SCIENCE CHINA
Information Sciences

February 2021, Vol. 64 120102:1–120102:12
https://doi.org/10.1007/s11432-020-2900-x

• RESEARCH PAPER •
Special Focus on Deep Learning for Computer Vision

Learning efficient text-to-image synthesis via interstage cross-sample similarity distillation

Fengling MAO[1,2], Bingpeng MA[3*], Hong CHANG[2,3],
Shiguang SHAN[2,3,4] & Xilin CHEN[2,3]

[1]*School of Information Science and Technology, ShanghaiTech University, Shanghai* 201210*, China;*
[2] *Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190*, China;*
[3]*University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[4]*CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai* 200031*, China*

Received 21 January 2020/Revised 8 March 2020/Accepted 26 April 2020/Published online 17 November 2020

Abstract For a given text, previous text-to-image synthesis methods commonly utilize a multistage generation model to produce images with high resolution in a coarse-to-fine manner. However, these methods ignore the interaction among stages, and they do not constrain the consistent cross-sample relations of images generated in different stages. These deficiencies result in inefficient generation and discrimination. In this study, we propose an interstage cross-sample similarity distillation model based on a generative adversarial network (GAN) for learning efficient text-to-image synthesis. To strengthen the interaction among stages, we achieve interstage knowledge distillation from the refined stage to the coarse stages with novel interstage cross-sample similarity distillation blocks. To enhance the constraint on the cross-sample relations of the images generated at different stages, we conduct cross-sample similarity distillation among the stages. Extensive experiments on the Oxford-102 and Caltech-UCSD Birds-200-2011 (CUB) datasets show that our model generates visually pleasing images and achieves quantitatively comparable performance with state-of-the-art methods.

Keywords generative adversarial network (GAN), text-to-image synthesis, knowledge distillation

Citation Mao F L, Ma B P, Chang H, et al. Learning efficient text-to-image synthesis via interstage cross-sample similarity distillation. Sci China Inf Sci, 2021, 64(2): 120102, https://doi.org/10.1007/s11432-020-2900-x

1 Introduction

Image generation [1–3] has achieved remarkable progress owing to the flourishing development of deep learning. Many applications of image generation [4–11], such as style-transfer [5], video generation [6], image-to-image translation [8,9], image inpainting [7], and text-to-image synthesis [12–17], have attracted increasing attention. For a given text, the text-to-image synthesis task aims at producing images that are of high quality and semantically consistent with the given text.

Serveral methods [12–17] for text-to-image synthesis have been proposed. Reed et al. [12] proposed the classic single-stage generative adversarial network (GAN) framework based on conditional deep convolutional GAN (DCGAN) [18]. Subsequent multistage GAN models [13–15, 17] generate images with a 256×256 resolution conditioned on the given text in a coarse-to-fine manner. These models first generate an initial image from noise and text. In the following stages, they utilize the image from the previous stage and the text condition to produce relatively fine-grained images. Although much progress has been made in this area, problems still exist in available methods. Specifically, two unsolved problems hinder efficient text-to-image synthesis.

One problem is that these methods do not generally consider interactions and the full transfer of useful information among stages. They generate images with coarse-to-fine quality stage-by-stage. In the last stage, the refined images contain vivid visual and semantic information, which is vital for discriminators

[*] Corresponding author (email: bpma@ucas.ac.cn)

to distinguish real images from generated ones. However, some information is lacking in the coarse images generated in the initial stages. Such shortcoming may cause inaccurate discrimination and then adversely affect the image generation in these stages. Therefore, previous methods may conduct inefficient text-to-image synthesis.

The other problem is that these methods do not enforce the consistent cross-sample relations of images generated in different stages. Intuitively, these images are generated in a coarse-to-fine manner. They differ in terms of resolution, with local details ranging from coarse to refined. Meanwhile, the cross-sample relations of these images should remain the consistent. Under this circumstance, the relation among a batch of coarse images should be consistent with that of the refined images. However, existing methods do not constrain the required consistency of cross-sample relations, which may thus cause unstable network training.

In this study, we propose an interstage cross-sample similarity distillation GAN (ICSD-GAN), which can transfer useful knowledge and constrain the consistency of cross-sample relations for multistage text-to-image generation. Motivated by knowledge distillation [19], which can conduct the transfer of knowledge from a teacher network to a student network, we propose interstage knowledge distillation to strengthen interactions and the full transfer of useful information among stages. With novel cross-sample similarity distillation (CSD) blocks, we achieve efficient interstage interaction and information transfer. To enhance the constraint on cross-sample relations, we adopt knowledge distillation of the cross-sample similarity in the CSD blocks. We conducts efficient text-to-image synthesis with ICSD-GAN.

Thorough experiments on the Oxford-102 [20] and Caltech-UCSD Birds-200-2011 (CUB) [21] datasets validate the effectiveness of our model. We achieved excellent visualization and generalizability performance. Our results are comparable to those of state-of-the-art methods in terms of the evaluation metrics of common inception score [22] and Fréchet inception distance (FID) [23].

2 Related work

2.1 Generative adversarial networks

GAN [2] has been widely explored recently. It learns the mapping from a random noise distribution to a realistic image distribution with a generator and a discriminator, both of which are trained in an adversarial way. Followed by the conditional version, the conditional GAN [24] generates images conditioned on class labels. DCGAN [18] is one of the most classical models in the GAN family. Built upon these, many applications of GAN have also become research hotspots [5–9, 17, 25–27]; examples include style-transfer [5], video generation [6], image-to-image translation [8, 9, 26], person image generation [27], image inpainting [7], and text-to-image synthesis [17]. In this study, we focus on the text-to-image synthesis task and aim to improve the quality of generated images and strengthen their semantic consistency with a given text.

