LFR-GAN: Local Feature Refinement based Generative Adversarial Network for Text-to-Image Generation

ZIJUN DENG, XIANGTENG HE, YUXIN PENG*, Wangxuan Institute of Computer Technology, and National Key Laboratory for Multimedia Information Processing, Peking University, China

Text-to-image generation aims to generate images from text descriptions. Its main challenge lies in two aspects: (1) Semantic consistency, i.e., the generated images should be semantically consistent with the input text; (2) Visual reality, i.e., the generated images should look like real images. To ensure text-image consistency, existing works mainly learn to establish the cross-modal representations via a text encoder and image encoder. However, due to the limited representation capability of the fixed-length embeddings and the flexibility of the free-form text descriptions, the learned text-to-image model is incapable of maintaining the semantic consistency between image local regions and fine-grained descriptions. As a result, the generated images sometimes miss some fine-grained attributes of the generated object, such as the color or shape of a part of the object. To address this issue, this paper proposes a **Local Feature Refinement Based Generative Adversarial Network (LFR-GAN)**, which first divides the text into some independent fine-grained attributes and generates an initial image, then refines the image details based on these attributes. The main contributions are three-fold: (1) **An attribute modeling** approach is proposed to model the fine-grained text descriptions by mapping them into representations of independent attributes, which provides more fine-grained details for image generation. (2) **A local feature refinement** approach is proposed to enable the generated image to form a complete reflection of the fine-grained attributes contained in the text description. (3) **A multi-stage generation** approach is proposed to realize the fine-grained manipulation of complex images progressively, which aims to improve the performance of the refinement and generate photo-realistic images. Extensive experiments on the CUB and Oxford102 datasets show the effectiveness of our LFR-GAN approach in both text-to-image generation and text-guided image manipulation tasks. Our LFR-GAN approach shows superior performance to the state-of-the-art methods. The codes will be released at https://github.com/PKU-ICST-MIPL/LFR-GAN_TOMM2023.

CCS Concepts: • **Computing methodologies** → **Computer vision**; • **Information systems** → *Multimedia information systems*.

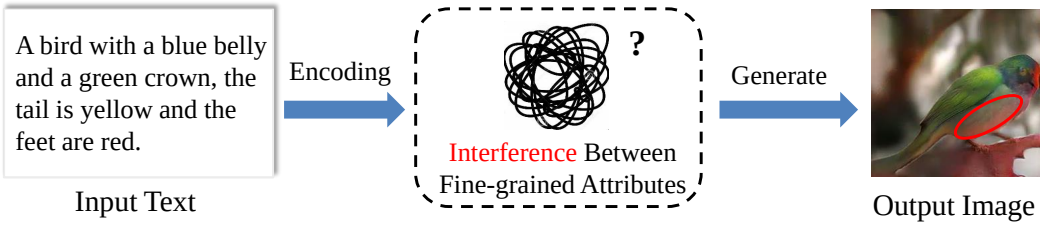
Additional Key Words and Phrases: Local Feature Refinement, Text-to-image Generation, Generative Adversarial Network

ACM Reference Format:
Zijun Deng, Xiangteng He, Yuxin Peng. 2023. LFR-GAN: Local Feature Refinement based Generative Adversarial Network for Text-to-Image Generation. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1, Article 1 (March 2023), 18 pages. https://doi.org/XXXXXXX.XXXXXXX

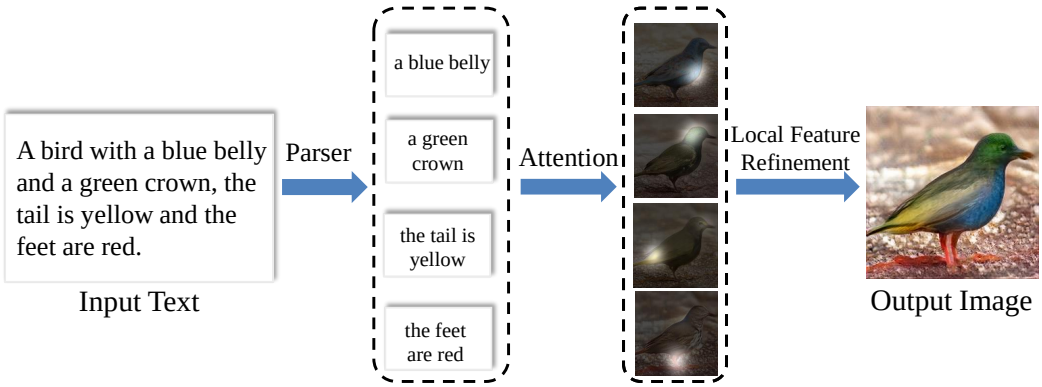
*Corresponding author.

This work was supported by the grants from the National Natural Science Foundation of China (62132001, 61925201, U21B2025, 62272013, U22B2048).
Author's address: Zijun Deng, Xiangteng He, Yuxin Peng, Wangxuan Institute of Computer Technology, and National Key Laboratory for Multimedia Information Processing, Peking University, Beijing, 100871, China, pengyuxin@pku.edu.cn.

1551-6857/2023/3-ART1 $15.00
https://doi.org/XXXXXXX.XXXXXXX



(a) Existing methods



(b) Our LFR-GAN approach

Fig. 1. An overview of our LFR-GAN approach and existing methods. The existing methods fail to generate the blue belly, which is marked in a red circle.

1 INTRODUCTION

Text-to-image generation aims to generate images according to the natural language input text. In recent years, due to the advances in the Generative Adversarial Networks (GANs) [6], great progress has been made in synthesizing photo-realistic images. Due to the wide potential applications of this task, such as visual reading [2] and graphic design [9], text-to-image generation has become one of the most active research areas.

The main challenge of the text-to-image generation lies in two aspects:(1) Semantic consistency, i.e., the generated images should be semantically consistent with the input text; (2) Visual reality, i.e., the generated images should look like real images. To ensure text-image consistency, the existing methods usually use text encoder and image encoder, commonly long short-term memory (LSTM) [8] and convolutional neural network (CNN) [11], to explore the cross-modal semantic association between the image and text. In recent years, a series of models [3, 33, 34, 38, 40] use the StackGAN architecture [37] to generate images, which pre-train an LSTM text encoder and a CNN image encoder to learn text-image cross-modal representation to address the problem of semantic consistency.

However, the representation capability of the fixed-length embeddings pales before the highly complicated and flexible natural language [14]. The text-to-image model can easily learn to handle a single attribute but struggle in understanding the combination of multiple attributes and mapping them into representations. Therefore, the learned text-to-image model is incapable of maintaining

the semantic consistency between the image local region and fine-grained description. The fine-grained description sometimes even interferes with each other in the encoding process, resulting in information loss and image distortion in generation as shown in Fig. 1(a).

