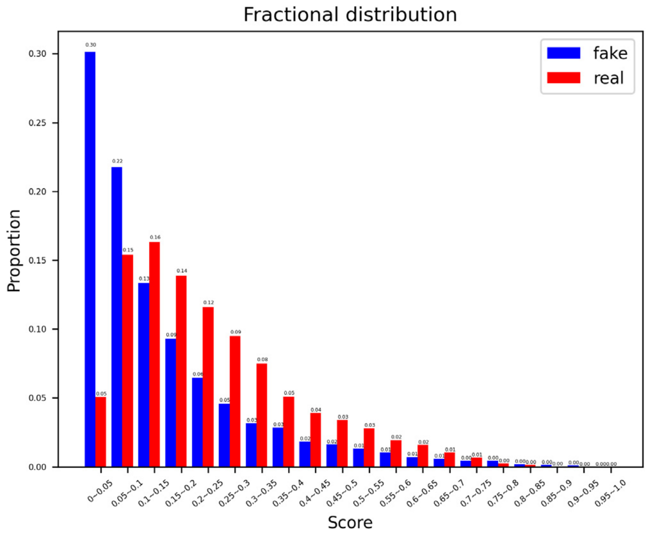
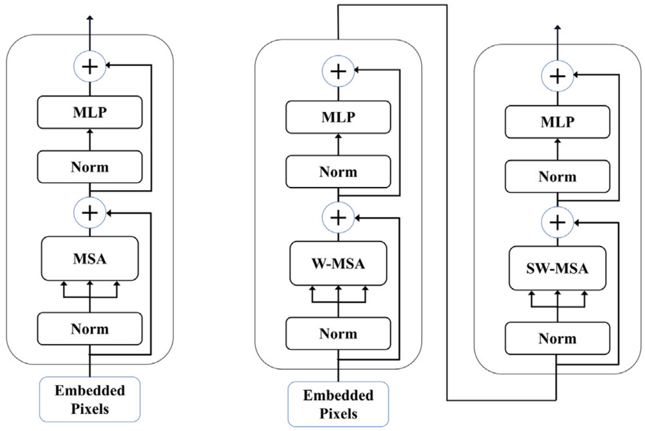


Fig. 3. Score distribution of the generated and real images. Blue and red represent the ratio of the generated and real images, respectively.



(a) Transformer Encoder (b) SwinTransformer Encoder

Fig. 4. Transformer and SwinTransformer encoders.

3.3. Transformer encoder

The transformer encoder plays an essential role as the core component of all transformer-based models in the field of NLP and CV. Moreover, many versions have been recently derived, and the SwinTransformer encoder is one of the most representative. In our model, both encoders are used simultaneously (Fig. 4). Fig. 4(a) shows the transformer encoder, which is a single-layer structure; it consists of alternating layers of multiheaded self-attention (MSA) and multi-layered perceptron (MLP) blocks. Layernorm (LN) is applied before every block and residual connections after every block. The MLP contains two layers with a GELU [29] nonlinearity. Fig. 4(b) shows the SwinTransformer encoder, which is a double-layer structure; the left side is the same as the transformer encoder, and the right side uses shifted windows MSA (SW-MSA) replaces MAS in the transformer encoder.

The MAS in the transformer encoder directly focuses on the information of the current window. The W-MSA and MSA in the SwinTransformer encoder only focus on the information of the current window. In contrast, the SW-MSA slides to the neighborhood window and obtains the information about the window. This feature of the SwinTransformer encoder allows the generator to pay attention to the information on the current and neighborhood windows when the grid is generated, which is beneficial for the grid to refer to the information in the neighborhood grid while drawing itself to improve the rationality of the overall layout, which is explained in detail in Section 3.4.

3.4. Shift self-attention

Compared with CNN, the advantage of transformer lies in its ability to capture global information because of its self-attention mechanism; however, this mechanism has disadvantages. Moreover, large-scale calculations often occur when dealing with long sequences or high-resolution images, which seriously reduce the reasoning efficiency. TransGAN [19] uses grid self-attention technology to solve this problem (Fig. 5(b)). Compared with the standard self-attention in Fig. 5(a), the grid self-attention divides the entire feature map into several grids of the same size and applies the standard self-attention to each grid. Each grid attention only focuses on the information in the current grid. Its advantage is that it can reduce the computational cost and is conducive to describing local details. However, its disadvantage is that information communication between grids is impossible, which is not conducive to improving the overall image quality. To solve this problem, we introduce SSA (Fig. 5(c)) by offsetting the grid in different layers to obtain the information in the neighborhood grid. The standard self-attention is still applied in each grid; its advantage is that each grid refers to the information in other grids while drawing itself, which is essential in improving the overall image quality.