2.2 Text-to-image synthesis

As a result of the rapid development of computer vision [2, 18, 24, 28, 29], methods based on the GAN model have been widely explored for text-to-image synthesis tasks. Reed et al. [12] made initial attempts to conduct text-to-image synthesis with the conditional DCGAN model. Conditioned on the text embeddings, this model outputs vivid images with a 64 × 64 resolution. Note that text embeddings can be extracted from models of cross-modal matching [30, 31]. Furthermore, Zhang et al. [13] proposed a two-stage StackGAN model to synthesize images of high resolution conditioned on the input text in a coarse-to-fine manner. The first stage generates 64 × 64 initial images conditioned on the input text, and the second stage inputs the given text and the initial images and outputs 256 × 256 refined images. StackGANv2 [14] adopted a multistage model, which divides the generation process into several sophisticated steps. It consists of multiple generators and discriminators, and the multiply generators are organized in a tree-like structure to share the most parameters. Similar to the multistage model, the HDGAN model [16] consists of a single stream generator and hierarchically-nested discriminators. Despite their remarkable improvements, these models are conditioned on global sentence embeddings extracted from the char CNN-RNN text encoder [30] without fully utilizing the semantic information at the word level. To solve this issue, Xu et al. [15] proposed AttnGAN to introduce attention between local image region features and word embeddings in the input layer of succeeding stages. Moreover, Xu et al. extracted text embeddings with a deep attentional multimodal similarity model, which obtains visual-relevant text features. Qiao et al. [32] added sentence-level attention based on AttnGAN and proposed a text-to-image-to-text model for semantic preservation. Qiao et al. [33] further proposed co-embedding model of prior visual and layout knowledge, which includes a text-image encoder and a text-mask encoder. The multi prior text embeddings extracted from the co-embedding model are aggregated as the condition of cascade attentive generation networks to produce images. Li et al. [34] divided the process of text-to-image synthesis into two steps: layout generation and image generation. They proposed the object-driven attentive image generator to generate salient objects from layout and text. Compared with the above methods, the method proposed in this study pays more attention to the self-learning of networks. We conduct coarse-to-fine generation and transfer abundant knowledge from the features of high-resolution images to those of low-resolution images.

2.3 Knowledge distillation

Hinton et al. [19] first proposed the concept of knowledge distillation as a model compression method, which learns a compact student network from a complicated teacher network without much performance loss. It tries to transfer the prediction knowledge of the teacher network to the student network. Built upon this method, subsequent studies [35–41] adopted other types of knowledge distillation. Some studies [36,37,39] matched the intermediate features between teacher networks and student networks. Paying attention to the important regions, Zagoruyko et al. [38] conducted knowledge transfer via the attention map of intermediate layers. Huang et al. [35] transferred the knowledge of neuron selectivity feature distributions. Chen et al. [41] proposed the DarkRank method to distill the knowledge of cross-sample similarity. Yuan et al. [40] proposed a symmetrical distillation network, which consists of a source network as the discriminator and a target network as the generator. The feature knowledge is transferred from each layer of the discriminator to that of the generator. These existing methods conduct knowledge distillation with two separate models, whereas our proposed method distills the knowledge from generated high-resolution generated images to low-resolution ones in different stages of the same multistage generation network.

3 ICSD-GAN

In this section, we will first introduce the overall architecture of the proposed model. Second, we describe the interstage cross-sample similarity distillation method in detail. Finally, we introduce the objective functions and the training algorithm of our model.

3.1 Overall architecture

Previous methods [13–15,17] do not consider the information transfer among stages and the constraint on the consistent cross-sample relations of generated images in different stages. To address these problems, we consider enhancing information transfer via interstage knowledge distillation and by enforcing consistent cross-sample similarity among stages to learn efficient text-to-image generation.

In this study, we propose the ICSD-GAN model. Figure 1 shows the detailed architecture of our model. The whole generation is divided into three stages (i.e., Stage^0, Stage^1 and Stage^2) in a coarse-to-fine manner. Stage^i, $i \in [0, 1, 2]$ consists of an attention-modulated generator G_i and a similarity-aware discriminator D_i [17].

G_i inputs the text information $t = [t_s, t_w]$, noise z (in G_0), and previous image feature (in G_1, G_2). The corresponding generated image is denoted as I_i. t_s is the sentence feature for all the stages, and t_w is the word feature for the last two stages. Table 1 [17,42] shows the structure of our multistage generators. Some attention-modulation (AM) blocks are applied to the intermediate layers of the generators to modulate the visual features of these layers with the given text.

In the similarity-aware discriminator D_i, the generated image I_i is downsampled to a high-level feature map $F_i \in \mathbb{R}^{N\times C\times H\times W}$ with several convolutional blocks. N, C, H, and W denote the batch size, number of channels, and the height and width of the feature map, respectively. The discrimination block discriminates not only True/False, but also the similarity between images and the given text information. It outputs the True/False score and the cross-modal similarity score, which is denoted as d. One can

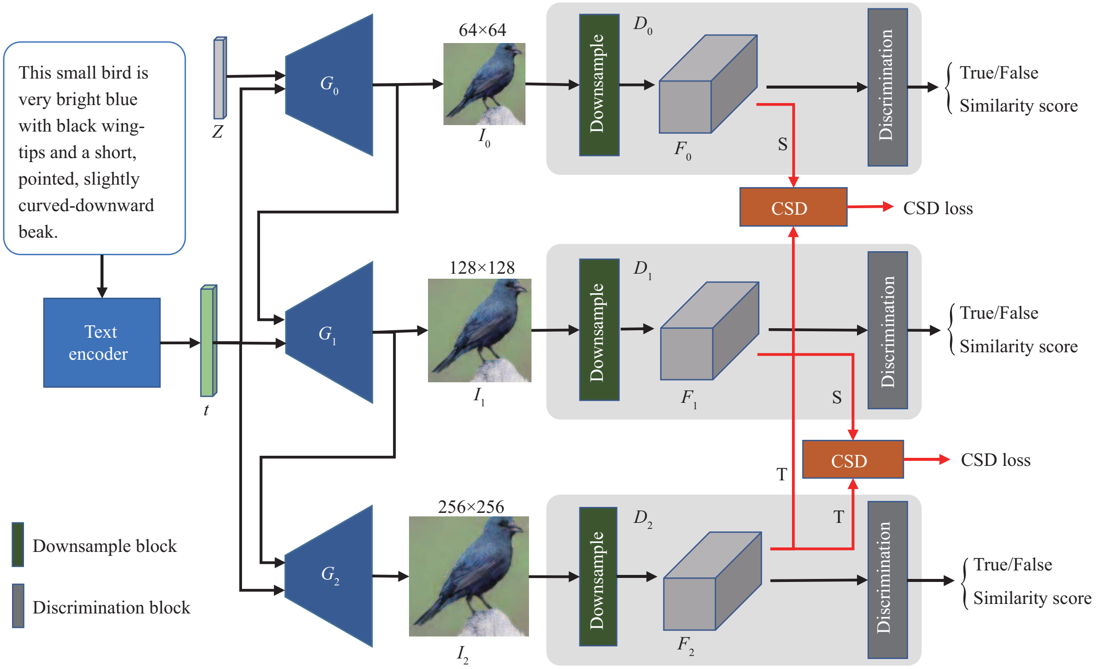

Figure 1 (Color online) Architecture of the proposed ICSD-GAN model. The whole network contains three stages (i.e., Stage^0, Stage^1 and Stage^2) and generates images with 64 × 64, 128 × 128, and 256 × 256 resolutions. G_i and D_i are the generator and discriminator for Stage^i, respectively. We conduct interstage knowledge distillation between Stage^2 and Stage^0/Stage^1 via the CSD block. On the red line, "T" means the teacher branch, and "S" means the student branch.