To address this issue, this paper proposes a Local Feature Refinement Based Generative Adversarial Network (LFR-GAN). The idea of our approach is simple yet effective: mitigate the encoding difficulty by dividing the free-form text descriptions into independent fine-grained attributes of the generated object. We combine the grammar analysis and text chunking methods to parse the input text into independent attributes to ensure the robustness and accuracy of the attribute modeling. The parsed independent attributes then forward separately into an encoder module and generation module to generate local feature maps. In the next step, StyleGAN [10] is used to merge these feature maps into the resulting image. We generate an initial image and refine its local features progressively via multi-stage generation by manipulating the latent feature of StyleGAN. As shown in Fig. 1(b), We propose to divide the text description into several independent fine-grained attributes and encode them separately to obtain the corresponding attention maps. The above design ease the text modeling and better reveal the complete fine-grained textual information. The contributions of this paper can be summarized as follows:

- **An attribute modeling** approach is proposed to model the fine-grained text descriptions by mapping them into representations of independent attributes. Specifically, we design a sentence parser to obtain independent fine-grained attributes and encode them separately, which could capture the sentence information more completely and provide more fine-grained details for image generation.
- **A local feature refinement** approach is proposed to enable the generated image to form a complete reflection of the fine-grained attributes contained in the text description. Compared with previous methods, we refine the images guided by attention maps and feature maps of the independent fine-grained attributes, which enables the generated image to have more vivid details consistent with the input text.
- **A multi-stage generation** approach is proposed to realize the fine-grained manipulation of complex images progressively by decomposing the manipulation into three stages that each one conducts a relatively simple manipulation. Specifically, the first stage controls the shape and posture of the object in the image; the second stage manipulates the regional feature of the image; the last stage enriches the detail of the image. By decomposing the image generation into multiple stages, the refinement difficulty of each stage is relatively reduced, which can further improve the performance of the refinement and also generate more photo-realistic images.

Extensive experiments on the CUB and Oxford102 datasets show the effectiveness of our LFR-GAN approach in both text-to-image generation and text-guided image manipulation tasks. Our LFR-GAN approach shows superior performance to the state-of-the-art methods.

2 RELATED WORKS

In this section, we briefly review the related works of text-to-image generation and text-guided image manipulation. Since our approach also leverages text structure modeling technics in NLP, we also discuss its related works in this section.

2.1 Text-to-image Generation

Text-to-image generation aims to generate images from text descriptions. Generative adversarial network is widely used in the text-to-image generation task. StackGAN [37] and StackGAN++ [38] propose a two-stage generation model in which the first stage sketches the overall shape and colors

of the object, and the second stage adds high-resolution details. AttnGAN [33] introduces a cross-modal attention model to generate fine-grained detail consistent with the corresponding words in the text. DM-GAN [40] introduces a dynamic memory to highlight relevant words and correct wrong-generated features. MA-GAN [34] introduces single sentence generation and multiple sentence discrimination (SGMD) modules to improve the reliability of the generated results by reducing the difference of images generated by similar sentences. DAE-GAN [22] represents text information from multiple granularities such as sentence level, word level, and aspect level. Sentence-level text representation is used to generate low-resolution images, and then word-level and aspect-level representations are used to refine the images. RiFeGAN [3] exploits a caption-matching model to extract and refine captions from prior knowledge. Recently, as StyleGAN [10] achieves great success in the image-generation task, StyleGAN-based methods occur in the field. StyleGAN uses a coarse-to-fine convolution structure to adjust the attributes of the image with the latent code, which realizes the control of the image style at different scales. The scale-specific control of the synthesis enables StyleGAN architecture to better disentangle the generated attributes. CI-GAN [27] uses StyleGAN as the generator backbone to synthesize high-quality images and introduced GAN-inversion and latent space alignment model to ensure image-text consistency. Lafite [39] develops a CLIP+StyleGAN architecture, which first maps the input text into StyleGAN latent space, then employs StyleGAN structure to generate output images.

The above methods introduce fine-grained text information into the generation process to improve text-image consistency. However, these methods ignore the interference of text information in the encoding process as well as the grammar structure of the input text. Our approach proposes a sentence parser to obtain fine-grained attributes and encode them separately, which could capture the sentence information more completely and provide more fine-grained details for image generation.

2.2 Text-guided Image Manipulation

Text-guided image manipulation aims to modify images according to the given natural language descriptions. Existing works propose various methods to establish the text-image correlation and realize image manipulation controlled by text. SIS-GAN [4] uses the GAN-based encoder-decoder architecture to transform the image concerning the text description while preserving the image feature that is irrelevant to the text. TA-GAN [17] introduces an adaptive word-level local discriminator to classify and modify the fine-grained image attributes. ManiGAN [12] proposes an affine combination module (ACM) and a detail correction module (DCM) to select and modify image regions relevant to input text, which achieves finer manipulation. Lightweight-GAN [13] proposes a lightweight architecture to achieve competitive manipulation performance with a much smaller number of parameters. In recent years, due to the impressive capability shown by StyleGAN in image style manipulation, a lot of works employ StyleGAN structure to conduct text-guided image manipulation. StyleCLIP [19] adopts CLIP to capture the text-image semantic space and establish the connection to StyleGAN latent space, thus allowing faster and more stable text-based manipulation. TediGAN [31] trains a text encoder to map the text into the StyleGAN latent space to control the image style with the input text.

Although the above approaches can establish the text-image connection and manipulate images with the input text, they still lack the capability in dealing with complex text descriptions. When the input text is complicated, modification omission or image distortion might occur after the manipulation. Besides, most StyleGAN-based approaches mainly focus on finding meaningful manipulation by exploring the latent space of StyleGAN, rather than finding general manipulating methods for custom user input. Therefore, the existing approaches are unsatisfactory in fine-grained manipulation guided by free-form input text, especially on images with complex semantic content. Compared with previous methods, we first parse the guided text into independent fine-grained

attributes, then refine the images guided by attention maps and feature maps of these attributes, which realizes fine-grained manipulation guided by complex input text progressively.

2.3 Text Modeling

2.3.1 Text Structure Modeling

. Text structure modeling aims to analyze the semantic structure of the input text. Existing NLP works mainly use grammar analysis and text chunking to model the structure of textual input.

Grammar analysis is the NLP task to analyze the grammatical components of a given sentence. Stanford CoreNLP [15] proposes a grammar analysis process: first, divide the text into sentences so that each sentence can be analyzed separately; second, segment the sentence into word sequence; third, carry out part of speech analysis and mark the part of speech of each word; Finally, the grammar analysis is carried out to form the grammar tree of the sentence from bottom to top based on the part of speech of the word.

Text chunking is the NLP task to divide the text into grammatically related and non-overlapping phrases. Sang et al. [24] first propose the task of text chunking and constructed the conll2000 dataset for this task. The general method of text chunking is to embed words first to get the part of speech mark of each word. Then the phrase is marked according to the part of speech and context of the word to get the final text chunking result. Flair [1] proposes a unified framework that can embed words and any word combination without engineering efforts for special texts.

