Text-to-Image Synthesis based on Object-Guided Joint-Decoding Transformer

Fuxiang Wu[1,2], Liu Liu[3], Fusheng Hao[1,2], Fengxiang He[4], Jun Cheng[1,2*]
[1] Guangdong Provincial Key Laboratory of Robotics and Intelligent System,
Shenzhen Institute of Advanced Technology, CAS, China
[2] The Chinese University of Hong Kong, Hong Kong, China
[3] The University of Sydney, Australia
[4] JD.com, Beijing, China
{fx.wu1, fs.hao, jun.cheng}@siat.ac.cn, liu.liu1@sydney.edu.au, hefengxiang@jd.com.

Abstract

Object-guided text-to-image synthesis aims to generate images from natural language descriptions built by two-step frameworks, i.e., the model generates the layout and then synthesizes images from the layout and captions. However, such frameworks have two issues: 1) complex structure, since generating language-related layout is not a trivial task; 2) error propagation, because the inappropriate layout will mislead the image synthesis and is hard to be revised. In this paper, we propose an object-guided joint-decoding module to simultaneously generate the image and the corresponding layout. Specially, we present the joint-decoding transformer to model the joint probability on images tokens and the corresponding layouts tokens, where layout tokens provide additional observed data to model the complex scene better. Then, we describe a novel Layout-VQGAN for layout encoding and decoding to provide more information about the complex scene. After that, we present the detail-enhanced module to enrich the language-related details based on two facts: 1) visual details could be omitted in the compression of VQGANs; 2) the joint-decoding transformer would not have sufficient generating capacity. The experiments show that our approach is competitive with previous object-centered models and can generate diverse and high-quality objects under the given layouts.

1. Introduction

Text-to-image synthesis is an important task in computer vision [10, 16, 25, 27, 30, 37], which generates images from textual descriptions. Recently, GAN-based methods have achieved many promising results [36, 39, 42]. However, GANs, which include both generators and discriminators, are known to have difficulty in reaching the stable convergence to simulate complex distributions over images conditioned on the text. For complex scenes with multi objects, the synthesizing results by GAN-based models are far from satisfactory. Because a complex scene may include various objects with different viewpoints and sizes, which is usually not mentioned in captions.

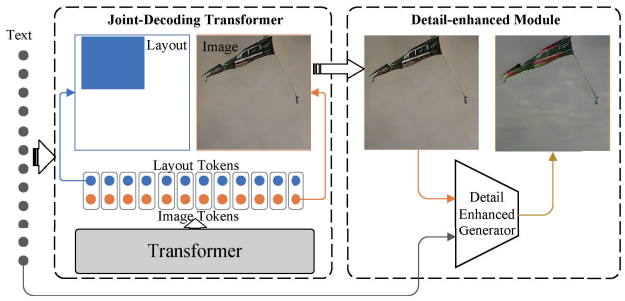


Figure 1. The proposed model includes an object-guided joint-decoding transformer and a detail-enhanced module: 1) the joint-decoding transformer simultaneously handles the object-centered layout tokens and image tokens, 2) the detail-enhanced module enriches the language-related finer-grained details to obtain a more realistic image.

The layout, consistinig of bounding boxes and object labels, can provide semantic information of the scene. The layout information is beneficial to model the corresponding image. The previous models [8, 9, 14] are usually built by two-step structures: firstly, the model generates the layout and then synthesizes images from the layout and captions. Such structures are complex and would suffer from the problem of error propagation. Nonetheless, the auto-regressive models could jointly handle the layout and image, denoted as one-step, and layouts can provide additional semantic information to model the complex scene. For example, as shown in the left part of Figure 1, given "A *colorful* kite flying through a *cloudy blue sky*", the model would

*J. Cheng is the corresponding author.

auto-regressively predicts the image tokens and the layout tokens simultaneously *under* both the previously predicted image tokens and the predicted layout tokens. Then, the image tokens and the layout tokens can be decoded into the image and the layout, respectively.

Recent auto-regressive generative models, like Generative Pre-Training (GPT) models [2, 23], exploit Transformers [34] to promote the performance of natural language generation. To reduce the computation of modeling the probability density function on an image, recent methods exploit the framework of Vector Quantized Variational AutoEncoders (VQ-VAE) [32] to transform and compress the density image into a low-dimensional discrete latent space, which is affordable to be modeled by the Transformers. CogView [6] and DALL-E [24] are jointly trained on large-scale text and image (from VQ-VAE or VQ-GAN) tokens and achieve promising results.

However, on one side, their models did not consider the layout information and may not properly decompose and understand the complex scene, which may lead to some unrealistic distortion. Besides, without layout, the model is hard to control the synthesis to meet some user-preferring. On another side, since the tokens of the image are generated by the compressor: VQ-VAE or VQ-GAN, some visual details would be lost, and the lost details will degrade the decoded images. Besides, the transformer is hard to model the massive finer-grained visual details with the finite computing resources and text-image dataset. For example, as shown in the right part of Figure 1, in the image generated by object-guided joint-decoding transformer, the kite is gray, and the sky is light pink, which needs to be manipulated to include the language-related visual features of "*colorful skite*" and "*cloudy blue sky*".