Table 1 Structure of generators$^{a)}$

G_0	G_1/G_2
Concat, FC, BN, GLU, Reshape	Attn, Concat
UpSample(2), Conv($3 \times 3/1$), BN, GLU}×4	AM block } × N_{AM}
Conv($3 \times 3/1$), Tanh	UpSample(2), Conv($3 \times 3/1$), BN, GLU
	Conv($3 \times 3/1$), Tanh

a) AM block denotes the attention-modulation block [17]. GLU denotes the gated linear units layer [42]. UpSample(2) means that the upsampling stride is 2. Conv($3 \times 3/1$) means that kernel size is 3 and stride is 1 for the convolutional layer.

Table 2 Structure of Downsample blocks$^{a)}$

Downsample block in D_0	Downsample block in D_1	Downsample block in D_2
Conv($4 \times 4/2$), LeakyReLU	Conv($4 \times 4/2$), LeakyReLU	Conv($4 \times 4/2$), LeakyReLU
Conv($4 \times 4/2$), BN, LeakyReLU} ×3	Conv($4 \times 4/2$), BN, LeakyReLU} ×4	Conv($4 \times 4/2$), BN, LeakyReLU} ×5
	Conv($3 \times 3/1$), BN, LeakyReLU	Conv($3 \times 3/1$), BN, LeakyReLU} ×2

a) Conv($4 \times 4/2$) means that kernel size is 3 and stride is 1 for the convolutional layer.

refer to MS-GAN [17] for further details about the discrimination block. Table 2 shows the structure of the downsample blocks in D_0, D_1, and D_2.

Through multistage generation, the model produces images with progressively high quality. The refined image I_2 contains more vivid details than the coarse images I_0 and I_1. To boost the performance of our model, we propose an interstage cross-sample similarity distillation method with CSD blocks among stages. This method transfers the knowledge of I_2 to I_0 and I_1. Moreover, it constrains the cross-sample relation of these images. As shown in Figure 1, we consider the Stage^2 model as the teacher network and the previous stage models as the student networks. We conduct knowledge distillation between the teacher network and the student networks via the CSD blocks. In this way, we generate coarse-to-fine images from I_0 to I_2, and then utilize the fine-grained image I_2 to guide the previous stages in improving the training in the next iteration. Therefore, we can improve the interaction and information transfer among stages to learn efficient text-to-image generation. Furthermore, we choose to distill the cross-sample similarity in the CSD blocks to enforce the consistency of the cross-sample relations and stabilize the training of our multistage model. We will subsequently present the proposed CSD method in detail.

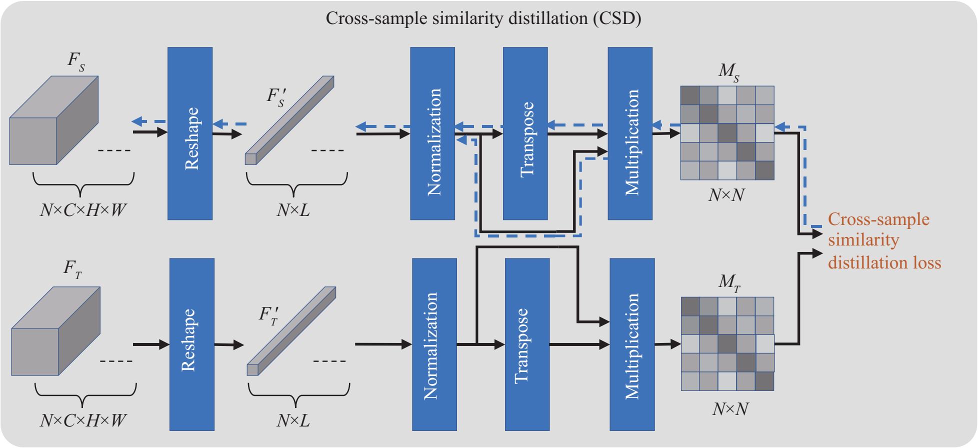

Figure 2 (Color online) Detailed structure of CSD block. This block takes student feature F_S and teacher feature F_T as input, and outputs the CSD loss. In this figure, the black solid line denotes forward-propagation and the blue dotted line denotes backpropagation. We optimize the output loss without backpropagation to the teacher branch.

3.2 Interstage cross-sample similarity distillation

We propose the interstage cross-sample similarity distillation method to transfer knowledge from refined images to coarse images. Specifically, we propose the CSD block and adopt knowledge distillation of cross-sample similarity. This approach provides a way to transfer the knowledge among stages, while constraining the consistency of the cross-sample relations of the images generated in different stages.

Figure 2 shows the detailed structure of the CSD block. The CSD block takes the feature map $F_S \in \mathbb{R}^{N\times C\times H\times W}$ and $F_T \in \mathbb{R}^{N\times C\times H\times W}$ as input. It outputs the CSD loss. F_S is the feature map extracted from image I_S by a student network. F_T is the feature map extracted from image I_T by a teacher network. As mentioned previously, we regard the Stage^2 model as the teacher network, and the previous stage models as the student networks. Therefore, in this study, F_T can be F_2. F_S can be F_0 and F_1.

Before distillation, the feature map F_S is reshaped into $F'_S \in \mathbb{R}^{N\times L}$, where $L = C \times H \times W$. In the knowledge distillation of the cross-sample similarity process, we consider the distillation of the cross-sample similarity matrix inspired by Gu et al. [39]. As shown in Figure 2, we normalize the feature for each sample in F'_S with a normalization layer. The cross-sample similarity matrix M_S is computed by matrix multiplication between the normalized F'_S and its transposed matrix. M_S can be formulated as

$$M_S = (m_{ij}) \in \mathbb{R}^{N\times N}, \quad m_{ij} = \frac{F'_{S,i} \cdot F'^{\mathrm{T}}_{S,j}}{||F'_{S,i}|| \cdot ||F'_{S,j}||}, \tag{1}$$

where m_{ij} denotes the cosine similarity between the feature of i-th and j-th samples. We get the cross-sample similarity matrix M_T for F_T in the same way. To constrain the consistency of cross-sample relations, we enforce the cross-sample similarity matrix M_S to be close to M_T. Therefore, the CSD loss between I_S and I_T can be formulated as follows:

$$L_{\mathrm{CSD}}(I_S, I_T) = \frac{1}{N^2}||M_S - M_T||_F^2. \tag{2}$$

3.3 Objective functions

In our model, each discriminator $D_i, i \in [0, 1, 2]$ outputs both the discrimination of True/False and the similarity, which is computed between images and the given text information. With these discriminations, the discriminator and generator loss in Stage^i can be formulated as follows:

$$L_{D_i} = L_{D_i}^{\mathrm{TF}} + \lambda L_{D_i}^{\mathrm{ASL}}, \qquad L_{G_i} = L_{G_i}^{\mathrm{TF}} + \lambda L_{G_i}^{\mathrm{ASL}}, \tag{3}$$