The goal of the attribute modeling in this paper is similar to text chunking, but not the same. The common part is that the text is divided into semantically related parts. The difference is that each block of the text chunking is a phrase, while the goal of this paper is that each block is an independent fine-grained attribute of the generated object. Each independent attribute may be a phrase, such as "black feathers"; a combination of multiple phrases, such as "dark grey on the back of the head"; or a short sentence, such as "The bill is short and pointed.". In the last two cases, a single phrase can not represent the full meaning of the characteristics. Therefore, it's necessary to analyze the independent attributes in the text. Our approach combines grammar analysis and text chunking methods to ensure the robustness and accuracy of the attribute modeling.

2.3.2 Text attribute modeling

. Text attribute modeling aims to analyze the linguistic attribute in the sentence. Existing NLP works mainly use text attribute transfer to analyze and control the attributes in the text.

Text attribute transfer is the NLP task to analyze the linguistic attribute (eg. emotion) in the sentence and alter them from one type to another, for example, transfer from positive to negative. Melnyk et al. [16] use a collaborative classifier to find linguistic attributes in the sentence and disentangle them from other contents in the sentence, which enables transferring of text attributes without modifying other text contents. Yang et al. [35] leverage language models as discriminators to provide token-level responses in the target attribute during training, which helps achieve higher accuracy in attribute modeling tasks. Fu et al. [5] find emotional words play an important role in sentences' attributes. Therefore, they propose to locate and modify the pivot words in the sentence to achieve accurate attribute transfer. Wang et al. [29] propose a flexible text attribute transfer framework to control the degree of transfer. They first use a Transformer-based autoencoder to learn the latent representation of text and then edit the latent representation to realize controllable modification of the text attribute. Yi et al. [36] further employ generative flow to establish instance-level underlying attributes, which form a more discriminative latent space for sentences' style. Xu et al. [32] develop a bi-directional reinforcement learning algorithm for Chinese text attribute transfer, which designs a style transfer reward to promote the capture of the text style, and a content preservation reward to preserve the other text content from missing.

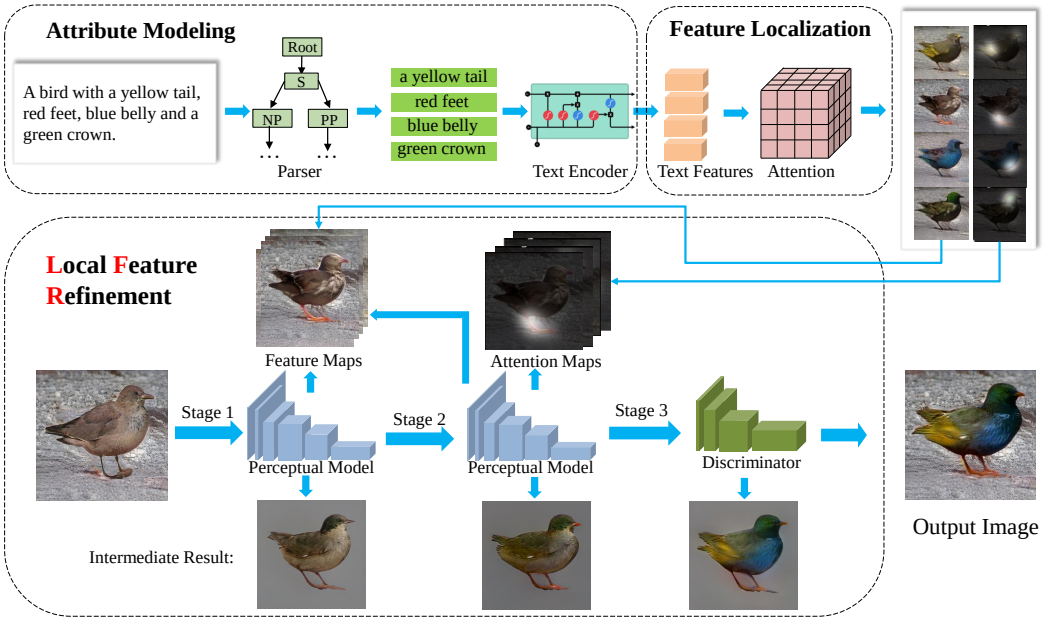


Fig. 2. The framework of our LFR-GAN approach.

The attribute modeling in our approach is different from the attribute modeling technique in text attribute transfer in two folds: (1) Our approach focuses on the attributes of the generated objects, while text attribute transfer focuses on linguistic attributes like emotion and sentiment. (2) The attributes extracted in our approach are independent, as we assume that each of them describes a different part of the generated object. However, the linguistic attributes may be related to each other (eg. happy and positive).

3 OUR LFR-GAN APPROACH

As shown in Fig. 2, our LFR-GAN approach consists of 3 components: attribute modeling, feature localization, and local feature refinement. In attribute modeling, our approach first divides the input text into several independent fine-grained attributes. Then, the feature maps and attention maps are generated for the attributes respectively in the following feature localization processing. In the local feature refinement, our approach refines the fine-grained features of the initial image in the multi-stage generation process. The feature maps and attention maps are used in different stages of the synthesis process.

3.1 Attribute Modeling

Our attribute modeling aims to divide the input text into several independent fine-grained attributes as mentioned in the previous section. The word *independent* means excluding any word from an independent attribute would damage its meaning.

We notice that the noun phrases in the text description, like "a blue belly", and "a green crown" are the essential parts of the text. Since these noun phrases play an important role to generate meaningful and independent attributes. Therefore, we first use text chunking in FLAIR [1] to extract these noun phrases from the text. Compared with the grammar tree which also contains noun phrases, the text chunking exhibit superior performance of noun phrase analysis. Thus, we first adopt text chunking to further improve the robustness of the attribute modeling.

However, we also observe that sometimes a single noun phrase cannot form an independent attribute alone. For example, in the sentence "The feet are red.", the verb phrase "are red" and the noun phrase "The feet" together form an independent fine-grained attribute. Therefore, based on the extracted noun phrases, we combine them with other components of the sentence to form an independent attribute. We design an algorithm based on the Stanford CoreNLP grammar tree [15] to combine the core noun phrase in the sentence with its adjacent verb phrase (VP) and prepositional phrase (PP), to form an independent attribute as shown in Algorithm 1. Our algorithm sets a standard that sentence components are independent attributes (noun phrases with adjectives), recursing the syntax tree from bottom to top. When traversing a node, if the node meets the standard, it will be put into the parse result. Otherwise, the syntax components of sibling nodes will be merged. Through the merging algorithm, our proposed text modeling can realize the extraction of independent attributes in the text.