We apply the standard self-attention to each grid. In the standard self-attention, we use relative position-encoded attention, given as follows:

$$Attention(Q,K,V) = softmax(((\frac{QK^T}{\sqrt{d_k}} + E)V)) \tag{2}$$

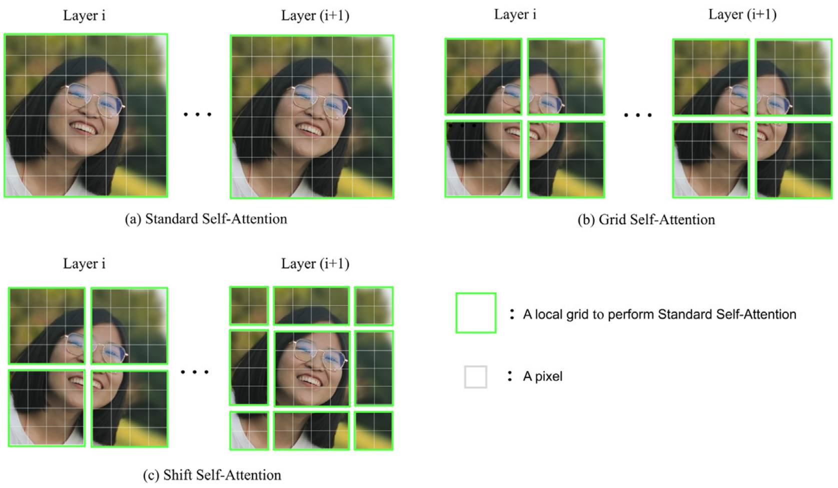
Here, $Q, K, V \in \mathbb{R}^{H\times W\times C}$ represent the query, key, and value matrixes, respectively. H, W, C represent the height, width of the image, and dimension of the feature map embedding. The difference in coordinate between each query and key on H axis lies in the range of $[-(H-1), H-1]$, and similar for W axis. By simultaneously considering the H and W axes, the relative position can be represented by a parameterized matrix $M \in \mathbb{R}^{(2H-1)\times(2W-1)}$. The relative position encoding E is taken from matrix M for each coordinate and added to the attention map QK^T as a bias term.

3.5. Positional embedding

In NLP, the most important thing in position coding is to add position information to reflect the different positions of each word. Position information is artificially set for each word because the same word has different meanings in different positions. In GAN, we reexamine the requirements of position encoding. First, we hope to reflect the meaning of the same pixel value in different positions. Second, we hope that pixels and pixels reflect a certain relative order relationship. Compared with establishing the relationship between words, artificially establishing the relationship between pixels is extremely large and complex. We draw on the experience of TransGAN [19] and use a learnable one-dimensional (1D) position embedding, which is different from it. We only use it when generating low-resolution images. When grid generation is needed, considering the mobility of the grid, we cancel the location embedding.

3.6. Generator

We take the first stage of generating a 64 × 64 resolution image as an example. The generator in this stage consists of Parts I and II. Part I generates the feature map with a resolution of not more than 32, whereas Part II generates the feature map with a resolution of more than 32. Fig. 6 describes the process of generating images using the generator, including initialization of input features, location



(a) Standard Self-Attention

(b) Grid Self-Attention

(c) Shift Self-Attention

Fig. 5. Three ways to calculate attention.

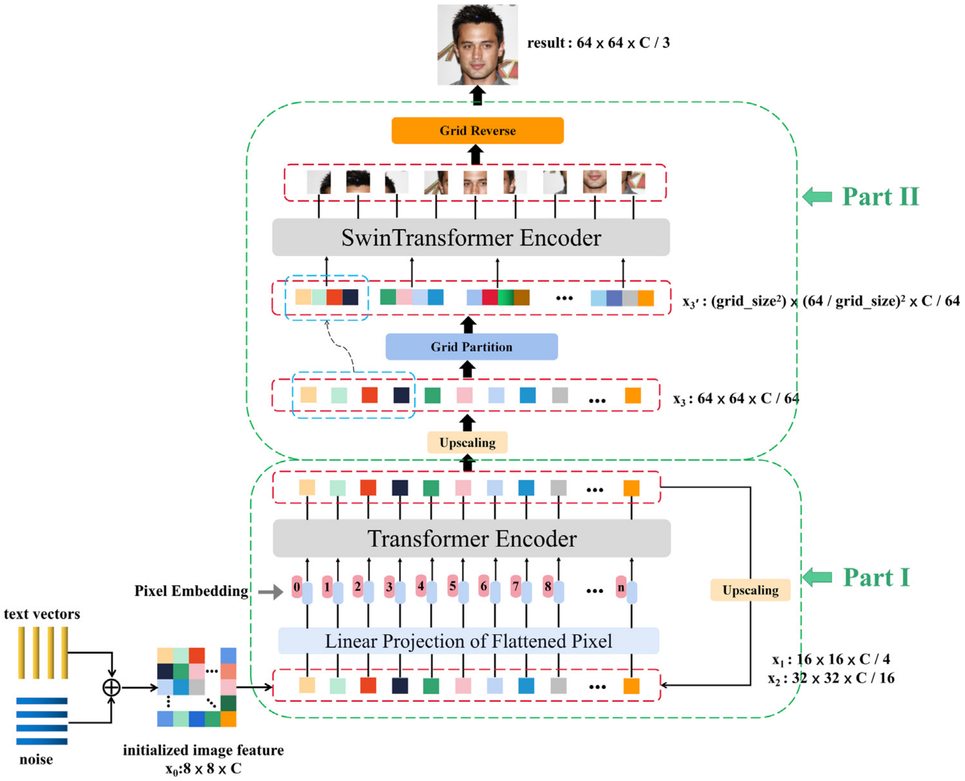


Fig. 6. Pipeline of the pure transform-based generator of the first-stage CT-GAN. We take the 64 × 64 resolution image generation task as a typical example of the main procedure. Each small square in the image represents a pixel. When the resolution is less than or equal to 32, we loop Part I. In this part, we use the standard self-attention. When the resolution is greater than 32, we need to use the grid generation. We execute Part II and use shift self-attention in this part.

embedding, iteration of feature maps, sampling of feature maps, and mapping and outputting of feature maps.