where $L_{D_i}^{\mathrm{TF}}$ and $L_{G_i}^{\mathrm{TF}}$ denote the losses for True/False discrimination. $L_{D_i}^{\mathrm{ASL}}$ and $L_{G_i}^{\mathrm{ASL}}$ denote the losses for the similarity discrimination with coefficient λ. Moreover, we propose interstage cross-sample similarity distillation in this paper. The corresponding distillation loss is denoted as L_{CSD}. The full loss of our multi-stage discriminator and generator can be formulated as follows:

$$L_D = \sum_{i=0}^{2} L_{D_i}, \quad L_G = \sum_{i=0}^{2} L_{G_i} + \lambda_1 L_{\mathrm{DAMSM}} + L_{\mathrm{CA}} + L_{\mathrm{CSD}}, \tag{4}$$

where L_{DAMSM} is the deep attentional multimodal similarity model (DAMSM) loss [15] with the coefficient λ_1. This loss is fine-grained image-text matching loss from the DAMSM [15]. L_{CA} is the loss for conditional augmentation [13]. Zhang et al. [13] augmented the text space by resampling the input sentence feature and added regularization on the standard Gaussian distribution and the conditioning Gaussian distribution $L_{\mathrm{CA}} = D_{\mathrm{KL}}\left(\mathcal{N}\left(\mu\left(t\right), \Sigma\left(t\right)\right) || \mathcal{N}\left(0, I\right)\right)$.

Discrimination of True/False. For discrimination of True/False, we adopt the widely used conditional and unconditional GAN loss for D_i, which can be formulated as

$$\begin{aligned} L_{D_i}^{\mathrm{TF}} = & -\mathbb{E}_{I_i^r \sim p_{r_i}}\left[\log D_i\left(I_i^r, t\right)\right] - \mathbb{E}_{I_i^g \sim p_{g_i}}\left[\log\left(1 - D_i\left(I_i^g, t\right)\right)\right] \\ & - \mathbb{E}_{I_i^r \sim p_{r_i}}\left[\log D_i\left(I_i^r\right)\right] - \mathbb{E}_{I_i^g \sim p_{g_i}}\left[\log\left(1 - D_i\left(I_i^g\right)\right)\right]. \end{aligned} \tag{5}$$

The first two terms denote the conditional component of $L_{D_i}^{\mathrm{TF}}$. The last two terms denote the unconditional component of $L_{D_i}^{\mathrm{TF}}$. p_{r_i} and p_{g_i} denote the real data distribution and generated data distribution in Stage^i, respectively. The corresponding conditional and unconditional GAN loss for G_i are

$$L_{G_i}^{\mathrm{TF}} = -\mathbb{E}_{I_i^g \sim p_{g_i}}\left[\log D_i\left(I_i^g, t\right)\right] - \mathbb{E}_{I_i^g \sim p_{g_i}}\left[\log D_i\left(I_i^g\right)\right]. \tag{6}$$

Discrimination of similarity. For discrimination of the similarity between images and the given text information, we utilize the adversarial similarity loss [17], which can enforce the model to generate images more semantically consistent with the given text. The adversarial similarity losses for D_i and G_i are as follows:

$$\begin{aligned} & L_{D_i}^{\mathrm{ASL}} = \mathbb{E}_{t \sim p_r}[\max\left(0, 1 - d\left(t, f_{\mathrm{im}}^{\mathrm{r}}\right)\right) + \left(\max\left(0, 1 + d\left(t, f_{\mathrm{im}}^{\mathrm{w}}\right)\right) + \max\left(0, 1 + d\left(t, f_{\mathrm{im}}^{\mathrm{g}}\right)\right)\right)/2], \\ & L_{G_i}^{\mathrm{ASL}} = -\mathbb{E}_{t \sim p_r}\left[d\left(t, f_{\mathrm{im}}^{\mathrm{g}}\right)\right], \end{aligned} \tag{7}$$

where t is the text embeddings. $f_{\mathrm{im}}^{\mathrm{r}}$, $f_{\mathrm{im}}^{\mathrm{w}}$ and $f_{\mathrm{im}}^{\mathrm{g}}$ are the features of real, wrong and generated images respectively. d is the the similarity score provided by the discriminators. For t and a visual feature f_{im}, the similarity score can be computed as

$$d\left(t, f_{\mathrm{im}}\right) = \left(\left(W_t t\right)^{\mathrm{T}}\left(W_{\mathrm{im}} f_{\mathrm{im}}\right)\right) / \left(||W_t t|| \cdot ||W_{\mathrm{im}} f_{\mathrm{im}}||\right), \tag{8}$$

where t and f_{im} are projected into the common feature space with W_t and W_{im}. By minimizing $L_{D_i}^{\mathrm{ASL}}$, the discriminator tries to increase the similarity between the text and real images, while decreasing that of the wrong and generated images. By minimizing $L_{G_i}^{\mathrm{ASL}}$, the generators adversarially increase the similarity between the text and generated images.

Distillation loss. Furthermore, we propose to conduct interstage knowledge distillation to transfer useful information of I_2 to I_0 and I_1. To constrain the cross-sample relation at the same time, we adopt the interstage cross-sample similarity distillation. Therefore, we add two extra loss term $L_{\mathrm{CSD}}(I_0, I_2)$ and $L_{\mathrm{CSD}}(I_1, I_2)$ to the generator loss. $L_{\mathrm{CSD}}(I_0, I_2)$ denotes the interstage cross-sample similarity distillation between I_0 and I_2. $L_{\mathrm{CSD}}(I_1, I_2)$ denotes the interstage cross-sample similarity distillation between I_1 and I_2. The total distillation loss is as follows:

$$L_{\mathrm{CSD}} = L_{\mathrm{CSD}}(I_0, I_2) + L_{\mathrm{CSD}}(I_1, I_2). \tag{9}$$

During training, we first optimize the discriminator stage-by-stage. We subsequently optimize the whole generator in an adversarial way. Adam optimizer [43] is the default optimizer in this paper. The training process of our proposed ICSD-GAN model is shown in Algorithm 1. Note that we optimize the loss $L_{\mathrm{CSD}}(\cdot, \cdot)$ without back-propagation in the teacher network (Stage^2). During testing, the generator takes random noise and the test text embedding as input. It produces images of 64 × 64, 128 × 128 and 256 × 256 resolution.