To alleviate the above issues, we propose an object-guided joint-decoding module to simultaneously generate the image and the corresponding layout. Specifically, to handle the layout tokens and image tokens jointly, we propose an auto-regressive joint-decoding transformer, where each image token will be better predicted from the more abundant conditions involving the historical image tokens and layout tokens. Moreover, to obtain high-quality layout tokens for the joint-decoding transformer, we introduce a novel Layout-VQGAN to handle the layout information with class data, in which the layout is compressed into layout tokens. In addition, we propose a detail-enhanced GAN based on affine combination modules (ACM) [12] to improve language-related visual details, which may reduce the requirement of the transformer through synthesizing the raw images without many finer-grained details. Furthermore, since language-based image editing is not a trivial task, an edited image may be worse than the original image. Thus, we introduce a global ranking to select the best image from images and the corresponding enhanced images, which are generated by the joint-decoding transformer and handled by the detail-enhanced GAN, respectively. The main contributions of this paper are three-fold:

- To improve the synthesizing quality of complex scenes, we propose an one-step object-guided joint-decoding transformer to simultaneously decode the image tokens and decode layout tokens, where the object-centered layout can be altered to control the scene.
- To remedy the omitted visual details induced by the compression in VQGAN and the limited capacity of the joint model, we introduce a detail-enhanced module based on the ACM to enrich the finer-grained language-related visual details.
- We conduct extensive experiments on MS-COCO dataset to verify the object-centered generating ability of the auto-regressive joint-decoding transformer and the effectiveness of the detail enhancement.

2. Related Work

Text-to-image generation: Stacked GANs *et al*. [25, 39, 40] are proposed to decompose the complex task into relatively simple tasks and gradually synthesize images. Xu *et al.* [36] and Zhu *et al.*. [42] exploit attentional models to synthesize different parts by focusing on different words to improve the quality of generated images. Chen *et al.* [5] introduce a RiFeGAN to enrich the given caption and improve the semantic meanings of text descriptions. Wu *et al.* [35] exploited the attribute pairs to synthesize images and improve the controllability. Qiao *et al.* [22] introduce a MirrorGAN with a captioner to re-describe the synthesized image and improve the semantic consistency. Tan *et al.* [31] proposed a knowledge-transfer GAN (KT-GAN) to bridge the cross-domain gap and improve the quality of synthesized images. Yuan and Peng [38] propose Bridge-GAN to construct a transitional space for associating text and image.

Object-Centered Text-to-image generation: Hinz *et al*. [8, 9] introduced a new framework that exploits an object-level synthesizing generator to utilize bounding boxes and model the complex scenes with multiple objects. Li *et al*. [14] exploited two-step object-driven attentive GANs to fuse the information of bounding boxes and object shapes to improve the synthesizing quality. Sylvain *et al*. [29] proposed an object-centric generator to utilize object layouts, where the information of objects is enhanced by a scene-graph similarity module. By exploiting the layouts of objects to guide the synthesis, many works [13, 15, 21, 28] generate high-quality images by implicitly decomposing the complex scene.

Text-Guided Image Manipulation: Nam *et al*. [19] propose a TAGAN to manipulate the image by exploiting the text-adaptive discriminator. Chen *et al*. [3] fuse visual fea-

tures of a source image and the language features by proposing a generic recurrent attentive modeling framework. Li *et al*. [11] propose a ControlGAN to effectively generate images, which consists of a spatial and channel-wise attentional generator and a word-level discriminator. Zhou *et al*. [41] introduce the text-based pose generation and visual appearance transferring to edit the person images. Liu *et al*. [18] introduce an IR-GAN consisted of the word-level and instruction-level instruction encoders and a reasoning discriminator to improve the consistency between the image and linguistic instruction. Li *et al*. [12] propose the text-image affine combination module and the detail correction module to manipulate the images based on a given description.

Transformer-based Image Synthesis: Chen *et al*. [4] introduce the ImageGPT, a pixel-level auto-regressive model, to synthesize images at a max resolution of 96×96. To generate images at a high-resolution, Vector Quantized Variational AutoEncoders (VQ-VAE) [32] is exploited to compress the dense image into a low-dimensional discrete latent vector that can be recovered by a decoder. Given the discrete latent vector, PixelCNN [33] can be used to model the prior and generate the latent vector. Following this framework, recent work [6,7,24] used Transformer to fit the prior and greatly improve the performance of image synthesis.

***Difference to Existing Works*:** OPGAN [8] utilize an object pathway in GANs to provide the object-center information like the bounding boxes and the corresponding labels. Their work needs a bounding box generator to generate the bounding boxes from a given caption as in [14] first. CogView [6] and DALL-E [24] are trained on a large-scale dataset and based on a much larger Transformer, and they can integrate into our works naturally to improve the synthesizing quality. In contrast, our work integrates the layout into the transformer and builds a one-step object-guided joint transformer to improve the synthesizing quality without the requirement of the additional bounding box generator. Then, we utilize a detail-enhanced GAN to recover the omitted visual details because of the compression in VQGAN and the limited capacity of the joint model.