In the first step, we use MLP to fuse random noise and text vectors. We initialize them into 2D image features denoted as $x_0 \in \mathbb{R}^{H\times W\times C}$. For the initialized x_0, H, W are relatively small, and C is relatively large,so as to provide sufficient resources for subsequent upsampling operations.

In the second step, since the transformer encoder cannot directly process 2D information, the 2D feature map x_0 needs to be flattened into a 1D feature map sequence $x_0 \in \mathbb{R}^{(H\times W)\times C}$, each small square in Fig. 6 represents a pixel. There are $H \times W$ pixels in total, and the number of channels of each pixel is C.

In the third step, to preserve the spatial position information between pixels, we embed a learnable position code for the flattened x_0, as follows:

$$x_0 = [Pix_1, Pix_2, \dots, Pix_n] + E_{pos}, \\ n = H \times W, E_{pos} \in \mathbb{R}^{n\times c} \tag{3}$$

where n is the number of all pixels in the entire feature map sequence. The random initialization is between 0 and 1, and the dimension is consistent with the input feature map sequence.

The fourth step is to send the feature map sequence x_0 embedded with location information into the transformer encoder. At this time, the width and height of the feature map are not greater than 32.

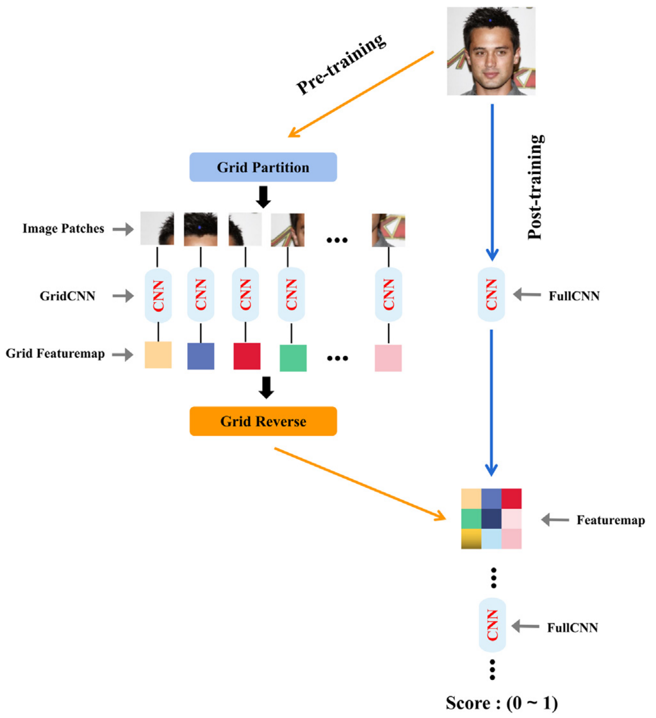


Fig. 7. Pipeline of the pure convolution discriminator in the first stage CT-GAN. We take the discrimination of 64 × 64 resolution images as an example of the main process. There are two discrimination modes in the figure: local discrimination (GridCNN) and global discrimination (FullCNN). Local discrimination is used in the early stage of training, whereas global discrimination is used in the late stage of training.

Therefore, there is no need to divide the grid, and the transformer encoder is used to update the iterative feature map directly.

In the fifth step, feature map upsampling, we reshape the 1D feature map sequence x_0 to 2D images feature x_0, upsample x_0 to obtain $x_1 \in \mathbb{R}^{2H\times 2W\times C/4}$, and then loop all operations of Part I until the feature figure size is greater than 32.

The last step is to reshape the k 1D token output using the Swin-Transformer encoder into k 2D grids, then reset the k 2D grids according to the window position to obtain a feature map of a given size, and finally project the number of channels of the feature map to three to get the final RGB image result $\in \mathbb{R}^{H\times W\times 3}$.

3.7. Discriminator

We refer to the structure of the generator and design the discriminator based on symmetry. Moreover, to better balance the performance of the generator and discriminator, we propose two training modes: local and global discriminations.

The idea of symmetry is reflected as follows. Assuming that the number of layers of the generator and discriminator is n, the generator adopts the method of sub-grid generation when generating a 64 × 64 resolution image at the ith layer. Then, the discriminator identifies the 64 × 64 resolution image at the $n-i$th layer. The method of sub-grid identification should also be used.

As an example, we take the first stage to discriminate 64 × 64 resolution images. Fig. 7 describes the process of the discriminator to discriminate images.

In the early stage of training, since the generator is difficult to learn and inefficient, and the discriminator has a low learning difficulty and high efficiency, the performance of the generator and discriminator is unbalanced, which is also the root cause of mode collapse. Therefore, to balance their performance, we used the grid discrimination mode in the early stages of training. In this mode, the discriminator will not discriminate the image from the overall perspective but discriminate the image from the perspective of each grid; thus, reducing the real image. This is different from the generated images; thus, avoiding mode collapse. From the generator's perspective, the grid discrimination mode can guide the generator more quickly on what each grid should generate to quickly outline the prototype.

