IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 April 2023, accepted 18 April 2023, date of publication 20 April 2023, date of current version 26 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3268869

RESEARCH ARTICLE

Enhanced Text-to-Image Synthesis With Self-Supervision

YONG XUAN TAN[1], CHIN POO LEE[1], (Senior Member, IEEE), MAI NEO[2], KIAN MING LIM[1], (Senior Member, IEEE), AND JIT YAN LIM[1]

[1]Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia
[2]Faculty of Creative Multimedia, Multimedia University, Persiaran Multimedia, Cyberjaya, Selangor 63100, Malaysia

Corresponding author: Chin Poo Lee (cplee@mmu.edu.my)

This work was supported in part by the Fundamental Research Grant Scheme of the Ministry of Higher Education under Grant FRGS/1/2021/ICT02/MMU/02/4, and in part by Telekom Malaysia (TM) Research and Development under Grant RDTC/190995.

ABSTRACT The task of Text-to-Image synthesis is a difficult challenge, especially when dealing with low-data regimes, where the number of training samples is limited. In order to address this challenge, the Self-Supervision Text-to-Image Generative Adversarial Networks (SS-TiGAN) has been proposed. The method employs a bi-level architecture, which allows for the use of self-supervision to increase the number of training samples by generating rotation variants. This, in turn, maximizes the diversity of the model representation and enables the exploration of high-level object information for more detailed image construction. In addition to the use of self-supervision, SS-TiGAN also investigates various techniques to address the stability issues that arise in Generative Adversarial Networks. By implementing these techniques, the proposed SS-TiGAN has achieved a new state-of-the-art performance on two benchmark datasets, Oxford-102 and CUB. These results demonstrate the effectiveness of the SS-TiGAN method in synthesizing high-quality, realistic images from text descriptions under low-data regimes.

INDEX TERMS Text-to-image synthesis, generative model, GAN, self-supervised learning, generative adversarial networks.

I. INTRODUCTION

Text-to-image synthesis is a challenging field that aims to generate visually realistic and semantically consistent images from a text description. This task requires the integration of image and text modalities, both of which are highly creative and flexible. One popular approach to text-to-image synthesis is using Generative Adversarial Networks (GANs) [1]. GANs consist of two components: a generator and a discriminator. The generator synthesizes images to fool the discriminator, while the discriminator evaluates the realism of the received images. The two networks are trained in a min-max game where they aim to maximize their respective objectives.

In order to condition the image synthesis on a given text description, a new variant of GANs, called Conditional Generative Adversarial Networks (cGANs) [2], is often used. cGANs can receive additional input as a conditioning variable, making it a suitable choice for text-to-image synthesis. The use of cGANs has been a major advancement in the field, leading to the development of various successful text-to-image synthesis models.

The challenge of synthesizing visually realistic images with limited training samples is addressed by exploring self-supervision in the text-to-image synthesis field. Self-supervision has been shown to be effective in other fields, such as computer vision, in mitigating the effects of low-data regimes [3]. One way self-supervision has been applied in text-to-image synthesis is through the use of multiple rotation variants of input images, which increases the size of the training sample and helps the model learn better structural features and explore the semantic content of the images. This is especially important when synthesizing complex objects like birds.

Multi-stage architecture is a popular approach in the existing works due to the reason that it is better at synthesizing large-scale realistic images. A new text-to-image

The associate editor coordinating the review of this manuscript and approving it for publication was Junhua Li.

synthesis approach named Self-Supervision Text-to-Image Generative Adversarial Networks (SS-TiGAN) is proposed, which combines self-supervision and bi-level architecture. Unlike multi-stage architecture, bi-level architecture only utilizes two-level GANs to synthesize images from 64 × 64 pixels to 128 × 128 pixels. By integrating self-supervision, the learned representation is diversified, leading to enhanced visual realism in the synthesized images.

To further address the challenges in training GANs, various techniques have been integrated to improve the training stability. To tackle the non-convergence issue, feature matching is applied as an extra objective in the generator. This mechanism aligns the focus of the generator from simply generating plausible images to deceive the discriminator, to producing images that closely match the real images, by minimizing the difference between the features of the synthesized images and the real images. In addition, the use of L1 distance loss can mitigate the issue of mode collapse in GANs. The L1 distance loss enforces the generator to reduce the discrepancy between the synthesized images and real images, thereby preventing the generator from learning incorrect features and producing low output variants. To address the overconfidence issue in the discriminator, one-sided label smoothing is introduced. One-sided label smoothing penalizes the discriminator when the predicted probability exceeds a certain threshold, thereby avoiding the discriminator from becoming overly confident in its predictions. With the above-mentioned techniques, the main contributions of this paper are:

- A novel approach to text-to-image synthesis that utilizes self-supervision and a bi-level GAN architecture to overcome the challenges faced in low-data regimes. The introduction of self-supervision with multiple rotation variants of input images as part of the training sample not only increases the data size but also forces the model to learn better structural features and semantic content of the images, leading to the synthesis of more complex objects.
- Enhancement techniques to mitigate the common problems faced during GANs training such as non-convergence, mode collapse, and overconfidence of the discriminator. The use of feature matching and L1 distance loss functions helps the generator to focus on mimicking real images and reduces the difference between the synthesized and real images. The application of one-sided label smoothing penalizes the discriminator when its prediction exceeds a certain threshold, preventing it from being overconfident.

II. RELATED WORK

In recent years, text-to-image synthesis has become a popular research topic in the field of computer vision. Many works have been proposed to address different aspects of the problem, including generating complex scenes from text descriptions, ensuring semantic consistency, and handling the variance of linguistic expression for the same image.

One of the earliest models proposed for text-to-image synthesis is GAN-INT-CLS [4]. It introduced the use of a deep convolutional GAN (DCGAN) with additional text input to generate images based on textual information. However, this model had limitations in synthesizing objects in desired locations and poses. To address this limitation, Reed et al. [5] proposed the Generative Adversarial What-Where Network (GAWWN) which allows for controlling the object's location and pose through additional inputs in the form of bounding boxes or coordinates.

Nguyen et al. [6] presented the Plug and Play Generative Network (PPGN), which iteratively produces a noise vector that maximizes the diversity of the synthesized image using a condition network. PPGN has shown promising results by synthesizing diverse images based on various textual descriptions. The role of auxiliary classification in enhancing the image synthesis process was demonstrated in Odena et al. [7]. Dash et al. [8] further improved this by introducing the Text Conditioned Auxiliary Classifier GAN (TAC-GAN), which used class information to improve the structural coherence of the generated images.

Recent models such as Text-conditioned Semantic Classifier GAN (Text-SeGAN) [9] and Dynamic Memory Generative Adversarial Networks (DM-GAN) [10] focused on improving the visual quality of synthesized images by incorporating complex multi-stage architectures. Text-SeGAN uses a triplet selection strategy during training to identify mismatched text-image pairs between real or fake images with different descriptions, while DM-GAN adds a memory module to handle the image generated after the first stage of generation.

Other recent models have also focused on improving the architecture of text-to-image models. Gao et al. [11] presented the Perceptual Pyramid Adversarial Networks (PPAN) that integrated a pyramid framework into the generator architecture to produce multi-scale images. Another architecture, Hierarchically-fused Generative Adversarial Network (HfGAN) by Huang et al. [12] employed a single discriminator and adaptively fused multi-scale visual features from different layers to synthesize final images.

In contrast, some works have simplified the architecture of text-to-image models. Souza et al. [13] proposed a simpler architecture that was trained directly on 256 × 256 images without involving multiple generators and discriminators. They also introduced a novel sentence interpolation strategy for smoother conditional space.