Algorithm 1 Independent attribute extraction

Input: A grammar tree T, noun phrases in the sentence $\{np_1, np_2, ...\}$;
Output: the parse result A=$\{C_1, C_2, ...\}$;

```
// verb phrase = VP, prepositional phrase = PP, noun phrase = NP
function DFS(node, A)
    cnt ⇐ count of NP child in the current node; // Recursive implementation
    for child C_i ∈ node do
        DFS(child)
    if cnt==0 then
        if current node is NP then
            put child C_i in A;
        if current node is VP and PP then
            merge current node with last noun phrase without an adj in A;
    return ;
delete the child of noun phrases in T;
A ⇐ ∅ // initialize A;
DFS(T.root, A) // put independent attributes into A by DFS;
return A;
```

3.2 Feature Localization

The feature localization in our model aims to mark the area of local feature which provides references for our refinement. We use a stack model here to generate the feature maps and attention maps. The low-resolution feature is generated from the initial image and is fixed in this process. The high-resolution details are added according to the embedding of each independent attribute obtained from the text modeling. Attention maps are calculated according to the similarity of regional image features and the text features, by using [40]. To eliminate the influence of redundant overlapping areas in the obtained attention maps, we assume that each independent attribute should focus on a different part of the object. Therefore, a new algorithm is designed to handle the overlapping areas in attention maps.

Our proposed algorithm assumes that a bigger value in the attention maps represents a more distinct feature in the feature maps. Therefore, we always eliminate points in the attention map that has a smaller value than other attention maps. For the i-th description, the attention areas in its

attention map are represented as a_i. For the attention map i, we first calculate the total area A_{i-1} in the first $i-1$ maps:

$$A_{i-1} = a_1 \cup a_2 \cup ... \cup a_{i-1} \tag{1}$$

Note that the $\cup$ operator denotes selecting the bigger value of the two attention maps at each point. After that, we select the attention maps that have less overlapping area. If the attention areas in the attention map overlap too much with the previous areas, this attention map will be considered invalid and directly ignored. Specifically, attention map i will be ignored if:

$$\begin{cases} S(a_i > A_{i-1})/S(a_i) < \alpha & (2) \\ S((A_i > \mu) \cap (a_i > \mu))/S(a_i) > \beta & (3) \end{cases}$$

Where $S(.)$ means the measure of the area of the region a, the $\cap$ operator denotes the overlapping area of the two attention maps, and the α and β are the distinct ratio and overlapping ratio. $a_i > A_{i-1}$ means the area that a_i has a bigger value than A_{i-1} in the attention map. Equation 2 demands the attention area should have distinct features as we assume that each independent attribute should be different from the others. Equation 3 demands the attention area should not overlap too much with other areas as we assume that each independent attribute should focus on a different part of the object.

After the selection step, there are m areas left, which can be represented as: $a_{l_1}, a_{l_2}, ..., a_{l_m}$. The overlapping areas can be further eliminated using the following operation:

$$a_{l_i} = a_{l_i} \cap (a_{l_i} < A_{l_m}) \tag{4}$$

During the formal steps, each point in A_{l_m} is the biggest one among $a_{l_1}, a_{l_2}, ..., a_{l_m}$. The principle of this step is that if the value of a certain point in the a_{l_i} is smaller than A_{l_m}, then there must be another area that has a bigger value in this point, which means this point is in the overlapping areas. Therefore, all the points of a_{l_i} in the overlapping area are eliminated in this way.

3.3 Local Feature Refinement

Based on the feature localization results, the proposed local feature refinement aims to modify the regional feature of the initial image, to make it as close to the features in the attention area as possible. Our approach proposes to modify the feature by optimizing the latent space in StyleGAN of the image. To do so, three different losses including a shape loss L_s, an attention loss L_a, and a discriminator loss L_d are used in different stages of the refinement. These three losses are formulated as below to control the shape information, local features, and the details of the image respectively:

$$L_s = \|F(x) - F(t)\|_2 \tag{5}$$

$$L_a = \sum \|F(x \cdot mask_i) - F(t_i \cdot mask_i)\|_2 \tag{6}$$

$$L_d = softplus(-D(x)) \tag{7}$$

where $F(.)$ denotes the VGG [25] feature, $D(.)$ denotes the discriminator of StyleGAN, x denotes the manipulated image, t denotes the target image, which will be further explained later, t_i and $mask_i$ denote the feature maps and attention maps respectively. The $softplus(.)$ denotes:

$$softplus(x) = log(1 + e^x) \tag{8}$$

Although various methods [19, 30, 31] are proposed to manipulate the latent space of StyleGAN, they mainly focus on finding meaningful manipulation by exploring the latent space of StyleGAN, rather than finding general manipulating methods for custom user input. Therefore, they can successfully do certain coarse-grained manipulations but are unsatisfactory in the more challenging fine-grained manipulation guided by custom text input, especially on images with complex contents.

Since the fine-grained manipulation of complex images is difficult, we divide the manipulation into three stages that each one conducts a relatively simple manipulation. Specifically, the first stage uses shape loss to control the shape and posture of the object in the image; the second stage adds attention loss to manipulate the regional feature of the image; the last stage use discriminator loss to enrich the detail of the image. To further reduce the difficulty of manipulation, the whole refinement process is carried out without background. To achieve this, we use U2Net [21] to remove the background of the target image. When the refinement of the object is finished, the removed background will be added back to obtain the final results.

In the first stage, the target image t is set as a "reference image" spliced according to the feature maps and the attention maps. This reference figure contains all the features in the text description, which may be discordant with each other. Since we just aim to modify the shape of the generated object in this stage, we propose to handle this discordant feature issue in the following stages. The loss L_1 in this stage can be expressed as:

$$L_1 = L_s \tag{9}$$

In the second stage, the target image t remains unchanged. Our approach further aims to correct the local features of the image as well as keep the shape of the object unchanged. To achieve this, the attention loss is jointly optimized with the shape loss in this stage which can be expressed as:

$$L_2 = L_s + \lambda_a L_a \tag{10}$$

In the third stage, the target image t is set to the last modified image of the previous stage. This design considers that this stage only modifies the texture details in the image to make the image more realistic. Therefore, the modification of this stage should be relatively small. The loss L_3 in this stage can be expressed as:

$$L_3 = L_s + \lambda_d L_D \tag{11}$$

In the above formula, λ_a and λ_d are hyper-parameters to balance different loss terms. After three-stage refinement, the background of the initial image is added to the modified image again to get the final result.

4 EXPERIMENT

4.1 Datasets

We conduct our experiments on the Caltech-UCSD Birds-200-2011 (CUB) [26] and Oxford-Flower-102 (Oxford102) [18] datasets. The statistics of the two datasets are summarized in Table 1, and their detailed information is as follows:.

- **CUB dataset** is a bird image dataset. It contains 200 bird categories with 11788 images, in which 150 categories with 8855 images are used for training, and the other 50 categories with 2933 images for testing. There are ten captions for each image in the CUB dataset.
- **Oxford102 dataset** is a flower image dataset. It includes 102 categories with 8189 images, with 20 categories for testing and the others for training. Each image in the Oxford102 has 10 captions.

Table 1. Statistics of datasets.