Table 3 The details of CUB and Oxford-102 datasets

Dataset	#images	Captions per image	#categories	#train categories	#test categories
CUB [21]	11788	10	200	150	50
Oxford-102 [20]	8189	10	102	82	20

Algorithm 1 ICSD-GAN training algorithm

```
Require: Training set, number of stages $n_{\text{stage}} = 3$, number of training iterations $T$, batch size $N$, learning rate of generator $\alpha_g$, learning rate of discriminator $\alpha_d$, coefficient $\lambda$ and $\lambda_1$ for loss terms, Adam hyperparameters $\beta_1$ and $\beta_2$.
Ensure: Generator parameters $\theta_g$.
Initialize discriminator parameters $\theta_d = [\theta_{d_0}, \theta_{d_1}, \theta_{d_2}]$ and generator parameters $\theta_g = [\theta_{g_0}, \theta_{g_1}, \theta_{g_2}]$;
for $t = 1, \dots, T$ do
    Sample $x$, text $t$, mis-matching images $x_w$ and random noise $z$;
    $\hat{x} = G(z, t)$ ($\hat{x} = [\hat{x}_0 \in \mathbb{R}^{N\times3\times64\times64}, \hat{x}_1 \in \mathbb{R}^{N\times3\times128\times128}, \hat{x}_2 \in \mathbb{R}^{N\times3\times256\times256}]$);
    for $i = 0, \dots, 2$ do
        $L_{D_i} \leftarrow L_{D_i}^{\text{TF}} + \lambda L_{D_i}^{\text{ASL}}$;
        $\theta_{d_i} \leftarrow \text{Adam}(L_{D_i}, \theta_{d_i}, \alpha_d, \beta_1, \beta_2)$;
    end for
    $L_G \leftarrow \sum_{i=0}^{2} L_{G_i} + \lambda_1 L_{\text{DAMSM}} + L_{\text{CA}} + L_{\text{CSD}}(\hat{x}_0, \hat{x}_2) + L_{\text{CSD}}(\hat{x}_1, \hat{x}_2)$;
    $\theta_g \leftarrow \text{Adam}(L_G, \theta_g, \alpha_g, \beta_1, \beta_2)$;
end for
return $\theta_g$.
```

4 Experiments

In this section, we will first introduce the datasets and evaluation metrics, as well as the implementation details in our experiments. Secondly, we will show the quantitive results and visualization of generated images compared with state-of-the-art methods. Finally, we will analyze the performance of our ICSD-GAN model in detail.

4.1 Experimental setup

Datasets. We conduct experiments on CUB [21] and Oxford-102 datasets [20]. Table 3 [20, 21] reports the statistics of the two datasets. Following the previous studies [12–14, 16, 17], we split the datasets into train and test dataset with separate categories.

Evaluation metrics. For the equal comparison of the proposed method with other state-of-the-art methods, we choose the commonly used evaluation metrics of Inception Score [22] and FID [23] to measure the quality and diversity of the generated images. For the inception score, we input the generated images into a pretrained inception model to obtain the condition label distribution $p(y|x)$. Note that the pretrained inception model [44] is the same as that used in previous studies [12–17]; it is first pretrained on ImageNet [45] and then fine-tuned on the CUB and Oxford-102 datasets. The inception score is computed as $\exp(\mathbb{E}_x \text{KL}(p(y|x)||p(y)))$. Given one high-quality image, the probability that it belongs to a certain category may be extremely high. The conditional label distribution $p(y|x)$ may have a low entropy. Take for example generated images with high diversity, their distribution $p(y)$ may have high entropy. Therefore, we obtain a high inception score. We utilize the same evaluation code as that used in previous studies [13–16] to compute the inception score. For the FID, we input the generated images and real images into the feature extractor ϕ, i.e., the pretrained inception model [44]. $\phi(p_g)$ and $\phi(p_r)$ are modeled as two multivariate Gaussian distributions $\mathcal{N}(\mu_g, C_g)$ and $\mathcal{N}(\mu_r, C_r)$, respectively. μ_g and μ_r are the mean of the feature distributions. C_g and C_r are the covariance. The distance is formulated as $\text{FID}(p_r, p_g) = ||\mu_r - \mu_g||_2^2 + \text{Tr}(C_r + C_g - 2(C_r C_g)^{1/2})$. We utilize the official implementation in TensorFlow [46] to compute the FID of 30000 generated images and real images. We evaluate our model with 256×256 generated images by default.

Implementation details. We adopt MS-GAN [17] as our baseline model for both datasets. To extract text embeddings, we utilize the pretrained bi-LSTM text encoder proposed by Xu et al. [15]. In our experiments, the generator outputs images with 64×64, 128×128, and 256×256 resolutions. The two datasets slightly differ in terms of the number of AM blocks N_{AM}, which is set to 2 for the Oxford-102 dataset and set to 1 for the CUB dataset. In each training iteration, we take turns in training D_0, D_1, and D_2. The generator is subsequently trained in an adversarial way. We employ the Adam optimizer [43] to train our network, with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. In all experiments, the learning rate is set to 0.0002 for the generator and discriminator. The batch size is set to 22. The maximum number of iterations is

Table 4 Quantitive comparison with state-of-the-art methods on CUB and Oxford-102 datasets

Method	CUB		Oxford-102	
	Inception score (↑)	FID (↓)	Inception score (↑)	FID (↓)
GAN_CLS_INT [12]	2.88 ± 0.04	68.79	2.66 ± 0.03	79.55
GANWN [47]	3.62 ± 0.07	67.22	–	–
StackGAN [13]	3.70 ± 0.04	51.89	3.20 ± 0.01	55.28
StackGAN-v2 [14]	4.04 ± 0.05	15.30	3.26 ± 0.01	48.68
HDGAN [16]	4.15 ± 0.05	18.23	3.45 ± 0.07	–
AttnGAN [15]	4.36 ± 0.03	10.65	3.75 ± 0.02	–
MirrorGAN [32]	4.56 ± 0.05	–	–	–
LeicaGAN [33]	4.62 ± 0.06	–	3.92 ± 0.02	–
MS-GAN [17]	4.56 ± 0.02	10.41	**3.95 ± 0.03**	36.24
ICSD-GAN	**4.66 ± 0.04**	**9.35**	3.87 ± 0.05	**32.64**

set to 800. We report the best performance across all iterations. The weight of the adversarial similarity loss is set to 1.0. λ is set to 1.0. λ_1 is set to 5.0.

4.2 Comparison results

For an equal quantitative comparison, we compare our model with GAN_CLS_INT [12], GANWN [47], StackGAN [13], StackGAN-v2 [14], HDGAN [16], AttnGAN [15], MirrorGAN [32], LeicaGAN [33], and MS-GAN [17]. Table 4 [12–17,32,33,47] reports the inception score and FID of these models. By default, the results of each model are based on the original works. However, the results of GAN_CLS_INT are given in the paper on StackGAN-v2. The inception score of AttnGAN for the Oxford-102 dataset is from the paper on LeicaGAN. The FID results of HDGAN and AttnGAN are from the paper on MS-GAN. For our model, we compute the inception score and FID in the same way as that used for the compared methods. Table 4 shows that among all compared methods, our model achieves the highest inception score and lowest FID on CUB and Oxford-102 datasets. Relative to our baseline, that is, MS-GAN, our model improves the inception score and FID on CUB dataset by 2% and 10%, respectively. On the Oxford-102 dataset, our model achieves comparable inception score and improves FID by 9.9%. The improvement of our model relative to the baseline model indicates that our model with interstage cross-sample similarity distillation achieves efficient text-to-image synthesis. Furthermore, the quantitative results demonstrate that our model generates images with higher quality in comparison with the other methods. Our model also outperforms other models for the CUB and Oxford-102 datasets.