In the realm of synthesizing complex scenes from text descriptions, several works have been proposed to address different aspects of the problem. Hong et al. [14] introduced a hierarchical approach that infers the image layout to generate scenes. Hinz et al. [15] proposed an object pathway method that allows for the generation of complex scenes with multiple objects using bounding boxes and object labels. Li et al. [16] introduced the Object-driven attentive

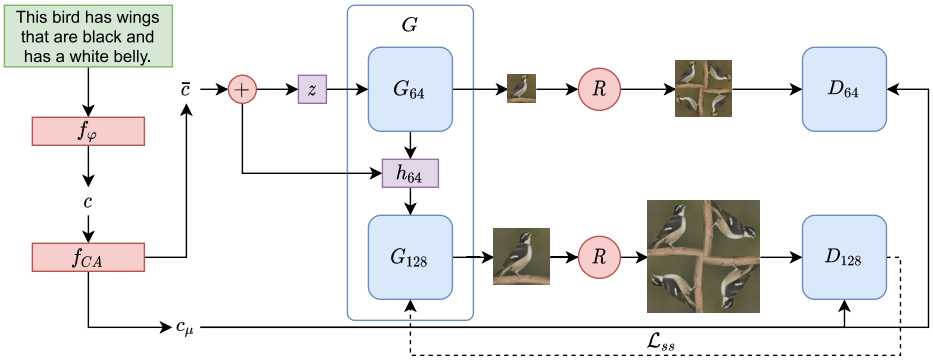


FIGURE 1. The overall architecture of the proposed SS-TiGAN. $\mathcal{L}_{ss}$ denotes the back propagation of self-supervision loss.

GAN (Obj-GAN), which uses attention mechanisms and semantic layouts to focus on objects. Hinz et al. [17] leveraged both a global pathway and object pathway to construct the image background structure and objects, respectively, based on the text description. Sharma et al. [18] developed Chatpainter, which used dialogue to generate high-quality images with multiple objects.

To ensure semantic consistency in image generation, Wang et al. [19] proposed an end-to-end approach that fuses semantic layouts, text semantics, and hidden visual features. Xu et al. [20] presented the Attentional Generative Adversarial Network (AttnGAN), which uses attention mechanisms and a multi-stage architecture to synthesize fine-grained images with improved semantic consistency. Qi et al. [21] developed the Multi-resolution Parallel Generative Adversarial Networks (MRP-GAN), which focuses on ensuring semantic consistency of the generated images in the early stage. Sah et al. [22] proposed the Multi-Modal Vector Representation (MMVR), which involves a two-way generation between images and text descriptions. MirrorGAN [23] modified AttnGAN to improve semantic consistency by using a re-description from the generated image.

Yin et al. [24] and Tan et al. [25] used a Siamese network to capture semantic commonality from different text descriptions to maintain consistency in the image generation process. Wang et al. [26] proposed the Textual-Visual Bidirectional Generative Adversarial Network (TVBi-GAN), which contains several semantic-related modules to utilize exact semantic features during the image synthesis process, thereby improving semantic consistency. Finally, to address the limitation of limited texture information in images generated from a single caption, Cheng et al. [27] developed the Rich Feature generation text-to-image synthesis (RiFeGAN), which retrieves several related text descriptions and utilizes the text features to enrich the input vector that is used to synthesize images.

In summary, recent works have made significant progress in text-to-image synthesis, and various approaches have been proposed to address different challenges.

III. SELF-SUPERVISION TEXT-TO-IMAGE GENERATIVE ADVERSARIAL NETWORKS (SS-TIGAN)

Self-Supervision Text-to-Image Generative Adversarial Networks (SS-TiGAN) is a generative model that leverages Generative Adversarial Networks (GANs) to synthesize high-quality images from textual descriptions. The architecture of the SS-TiGAN model is composed of a stacked generator G and two discriminators, D_{64} and D_{128}, which produce images with resolutions of 64×64 pixels and 128×128 pixels, respectively. Figure 1 illustrates the architecture of the SS-TiGAN model, highlighting the generator and the two discriminators.

A. CONDITIONING AUGMENTATION

SS-TiGAN addresses the challenge of transforming high-dimensional text features (1024d) into a smaller latent embedding (< 100d) for GANs training, which could result in a loss of information and affect the generator's performance, by introducing the text conditioning augmentation function f_{CA} [28], [29]. f_{CA} synthesizes more text embedding samples from a small number of original samples by encoding the text description t into a 1024-dimensional text embedding c through the pre-trained char-CNN-RNN text encoder f_φ [30]. c is then transformed into two 256-dimensional mean c_μ and covariance c_σ using a fully connected layer activated by the GLU function, as depicted in Figure 2. This transformation preserves crucial text information for the following learning stage, alleviating the issue of data discontinuity.

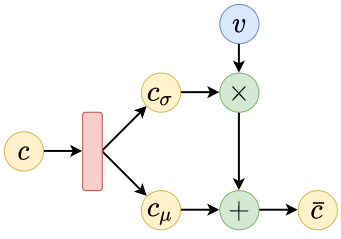


FIGURE 2. The architecture of f_{CA}.

The augmented text embedding $\bar{c}$ is computed using v, a random variable sampled from a normal distribution, which is element-wise multiplied with c_σ before being added to c_μ:

$$\bar{c} = v \times c_\sigma + c_\mu \tag{1}$$

To ensure that semantically related text embeddings are associated with the synthesized images, a regularization term $\mathcal{L}_{CA}$ is introduced into the generator's objective function, which maintains the smoothness of the transformation process:

$$\mathcal{L}_{CA} = D_{KL}\left[N\left(c_\mu, c_\sigma\right) \parallel N\left(0, 1\right)\right] \tag{2}$$

In GANs, the generator uses $\bar{c}$ as the conditioning variable while the discriminators use c_μ as the conditioning variable. By leveraging f_{CA}, SS-TiGAN enhances the semantic consistency performance of the generator.

B. NETWORK ARCHITECTURE

The SS-TiGAN is a network architecture designed to generate high-quality images from text descriptions, comprising of two stacked generators: G_{64} and G_{128}. Each generator contains a learning model, F_i^e, and an output model, F_i^g, with $i = \{64, 128\}$. The learning models extract the critical features that contribute to the realism of the generated images, while the output models convert these features into actual images. The detailed architecture of the stacked generator is presented in Table 1, with the notation following the format of $m \times m \times n$, where m denotes the width and height, while n represents the number of channels.

TABLE 1. The stacked generator architecture of the proposed SS-TiGAN.

Stages	Components		Configurations
G_{64}	F_{64}^e	FC+reshape	linear 32768 ($4 \times 4 \times 2048$), batchnorm, GLU
		Upsampling	upsample 2×2, conv $3 \times 3 \times 1024$, batchnorm, GLU
		Upsampling	upsample 2×2, conv $3 \times 3 \times 512$, batchnorm, GLU
		Upsampling	upsample 2×2, conv $3 \times 3 \times 256$, batchnorm, GLU
		Upsampling	upsample 2×2, conv $3 \times 3 \times 128$, batchnorm, GLU
	F_{64}^g	Conv+Tanh	conv $3 \times 3 \times 3$, Tanh
G_{128}	F_{128}^e	Joining	conv $3 \times 3 \times 128$, batchnorm, GLU
		Residual	conv $3 \times 3 \times 128$, batchnorm, GLU, conv $3 \times 3 \times 64$, batchnorm
		Residual	conv $3 \times 3 \times 128$, batchnorm, GLU, conv $3 \times 3 \times 64$, batchnorm
		Upsampling	upsample 2×2, conv $3 \times 3 \times 64$, batchnorm, GLU
	F_{128}^g	Conv+Tanh	conv $3 \times 3 \times 3$, Tanh

The learning model F_{64}^e of G_{64} is composed of a fully connected (FC) layer followed by a reshape block and four upsampling blocks. The FC+reshape block consists of a linear layer with 32,768 output neurons and a batch normalization layer, activated by the gated linear unit (GLU) function. This block transforms the input into a $4 \times 4 \times 2048$ tensor, which serves as the input for the upsampling blocks. The upsampling blocks consist of an upsample layer, a convolutional layer, and a batch normalization layer, activated by the GLU function at the end. These blocks work together to upscale the input tensor and refine the image features.