In the later stage of training, after improving the performance of the generator, the grid-based discrimination mode has limited the generator. We are not forcing the generator to generate something for each grid. As long as the entire image is reasonable, it can be regarded as a good image. This requires the discriminator to discriminate from an overall perspective. Thus, we adopt a global discriminative mode in the later stages of training.

This method can improve the stability of model training, speed up model convergence, and improve the quality of the final generated image.

3.8. Objective function

As shown in Fig. 1, in the first stage, the pretrained text-encoder is used to encode the text description to obtain the conditional variable c_0 that obeys the Gaussian distribution. The conditional variable c_0 and the random noise z are fused and sent to G_0 to generate a low-resolution image s_0, and then fuse c_0 with s_0 and the real sample I_0, respectively, and pass it into the discriminator D_0. This process can be represented by the objective functions $\mathcal{L}_{D_0}$ and $\mathcal{L}_{G_0}$. Stage-I trains discriminator D_0 and generator G_0 by maximizing $\mathcal{L}_{G_0}$ and minimizing $\mathcal{L}_{D_0}$, respectively, as follows.

$$\begin{aligned}\mathcal{L}_{D_0} = {} & \mathbb{E}_{(I_0,t)\sim p_{data}}[\log\ D_0(I_0,c_0)]+ \\ & \mathbb{E}_{z\sim p_z,t\sim p_{data}}[\log\ (1-D_0(G_0(z,c_0),c_0))]\end{aligned} \tag{4}$$

$$\mathcal{L}_{G_0} = \mathbb{E}_{z\sim p_z,t\sim p_{data}}[\log\ (1-D_0(G_0(z,c_0),c_0))] \tag{5}$$

The real image I_0 and text description t obey the real sample distribution p_{data}, and z is random noise obeying the Gaussian distribution.

Entering the second stage, we need to introduce G_0 and F_0, use G_0 to generate a low-resolution image s_0, and then score through the filter F_0, select the image s_0 with the highest score. After downsampling the image s_1 to an appropriate size, it is fused with the conditional information c_1 and sent to G_1 to generate a high-resolution image. For the discriminator D_1, the same as in the first stage, c_1 is sent to D_1 combined with the real sample I_1, and the corresponding results are obtained. The discriminator D_1 and generator G_1 in Stage-II GAN are trained by alternatively maximizing $\mathcal{L}_{G_1}$ and minimizing $\mathcal{L}_{D_1}$, respectively, as follows.

$$\begin{aligned}\mathcal{L}_{D_1} = {} & \mathbb{E}_{(I_1,t)\sim p_{data}}[\log\ D_1(I_1,c_1)]+ \\ & \mathbb{E}_{s_0\sim p_{G_0},t\sim p_{data}}[\log\ (1-D_1(G_1(F_0(s_0,c_1),c_1),c_1))]\end{aligned} \tag{6}$$

$$\mathcal{L}_{G_1} = \mathbb{E}_{s_0\sim p_{G_0},t\sim p_{data}}[\log\ (1-D_1(G_1(F_0(s_0,c_1),c_1),c_1))] \tag{7}$$

Unlike the first stage, the second stage does not use random noise as input; however, it uses the low-resolution image generated in the first stage as input, which also has randomness.

4. Experiments

4.1. Datasets

We performed conditional image generation in the CUB and COCO datasets. The CUB dataset contains 11,788 images of 200 bird species, of which 150 categories and 8855 images are used for training, and the remaining 50 categories and 2933 images are used for testing. Each image has ten text descriptions. The COCO dataset consists of 80k and 40k images for the training and test sets, respectively. Each image has five text descriptions. we also conducted unconditional image generation on CelebA and LSUN church datasets.

4.2. Implementation details

Before the training, we first use the pretrained bidirectional LSTM [28] to encode all the text information, obtain the text embedding, and use the image and text embedding as input during training. Image preprocessing is also crucial for CT-GAN. For single-category datasets, we use the Boundingbox that comes with the dataset to crop the images so that the model pays more attention to the target instead of the background.

Due to the memory limitation, we only have two stages of image generation: 64 × 64 and 128 × 128 resolution images.

When we first built the model, we first tried using the pure transformer architecture in the generator and discriminator. The experimental results confirm that this architecture effectively generates low-resolution (not more than 48) images. When the resolution is further improved, the discriminator of the transformer architecture cannot play a good role. We replaced the discriminator with a convolution structure. Consequently, we found that the discriminator of convolution structure could make the model training more stable and generate recognizable images.

For the loss function, in the case of conditional training, we need to divide the positive and negative samples. Positive samples pair real images with real labels. In contrast, negative samples pair not only fake images with real labels but also fake images. Paired with the wrong labels, we use a binary cross-entropy loss function. The generator and discriminator use the Adam optimizer with a learning rate of $2e{-}4$, β_1 of 0.5, β_2 of 0.999. However, we do not need to divide positive and negative samples for unconditional training. The generator and discriminator use the Adam optimizer with a learning rate of $2e{-}4$, β_1 of 0.5, and β_2 of 0.999. In the first stage, we set the batch size of the generator and discriminator to 128. Meanwhile, in the second stage, we set the batch size to 64. We used 8 P100 GPUs and the open source Python library Pytorch to implement the CT-GAN model.