3. Methodology

In this section, we propose an object-guided joint-decoding module to simultaneously generate the image and the corresponding layout. In Sec 3.1, we present the joint-decoding transformer to synthesize images tokens and the corresponding layouts tokens at one-step. Then, we describe a novel Layout-VQGAN for layout encoding and decoding for providing more information about the complex scene in Sec 3.2. After that, in Sec 3.3, we present the detail-enhanced module to enrich the language-related details based on two facts: 1) visual details could be omitted in the compression of VQGANs [7]; 2) the joint-decoding transformer would not have sufficient generating capacity.

3.1. Joint-Decoding Transformer

We introduce the joint-decoding transformer involving both image tokens and layout tokens. Firstly, we present the evidence lower bound of the joint-decoding transformer to derive the Negative Log-Likelihood (NLL) loss for image tokens and layout tokens. Secondly, we proposed a joint autoregressive decoding for both image tokens and layout tokens.

3.1.1 Evidence Lower Bound

The process of the joint-encoding transformer is the maximizing problem of the Evidence Lower BOund (ELBO) over images, corresponding layouts, and corresponding captions. Given an image x in the dataset, the corresponding layout m, and the corresponding caption t, The ELBO of the joint distribution $p_{\theta,\phi_x,\phi_m}(x,m,t)$ is

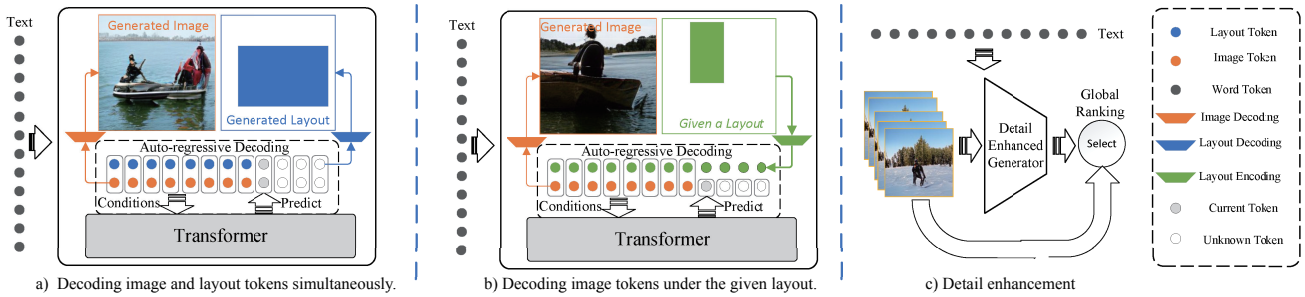
$$
\begin{aligned}
&\log p_{\theta,\phi_x,\phi_m}(x,m,t) \\
&= \log p_\theta(t) + \log p_{\theta,\phi_x,\phi_m}(x,m|t) \\
&\geq -\Bigg(\underbrace{-\log p_\theta(t)}_{\text{NLL loss for text}} + \underbrace{\mathbb{E}_{z^x\sim q_{\gamma_x}(z^x|x)}[-\log p_{\phi_x}(x|z^x)]}_{\text{image reconstruction loss}} \\
&\quad + \underbrace{\mathbb{E}_{z^m\sim q_{\gamma_m}(z^m|m)}[-\log p_{\phi_m}(m|z^m)]}_{\text{layout reconstruction loss}} \\
&\quad + \underbrace{\mathrm{KL}(q_{\gamma_x,\gamma_m}(z^x,z^m|x,m)||p_\theta(z^x,z^m|t))}_{\text{KL between } q \text{ and text conditional prior}} \Bigg) \qquad (1)
\end{aligned}
$$

where θ is the parameter of the prior $p_\theta(t)$; γ_m and ϕ_m are an encoder and a decoder of the VQGAN for the layout m; γ_x and ϕ_x are an encoder and a decoder of the VQGAN for the image x; z^x and z^m are the latent variables regarding the image and layout; KL is the Kullback–Leibler divergence. NLL denotes the Negative Log-Likelihood.

Recent methods usually utilize the powerful transformers to deal with the discrete tokens z^x and z^m, which can be set as $z^m = \arg\max_{z^m} q_{\gamma_m}(z^m|m)$ and $z^x = \arg\max_{z^x} q_{\gamma_x}(z^x|x)$ [6]. The Eq. (1) can be rewritten as NLL loss for tokens,

$$
\mathrm{KL}(q_{\gamma_x\gamma_m}(z^x,z^m|x,m)||p_\theta(z^x,z^m|t)) = -\log p_\theta(z^x,z^m|t).
$$

Thus, we can exploit a transformer to model the NLL losses by using the tokens t, z^x, and z^m, then utilize two VQGANs for the image and the layout to generate images and layouts. We utilize the Image VQGAN built by Esser *et al*. [7] for image encoding and decoding, and propose a novel VQGAN for layout encoding and decoding in Sec 3.2.



a) Decoding image and layout tokens simultaneously. b) Decoding image tokens under the given layout. c) Detail enhancement

Figure 2. The synthesizing framework: a) the joint-decoding transformer decodes image tokens and layout tokens simultaneously; b) the joint-decoding transformer decodes image tokens according to the given layout tokens; c) the detail enhanced module consists of a detail enhanced GAN to edit the input image and a global ranker to select the best candidate.

3.1.2 Joint Autoregressive Decoding in One-Step

Previous two-step methods like [8], which need first generate a layout and then generate the image from the layout, could trigger error propagation when the generated layout did not fit the given caption. What's more, the layout could not be changed once generated.