For an equal qualitative comparison, we visualize the images generated by our model and those by StackGAN, HDGAN, and MS-GAN. Figure 3 shows the visualization results for the CUB and Oxford-102 datasets. Given a text, our model can produce impressive images with high quality and semantic consistency with the text. For example, given the text “this bird has wings that are grey and has a spotted body and red throat.”, StackGAN and HDGAN generate images that contain many artifacts and are semantically inconsistent with the text. MS-GAN generates a visually pleasing image, but the bird in the image does not have a spotted body as described in the text. Compared with these methods, our model generates a more photographic image that contains vivid details and is consistent with the text. Visualization results also demonstrate that our model learns efficient text-to-image synthesis. As a result, our model generates images high quality.

4.3 Performance analysis

In this subsection, we conduct several experiments to evaluate our model’s generalization ability, the performance of its different stages, and the performance of different variants.

Performance of generalizability. To evaluate the generalizability of our model, we modify the given text and generate images with the modified text on the CUB dataset and Oxford-102 datasets. In Figure 4, we show the images generated with each given text and the images with the corresponding modified text on the CUB dataset. Figure 4 shows the powerful generalizability of our model. Our model is capable of capturing changes in the given text and generating semantically consistent images. For example, in the first pair of results in Figure 4, “brown wings” in the original test text is modified to “blue wings”. Our model is sensitive to this change and proceeds to generate an image of a bird with “blue wings”. Figure 5 shows that our model also performs well for the Oxford-102 dataset when the test

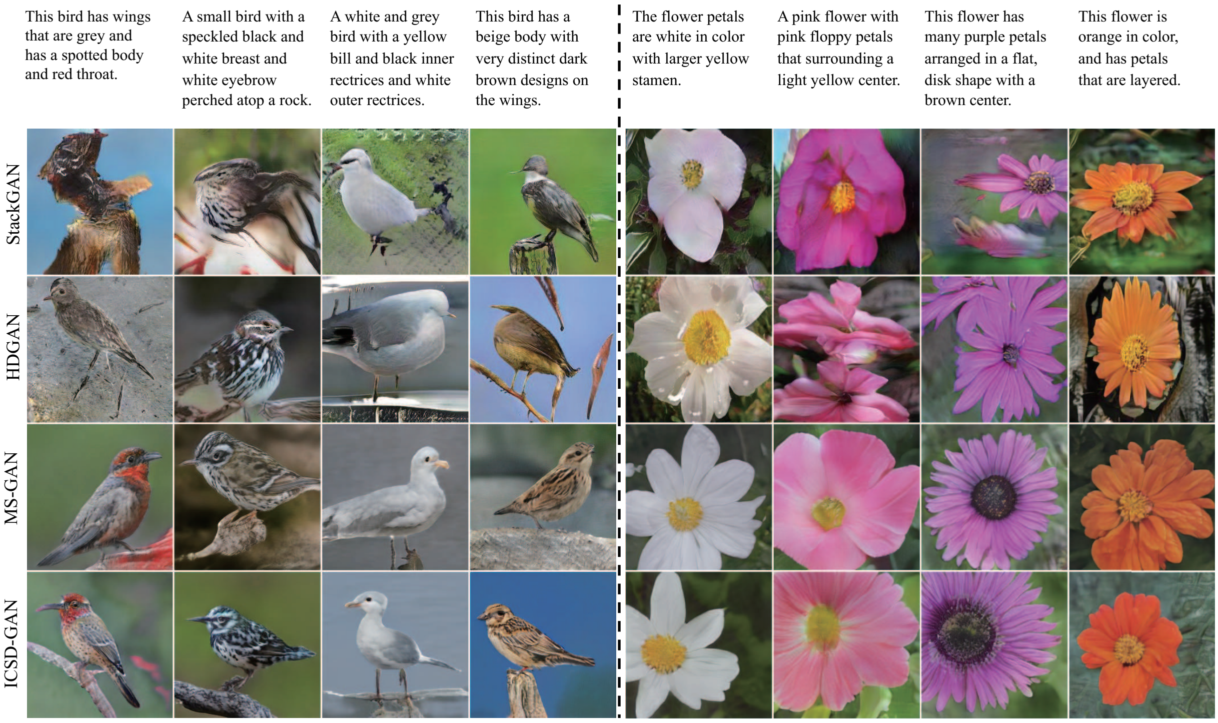
This bird has wings that are grey and has a spotted body and red throat.
A small bird with a speckled black and white breast and white eyebrow perched atop a rock.
A white and grey bird with a yellow bill and black inner rectrices and white outer rectrices.
This bird has a beige body with very distinct dark brown designs on the wings.
The flower petals are white in color with larger yellow stamen.
A pink flower with pink floppy petals that surrounding a light yellow center.
This flower has many purple petals arranged in a flat, disk shape with a brown center.
This flower is orange in color, and has petals that are layered.
StackGAN
HDGAN
MS-GAN
ICSD-GAN

Figure 3 (Color online) Visualization results of 256 × 256 images generated by StackGAN, HDGAN, MS-GAN and our ICSD-GAN model on CUB and Oxford-102 dataset.

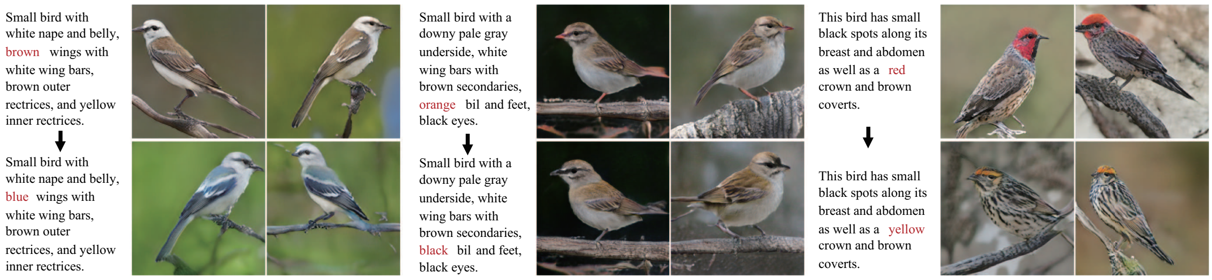
Small bird with white nape and belly, brown wings with white wing bars, brown outer rectrices, and yellow inner rectrices.
Small bird with white nape and belly, blue wings with white wing bars, brown outer rectrices, and yellow inner rectrices.
Small bird with a downy pale gray underside, white wing bars with brown secondaries, orange bil and feet, black eyes.
Small bird with a downy pale gray underside, white wing bars with brown secondaries, black bil and feet, black eyes.
This bird has small black spots along its breast and abdomen as well as a red crown and brown coverts.
This bird has small black spots along its breast and abdomen as well as a yellow crown and brown coverts.