4.3. Evaluation criteria

We choose the widely used inception score (IS) [30] and Frechet inception distance (FID) [31] as evaluation criteria. IS examines the performance of the generated model from two aspects to determine whether the quality of the generated image is clear. The larger the IS value, the more representative the image. The higher the clarity of the image, the higher the quality of the image. The second is to determine whether the generated image has diversity, i.e., the generated image contains as many categories as possible. FID is another evaluation criterion that considers more connections between the generated and real images. It determines whether the image quality is good or bad by calculating the Frechet distance between the synthetic and real-world image distributions in the feature space. Contrary to IS, the smaller the value of FID, the more realistic the generated image. We generated 30k sample images on the conditional datasets using the text descriptions of the untrained test set for computing FID and IS. In contrast, we used the trained model to randomly generate 50k sample images on an unconditional datasets to calculate FID and IS.

4.4. Experiment results

We compare the most advanced text image synthesis methods [7,8] on CUB and COCO datasets with text descriptions. We obtain the IS and FID of these state-of-the-art models on the conditional datasets from the official pretrained models (Table 1). The results show that the proposed CT-GAN achieves the highest score on IS compared with other leading models, with the second best score on FID, just behind DF-GAN [8]. Compared with the current state-of-the-art DF-GAN [8], we improved the IS from 5.10 to 5.37 and 30.49 to 33.01 on the CUB and COCO datasets, respectively.

Table 1
Results of IS and FID compared with state-of-the-art methods on CUB and COCO datasets. The best results are in bold.

Methods	CUB		COCO	
	IS	FID	IS	FID
StackGAN	3.70 ± 0.04	(–)	8.45 ± 0.03	(–)
AttentionGAN	4.36 ± 0.03	23.98	25.89 ± 0.47	35.49
DM-GAN	4.75 ± 0.07	16.09	30.49 ± 0.57	32.64
DF-GAN	5.10 ± 0.03	**14.81**	(–)	**21.42**
Ours	**5.37** ± 0.04	15.06	**35.26** ± 0.43	32.36

Table 2
Results of FID compared with state-of-the-art methods on the CelebA dataset.

Methods	COCOGAN	StyleGANv2	TransGAN	Ours
CelebA	5.74	5.59	5.28	**5.14**

Table 3
Results compared with state-of-the-art models on Params, Memory, GFLOP and FPS metrics.

Methods	Params(M)	Memory (MB)	GFLOPs	FPS
StackGAN	97.55	573.24	15.92	184
AttentionGAN	29.77	433.38	17.96	152
DM-GAN	87.20	456.07	33.93	57
DF-GAN	32.41	387.06	13.22	233
TransGAN	180.15	2785.28	102.62	8
Ours	99.20	952.96	30.73	86

The three state-of-the-art image synthesis methods [19,32,33] are compared on the CelebA dataset without textual information. we obtain the FID of these state-of-the-art models from the official pretrained models (Table 2). We compared our model and current state-of-the-art models. Compared with TransGAN [19], we drop the FID value from 5.28 to 5.14.

We also compared the relevant metrics of the proposed CT-GAN and other models when generating the image of 128 × 128 resolution by using the same setting (Table 3). It can be seen that under the number of parameters, the memory and calculation amount (GFLOPS) consumed by CT-GAN are much higher than StackGAN, which indirectly leads to the lag in terms of speed (FPS). The reason for this result is that the computation mode of Transformer and convolution is different. Compared with TransGAN, the same Transformer architecture, we have obvious advantages in each metrics, but there is still a lot of room to rise over the state-of-the-art convolution model.

The experimental results show that the generator based on the transformer architecture has more advantages in IS than that with the convolutional structure. This advantage is mainly reflected in the diversity of the generated images.

4.5. Visual quality

For qualitative evaluation, Fig. 8 shows an example of text-to-image synthesis generated using the proposed CT-GAN and state-of-the-art models. Generally speaking, compared with convolution model, our CT-GAN synthesizes the semantic information of images more accurately in most cases, because we use the generator of pure Transformer. The self-Attention mechanism in Transformer allows long-distance pixels or words to contact directly, which makes it easier for the model to learn the long-distance dependence of sequences, which is essential for synthesizing images from texts.