However, in the decoding process of the joint-decoding transformer, our model simultaneously generates jth image tokens $z_j^x \in z^x$ and jth layout tokens $z_j^m \in z^m$ under the previously generated image tokens $z_{1:j-1}^x \subset z^x$ and layout tokens $z_{1:j-1}^m \subset z^m$. In particular, the probability is formulated as,

$$p_\theta(z_j^x, z_j^m, z_{1:j-1}^x, z_{1:j-1}^m|t) = \prod_{i=1}^{j} p_\theta(z_i^x, z_i^m|z_{1:i-1}^x, z_{1:i-1}^m, t).$$

Then, the z_j^x and z_j^m can be obtained as following,

$$z_j^x, z_j^m = \arg\max_{z_j^x, z_j^m} p_\theta(z_j^x, z_j^m, z_{1:j-1}^x, z_{1:j-1}^m|t). \quad (2)$$

After obtaining the layout tokens z^m and image tokens z^x, we can generate the corresponding layout m and image x through *Layout-VQGAN* (Sec. 3.2) and Image-VQGAN [7]. Figure 2 a) shows the process.

Moreover, if the layout tokens $\hat{z}_i^m$ are given, we could generate the corresponding image satisfying the given layout. Similarity, the probability is

$$p_\theta(z_j^x, z_j^m, z_{1:j-1}^x, \hat{z}_{1:j-1}^m|t) = \prod_{i=1}^{j} p_\theta(z_i^x, z_i^m|z_{1:i-1}^x, \hat{z}_{1:i-1}^m, t).$$

We will ignore the prediction z_j^m because of the given $\hat{z}^m$. the z_j^x can be obtained as,

$$z_j^x = \arg\max_{z_j^x, z_j^m} p_\theta(z_j^x, z_j^m, z_{1:j-1}^x, \hat{z}_{1:j-1}^m|t). \quad (3)$$

Figure 2. b) shows the process.

3.2. Layout VQGAN

As the joint-decoding transformer need the layout tokens to guide the decoding training, we rasterize a layout structure into a layout image, where values of each pixel denote the class of the corresponding object. Then, we present a new Layout-VQGAN to transfer the layout image into the layout tokens. However, different from VQGAN [7] handling the RGB data, the Layout-VQGAN processes the class information, where the set of class labels is unordered.

Specially, given layout images m, we can calculate the closest codebook entries (Layout Tokens) z^m and reconstruction $\hat{m}$ as,

$$\begin{cases} z^m = \mathbf{Q}(\gamma_m(m)) = \arg\min_{z^m \in \mathcal{Z}} \|\gamma_m(m) - z^m\|_2^2, \\ \hat{m} = \phi_m(z^m) = \phi_m(\mathbf{Q}(\gamma_m(m))). \end{cases} \quad (4)$$

where $\mathbf{Q}(\cdot)$ is an element-wise quantization function, $\mathcal{Z} \subset \mathbb{R}^{N_z}$ is the learned codebook, where N_z is the dimensionality of code, γ_m and ϕ_m are the encoder and decoder of the Layout-VQGAN for layout.

The loss function of the Layout-VQGAN is

$$\begin{aligned} &\mathcal{L}_{\text{VQ}}(\gamma_m, \phi_m, \mathcal{Z}) \\ =&\mathcal{L}_{\text{quant}}(\gamma_m, \mathcal{Z}) - \sum_{i,j}\sum_{c=1}^{M} m_{i,j}^c \log \hat{m}_{i,j}^c, \end{aligned} \quad (5)$$

where the last item is the cross entropy between the probability of the predicted class $\hat{m}_{i,j}$ and the ground-truth $m_{i,j}$, and $m_{i,j}^c$ and $\hat{m}_{i,j}^c$ return the corresponding probability of the pixel at the ith row and the jth column at c class; $\mathcal{L}_{\text{quant}}$ is the loss for the codebook and the commitment loss [32] formulated as,

$$\begin{aligned} &\mathcal{L}_{\text{quant}}(\gamma_m, \mathcal{Z}) \\ =&\|\text{STG}(\gamma_m(m)) - z^m\|_2^2 + \|\text{STG}(z^m) - \gamma_m(m)\|_2^2, \end{aligned} \quad (6)$$

where $\text{STG}(\cdot)$ denotes the stop-gradient operation. Backpropagation through the quantization function is implemented by simply copying the gradients through the computing graph and can be done end-to-end. In addition, to

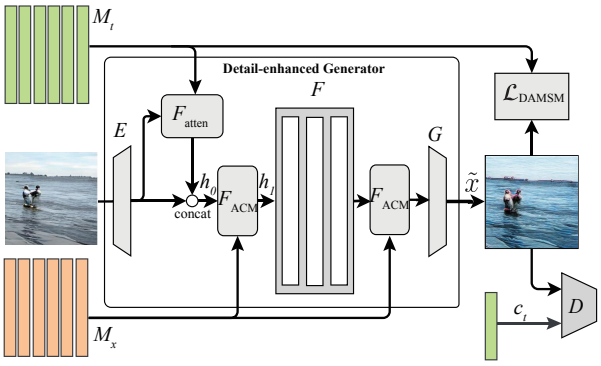


Figure 3. Detail-enhanced GAN for the language-related detail enhancement.

learn a rich codebook, we also exploit a patch-based discriminator to train the encoder and decoder adversarially as Esser *et al*. [7].