Dataset	Train set		Test set		Captions per image
	Class	Images	Class	Images	
CUB	150	8855	50	2933	10
Oxford102	82	6149	20	2040	10

4.2 Evaluation Metrics

To fairly compare our LFR-GAN approach with state-of-the-art methods, following [33, 37, 40], we quantify the performance of our LFR-GAN approach in terms of Inception Score (IS) [23] and Frechet Inception Distance (FID) [7]. IS calculates the KL-divergence between the conditional class distribution and the marginal class distribution of predicted image labels by a fine-tuned inception-v3 network. A higher IS means the generated images have higher diversity, that the num of images of each class is closer, meanwhile each image more clearly belongs to a category. FID calculates the distance of the inception-v3 feature between the synthesis images and real images. Lower FID means that the generated images are closer to the real images.

4.3 Implementation Details

The sentence parser in the attribute modeling approach takes all 10 captions of one image as the input to acquire more information about the synthesis object. To eliminate repeated semantic content, we first find out the key noun words in each independent attribute, such as "wings", "feet" and so on. We use the DAMSM [33] to calculate the word embeddings of these words. Two keywords will be considered to have duplicated semantics if the cosine similarity is higher than 0.75. In that case, one description will be removed from the encoding process.

The generator framework of our approach is based on Lafite [39]. The initial resolution in the feature localization part starts from 64x64. Then, we refine the feature maps to the resolution of 128x128 and 256x256. The attention maps are acquired from the second refining step. The α, β and μ in the feature localization are set to 0.4, 0.9, and 0.1 respectively. We use the StyleGAN2 network to manipulate the local feature of the images. The local feature manipulation is conducted in w^+ of the StyleGAN latent space [10]. The λ_a and λ_d in the local feature refinement are set to 3 and 0.08 respectively. The training of the three stages of local feature refinement lasts 300 epochs, 500 epochs, and 200 epochs, respectively.

4.4 Comparison with the State-of-the-art

To verify the performance of our LFR-GAN approach, we first conduct quantity experiments on text-to-image generation and text-guided image manipulation tasks. Our LFR-GAN approach shows superior performance to the state-of-the-art methods in both two tasks.

4.4.1 Experiments on Text-to-image Generation Task.

Table 2. The results of text-to-image generation task.

Method	CUB		Oxford-102	
	IS ↑	FID ↓	IS ↑	FID ↓
StackGAN [ICCV 2017] [37]	3.70	51.89	3.20	55.28
StackGAN-v2 [TPAMI 2018] [38]	4.04	15.30	3.26	48.68
attnGAN [CVPR 2018] [33]	4.36	23.98	3.91	-
DM-GAN [CVPR 2019] [40]	4.75	16.09	4.03	<u>41.39</u>
MirrorGAN [CVPR 2019] [20]	4.56	-	-	-
RiFeGAN [CVPR 2020] [3]	5.23	-	<u>4.53</u>	-
MA-GAN [TIP 2021] [34]	4.76	21.66	-	-
DAE-GAN [ICCV 2021] [22]	4.42	15.19	-	-
Lafite [CVPR 2022] [39]	<u>5.97</u>	<u>10.48</u>	-	-
LFR-GAN (Ours)	**6.15**	**9.96**	**4.70**	**35.27**

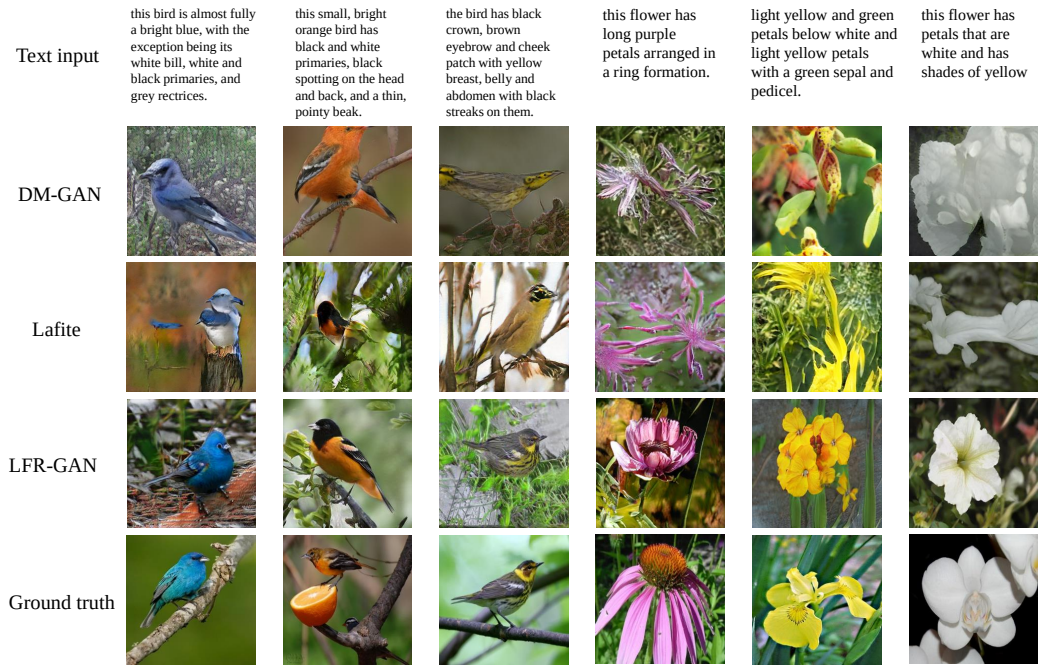


Fig. 3. Examples of images synthesized by DM-GAN, Lafite, and our LFR-GAN approach conditioned on text descriptions from the test set of CUB and Oxford102 datasets. The corresponding real images are also shown for reference denoted as "Ground truth" in the last row.

To verify the performance of our LFR-GAN approach, we compare our LFR-GAN approach with the state-of-the-art methods of the text-to-image generation task on the CUB and Oxford102 test datasets. As shown in Table 2, our approach achieves the best IS and FID on the CUB dataset and Oxford102 dataset. On the CUB dataset, compared with Lafite, our LFR-GAN approach improves the IS from 5.97 to 6.15, and improves the FID from 10.48 to 9.96. The performance improvement is mainly acquired by improving the representation of complex text. The previous SOTA Lafite method just uses a CLIP text encoder to acquire the text representation. Our LFR-GAN approach parses the complex text into independent fine-grained attributes and encodes them separately, which could capture the sentence information more completely and provide more fine-grained details for image generation. Besides, our local feature refinement and multi-stage generation enable the synthetic images to have more vivid fine-grained details, which improves the image quality of the image and achieves better FID and IS. On the Oxford102 dataset, compared with RiFeGAN, our LFR-GAN approach improves the IS from 4.53 to 4.70; compared with DM-GAN, our LFR-GAN approach improves the FID from 41.39 to 35.27. The performance improvement is mainly achieved due to better text information modeling of our attribute modeling approach. Our local feature refinement and multi-stage generation also contribute to performance improvement by generating more vivid fine-grained details.