The learning model F_{128}^e of G_{128} is designed to enhance the image details further. It consists of a joining block, two residual blocks, and an upsampling block. The joining block combines the output from F_{64}^e with the input text embedding and contains a convolutional layer and a batch normalization layer activated by the GLU function. The residual blocks capture more complex representations of the input tensor using a deeper network architecture, as illustrated in Figure 3. The upsampling block then refines the image details before the final image is produced by the output model. This block has a similar structure as the upsampling blocks in F_{64}^e and is composed of an upsample layer, a convolutional layer, and a batch normalization layer, activated by the GLU function. Both output models F_{64}^g and F_{128}^g consist of a single convolutional layer activated by the hyperbolic tangent (Tanh) function.

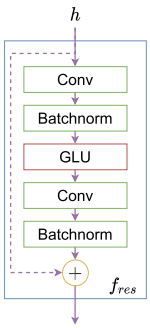


FIGURE 3. The architecture of the residual block f_{res}, h denotes the input tensor.

The discriminator comprises a feature extractor and multiple classifiers, which work together to judge the authenticity of the generated images. The feature extractor f_{θ_i} plays a critical role in extracting essential features from the generated images, enabling the classifiers to make informed decisions. To handle larger image sizes, D_{128} has six convolutional blocks in its feature extractor, while D_{64} has only four.

Both classifiers in the discriminators have similar structures but differ slightly in their specific tasks. The first classifier, f_{cls}, consists of two convolutional blocks and aims to classify the relationship between the text and images. The second classifier, f_{ucls}, is composed of a single convolutional block and determines whether the image is genuine or fake. The final classifier, f_{ss}, has two linear layers and aims to match the generated images with their corresponding rotation angles. The D_{64} discriminator has two classifiers, while the D_{128} discriminator has three, as shown in Table 2.

C. PROCESS

This section explains the learning process in the proposed SS-TiGAN. The process starts by sampling a 100-dimensional

TABLE 2. The discriminators architecture of the proposed SS-TiGAN.

Stages	Components	Configurations
D_{64}	$f_{\theta_{64}}$	conv $4 \times 4 \times 64$, leakyReLU, conv $4 \times 4 \times 128$, batchnorm, leakyReLU, conv $4 \times 4 \times 256$, batchnorm, leakyReLU, conv $4 \times 4 \times 512$, batchnorm, leakyReLU
	f_{cls}	conv $3 \times 3 \times 512$, batchnorm, leakyReLU, conv $4 \times 4 \times 1$, sigmoid
	f_{ucls}	conv $4 \times 4 \times 1$, sigmoid
D_{128}	$f_{\theta_{128}}$	conv $4 \times 4 \times 64$, leakyReLU, conv $4 \times 4 \times 128$, batchnorm, leakyReLU, conv $4 \times 4 \times 256$, batchnorm, leakyReLU, conv $4 \times 4 \times 512$, batchnorm, leakyReLU, conv $4 \times 4 \times 1024$, batchnorm, leakyReLU, conv $3 \times 3 \times 512$, batchnorm, leakyReLU
	f_{cls}	conv $3 \times 3 \times 512$, batchnorm, leakyReLU, conv $4 \times 4 \times 1$, sigmoid
	f_{ucls}	conv $4 \times 4 \times 1$, sigmoid
	f_{ss}	linear 64, linear 4

random noise vector z from a Gaussian distribution, which is concatenated with the text embedding $\bar{c}$ to form the input for the first stage of the generator, G_{64}. The learning model F_{64}^{e} in G_{64} takes the input and produces the initial image features h_{64}, which are then fed into the output model F_{64}^{g} to synthesize the initial image $\bar{x}_{64}$ with a resolution of 64×64 pixels. The two processes in G_{64} can be described as follows:

$$G_{64}: F_{64}^{e}(z, \bar{c}) \rightarrow h_{64} \qquad F_{64}^{g}(h_{64}) \rightarrow \bar{x}_{64} \tag{3}$$

The synthesizing process can be expressed as:

$$\bar{x}_{64} = G_{64}(z, \bar{c}) \tag{4}$$

The synthesized image $\bar{x}_{64}$ is then used for the learning process of the discriminator D_{64}. The initial image features h_{64} are then concatenated with the augmented text embedding $\bar{c}$ and fed into the learning model F_{128}^{e} in the second stage of the generator, G_{128}. The learning model F_{128}^{e} expands and processes the input to produce the final image features h_{128}, which are then passed into the output model F_{128}^{g} to produce the final image $\bar{x}_{128}$ with a resolution of 128×128 pixels. The two processes in G_{128} can be expressed as follows:

$$G_{128}: F_{128}^{e}(h_{64}, \bar{c}) \rightarrow h_{128} \qquad F_{128}^{g}(h_{128}) \rightarrow \bar{x}_{128} \tag{5}$$

The synthesizing process can be expressed as:

$$\bar{x}_{128} = G_{128}(h_{64}, \bar{c}) \tag{6}$$

Finally, the final image $\bar{x}_{128}$ is fed into the discriminator D_{128} to evaluate its realism. The loss functions of the generator G_{64} and the discriminator D_{64} are updated accordingly, then the process is repeated until convergence.

The quality of the images synthesized in text-to-image synthesis is significantly impacted by the richness of the learned representation. To enhance the learned representation and produce more detailed images $\bar{x}_{128}$, the model incorporates the self-supervised rotation prediction task [31]. In this task, each image is rotated to $r \in 90, 180, 270$ degrees, expressed as:

$$\hat{\bar{x}}_{128} = R(\bar{x}_{128}) \tag{7}$$

where R represents the rotation function. The resulting rotated image tensor $\hat{\bar{x}}_{128}$ is created by concatenating $\{\bar{x}_{128}, \bar{x}_{128}^{90}, \bar{x}_{128}^{180}, \bar{x}_{128}^{270}\}$. The same rotation procedure is applied to real images x as well. To accommodate the increased number of training images, the embedding c_{μ} is replicated three more times and combined to form $\hat{c}_{\mu}$.

In both discriminators D_{64} and D_{128}, the objective is to determine the realism of the received images $X_i \in \{\hat{x}_i, \hat{x}_i', \hat{\bar{x}}_i\}$ with $i \in \{64, 128\}$ and the semantic consistency between the images and the transformed text embedding $\hat{c}_{\mu}$. To achieve these goals, the models learn from three sample pairs:

1) $\hat{c}_{\mu}$ paired with matched real images $\hat{x}_i$.
2) $\hat{c}_{\mu}$ paired with unmatched real images $\hat{x}_i'$.
3) $\hat{c}_{\mu}$ paired with synthesized images $\hat{\bar{x}}_i$.