3.3. Detail-enhanced Module

In this subsection, we present the detail-enhanced module to enrich the language-related details based on two facts: 1) visual details could be omitted in the compression of VQ-GANs [7]; 2) the joint-decoding transformer would not have sufficient generating capacity. Besides, the transformer is hard to model all finer-grained visual details based on the finite computing resources and text-image dataset. However, the visual details could be modeled by a GAN-based language-guided image manipulation under the limited resource.

Specifically, in Figure 3, given a generated image $\hat{x}$ from the joint transformer, VGG is exploited to extract regional features M_x. For a target text description t, an LSTM-RNN model is employed to get the word embeddings M_t and sentence code c_t. Then, a feature extractor E extracts the inner features from $\hat{x}$, followed by the attentional module F_{Atten}, the spatial and channel-wise attention as defined in ManiGAN [12], to fuse the textual features and output inner features h_0. Next, we exploit an Affine Combination Module (ACM) to retain more visual details as,

$$h_1 = F_{\text{ACM}}(h_0, M_I), \tag{7}$$

where F_{ACM} denotes as process of ACM,

$$F_{\text{ACM}}(h, M) = h \odot W(M) + b(M), \tag{8}$$

where $W(M)$ and $b(M)$ compute weights and biases by given regional features M; $\odot$ is a Hadamard element-wise product. Given h_1, we use an upsampling module F_0, consisting of several residual net and an upsampling layer, to compute the features at a high-resolution. Finally, we utilize another ACM to try to retain original features more, then a generator G to transfer the inner features into final RGB images $\tilde{x}$.

3.3.1 Loss Function and Training

Given an enhanced image $\tilde{x}$, a input image x, and the corresponding caption t, the loss for generator G is,

$$\begin{aligned}\mathcal{L}_G(x,t) = &-\mathbb{E}_{\tilde{x}\sim P_G(x,t)}\{\log D(\tilde{x}|t) + \mathcal{L}_{\text{DAMSM}}(\tilde{x},t) \\ &+ \|x-\tilde{x}\|^2 + \|\text{VGG}(x) - \text{VGG}(\tilde{x})\|^2\},\end{aligned}$$

where D includes the conditional and unconditional output of the discriminator; $\mathcal{L}_{\text{DAMSM}}$ is defined in AttnGAN. $\text{VGG}(\cdot)$ is the feature extractor. To avoid over-editing and retain image quality, we use $\|x-\tilde{x}\|^2 + \|\text{VGG}(x) - \text{VGG}(\tilde{x})\|^2$ in the generator loss. Besides, we employ $\mathcal{L}_{\text{DAMSM}}$ to encourage the text-related finer-grained editing and prevent the identity mapping. To exploit the features of a generated image $\hat{x}$ from the joint-decoding transformer and ground-truth x, we train the generator as,

$$\mathcal{L}_{\text{AllG}} = \mathcal{L}_G(\hat{x}, t) + \mathcal{L}_G(x, t). \tag{9}$$

In Eq. (9), we train the generator to alleviate the error propagation by jointly considering ground-truth image x.

Similarly, the loss for the discriminator D is,

$$\mathcal{L}_D(x,t) = \mathbb{E}_{\tilde{x}\sim P_G(x,t)} \log D(\tilde{x}|t) - \mathbb{E}_{x\sim P_{\text{data}}} \log D(x|t).$$

We train the discriminator to distinguish the fake images $\tilde{x}$ from the input images, which will guide the generator to remedy the distorted features.

3.3.2 Global Ranking

Because of the ambiguousness and abstractive property of natural language and the uncertainty of synthesizing performance of GANs, Detail-enhanced generator may fail to enhance the images generated by the joint-decoding transformer, thus it is better to evaluate the quality of the enhanced images. In CogView [6] and DALL-E [24], they generate multi images for each caption and select the best one. Because AttnGAN [36] and DM-GAN [42] exploit the text-image similarity ($\mathcal{L}_{\text{DAMSM}}$) to train the generator, we exploit this similarity to select the best images from the multi images made by the joint-decoding transformer and their corresponding enhanced images.

$$\hat{x}_{best} = \underset{x\in G(\hat{\mathcal{X}})\cup\hat{\mathcal{X}}}{\arg\max} \cos(f_{\text{rnn}}(t), f_{\text{cnn3}}(x)),$$

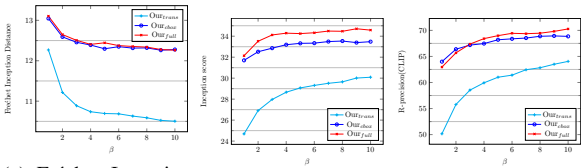
where $\hat{\mathcal{X}}$ is a set of images of the given caption t generated by the joint-encoding transformer, $G(\hat{\mathcal{X}})$ is a set of images enhanced by the detail-enhanced module, f_{rnn} and f_{cnn3} are the text encoder and the image encoder in AttnGAN [36] pre-trained on MS-COCO to represent the similarity on the local domain, $\cos(\cdot)$ is the cosine similarity function.