4.4.2 Experiments on Text-guided Image Manipulation Task.

To further evaluate the performance of our LFR-GAN approach, we conduct experiments on a different task, text-guided image manipulation. Text-guided image manipulation aims to modify

images according to the given natural language descriptions. Our local feature refinement approach aims at modifying the initial image based on the text description, which is consistent with the text-guided image manipulation task. As shown in Table 3, the results of ManiGAN and Lightweight-GAN are from [28], and the rest are from their paper. The FID result is not included in the table as the previous methods do not report their FID.

Table 3. The results of text-guided image manipulation task on CUB dataset.

Method	IS ↑
SISGAN [ICCV 2017] [4]	2.24
TAGAN [NeurIPS 2018] [17]	3.32
ManiGAN [CVPR 2020] [12]	4.19
Lightweight-GAN [NeurIPS 2020] [13]	4.66
LFR-GAN (Ours)	**5.37**

Our LFR-GAN approach achieves the best IS in the text-guided image manipulation task, which is 0.71 higher than the previous best result of 4.66. The performance improvement of our approach is mainly brought about by the optimization of complex text representation problems by text modeling. Previous methods send the whole complex text into the text encoder, resulting in distortion or missing information. Differently, the complex text is divided into independent attributes by our approach and sent to the text encoder separately to avoid this problem.

4.5 Visual Quality Comparison Results

For qualitative evaluation, Fig. 3 shows text-to-image synthesis examples generated by our LFR-GAN approach and the state-of-the-art methods. Our approach generates images with more vivid details as well as clearer backgrounds compared with the exhibited methods. For example, in the second column, our generated image clearly demonstrates the attribute "a thin, pointed beak", while other methods fail to generate this attribute. Besides, in this example, the bird in our generated image can clearly be seen standing on a leafy branch, while the background of other images is vague and unrealistic. The superior performance of our approach is because our approach encodes the text input more accurately and can better express the text information. Moreover, the results shown in Fig. 3 also demonstrate that images generated by our approach are closer to the real image as our approach can better model the input text and generate fine-grained details.

To evaluate the effectiveness of our multi-stage refinement process, as shown in Fig. 4, we demonstrate images generated from different stages in the refinement process that each image is 100 epochs apart. Recall our approach, we divide the refinement process into 3 stages, to manipulate the object shape, regional features, and detail of the image respectively. Images in stage 1 demonstrate that the shape of the bird can be modified gradually into the correct shape. Images in stage 2 vividly display the refinement of local features. The characteristics of the bird, including wings, feet, crown, and belly are modified gradually into the right color. Images in stage 3 present the detail change of the bird, that the texture of the bird is enriched in the marked area.

To evaluate the effectiveness of the attribute modeling, we demonstrate examples of independent fine-grained attributes parsed by our attribute modeling approach in Table 4. The text descriptions shown in the table are from the CUB dataset. To clearly display the parsed result, the independent fine-grained attributes in the text descriptions are marked in bold. Due to the limitation of the width of the table, some sentences are partly shown in the table. The results show that our attribute modeling approach is capable of handling various situations, such as clauses (case 1, 4); prepositional phrases (case 2, 3, 5, 7, 10); special characters (case 4); multiple adjectives (case 3, 6); multiple nouns (case 8,

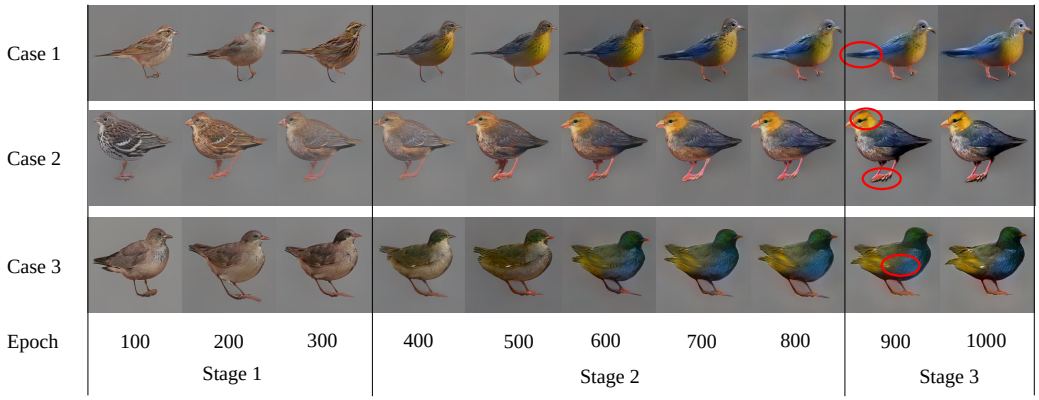


Fig. 4. Generated images in different stages of local feature refinement, each image is 100 epochs apart. The refinement lasts 1000 epochs. To reveal the detailed changes in the last stage, we mark the image areas that are modified in the refinement.

Table 4. Examples of independent fine-grained attributes. The independent fine-grained attributes in the text descriptions are marked in bold.

No.	Text descriptions and independent fine-grained attributes
1	this bird has **wings that are black** and has **a long black bill**.
2	a bird with **a large black bill with downward curve** and **white superciliaries**.
3	bird is **black with brown on its stomach** and has **a long, pointy beak**.
4	this bird has **a long neck that is grainy** and **a pastel orange/blue narrow beak**.
5	**this bird is all brown** with **slight cream colored speckles along its neck**.
6	the **black wings** have **brown wingbars**, **the bill is short and pointed**.
7	**the bird is brown** with **a white ring around the beak** and a **brown curved beak**.
8	**a medium bird** with **a gray body, feet, wings and bill**.
9	**a large bird** with **an expansive wing span** but with **a small head and beak**.
10	this is **a medium bird, grey** with **webbed feet**, **darker grey on the back of the head**.

9). Our parser accurately analyzes the input text and finds out the independent fine-grained attributes successfully in all these examples.

4.6 Ablation Study

Table 5. Effectiveness of each component in our LFR-GAN approach. The reported results are IS↑ metrics in CUB and Oxford102 datasets.

Architecture	CUB	Oxford102
LFR-GAN-w/o refinement	5.95	4.47
LFR-GAN-w/o attribute modeling	6.04	4.56
LFR-GAN-w/o feature localization	6.06	4.59
LFR-GAN-w/o eliminating overlapping areas	6.08	4.61
LFR-GAN	**6.15**	**4.70**

To verify the effectiveness of each component in our LFR-GAN approach, we conduct ablation studies on the test set of the CUB. Our LFR-GAN approach consists of three components including text modeling, feature localization, and local feature refinement. First, we define a baseline that removes the local feature refinement module from LFR-GAN, which is the "LFR-GAN w/o refinement" in Table 5. Second, we remove the attribute modeling process from the model, which is the "LFR-GAN w/o attribute modeling" in the table. The whole text is sent to the encoder for coding, and the attention map is generated for each word to refine the image feature. Third, we remove the feature localization from our architecture, which is the "LFR-GAN w/o feature localization" in the table. Finally, we remove the elimination of overlapping areas in the feature localization, which is the "LFR-GAN w/o eliminating overlapping areas" in the table.