The task of the discriminator is to predict sample pair (1) as a real sample pair and the rest as fake sample pairs. The inclusion of sample pair (2) allows the discriminator to distinguish the relationship between the image and text description. To evaluate the performance of the models, two classifiers are used: f_{cls} and f_{ucls}.

f_{cls} performs conditional classification to evaluate the semantic consistency between the image features $f_{\theta_i}(X_i)$ and $\hat{c}_{\mu}$ in all three sample pairs. The conditional process is defined as:

$$D_i(X_i, \hat{c}_{\mu}) = f_{cls}(f_{\theta_i}(X_i), \hat{c}_{\mu}) \tag{8}$$

f_{ucls} performs unconditional classification to evaluate the visual realism of the received images from sample pairs (1) and (3). The f_{ucls} discriminates whether X_i is a real or synthesized image without being conditioned on the text embedding. The unconditional process is defined as:

$$D_i(X_i) = f_{ucls}(f_{\theta_i}(X_i)) \tag{9}$$

The discriminator's fundamental objective in conditional and unconditional classification is expressed as:

$$\mathcal{L}_{D_i} = \frac{\mathcal{L}_{r_i}}{2} + \frac{\mathcal{L}_{f_i}}{3} \tag{10}$$

where $\mathcal{L}_{D_i}$ is the overall loss for the i-th discriminator, and $\mathcal{L}_{r_i}$ and $\mathcal{L}_{f_i}$ are the real sample loss and fake sample loss, respectively. These terms are computed as follows:

$$\begin{aligned} \mathcal{L}_{r_i} &= -\log\left[D_i(\hat{x}_i)\right] - \log\left[D_i(\hat{x}_i, \hat{c}_{\mu})\right] \\ \mathcal{L}_{f_i} &= -\log\left[1 - D_i\left(\hat{\bar{x}}_i\right)\right] - \log\left[1 - D_i\left(\hat{\bar{x}}_i, \hat{c}_{\mu}\right)\right] \\ &\quad - \log\left[1 - D_i(\hat{x}_i', \hat{c}_{\mu})\right] \end{aligned} \tag{11}$$

where $D_i(\cdot)$ is the i-th discriminator function, $\hat{x}_i$ is a real sample, $\hat{\bar{x}}_i$ is a generated sample, $\hat{x}_i'$ is a unmatched real sample, and $\hat{c}_{\mu}$ is the text embedding (if available). In this setup, the discriminator tries to maximize the real sample loss and minimize the fake sample loss, which ultimately leads

to the discriminator learning to distinguish between real and generated samples. The weights of $\frac{1}{2}$ and $\frac{1}{3}$ in Equation 10 balance the importance of the real and fake sample losses, respectively.

The SS-TiGAN model is designed to handle bi-scale image generation through a tree-like structure consisting of two generators, G_{64} and G_{128}, and two discriminators, D_{64} and D_{128}. The stacked generator $G = \{G_{64}, G_{128}\}$ and discriminators are trained end-to-end to optimize the approximation of the bi-scale image distribution.

The training objective is to minimize the common loss function $\mathcal{L}_G$, which considers the losses of both discriminators, as follows:

$$\mathcal{L}_G = \epsilon \cdot \mathcal{L}_{CA} + \sum_{i=\{64,128\}} \mathcal{L}_{G_i} \text{ where}$$
$$\mathcal{L}_{G_i} = -\log\left[D_i\left(\hat{\tilde{x}}_i, \hat{c}_\mu\right)\right] - \log\left[D_i\left(\hat{\tilde{x}}_i\right)\right] \quad (12)$$

where ϵ is the coefficient for the regularization term $\mathcal{L}_{CA}$.

The bi-scale design allows the generator and discriminator pairs to start with lower scale images and gradually refine into larger scale and finer images. For instance, G_{64} generates low resolution images with essential features such as color and object structure, while G_{128} focuses on refining visual details to generate high resolution images. The end-to-end training ensures that all generators share a common understanding of approximation across different scales, thereby stabilizing the overall network training.

1) FEATURE MATCHING

In GANs, the generator and discriminator play a two-player min-max game, with the goal of finding a Nash equilibrium where both are optimized. However, the optimization process can sometimes become unbalanced and the generator and discriminator do not converge. To address this issue, feature matching is introduced as a new objective to ensure that the generator focuses on synthesizing visually realistic images.

The feature matching objective in a proposed SS-TiGAN is defined as follows:

$$\mathcal{L}_{fm_i} = \left\| f_{\theta_i}\left(\hat{x}_i\right) - f_{\theta_i}\left(\hat{\tilde{x}}_i\right) \right\|_2^2 \quad (13)$$

where f_{θ_i} represents the feature extraction function for the i-th stage, $\hat{x}_i$ is the real image, and $\hat{\tilde{x}}_i$ is the generated image. The objective calculates the L2 distance between the mean of the feature maps of the real and generated images in all stages. This objective encourages the generator to synthesize images that are similar to the real images as perceived by the discriminator in each stage.

2) L1 DISTANCE LOSS

The phenomenon where the generator repeatedly produces a small set of similar outputs, known as mode collapse, results in synthesized images with high similarity and low diversity. To overcome this, the L1 distance loss is introduced as an objective function for both generators.

This objective aims to minimize the difference between real images x_i and generated images $\bar{x}_i$ in the pixel space. The L1 distance is computed as follows:

$$\mathcal{L}_{L1_i} = \left\| \hat{x}_i - \hat{\tilde{x}}_i \right\|_1 \quad (14)$$

The L1 distance loss encourages the generators to learn features from the real images and produce more diverse outputs, rather than synthesizing similar outputs repeatedly. This helps to overcome the problem of mode collapse and ensure that the generated images are diverse and similar to the real images.

3) ONE-SIDED LABEL SMOOTHING

GANs are prone to the issue of overconfidence, where the discriminator relies too heavily on a small set of features to predict the authenticity of an input image. To address this issue, one-sided label smoothing is introduced as a way to reduce the overconfidence of the discriminators.

One-sided label smoothing penalizes both discriminators when the predictions of the conditional or unconditional pairs of real images, $D_i(\hat{x}_i, \hat{c}_\mu)$ or $D_i\left(\hat{x}_i\right)$, are higher than 0.9. Instead of using a target label of 1.0, the true target label is set to 0.9 to introduce some uncertainty into the discriminator's predictions.

The objective of one-sided label smoothing is to encourage the discriminators to not rely too heavily on a small set of features and to consider a wider range of features in their predictions. This helps to make the discriminator's predictions more robust and reduces the risk of overconfidence.

4) SELF-SUPERVISION

Self-supervision is introduced into the GAN framework to enhance the quality and diversity of the generated images. In this approach, a classifier f_{ss} is added to the discriminator D_{128} to predict the rotation degree of the input images $\hat{x}_{128}$ and $\hat{\tilde{x}}_{128}$.

The self-supervision loss functions are defined as follows:

$$\mathcal{L}_{D_{ss}} = -\log\left[D_{128}\left(r \mid \hat{x}_{128}\right)\right]$$
$$\mathcal{L}_{G_{ss}} = -\log\left[D_{128}\left(r \mid \hat{\tilde{x}}_{128}\right)\right] \quad (15)$$

where r is the rotation degree of the image.

The classifier f_{ss} takes the extracted features $f_{\theta_{128}}\left(\hat{x}_{128}\right)$ and $f_{\theta_{128}}\left(\hat{\tilde{x}}_{128}\right)$ from the input images $\hat{x}_{128}$ and $\hat{\tilde{x}}_{128}$ as inputs, and predicts the corresponding rotation degree r as follows:

$$D_{128}\left(r \mid \hat{x}_{128}\right) = f_{ss}\left(r \mid f_{\theta_{128}}\left(\hat{x}_{128}\right)\right)$$
$$D_{128}\left(r \mid \hat{\tilde{x}}_{128}\right) = f_{ss}\left(r \mid f_{\theta_{128}}\left(\hat{\tilde{x}}_{128}\right)\right) \quad (16)$$

Through this self-supervision mechanism, the discriminator D_{128} is able to explore high-level semantic information, allowing the generator G_{128} to construct clearer object parts during the refinement process.