Figure 4 (Color online) Visualization of text modification to evaluate the generation ability of our ICSD-GAN on CUB dataset.

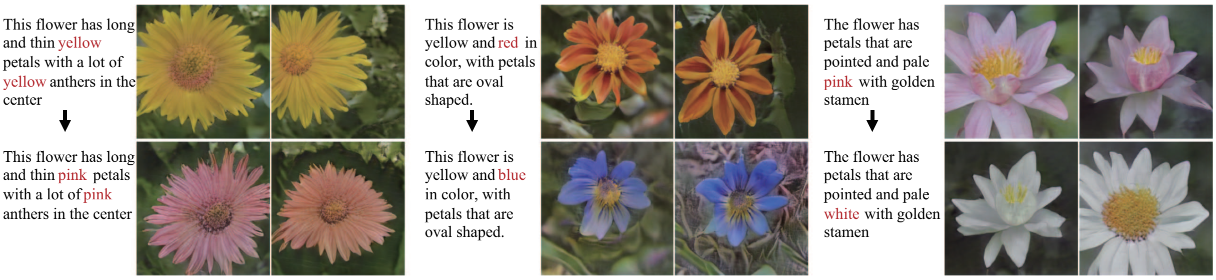
This flower has long and thin yellow petals with a lot of yellow anthers in the center
This flower has long and thin pink petals with a lot of pink anthers in the center
This flower is yellow and red in color, with petals that are oval shaped.
This flower is yellow and blue in color, with petals that are oval shaped.
The flower has petals that are pointed and pale pink with golden stamen
The flower has petals that are pointed and pale white with golden stamen

Figure 5 (Color online) Visualization of text modification to evaluate the generation ability of our ICSD-GAN on Oxford-102 dataset.

text is modified. The experimental results show the superior generalizability of the proposed ICSD-GAN model.

Performance of different stages of ICSD-GAN. We first evaluate the performance of the different stages of our model for the CUB dataset. Figure 6 shows the curve of the inception score for the generated images of different resolutions. As shown in Figure 6, our method outperforms StackGAN and HDGAN with respect to the inception score for the generated images of all resolutions, i.e., 64 × 64, 128 × 128, and 256 × 256. This result indicates the effectiveness of all stages in our model. What's more, the inception

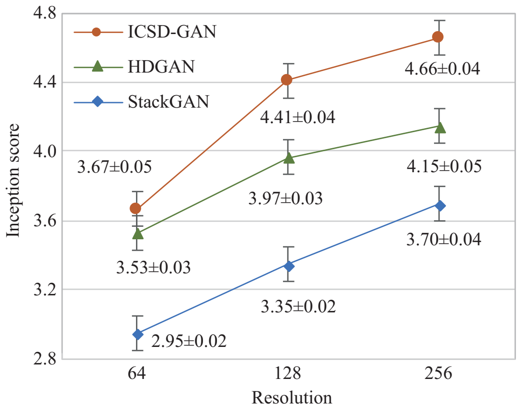

Figure 6 (Color online) Inception score for generated images with different resolutions on CUB dataset.

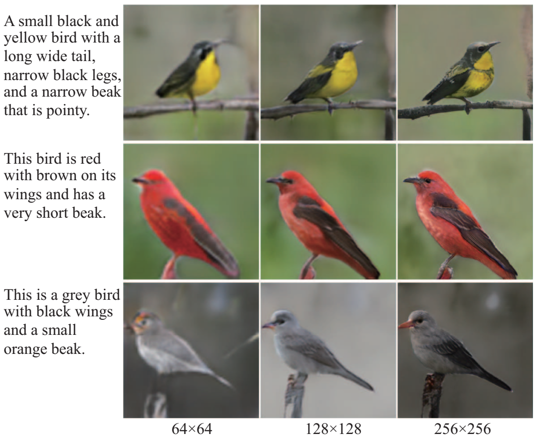
A small black and yellow bird with a long wide tail, narrow black legs, and a narrow beak that is pointy.
This bird is red with brown on its wings and has a very short beak.
This is a grey bird with black wings and a small orange beak.
64×64
128×128
256×256

Figure 7 (Color online) Visualization of generated images with different resolutions on CUB dataset.

Table 5 Performance of baseline model, model with $L_{\rm CSD}(I_0, I_2)$, model with $L_{\rm CSD}(I_1, I_2)$ and our full model with the two losses on CUB dataset

Method	Inception score (↑)	FID (↓)
Baseline	4.56 ± 0.02	10.41
With $L_{\rm CSD}(I_0, I_2)$	**4.72 ± 0.06**	10.97
With $L_{\rm CSD}(I_1, I_2)$	4.58 ± 0.04	9.58
ICSD-GAN	4.66 ± 0.04	**9.35**

score gradually increases from 64×64 images to 256×256 images; this trend is consistent with the coarse-to-fine mode of our model.

We also visualize the generated images of different scales in Figure 7. Other semantic details appear, and image quality increases with the resolution. The images of three resolutions differ in terms of vivid local details while other information remains the same. This result indicates the steady refinement of the text-to-image synthesis from the initial stage to the refined stage. Moreover, the result demonstrates the performance of our proposed cross-sample similarity distillation in terms of its enforcement of consistency among the generated images across stages.

Performance of different variants. In this study, we propose the interstage cross similarity distillation method in our multistage model. Eq. (9) shows two losses. One is $L_{\rm CSD}(I_0, I_2)$ between Stage^0 and Stage^2. The other is $L_{\rm CSD}(I_1, I_2)$ between Stage^1 and Stage^2. To evaluate the effectiveness of the two losses, we conduct experiments with different variants of our model for the CUB dataset. Table 5 shows the results of the baseline model, the model with $L_{\rm CSD}(I_0, I_2)$, the model with $L_{\rm CSD}(I_1, I_2)$, and our full model with both losses. The model with only $L_{\rm CSD}(I_0, I_2)$ shows improved inception score and FID. The model with only $L_{\rm CSD}(I_1, I_2)$ achieves a considerable increase in inception score, but its FID declines. Our full model outperforms these variants with respect to both the inception score and FID. The results demonstrate that the model with only one of the interstage cross similarity distillation loss cannot balance the quality and diversity of the generated images. On the contrary, our full model greatly improves the quality and diversity of the generates images. This result indicates the efficient text-to-image generation of ICSD-GAN. Therefore, the interstage distillations for Stage^0 and Stage^1 are of vital importance; it stabilizes the training and constrains the cross-sample similarity consistency for all stages of the multistage model.