The results in Table 5 show the effectiveness of each component in our LFR-GAN approach. The IS of LFR-GAN without refinement drops by 0.2, which means that our local feature refinement method enables the generated images to have more vivid details and better quality. The IS of LFR-GAN without attribute modeling drops 0.11, which indicates that our attribute modeling helps better express the input text information. The performance of LFR-GAN without feature localization is also poorer, which is due to our feature localization helping the refinement process better manipulate the fine-grained image features. The performance of LFR-GAN without eliminating overlapping areas drops to 6.08, which means eliminating overlapping areas helps to acquire better attention maps in feature localization. The results on the Oxford102 dataset also show the effectiveness of our components. The IS metrics of LFR-GAN without attribute modeling, feature localization, and refinement, which refers to “LFR-GAN-w/o attribute modeling”, “LFR-GAN-w/o feature localization” and “LFR-GAN-w/o refinement” in Table 5, falls by 0.14, 0.11, and 0.23, respectively. The attribute modeling extracts fine-grained object attributes from the input text. The feature localization finds the local region of these attributes and provides attention maps for the feature refinement. The local feature refinement refines the images’ local region according to the fine-grained attributes, which enriches the detail and generates more photo-realistic images. All components are indispensable for performance improvement, combining all of them to form our whole LFR-GAN model can achieve the best performance.

Table 6. Ablation study on text chunking and grammar analysis of attribute modeling. We report the IS ↑ metrics in CUB and Oxford102 datasets.

Architecture	CUB	Oxford102
LFR-GAN-w/o text chunking	6.01	4.52
LFR-GAN-w/o grammar analysis	6.03	4.57
LFR-GAN-w/o attribute modeling	6.04	4.56
LFR-GAN	**6.15**	**4.70**

To further analyze the effectiveness of attribute modeling, we remove the text chunking and grammar analysis from our architecture, which refers to “LFR-GAN-w/o text chunking” and “LFR-GAN-w/o grammar analysis” in Table 6. Then, we remove both of them, referring to “LFR-GAN-w/o attribute modeling”, which means the whole attribute modeling process is removed from our architecture. The results of LFR-GAN without the text chunking and grammar analysis fall to 6.01 and 6.03 in the CUB dataset, and drop to 4.52 and 4.57 in the Oxford102 dataset, lower than the results of attribute modeling. The text chunking finds the core noun phrases of the input sentences, which plays an important role in forming meaningful and independent object attributes. The grammar analysis combines these core noun phrases with other components of the sentence to enrich and

complete the information of the independent object attributes. Therefore, the attribute modeling without any of them cannot well function and will damage the performance.

Table 7. Ablation study on removing background operation in local feature refinement. The results are IS ↑ metrics in CUB and Oxford102 datasets.

Architecture	CUB	Oxford102
LFR-GAN-w/o refinement	5.95	4.47
LFR-GAN-w/o removing background	6.09	4.60
LFR-GAN	**6.15**	**4.70**

To study whether removing the background is helpful in local feature refinement, we remove this operation, which is the "LFR-GAN w/o removing background" in Table 7. The results of our approach drop to 6.09 without removing the background of the initial image, which shows removing the background indeed mitigates the difficulty of fine-grained refinement of complex images and helps generate images with more vivid details.

To analyze the sensitivity of the hyper-parameters of our approach, we conduct hyper-parameter analysis experiments on the test set of the CUB. The hyper-parameters include α, β, μ, λ_a, λ_d. The hyper-parameter α is the distinct ratio in the selection step of feature localization. A bigger α demands the attention map to have more distinct features. The hyper-parameter β is the overlapping ratio in the selection step of feature localization. A smaller β demands the attention map has fewer overlapping areas. The hyper-parameter μ is the threshold in equation 3 that filters the points that have a small value in the attention map. The hyper-parameter λ_a is used to balance the shape loss and attention loss. The hyper-parameter λ_g is used to balance the shape loss and discriminator loss.

The results in Table 8 show the influence of each hyper-parameter in our LFR-GAN approach on the results. Either increasing or decreasing the value of these hyper-parameters would damage the performance. A small variation of α and β causes IS of LFR-GAN to drop about 0.08, which is because a bigger α or smaller β would exclude necessary attention areas and lead to the omission of attributes. A smaller α or bigger β would include wrong-generated attention areas in the attention maps. The value of μ has little influence on the result, which shows that μ is less sensitive to the

Table 8. Hyper-parameter analysis of Local feature refinement. The variable of each experiment is underlined.

Hyper-parameters					IS ↑
α	β	μ	λ_a	λ_d	
<u>0.3</u>	0.9	0.1	3	0.08	6.09
<u>0.5</u>	0.9	0.1	3	0.08	6.06
0.4	<u>0.85</u>	0.1	3	0.08	6.10
0.4	<u>0.95</u>	0.1	3	0.08	6.07
0.4	0.9	<u>0.05</u>	3	0.08	6.11
0.4	0.9	<u>0.15</u>	3	0.08	6.13
0.4	0.9	0.1	<u>2</u>	0.08	5.99
0.4	0.9	0.1	<u>4</u>	0.08	5.97
0.4	0.9	0.1	3	<u>0.06</u>	6.12
0.4	0.9	0.1	3	<u>0.10</u>	6.11
0.4	**0.9**	**0.1**	**3**	**0.08**	**6.15**

performance of LFR-GAN, but deleting a proper number of small value points in the attention maps still benefits the results. The variation of λ_a causes the IS of LFR-GAN to drop by nearly 0.17, which shows that the value of λ_a is highly sensitive to the results. A smaller λ_a would lead to inadequate local feature refinement and damage the text-image consistency while a higher one cannot well maintain the shape of the generated object while refining the local feature. The change of λ_g has a limited impact on the results, which means the value of λ_g is also less sensitive to the performance, but adding discriminator loss to the model would still improve the performance.

4.7 Failure Case analysis

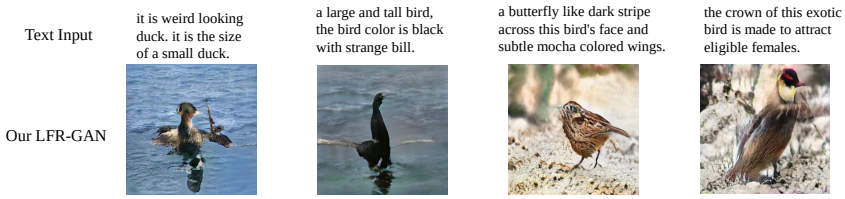
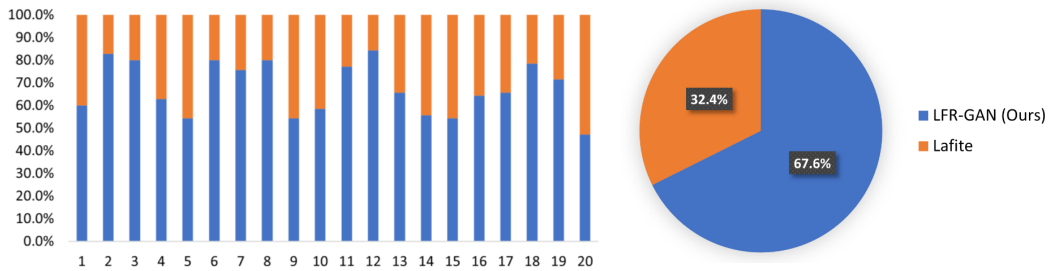


Fig. 5. Some failure cases of our approach.