It can be seen that the visual effect of the proposed CT-GAN on single-category CUB dataset is not inferior to that of the two most advanced convolution models, even superior to them in some scenes. For example, the transition of feather color is more natural and rich (columns 2, 4, and 7). Owing to the generator of transformer architecture and the use of SSA, things can be portrayed from a larger receptive field. For example, the description and integration of the background

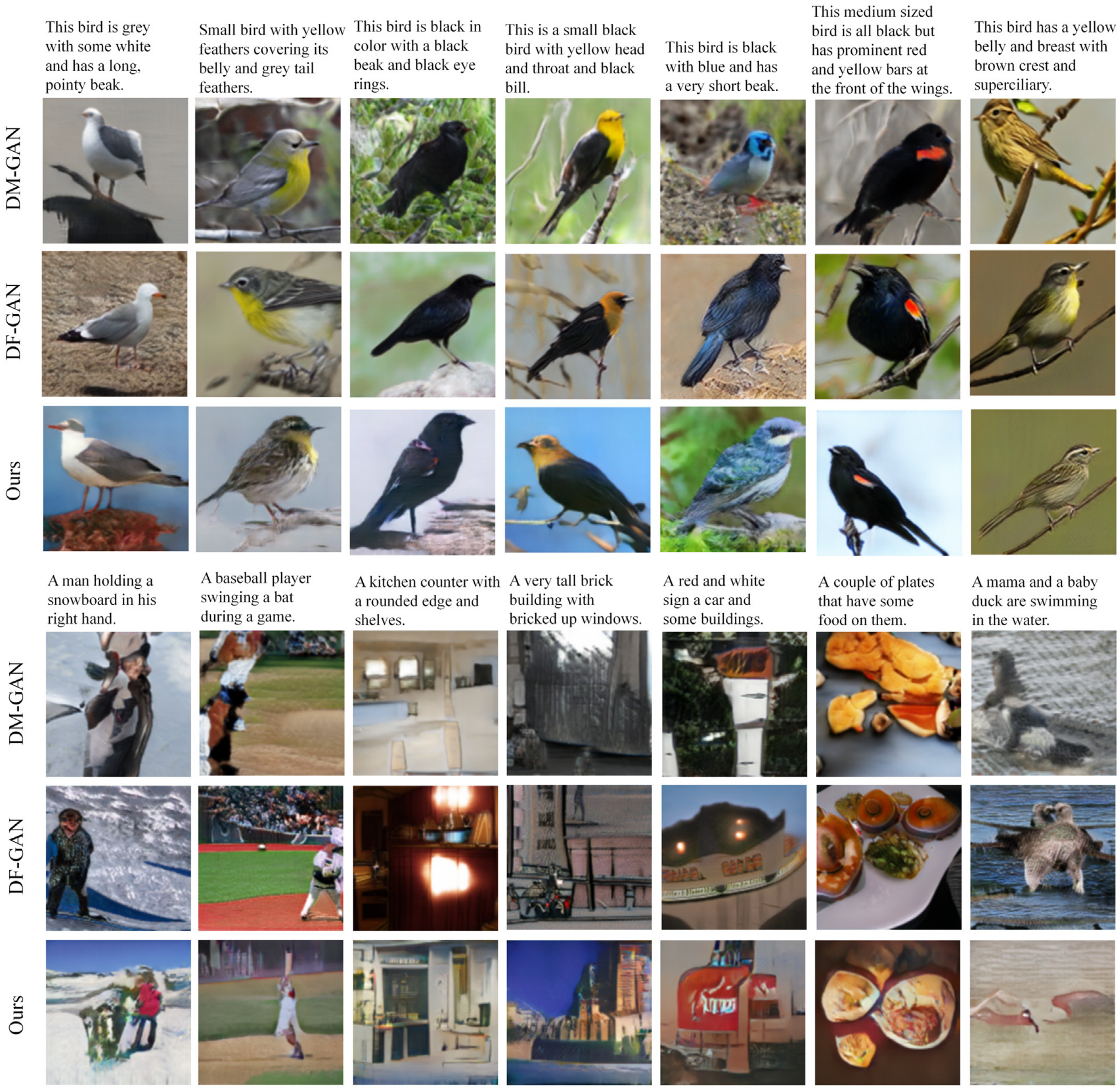

This bird is grey with some white and has a long, pointy beak.
Small bird with yellow feathers covering its belly and grey tail feathers.
This bird is black in color with a black beak and black eye rings.
This is a small black bird with yellow head and throat and black bill.
This bird is black with blue and has a very short beak.
This medium sized bird is all black but has prominent red and yellow bars at the front of the wings.
This bird has a yellow belly and breast with brown crest and superciliary.
DM-GAN
DF-GAN
Ours
A man holding a snowboard in his right hand.
A baseball player swinging a bat during a game.
A kitchen counter with a rounded edge and shelves.
A very tall brick building with bricked up windows.
A red and white sign a car and some buildings.
A couple of plates that have some food on them.
A mama and a baby duck are swimming in the water.
DM-GAN
DF-GAN
Ours

Fig. 8. Examples of images synthesized using DM-GAN [7], DF-GAN [8], and our proposed CT-GAN conditioned on text descriptions from the test set of CUB and COCO dataset.