5 Conclusion

In this study, we propose the ICSD-GAN model to transfer useful knowledge and constrain the consistent cross-sample relations of generated images within a multistage generation framework. With novel CSD blocks, we achieve knowledge distillation from the refined stage to the coarse stages. This process benefits efficient generation and discrimination. We adopt the knowledge distillation of cross-sample similarity, which achieves consistency in the cross-sample relations of generated images. We conduct extensive experiments on the Oxford-102 and CUB datasets. Our model achieves quantitatively remarkable performance relative to other state-of-the-art methods. Our model also generates photographic images, which are semantically consistent with the given text.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61876171, 61976203) and Fundamental Research Funds for the Central Universities.

References

1 Chen X, Duan Y, Houthooft R, et al. InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 2172–2180

2 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of Advances in Neural Information Processing Systems, 2014. 2672–2680

3 Kingma D P, Welling M. Auto-encoding variational Bayes. 2013. ArXiv: 1312.6114

4 Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. In: Proceedings of International Conference on Learning Representations, 2019

5 Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. 2018. ArXiv: 1812.04948

6 Xiong W, Luo W H, Ma L, et al. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 2364–2373

7 Xiong W, Lin Z, Yang J M, et al. Foreground-aware image inpainting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 5840–5848

8 Isola P, Zhu J-Y, Zhou T H, et al. Image-to-image translation with conditional adversarial networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 1125–1134

9 Zhu J-Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of IEEE International Conference on Computer Vision, 2017. 2223–2232

10 Miao Y W, Liu J Z, Chen J H, et al. Structure-preserving shape completion of 3D point clouds with generative adversarial network (in Chinese). Sci Sin Inform, 2020, 50: 675–691

11 Li Y H, Ao D Y, Dumitru C O, et al. Super-resolution of geosynchronous synthetic aperture radar images using dialectical GANs. Sci China Inf Sci, 2019, 62: 209302

12 Reed S, Akata Z, Yan X C, et al. Generative adversarial text to image synthesis. In: Proceedings of International Conference on Machine Learning, 2016

13 Zhang H, Xu T, Li H S, et al. StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of IEEE International Conference on Computer Vision, 2017. 5907–5915

14 Zhang H, Xu T, Li H S, et al. StackGAN++: realistic image synthesis with stacked generative adversarial networks. IEEE Trans Pattern Anal Mach Intell, 2018, 41: 1947–1962

15 Xu T, Zhang P C, Huang Q Y, et al. AttnGAN: fine-grained text to image generation with attentional generative adversarial networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1316–1324

16 Zhang Z Z, Xie Y P, Yang L. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 6199–6208

17 Mao F L, Ma B P, Chang H, et al. MS-GAN: text to image synthesis with attention-modulated generators and similarity-aware discriminators. In: Proceedings of British Machine Vision Conference, 2019

18 Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015. ArXiv: 1511.06434

19 Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. 2015. ArXiv: 1503.02531

20 Nilsback M, Zisserman A. Automated flower classification over a large number of classes. In: Proceedings of Indian Conference on Computer Vision, Graphics & Image Processing, 2008. 722–729

21 Wah C, Branson S, Welinder P, et al. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001. California Institute of Technology. 2011

22 Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training GANs. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 2234–2242

23 Heusel M, Ramsauer H, Unterthiner T, et al. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 6626–6637

24 Mirza M, Osindero S. Conditional generative adversarial nets. 2014. ArXiv: 1411.1784

25 Pang Y W, Xie J, Li X L. Visual haze removal by a unified generative adversarial network. IEEE Trans Circ Syst Video Tech, 2019, 29: 3211–3221

26 Mo S, Cho M, Shin J. InstaGAN: instance-aware image-to-image translation. In: Proceedings of International Conference on Learning Representations, 2019

27 Zhu Z, Huang T T, Shi B G, et al. Progressive pose attention transfer for person image generation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019

28 Zhang Z J, Pang Y W. CGNet: cross-guidance network for semantic segmentation. Sci China Inf Sci, 2020, 63: 120104

29 Liao M H, Song B Y, Long S B, et al. SynthText3D: synthesizing scene text images from 3D virtual worlds. Sci China Inf Sci, 2020, 63: 120105

30 Reed S, Akata Z, Lee H, et al. Learning deep representations of fine-grained visual descriptions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 49–58

31 Ji Z, Wang H R, Han J G, et al. Saliency-guided attention network for image-sentence matching. In: Proceedings of IEEE International Conference on Computer Vision, 2019

32 Qiao T T, Zhang J, Xu D Q, et al. MirrorGAN: learning text-to-image generation by redescription. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 1505–1514

33 Qiao T T, Zhang J, Xu D Q, et al. Learn, imagine and create: text-to-image generation from prior knowledge. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 885–895

34 Li W B, Zhang P C, Zhang L, et al. Object-driven text-to-image synthesis via adversarial training. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019

35 Huang Z H, Wang N Y. Like what you like: knowledge distill via neuron selectivity transfer. 2017. ArXiv: 1707.01219

36 Yim J, Joo D, Bae J, et al. A gift from knowledge distillation: fast optimization, network minimization and transfer learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 4133–4141

37 Romero A, Ballas N, Kahou S E, et al. Fitnets: hints for thin deep nets. 2014. ArXiv: 1412.6550

38 Zagoruyko S, Komodakis N. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. 2016. ArXiv: 1612.03928

39 Gu X Q, Ma B P, Chang H, et al. Temporal knowledge propagation for image-to-video person re-identification. In: Proceedings of IEEE International Conference on Computer Vision, 2019. 9647–9656

40 Yuan M K, Peng Y X. Text-to-image synthesis via symmetrical distillation networks. In: Proceedings of ACM International Conference on Multimedia, 2018

41 Chen Y T, Wang N Y, Zhang Z X. Darkrank: accelerating deep metric learning via cross sample similarities transfer. In: Proceedings of AAAI Conference on Artificial Intelligence, 2018

42 Dauphin Y N, Fan A, Auli M, et al. Language modeling with gated convolutional networks. In: Proceedings of International Conference on Machine Learning, 2017. 933–941

43 Kingma D P, Ba J. Adam: a method for stochastic optimization. In: Proceedings of International Conference on Learning Representations, 2015

44 Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2818–2826

45 Deng J, Dong W, Socher R, et al. Imagenet: a large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255

46 Abadi M, Barham P, Chen J M, et al. Tensorflow: a system for large-scale machine learning. In: Proceedings of Symposium on Operating Systems Design and Implementation, 2016. 265–283

47 Reed S E, Akata Z, Mohan S, et al. Learning what and where to draw. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 217–225