Some failure cases are presented in Fig. 5. The failure cases of our LFR-GAN approach are mainly due to the unclear attributes in the input text. These given attributes are vague and hard to refine in the generation. For example, "wield looking" in the leftmost case, "strange bill" in the second case, “butterfly like” in the third case, and "to attract females" in the rightmost case in Fig. 5. The unclear attributes in these cases are vague and hard to refine in the generation.

4.8 User Study

To further demonstrate the superiority of our proposed LFR-GAN approach, we conduct a user study and ask humans to evaluate the quality of the generated images, i.e., whether the image is photo-realistic and looks attractive. We collect the evaluation from 70 humans of 1400 generated bird image pairs from Lafite and our LFR-GAN approach since Lafite is the most competitive method in our comparison. The images are divided into 20 groups, each human evaluates 20 randomly chosen image pairs from each group. Here we present the human preference in each group and overall human preference in Fig. 6.



(a) Human preference in each group (b) Overall human preference

Fig. 6. User study results of comparison between Lafite and Our approach.

The results in Fig. 6 show that 67.6% of the images generated by our approach are preferred by humans. This is because Lafite uses a fix-length text encoder to encode the highly flexible text input

while our LFR-GAN approach divides the complex text into independent fine-grained attributes and encodes them separately, which could capture more attribute information and provide more fine-grained details for image generation. Therefore, our approach can generate images with more vivid details and be preferred by humans.

5 CONCLUSION

In this paper, we propose the LFR-GAN approach for the text-to-image generation task. It first divides the input text into several independent fine-grained attributes and generates an initial image, then modifies the local features of the initial image according to these attributes. Our LFR-GAN approach can capture the text information more completely and generate images with more vivid details consistent with the input text. Experiment results on two benchmark datasets have verified that the proposed LFR-GAN approach outperforms the other state-of-the-art methods.

The future work mainly lies in two aspects: First, we plan to add multi-object analysis in our attribute modeling to classify attributes from different generated objects. Second, we plan to introduce a more powerful image generation network to obtain better initial images.

REFERENCES

[1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Association for Computational Linguistics, 54–59.

[2] H Bouma. 1980. Visual reading processes and the quality of text displays. *IPO Annual Progress Report* 15 (1980), 83–90.

[3] Jun Cheng, Fuxiang Wu, Yanling Tian, Lei Wang, and Dapeng Tao. 2020. RiFeGAN: Rich feature generation for text-to-image synthesis from prior knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 10911–10920.

[4] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. 2017. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5706–5714.

[5] Yao Fu, Hao Zhou, Jiaze Chen, and Lei Li. 2019. Rethinking Text Attribute Transfer: A Lexical Analysis. In *Proceedings of the 12th International Conference on Natural Language Generation*. 24–33.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems* 30 (2017), 6629–6640.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[9] Paul Jobling, Paul Jobling, and David Crowley. 1996. Graphic design: reproduction and representation since 1800. (1996).

[10] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4401–4410.

[11] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1, 4 (1989), 541–551.

[12] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. 2020. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7880–7889.

[13] Bowen Li, Xiaojuan Qi, Philip Torr, and Thomas Lukasiewicz. 2020. Lightweight generative adversarial networks for text-guided image manipulation. *Advances in Neural Information Processing Systems* 33 (2020), 22020–22031.

[14] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345.

[15] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.

[16] Igor Melnyk, Cicero Nogueira dos Santos, Kahini Wadhawan, Inkit Padhi, and Abhishek Kumar. 2017. Improved neural text attribute transfer with non-parallel data. *arXiv preprint arXiv:1711.09395* (2017).

[17] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. 2018. Text-adaptive generative adversarial networks: manipulating images with natural language. *Advances in Neural Information Processing Systems* 31 (2018), 42–51.

[18] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 722–729.

[19] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2085–2094.

[20] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. 2019. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1505–1514.

[21] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. 2020. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition* 106 (2020), 107404.

[22] Shulan Ruan, Yong Zhang, Kun Zhang, Yanbo Fan, Fan Tang, Qi Liu, and Enhong Chen. 2021. Dae-gan: Dynamic aspect-aware gan for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13960–13969.

[23] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in Neural Information Processing Systems* 29 (2016), 2234–2242.

[24] Erik F Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. *arXiv preprint cs/0009008* (2000).

[25] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

[26] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. (2011).

[27] Hao Wang, Guosheng Lin, Steven CH Hoi, and Chunyan Miao. 2021. Cycle-Consistent Inverse GAN for Text-to-Image Synthesis. In *Proceedings of the 29th ACM International Conference on Multimedia*. 630–638.

[28] Jianan Wang, Guansong Lu, Hang Xu, Zhenguo Li, Chunjing Xu, and Yanwei Fu. 2022. ManiTrans: Entity-Level Text-Guided Image Manipulation via Token-wise Semantic Alignment and Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10707–10717.

[29] Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. *Advances in Neural Information Processing Systems* 32 (2019), 11036–11046.

[30] Zongze Wu, Dani Lischinski, and Eli Shechtman. 2021. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12863–12872.

[31] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. 2021. TediGAN: Text-Guided Diverse Face Image Generation and Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2256–2265.

[32] Minzhang Xu, Min Peng, and Fang Liu. 2022. Text style transfer between classical and modern chinese through prompt-based reinforcement learning. *World Wide Web* (2022), 1–18.

[33] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1316–1324.

[34] Yanhua Yang, Lei Wang, De Xie, Cheng Deng, and Dacheng Tao. 2021. Multi-Sentence Auxiliary Adversarial Networks for Fine-Grained Text-to-Image Synthesis. *IEEE Transactions on Image Processing* 30 (2021), 2798–2809.

[35] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. *Advances in Neural Information Processing Systems* 31 (2018), 7298–7309.

[36] Xiaoyuan Yi, Zhenghao Liu, Wenhao Li, and Maosong Sun. 2021. Text style transfer via learning style instance supported latent space. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 3801–3807.

[37] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5907–5915.

[38] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2018. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2018), 1947–1962.

[39] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. 2022. Towards language-free training for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17907–17917.

[40] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. 2019. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5802–5810.