With the components mentioned, the final losses of the discriminators D_{64} and D_{128} are defined as:

$$\mathcal{L}_{D_{64}} = \frac{\mathcal{L}_{r_{64}}}{2} + \frac{\mathcal{L}_{f_{64}}}{3}$$
$$\mathcal{L}_{D_{128}} = \frac{\mathcal{L}_{r_{128}}}{2} + \frac{\mathcal{L}_{f_{128}}}{3} + \delta \cdot \mathcal{L}_{D_{ss}} \quad (17)$$

where δ is the coefficient for the self-supervision loss term. The losses of the generators G_{64} and G_{128} are updated as:

$$\mathcal{L}_{G_{64}} = \mathcal{L}_{G_{64}} + \alpha \cdot \mathcal{L}_{fm_{64}} + \beta \cdot \mathcal{L}_{L1_{64}}$$
$$\mathcal{L}_{G_{128}} = \mathcal{L}_{G_{128}} + \alpha \cdot \mathcal{L}_{fm_{128}} + \beta \cdot \mathcal{L}_{L1_{128}} + \gamma \cdot \mathcal{L}_{G_{ss}} \quad (18)$$

where α, β, γ are the coefficients for different loss terms. The $\mathcal{L}_{Gi}$ term on the right-hand side is derived from the fundamental loss term in Equation 12. α and β are shared across both generators. The self-supervised training process for SS-TiGAN is described in detail in Algorithm 1.

Algorithm 1 The Training Flow of the Proposed SS-TiGAN.

Require: Mini batch b from the training dataset T, matched real images x_i, unmatched real images x'_i, text description t, rotation function R, generator G, discriminators D_i

for e iterations **do**
  for $b \sim T$ **do**
    $x_{64}, x_{128}, x'_{64}, x'_{128}, t \leftarrow b$
    $\bar{c}, c_\mu, c_\sigma \leftarrow f_{CA}\left(f_\varphi\left(t\right)\right)$
    $z \sim N\left(0, 1\right); \hat{c}_\mu \leftarrow \left\{c_\mu, c_\mu, c_\mu, c_\mu\right\}$
    $r \leftarrow \{0, 1, 2, 3\}$
    $\bar{x}_{64}, \bar{x}_{128} \leftarrow G\left(z, \bar{c}\right)$
    $\hat{x}_{64} \leftarrow R\left(x_{64}\right); \hat{x}'_{64} \leftarrow R\left(x'_{64}\right); \hat{\bar{x}}_{64} \leftarrow R\left(\bar{x}_{64}\right)$
    $\hat{x}_{128} \leftarrow R\left(x_{128}\right); \hat{x}'_{128} \leftarrow R\left(x'_{128}\right); \hat{\bar{x}}_{128} \leftarrow R\left(\bar{x}_{128}\right)$
    for i iterations **do**
      $\mathcal{L}_{r_i} \leftarrow -\log\left[D_i\left(\hat{x}_i\right)\right] - \log\left[D_i\left(\hat{x}_i, \hat{c}_\mu\right)\right]$
      $\mathcal{L}_{f_i} \leftarrow -\log\left[1 - D_i\left(\hat{\bar{x}}_i\right)\right] - \log\left[1 - D_i\left(\hat{\bar{x}}_i, \hat{c}_\mu\right)\right] - \log\left[1 - D_i\left(\hat{x}'_i, \hat{c}_\mu\right)\right]$
      $\mathcal{L}_{D_i} \leftarrow \frac{\mathcal{L}_{r_i}}{2} + \frac{\mathcal{L}_{f_i}}{3}$
      if i is 128 **then**
        $\mathcal{L}_{D_{ss}} \leftarrow -\log\left[D_{128}\left(r \mid \hat{x}_{128}\right)\right]$
        $\mathcal{L}_{D_{128}} \leftarrow \mathcal{L}_{D_i} + \delta \cdot \mathcal{L}_{D_{ss}}$
      end if
      $D_i \leftarrow D_i - \Delta\sigma \mathcal{L}_{D_i} / \sigma D_i$
    end for
    $\mathcal{L}_{fm_{64}} \leftarrow \left\| f_{\theta_{64}}\left(\hat{x}_{64}\right) - f_{\theta_{64}}\left(\hat{\bar{x}}_{64}\right) \right\|_2^2$
    $\mathcal{L}_{L1_{64}} \leftarrow \left\| \hat{x}_{64} - \hat{\bar{x}}_{64} \right\|_1$
    $\mathcal{L}_{fm_{128}} \leftarrow \left\| f_{\theta_{128}}\left(\hat{x}_{128}\right) - f_{\theta_{128}}\left(\hat{\bar{x}}_{128}\right) \right\|_2^2$
    $\mathcal{L}_{L1_{128}} \leftarrow \left\| \hat{x}_{128} - \hat{\bar{x}}_{128} \right\|_1$
    $\mathcal{L}_{G_{ss}} \leftarrow -\log\left[D_{128}\left(\hat{r} \mid \hat{\bar{x}}^r_{128}\right)\right]$
    $\mathcal{L}_{G_{64}} \leftarrow -\log\left[D_{64}\left(\hat{\bar{x}}_{64}, \hat{c}_\mu\right)\right] - \log\left[D_{64}\left(\hat{\bar{x}}_{64}\right)\right] + \alpha \cdot \mathcal{L}_{fm_{64}} + \beta \cdot \mathcal{L}_{L1_{64}}$
    $\mathcal{L}_{G_{128}} \leftarrow -\log\left[D_{128}\left(\hat{\bar{x}}_{128}, \hat{c}_\mu\right)\right] - \log\left[D_{128}\left(\hat{\bar{x}}_{128}\right)\right] + \alpha \cdot \mathcal{L}_{fm_{128}} + \beta \cdot \mathcal{L}_{L1_{128}} + \delta \cdot \mathcal{L}_{G_{ss}}$
    $\mathcal{L}_{CA} \leftarrow D_{KL}\left[N\left(c_\mu, c_\sigma\right) \parallel N\left(0, 1\right)\right]$
    $\mathcal{L}_G \leftarrow \mathcal{L}_{G_{64}} + \mathcal{L}_{G_{128}} + \epsilon \cdot \mathcal{L}_{CA}$
    $G \leftarrow G - \Delta\sigma \mathcal{L}_G / \sigma G$
  end for
end for

IV. EXPERIMENTS AND DISCUSSIONS

This section presents the details of the datasets used, implementation specifics, and a comprehensive analysis of the model performance compared to other existing approaches. The experiments are conducted on a NVIDIA GeForce RTX 2080 Ti using the Anaconda environment on a Windows 10 platform.

A. DATASETS

The evaluation of the model is performed using two popular text-to-image synthesis datasets: the Oxford-102 [32] and CUB [33] datasets. The Oxford-102 dataset comprises 8,189 images belonging to 102 different flower categories, with 82 classes used for training and 20 classes reserved for testing. The CUB dataset, on the other hand, contains 11,788 images of 200 bird species, with 150 classes for training and 50 for testing, following the established practices in existing works [4], [28], [29].

Each image in both datasets is paired with 10 captions, and all images are resized to either 64 × 64 or 128 × 128 pixels based on stages. The pixel values are normalized to the range of [−1, 1], and data augmentation is performed during the training stage using random cropping and horizontal flipping. Before training, the images in the CUB dataset are preprocessed to ensure that the ratio of the object to the image region is greater than 75% [4], [28], [29].