Table 1. Inception Score (IS), Fréchet Inception Distance (FID), R-precision(CLIP), and Semantic Object Accuracy on Class (SOA-C) and Image Average (SOA-I) on the MS-COCO dataset.

	IS↑	FID↓	R-precision(CLIP)↑	SOA-C↑	SOA-I↑
AttnGAN [36]	23.25 ± 0.29	34.98	48.64 ± 0.97	25.72	38.93
DM-GAN [42]	32.20 ± 0.33	26.73	62.85 ± 0.99	33.31	47.93
OPGAN [8]	27.58 ± 0.53	23.21	58.14 ± 1.23	35.17	49.50
DALL-E [24]	17.90	27.50	N/A	N/A	N/A
CogView [6]	18.20	27.10	N/A	N/A	N/A
AttnGAN* [36]	25.94 ± 0.62	32.38	54.08 ± 0.61	29.68	43.50
DM-GAN* [42]	34.05 ± 0.49	25.22	66.01 ± 0.85	36.80	51.52
OPGAN* [8]	28.70 ± 0.51	22.04	60.87 ± 1.07	37.73	52.15
Our_{full}	**34.58 ± 0.39**	**12.26**	**70.27 ± 0.53**	**52.63**	**63.51**

Table 2. Influence and comparison of different components on the MS-COCO dataset.

	IS↑	FID↓	R-precision(CLIP)↑
$\text{Our}_{\text{trans}}$	30.09 ± 0.32	**10.50**	64.02 ± 0.92
Our_{cbox}	33.48 ± 0.41	12.27	68.83 ± 0.89
Our_{full}	**34.58 ± 0.39**	12.26	**70.27 ± 0.53**



(a) Fréchet Inception Distance (b) Inception Score (c) R-precision(CLIP)

Figure 4. Diagrams of FID, IS, and R-precision(CLIP) on MS-COCO for our models by exploiting different β.

4. Experiments

We report the experiments by jointly decoding images and their layouts, denoted as the subscript "full", on the MS-COCO dataset [17]. The baselines are taken from AttnGAN [36], DM-GAN [42], OPGAN [8,9] provided by the authors, and "*" denotes the corresponding baselines with the same ranking as our models.

4.1. Datasets and Metrics

Datasets: In MS-COCO dataset [17], we exploited the 2014 dataset split, where the training part includes approximately 80,000 images, the testing part contains 40,000 images, and each image is described by 5 captions.

Evaluation Metrics: We adopt the following metrics,[1].

a) Inception score (IS): Inception score [26] is popular and tendentious to favor meaningful and diverse images. Although it has the notable flaws [1], we exploit it to evaluate the quality of the synthesized images, as in [9, 36, 39].

[1]For DALL-E [24] and CogView [6], we only list their Inception score and Fréchet Inception Distance without blurring for reference. Because they are trained on a much larger dataset, which does not include MS-COCO, and make use of the much larger transformers.

Figure 5. Generated examples: The target text description is above the corresponding image and the prominent edited characteristics are marked as bold.

b) Fréchet Inception Distance (FID): We only utilize the MS-COCO dataset for training and testing. It is suitable to compare the FID on MS-COCO with baselines. Thus, we report the FID to measure the distance between synthesized and real images. A lower FID indicates that the generated images have higher visual quality.

c) R-precision(CLIP): The text-image similarity of R-precision [36] is used in training the baseline. The CLIP model can provide their alignment information in a different manner and is pre-trained on a much large-scale dataset. Thus, similar to the work [20], we exploit the CLIP to extract features of captions and images and report the R-precision, denoted as R-precision(CLIP).

d) Semantic object accuracy: Since the object-center synthesis integrates the bounding boxes and object labels of objects, we can evaluate the class average semantic object accuracy (SOA-C) [8] and the image average semantic object accuracy (SOA-I) [8] to check whether the synthesizing image includes the given object.

4.2. Quantitative Comparison

In Table 1, compared with AttnGAN*, DM-GAN*, and OPGAN* trained with the same dataset MS-COCO, the FID of our model Our_{full} largely decreases at least **9.78**. The IS of Our_{full} increases **0.53** over DM-GAN*. The R-precision(CLIP) increases at least **4.26%**, respectively. The SOA-C and SOA-I increase at least **14.90** and **11.36**, re-