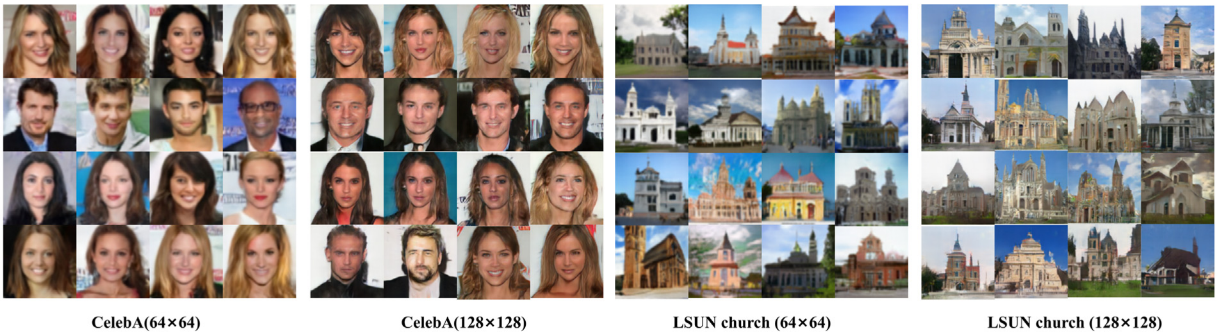


Fig. 9. Examples of images synthesized our proposed CT-GAN on CelebA and LSUN church datasets.

are purer and more natural (columns 1, 2, 3, 4, 6, and 7). There are no words related to the background in the text description. Meanwhile, the convolution-based model randomly produces more backgrounds, and the background of our model is simpler. This indicates that the transformer can grasp the key information in processing text information, learn the mapping relationship between text and image, and synthesize images from the text description perspective. We also conducted experiments on more challenging COCO dataset. The experimental results show that the proposed model is inferior to convolution-based models in detail control of multi-category datasets but superior to them in

Table 4
Performance of different architectures of the proposed CT-GAN on CUB and COCO datasets. Grid self-attention (Fig. 5(b)) technology is used as the baseline. GC and SSA represent GridCNN and shift self-attention, respectively.

Architecture	CUB	
	IS	FID
Baseline	5.04	24.36
Baseline+Filter	5.17	20.17
Baseline+GC	5.16	22.16
Baseline+SSA	5.33	18.13
Baseline+Filter+SSA+GC	**5.37**	**15.06**

semantic consistency and text understanding. For example, in the first column, DM-GAN [7] and DF-GAN [8] focus on depicting characters, ignoring the information about "snowboard" and our model captures this information, indicating that the transformer model is superior to convolution-based models in semantic image consistency.

Fig. 9 shows an example of image synthesis using the proposed CT-GAN without textual information. Compared with the most advanced TransGAN [19] based on transformer architecture, our advantage lies in the multi-stage method and the self-attention mode based on a sliding grid. The multi-stage method can synthesize images step by step, reducing the difficulty of synthesizing images, while sliding grid can increase the information communication between different grids and improve the overall quality of images.

It can be seen that the visual effect of the proposed CT-GAN on CelebA and LSUN church datasets, We can observe the skin color, hair color, expression and other details from the face images, and different architectural styles can be observed in the architectural images, all of which show pleasant visual details and diversity.

4.6. Ablation study

To further evaluate the effectiveness of the proposed components, we perform an ablation study, add these components to the base method separately, and report their IS and FID scores on the CUB dataset.

We define a baseline model, which excludes filter, GridCNN (GC), and SSA but uses a grid self-attention (Fig. 5(b)). To evaluate the effectiveness of these components separately, we add these components to baseline for comparison. As presented in Table 4, IS and FID obtained using the baseline model are 5.04 and 24.36, respectively. Compared with the baseline model, the filter improves the IS score from 5.04 to 5.17, the FID reduces from 24.36 to 20.17, and the GC improves the IS score from 24.36 to 20.17. The score increases from 5.04 to 5.16, and the FID decreases from 24.36 to 22.16.SSA increases the IS score from 5.04 to 5.33, and the FID decreases from 24.36 to 18.13. It can be seen that each component plays a corresponding role in the model, especially the filter, and SSA components play a key role. Finally, we combine all components to improve the IS score from 5.04 to 5.37 and reduce FID from 24.36 to 15.06.

In order to show the role of the filter more intuitively, we use the same text description to generate 5 images under the action of different noises. As shown in Fig. 10, the filter scores them according to the text description, and it can be seen that the images with higher recognition get higher scores, and the images with lower recognition get lower scores. When the images of the first stage are sent to the second stage, it can be seen that the images with high scores in the first stage finally produce higher image quality. This is enough to prove that when the filter is strong enough and the sample is generated enough, the bad image rate of the final generated image drops to very low, and this method can be extended to almost all multistage models.

Moreover, we also tested the convergence speed of the model in local and global discrimination modes. As shown in Fig. 11, in the 10th step at the beginning of the training, the contour of the face was captured by the generator in the local discrimination mode, but there was no change in the global discrimination mode. After a epoch of training, it can be observed that the quality of the synthesized picture in the global discrimination mode lags behind that in the local discrimination mode, which also shows that the convergence speed of the model is faster in the local discrimination mode.

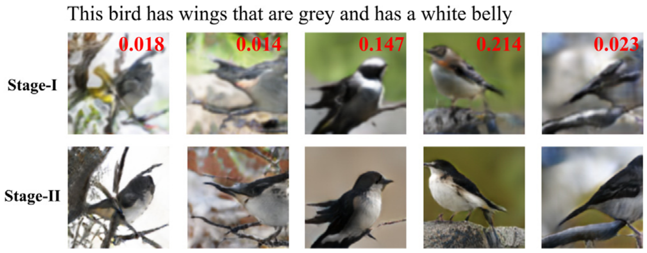


Fig. 10. The filter scores the images generated under different noise for the same text description, with the red font as the obtained score.