B. IMPLEMENTATION DETAILS

In this work, the deep learning model is trained using the Adaptive Moment Estimation (Adam) optimizer. The learning rate is set to 0.0002 and is kept constant throughout the training process, with *beta*1 and *beta*2 values of 0.5 and 0.999, respectively, following established practices in the field [4], [20], [28], [29]. The training is conducted over 600 epochs with a batch size of 32 samples. The hyperparameters α and β are fixed at 1.0 throughout the training process. The optimal result for the Oxford-102 dataset is achieved using $\epsilon = 1.0$, $\delta = 2.0$, $\gamma = 1.5$ while for the CUB dataset, the optimal result is obtained using $\epsilon = 5.0$, $\delta = 2.0$, $\gamma = 1.0$.

C. EVALUATION METRICS

This section describes the evaluation metrics used to assess the performance of the proposed model.

TABLE 3. The inception score and FID of comparison results.

Methods	Resolution	Oxford-102		CUB	
		Inception score ↑	FID ↓	Inception score ↑	FID ↓
GAN-INT-CLS [4]	64×64	2.66±0.03	79.55	2.88±0.04	68.79
GAWWN [5]	128×128	-	-	3.62±0.07	-
StackGAN [28]	256×256	3.20±0.01	55.28	3.70±0.04	51.89
StackGAN++ [29]	256×256	3.26±0.01	48.68	4.04±0.05	15.30
AttnGAN [20]	256×256	-	-	4.36±0.03	23.19
MLADIC [34]	64×64	3.13±0.04	-	-	-
FusedGAN [35]	256×256	-	-	3.92±0.03	-
SAM-GAN [36]	256×256	-	-	**4.61±0.03**	20.49
SSTIS [37]	128×128	3.41±0.06	42.95	3.93±0.04	15.80
SS-TiGAN (Proposed)	128×128	**3.45±0.04**	**40.54**	4.09±0.05	**14.20**

TABLE 4. The SSIM of comparison results.

Methods	Resolution	SSIM ↑	
		Oxford-102	CUB
GAN-INT-CLS [4]	64×64	0.1948	0.2934
GAWWN [5]	128×128	-	0.2370
StackGAN [28]	256×256	0.1837	0.2812
AttnGAN [20]	256×256	0.1873	0.3129
HDGAN [38]	256×256	0.1886	0.2887
SSTIS [37]	128×128	0.7290	0.7982
SS-TiGAN (Proposed)	128×128	**0.7353**	**0.8195**

1) INCEPTION SCORE

The Inception score is a commonly used metric for evaluating the quality of images generated by generative models. It utilizes a pre-trained Inception v3 model to compute the score, which reflects the distinctness of the objects in each image and the variety of objects in the image set. The Inception score is computed based on the probabilities returned by the fine-tuned Inception v3 model from [28]. A higher inception score indicates better object distinctness and variety in the generated images.

2) FRÉCHET INCEPTION DISTANCE (FID)

The Fréchet Inception Distance (FID) is used to measure the similarity between the distribution of the generated images and the real images in the feature space. This evaluation metric adopts the pre-trained Inception v3 model to extract features from both sets of images and calculates the Fréchet distance between the feature sets. The lower the FID score, the closer the generated images are to the real images in terms of distribution similarity.

3) STRUCTURAL SIMILARITY INDEX MATRIX (SSIM):

The Structural Similarity Index Matrix (SSIM) is a widely used metric to evaluate the similarity between two images. In text-to-image synthesis, SSIM can be used to indirectly measure the semantic consistency between the generated image and the corresponding text description. Since the synthesized images based on the particular text description should contain the similar visual features with the paired real images. Higher SSIM scores indicate a better semantic consistency between the generated images and the real images.

D. COMPARISON WITH EXISTING APPROACHES

The proposed SS-TiGAN method has been evaluated against several existing text-to-image synthesis methods using the Inception score and FID metrics on the datasets mentioned previously. The results presented in Table 3 indicate that SS-TiGAN outperforms most of the methods in synthesizing realistic images. This is due to the deeper network architecture that incorporates residual blocks, which helps alleviate the low Inception score problem faced by other approaches like GAN-INT-CLS and MLADIC when synthesizing low-resolution images. Furthermore, SS-TiGAN outperforms GAWWN, which synthesizes images at the same resolution but without residual architecture.

Interestingly, SS-TiGAN achieves better results than prior works such as StackGAN, StackGAN++, and FusedGAN by synthesizing smaller images (128 × 128) instead of larger images, which is the trend to obtain a better Inception score. This highlights the effectiveness of self-supervision in diversifying the model representation, allowing small-scale images to contain more diverse and realistic visual information. Despite the impressive performance of the proposed SS-TiGAN, it achieved a slightly lower Inception score compared to AttnGAN and SAM-GAN. The primary reason for this difference in performance is that both AttnGAN and SAM-GAN utilize an attention module to synthesize large-scale images, which results in generating fine-grained images that are more likely to achieve higher Inception scores compared to other methods.

Unlike Inception score, the proposed SS-TiGAN surpasses all the existing methods in terms of FID, as indicated by the lowest FID scores of 40.54 and 14.20 on the Oxford-102 and CUB datasets respectively. This low FID score implies that the synthesized images from SS-TiGAN are highly similar to the real images in the feature space. This is achieved through the use of self-supervision, which allows the model to explore the semantic context from the real images and replicate it during synthesis, as well as feature matching and L1 distance loss, which forces the model to mimic the real image features during synthesis. These techniques enable SS-TiGAN to achieve the lowest FID score compared to existing methods that synthesize large-scale images. Although AttnGAN and SAM-GAN generate highly realistic

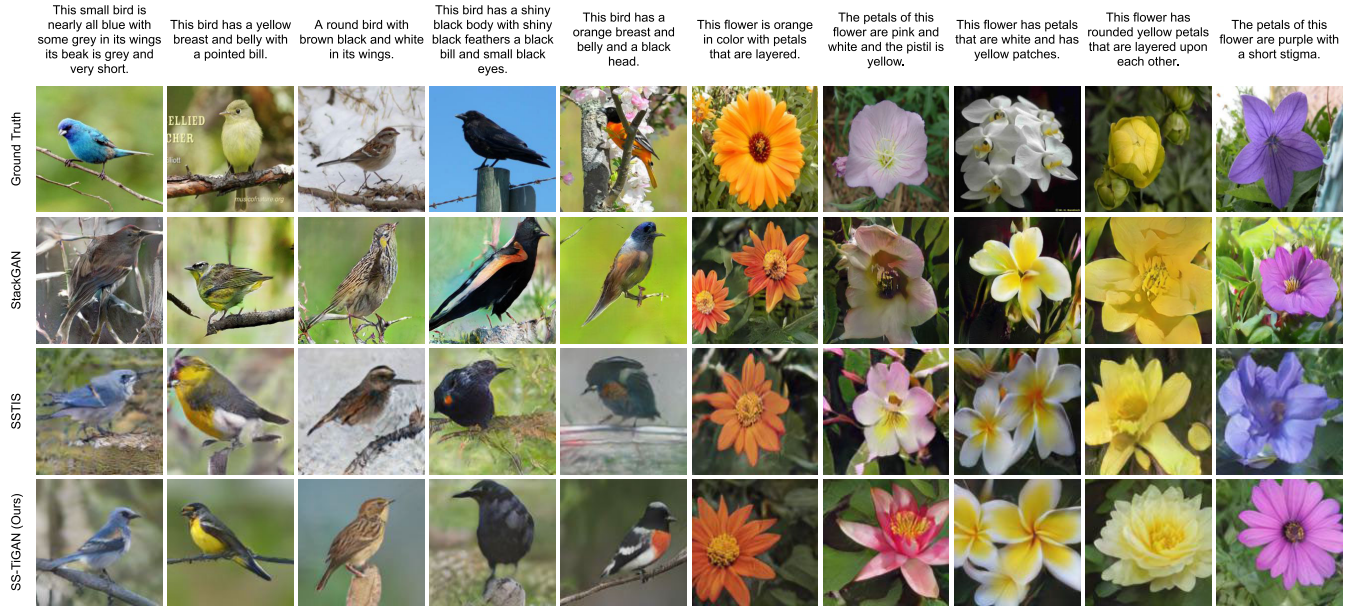


FIGURE 4. Some images synthesised from other methods with the proposed SS-TiGAN alongside real images (Ground Truth), given the text descriptions from CUB and Oxford-102 datasets.