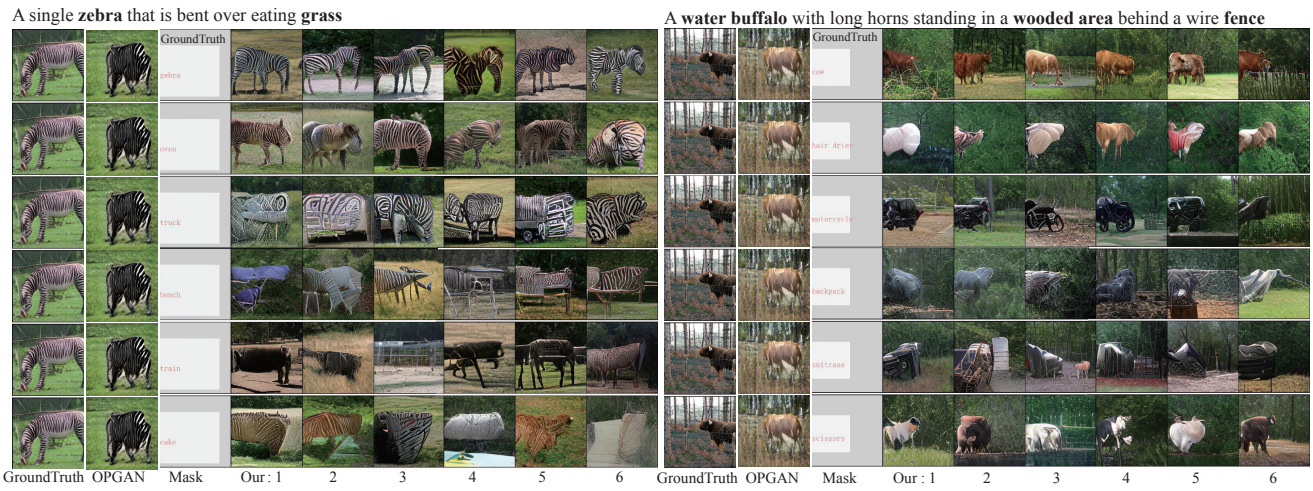


Figure 6. Generated examples with disturbing object labels: The target text description is above the corresponding image and the prominent edited characteristics are marked as bold.

spectively. The results indicate that our model can synthesize better images than the baselines in the semantic consistency and synthesizing quality. Compared with DALL-E and CogView trained with the much larger dataset, the IS of Our_{full} increases at least 16.38, and the FID decreases at least 14.84.

In Table 2, we analyze the effectiveness of different components, the subscript "trans" indicates that the output images are generated by the joint-decoding transformer and without detail enhanced; the subscript "cbox" indicates that the output images are enhanced by the detail-enhanced GAN and constrained by the predefined layout; and the subscript "full" indicates that the output images are enhanced by the detail-enhanced GAN. With detail enhancement, although the FID increase about **1.76**, the IS increases at least **4.49** and the R-precision(CLIP) increases at least **6.25%**, respectively. The results show that the detail enhancement can improve the synthesizing quality as well. To analyze the constraints of object layouts, we enforce the decoding following the guide of the given layout tokens. The generating quality of Our_{cbox} will be lower than that of the unconstrained model Our_{full} because of the left-to-right process in auto-regressive decoding, where constraining current layout tokens is hard to alter the previously generated tokens.Besides, without the jointly decoding by using the original transformer, the IS of $\text{Our}_{\text{full}}^{\text{w/oJ}}$ would decrease 0.73, the FID would increase 0.87, the R-precision(CLIP) would decrease 2.17%, respectively. The results show that the jointly decoding could improve the synthesizing quality. In the two-steps generator consisted of "text-layout" and "layout-image". The part "text-layout" should model a L_{text}+256 sequence, and another part "layout-image" should model a L_{text}+256+256 sequence, where L_{text} is the length of text. However, in the one-step, we only need to model a L_{text}+256 sequence. Thus, the one-step has lower complexity, which may be relatively easy to train and achieve better results.

In Figure 4, the results demonstrate the influence of β: the sample size used for reranking. With increasing β, the FID will be decreasing, and IS and R-precision(CLIP) will be increasing, which indicates that the quality of synthesized images will be improving. In Table 1 and Table 2, the results are computed by selecting the best one from 10 generated images, while DALL-E and CogView generate 512 images and 60 images for selecting, respectively. Our results can be improved further by exploiting more generated images like DALL-E.

4.3. Qualitative Comparison

In Figure 5, the subscript "full*" denotes that the output images are the inner images generated by the joint-decoding transformer before sending to the detail-enhanced GAN. The results show that images synthesized by transformer-based models are better than others and the transformer can synthesize better results on a relatively small dataset. Models handling the layout jointly may generate more realistic images than the original ones, which indicates that jointly modeling layout is beneficial to the synthesis. In addition, the detail-enhanced GAN could enhance the language-related details, providing more realistic features. For example, given "An old short train traveling through a wooded area." in the first row, transformer-based models can generate more realistic images with visual details of a train than the baselines, and models with considering the layout, $\text{Our}_{\text{trans}}$, $\text{Our}_{\text{full}*}$, Our_{full}, can synthesize the images with correct details about the train and the wooded area. Besides, the detail-enhanced GAN can alter the wrong ground color.

Influence of object labels in layout:

In Figure 6, we replace the original label in the layout by a randomly selected label to verify their influence in