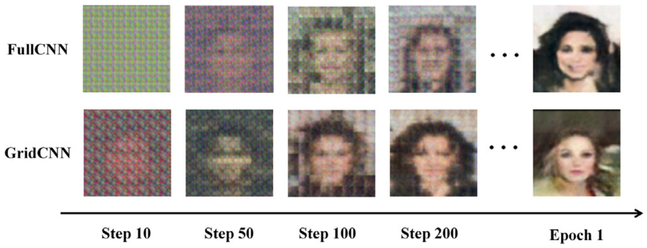


Fig. 11. The first stage generator visualizes the pre-training effect in two modes of global discrimination (FullCNN) and local discrimination (GridCNN).

5. Conclusions

In this study, we developed a new architecture called CT-GAN and applied the GAN generator based on the transformer architecture to the text-to-image tasks. We also propose a filtering module that can filter out low-quality images. Additionally, we design the generator and discriminator based on symmetry. In the generator, we propose an SSA technology to establish information communication between grids. We used two discrimination modes in the discriminator, local and global discriminations, to balance the performance of the generator and discriminator. Experimental results on several real datasets show that the proposed CT-GAN is superior to the most advanced convolution model in generating diversity and semantic consistency. In the future, we will try to choose more effective filtering modules and synthesize higher resolution images.

CRediT authorship contribution statement

Xin Zhang: Writing - Review, Editing. **Wentao Jiao:** Writing - Original Draft, Software. **Bing Wang:** Conceptualization, Methodology, Software. **Xuedong Tian:** Writing - Review, Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research work is partially supported by the National Natural Science Foundation of China (Grant No. 61375075), the Natural Science Foundation of Hebei Province of China (Grant No. F2019201329), the Special Project of Science and Technology Department of Hebei Province (Grant No. 2021H060306). The authors are grateful to the clinicians in Hebei University Affiliated Hospital and Hebei Key Laboratory of Machine Learning and Computational Intelligence for their support in providing data and clinical advice.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).

[2] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, 2016, arXiv preprint arXiv:1701.00160.

[3] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, arXiv preprint arXiv:1411.1784.

[4] S. Frolov, T. Hinz, F. Raue, J. Hees, A. Dengel, Adversarial text-to-image synthesis: A review, Neural Netw. 144 (2021) 187–209.

[5] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, S. Belongie, Stacked generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5077–5086.

[6] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D.N. Metaxas, Stackgan++: Realistic image synthesis with stacked generative adversarial networks, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2018) 1947–1962.

[7] M. Zhu, P. Pan, W. Chen, Y. Yang, Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5802–5810.

[8] M. Tao, H. Tang, S. Wu, N. Sebe, X.-Y. Jing, F. Wu, B. Bao, Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis, 2020, arXiv preprint arXiv:2008.05865.

[9] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).

[10] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: LSTM cells and network architectures, Neural Comput. 31 (7) (2019) 1235–1270.

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[13] K. Lagler, M. Schindelegger, J. Böhm, H. Krásná, T. Nilsson, GPT2: Empirical slant delay model for radio space geodetic techniques, Geophys. Res. Lett. 40 (6) (2013) 1069–1073.

[14] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-attention generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 7354–7363.

[15] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803.

[16] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, 2020, arXiv preprint arXiv:2010.04159.

[17] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, Z. Liu, Mobile-former: Bridging mobilenet and transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5270–5279.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[19] Y. Jiang, S. Chang, Z. Wang, Transgan: Two transformers can make one strong gan, 2021, arXiv preprint arXiv:2102.07074, 1.

[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[21] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, I.B. Ayed, Boundary loss for highly unbalanced segmentation, in: International Conference on Medical Imaging with Deep Learning, PMLR, 2019, pp. 285–296.

[22] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200–2011 dataset, 2011.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[24] Z. Liu, P. Luo, X. Wang, X. Tang, Large-scale celebfaces attributes (celeba) dataset, 2018, p. 11, Retrieved August 15.

[25] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, J. Xiao, Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2015, arXiv preprint arXiv:1506.03365.

[26] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, X. He, Attngan: Fine-grained text to image generation with attentional generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1316–1324.

[27] R. Durall, S. Frolov, J. Hees, F. Raue, F.-J. Pfreundt, A. Dengel, J. Keuper, Combining transformer generators with convolutional discriminators, in: German Conference on Artificial Intelligence (KÜNstliche Intelligenz), Springer, 2021, pp. 67–79.

[28] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw. 18 (5–6) (2005) 602–610.

[29] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), 2016, arXiv preprint arXiv:1606.08415.

[30] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, Adv. Neural Inf. Process. Syst. 29 (2016).

[31] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, Adv. Neural Inf. Process. Syst. 30 (2017).

[32] C.H. Lin, C.-C. Chang, Y.-S. Chen, D.-C. Juan, W. Wei, H.-T. Chen, COCO-GAN: Conditional coordinate generative adversarial network, 2018.

[33] Y. Viazovetskyi, V. Ivashkin, E. Kashin, Stylegan2 distillation for feed-forward image manipulation, in: European Conference on Computer Vision, Springer, 2020, pp. 170–186.