FIGURE 5. A wide variety of images synthesised by the proposed SS-TiGAN.

images that outperform the proposed SS-TiGAN in terms of Inception scores, the similarity between the synthesized and real image sets is lower for these models. In contrast, the proposed SS-TiGAN achieves a better FID score due to the effectiveness of feature matching and L1 distance loss in replicating the characteristics of the real image set. These techniques enable SS-TiGAN to generate images that closely resemble the real set, resulting in a better FID score.

Additionally, the proposed SS-TiGAN has achieved outstanding results in terms of SSIM, with values of 0.7353 and 0.8195 on the Oxford-102 and CUB datasets, respectively, as shown in Table 4. This high SSIM score indicates that the images generated by SS-TiGAN possess a high level of semantic consistency, making them visually appealing and plausible.

This success can be attributed to a combination of factors, including self-supervision which enriches the diversity of the learned representations, the L1 distance loss that enhances the visual realism of the images, and the use of one-sided label smoothing and feature matching, which streamlines the training process while simultaneously elevating the visual realism and semantic consistency of the generated images.

The qualitative results of the comparison between the proposed SS-TiGAN method and other existing text-to-image synthesis approaches are presented in Figure 4. The figure displays the images synthesized from randomly selected text descriptions from the testing set of the CUB and Oxford-102 datasets.

It is evident from the results that the images synthesized by the proposed SS-TiGAN show high semantic consistency with the corresponding text descriptions. They also exhibit a remarkable similarity with the real images (Ground Truth). The proposed SS-TiGAN is able to synthesize images with a rich diversity of heterogeneous visual content, which can be attributed to the combination of self-supervision and other loss functions used in the model.

When compared to existing methods like StackGAN and SSTIS, the images synthesized by the proposed SS-TiGAN are of higher quality and realism. Additionally, Figure 5 showcases the wide range of image contents that the

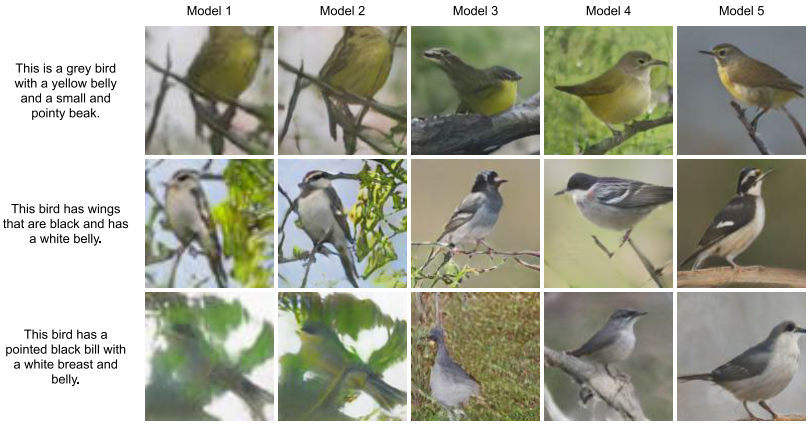


FIGURE 6. The ablation study of the proposed SS-TiGAN on the CUB dataset.

TABLE 5. The ablation study of the proposed SS-TiGAN on the CUB dataset.

Model	G_{64}	G_{128}	f_{res}	T	$\mathcal{L}_{ss}$	Inception score ↑
1	✓	-	-	-	-	3.51±0.05
2	✓	✓	-	-	-	3.91±0.04
3	✓	✓	✓	-	-	3.97±0.05
4	✓	✓	✓	✓	-	4.00±0.02
5	✓	✓	✓	✓	✓	**4.09±0.05**

proposed SS-TiGAN is capable of synthesizing. These results highlight the effectiveness of the proposed SS-TiGAN in synthesizing high-quality, realistic, and diverse images from text descriptions.

E. ABLATION STUDY

The aim of the ablation study in this section is to evaluate the impact of each component that contributes to the performance of the proposed SS-TiGAN. The evaluation of these components will provide insight into the key factors that drive the success of SS-TiGAN in synthesizing high-quality images from text descriptions.

The components evaluated in this study include the first stage generator G_{64}, the second stage generator G_{128}, the residual blocks f_{res}, various improvement techniques T such as feature matching, L1 distance loss, and one-sided label smoothing, and finally, the self-supervision component $\mathcal{L}_{ss}$. The results of the Inception score, a widely used evaluation metric in image synthesis tasks, are presented in Table 5. These results will provide quantitative evidence of the effectiveness of each component and highlight the contributions they make to the overall performance of SS-TiGAN.

Based on the results of the ablation study, it is evident that each component has played a significant role in improving the performance of the proposed SS-TiGAN. The initial stage of generator G_{64}, which synthesizes 64 × 64 pixel images, was not sufficient as it achieved the lowest Inception score (Model 1). However, incorporating the second stage of generator G_{128}, which synthesizes 128 × 128 pixel images, improved the Inception score to 3.91 (Model 2). The increased resolution allowed the image to carry more visual information, but residual blocks f_{res} were necessary to effectively learn better features from the high-resolution images without losing the propagated gradients (Model 3).

Further improvement was achieved through the introduction of various enhancement techniques T (Model 4), including the L1 distance loss, feature matching, and one-sided label smoothing. These techniques helped stabilize the performance, increase the inception score to 4.00, and prevent the model from synthesizing plausible but unrealistic images.

The highest Inception score of 4.09 was achieved by including self-supervision $\mathcal{L}_{ss}$ (Model 5). This component allowed the model to generate more training samples and explore high-level object information, resulting in a more diverse learned representation. Self-supervision was particularly useful in synthesizing complex objects like birds where the various parts of the object need to be carefully rendered. The results demonstrate the effectiveness of each component in improving the performance of SS-TiGAN.

Besides the inception score, each combination of model settings has synthesized some images from the text description, as depicted in Figure 6. It is observable that the images are getting more visually realistic and semantically consistent with the conditioned text description.

V. CONCLUSION

This paper introduces SSTiGAN, an innovative text-to-image synthesis method that utilizes self-supervision to generate vivid images. This method uses a bi-level architecture with two separate discriminators to maintain stability throughout the synthesis process. Self-supervision addresses the issue of low-data regime by augmenting the training data with rotated variants, and SSTiGAN incorporates various strategies to overcome the limitations of GANs during training. Experimental results demonstrate the superiority of SS-TiGAN compared to existing approaches on two well-known benchmark datasets, the Oxford-102 and CUB, highlighting its

effectiveness in generating high-quality images from text descriptions.

Despite the good performance, SS-TiGAN suffers from some limitations due to limited computing resources. SS-TiGAN was designed with a two-stage architecture, resulting in a maximum output resolution of 128 × 128 pixels. This is lower than the standard 256 × 256 pixels used in many existing works. Additionally, due to resource constraints, a more advanced text encoder, such as a sentence-level and word-level encoder, could not be implemented. As a result, only a sentence-level text encoder was used to obtain the global text embedding.

However, there are advantages to the two-stage architecture and the use of only a sentence-level encoder in SS-TiGAN. The two-stage architecture allows for faster training and inference times, as well as reducing the complexity of the model. Additionally, using a sentence-level encoder simplifies the text input process and can lead to more coherent image generation as it considers the overall meaning of the input text rather than individual words. The use of a sentence-level encoder also makes the model more robust to variations in word choice and grammar within the input text. Overall, while SS-TiGAN may have some limitations, the simplified architecture and use of a sentence-level encoder offer several advantages that make it a promising approach for text-to-image synthesis.

REFERENCES

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2, 2014, pp. 2672–2680.

[2] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[3] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Boosting few-shot visual learning with self-supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jun. 2019, pp. 8058–8067.

[4] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text-to-image synthesis," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1060–1069.

[5] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 217–225.

[6] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug and play generative networks: Conditional iterative generation of images in latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1–11.

[7] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[8] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal, "TAC-GAN–text conditioned auxiliary classifier generative adversarial network," 2017, *arXiv:1703.06412*.

[9] M. Cha, Y. L. Gwon, and H. Kung, "Adversarial learning of semantic relevance in text to image synthesis," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3272–3279.

[10] M. Zhu, P. Pan, W. Chen, and Y. Yang, "DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5795–5803.

[11] L. Gao, D. Chen, J. Song, X. Xu, D. Zhang, and H. T. Shen, "Perceptual pyramid adversarial networks for text-to-image synthesis," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 8312–8319.

[12] X. Huang, M. Wang, and M. Gong, "Hierarchically-fused generative adversarial network for text to realistic image synthesis," in *Proc. 16th Conf. Comput. Robot Vis. (CRV)*, May 2019, pp. 73–80.

[13] D. M. Souza, J. Wehrmann, and D. D. Ruiz, "Efficient neural architecture for text-to-image synthesis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[14] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical text-to-image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7986–7994.

[15] T. Hinz, S. Heinrich, and S. Wermter, "Generating multiple objects at spatially distinct locations," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–23.

[16] W. Li, P. Zhang, L. Zhang, Q. Huang, X. He, S. Lyu, and J. Gao, "Object-driven text-to-image synthesis via adversarial training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12166–12174.

[17] T. Hinz, S. Heinrich, and S. Wermter, "Semantic object accuracy for generative text-to-image synthesis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1552–1565, Mar. 2022.

[18] S. Sharma, D. Suhubdy, V. Michalski, S. Ebrahimi Kahou, and Y. Bengio, "ChatPainter: Improving text to image generation using dialogue," 2018, *arXiv:1802.08216*.

[19] M. Wang, C. Lang, L. Liang, S. Feng, T. Wang, and Y. Gao, "End-to-end text-to-image synthesis with spatial constrains," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 4, pp. 1–19, May 2020.

[20] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1–9.

[21] Z. Qi, C. Fan, L. Xu, X. Li, and S. Zhan, "MRP-GAN: Multi-resolution parallel generative adversarial networks for text-to-image synthesis," *Pattern Recognit. Lett.*, vol. 147, pp. 1–7, Jul. 2021.

[22] S. Sah, D. Peri, A. Shringi, C. Zhang, M. Dominguez, A. Savakis, and R. Ptucha, "Semantically invariant text-to-image generation," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3783–3787.

[23] T. Qiao, J. Zhang, D. Xu, and D. Tao, "MirrorGAN: Learning text-to-image generation by redescription," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1505–1514.

[24] G. Yin, B. Liu, L. Sheng, N. Yu, X. Wang, and J. Shao, "Semantics disentangling for text-to-image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2322–2331.

[25] H. Tan, X. Liu, X. Li, Y. Zhang, and B. Yin, "Semantics-enhanced adversarial nets for text-to-image synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10500–10509.

[26] Z. Wang, Z. Quan, Z.-J. Wang, X. Hu, and Y. Chen, "Text to image synthesis with bidirectional generative adversarial network," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2020, pp. 1–6.

[27] J. Cheng, F. Wu, Y. Tian, L. Wang, and D. Tao, "RiFeGAN: Rich feature generation for text-to-image synthesis from prior knowledge," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1–10.

[28] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1–9.

[29] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.

[30] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 49–58.

[31] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[32] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process.*, Dec. 2008, pp. 722–729.

[33] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD birds 200," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2010-001, 2010.

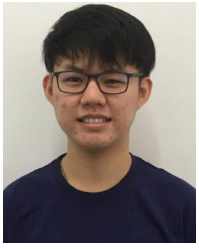
[34] M. Yang, W. Zhao, W. Xu, Y. Feng, Z. Zhao, X. Chen, and K. Lei, "Multitask learning for cross-domain image captioning," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 1047–1061, Apr. 2019.
[35] N. Bodla, G. Hua, and R. Chellappa, "Semi-supervised FusedGAN for conditional image generation," in *Proceedings Eur. Conf. Comput. Vis. (ECCV)*, pp. 669–683, 2018.
[36] D. Peng, W. Yang, C. Liu, and S. Lü, "SAM-GAN: Self-attention supporting multi-stage generative adversarial networks for text-to-image synthesis," *Neural Netw.*, vol. 138, pp. 57–67, Jun. 2021.
[37] Y. X. Tan, C. P. Lee, M. Neo, and K. M. Lim, "Text-to-image synthesis with self-supervised learning," *Pattern Recognit. Lett.*, vol. 157, pp. 119–126, May 2022.
[38] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6199–6208.

YONG XUAN TAN received the B.IT. degree (Hons.) in artificial intelligence from Multimedia University, where he is currently pursuing the Master of Science degree in information technology. His research interests include pattern recognition and computer vision.

CHIN POO LEE (Senior Member, IEEE) received the Master of Science and Ph.D. degrees in information technology in the areas of abnormal behavior detection and gait recognition. She is currently a Senior Lecturer with the Faculty of Information Science and Technology, Multimedia University, Malaysia. Her research interests include action recognition, computer vision, gait recognition, and deep learning.

MAI NEO is currently a Professor with the Faculty of Creative Multimedia, specializing in e-learning and digital learning curriculum and the former Director of Academic Development for Excellence in Programmes and Teaching (ADEPT) with Multimedia University. She is also the Director of the Award-Winning MILE Research Laboratory, the Founding Chairperson of the Centre for Adaptive Multimedia, Education and Learning cOntent Technologies (CAMELOT) Research Centre, and a Managing Editor of the *International Journal of Creative Multimedia* (IJCM). Her research interests include the design of constructivist learning environments, micro-learning, team-based learning, and web-based education. She was a recipient of the 2014 Excellent Researcher Award, an AKEPT Certified Trainer for Interactive Lectures (Level 1, 2, 3), an HRDF Certified Trainer, and certified in Team-Based Learning from the Team-Based Learning Collaborative, USA. She has been a recipient of several TMRnD grants, since 2010, and her projects have won several gold medals in innovation competitions (ITEX, IUCEL, and IPHEX).

KIAN MING LIM (Senior Member, IEEE) received the B.IT. degree (Hons.) in information systems engineering and the Master of Engineering Science (M.Eng.Sc.) and Ph.D. (I.T.) degrees from Multimedia University. He is currently a Lecturer with the Faculty of Information Science and Technology, Multimedia University. His research and teaching interests include machine learning, deep learning, computer vision, and pattern recognition.



JIT YAN LIM received the B.IT. degree (Hons.) in artificial intelligence from Multimedia University, where he is currently pursuing the Ph.D. (I.T.) degree, advised by Dr. Kian Ming Lim. His research interests include machine learning and computer vision.