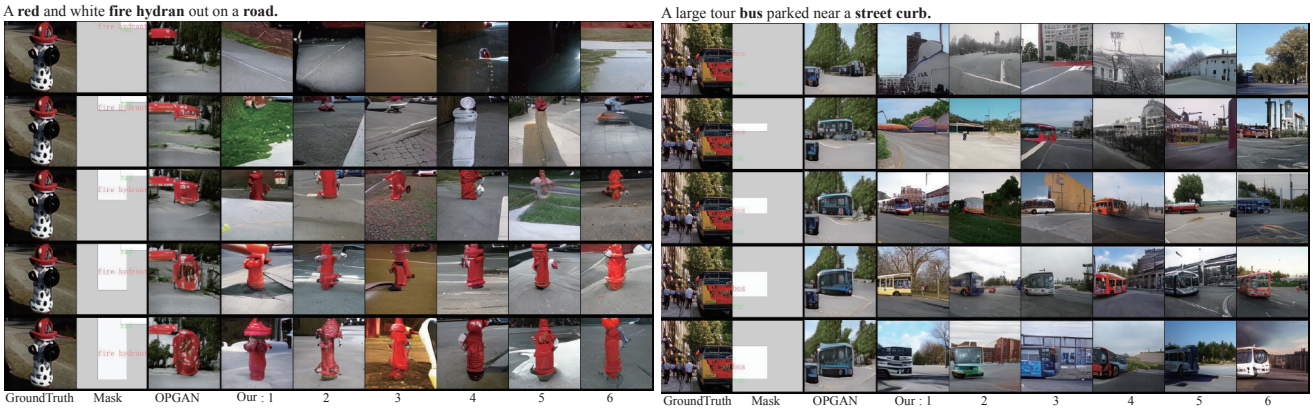


Figure 7. Generated examples with the largest object based on varied bounding boxes: The target text description is above the corresponding image and the prominent edited characteristics are marked as bold.

the synthesis: the controllability by enforcing models to obey the given layout (our model can also directly generate images without needing predicted layout as shown in Figure 2a) . Given "A single zebra that is bent over eating grass" in the first example, we replace the ground-truth label "zebra" with other labels, like "oven", "truck", "bench", "train", and "cake". In the third row of the first example, with replacing "zebra" by "truck", our joint transformer will synthesize an object with the truck shape and the zebra texture. Given "A water buffalo with long horns standing in a wooded area behind a wire fence", we replace the ground-truth label "cow" by "motorcycle", our model will try to generate motorcycle-like objects with two wheels. It is hard to imagine the combination of the features of "bench" and the textual features of the given caption for the model, thus, it may synthesize some degraded images. The results show that the object label can alter the synthesized result to some degree and will try to maintain the details corresponding to the description of the given caption.

Influence of the bounding boxes in layout:

In Figure 7, we alter the original shape of the largest object in the layout by gradually scaling the length of one side of the bounding box to verify their influence in the synthesis. Given "A red and white fire hydran out on a road." in the first example, we scale the height of the object "fire hydran" gradually, In the first row, the height is 0 and there are no "fire hydran" in our generated images. With increasing height, our joint transformer will provide more realistic details of "fire hydran". Given "A large tour bus parked near a street curb.", the behavior of our model is similar, but OPGAN fails to eliminate "bus" when the height is 0. The results indicate that our model can provide a more powerful object-level controllability and high-quality results.

5. Limitation and Discussion

Our work is trained on MS-COCO, which contains 0.328 million images with variant numbers of each object. It is better to utilize larger datasets like DALL-E (trained on 3.3 million text-image) and CogView (trained on 30 million high-quality text-image pairs). Our transformer contains about 0.305 billion parameters, which is much smaller than the parameters in DALL-E (up to 12 billion parameters) and CogView (4 billion parameters), and we believe enlarging our transformer as DALL-E and CogView can largely improve the performance. Finer-grained visual details are nearly inexhaustible, and it may be unfeasible to model them in a transformer with finite resources. We exploit the detail-enhanced GAN to model the finer-grained visual details and enrich the language-related features. The detail-enhanced GAN is a language-guided editing model. More sophisticated editing model will improve the final results, which will be a focus of future works.

6. Conclusion

To synthesize high-quality images from the textual description, we propose an object-enhanced joint-decoding transformer to auto-regressive generate images without pre-generating layouts. To complement the missing finer-grained visual details, we introduce a detail-enhanced GAN to enrich the language-related features and improve the semantic consistency between the given text and the synthesized images. The experimental results show that our approach can synthesize high-quality images while providing more object-centered controllability.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (U21A20487), in part by Australian Research Council Project (DP-180103424), in part by the Shenzhen Technology Project (JCYJ20200109113416531, JCYJ20180507182610734, KCXFZ20201221173411032), and in part by CAS Key Technology Talent Program.

References

[1] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018. 6

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proceedings of Advances in neural information processing systems, NeurIPS*, volume 33, pages 1877–1901, 2020. 2

[3] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8721–8729, 2018. 2

[4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of International Conference on Machine Learning, ICML*, pages 1691–1703, 2020. 3

[5] Jun Cheng, Fuxiang Wu, Yanling Tian, Lei Wang, and Dapeng Tao. Rifegan: Rich feature generation for text-to-image synthesis from prior knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 10908–10917, 2020. 2

[6] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, volume 34, 2021. 2, 3, 5, 6

[7] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 12873–12883, 2021. 3, 4, 5

[8] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1552–1565, 2022. 1, 2, 3, 4, 6

[9] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Generating multiple objects at spatially distinct locations. In *Proceedings of International Conference on Learning Representations, ICLR*, 2019. 1, 2, 6

[10] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1219–1228, 2018. 1

[11] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H.S. Torr. Controllable text-to-image generation. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, pages 2065–2075, 2019. 3

[12] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H.S. Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 7880–7889, 2020. 2, 3, 5

[13] Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 4220–4229, 2019. 2

[14] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 12174–12182, 2019. 1, 2, 3

[15] Yandong Li, Yu Cheng, Zhe Gan, Licheng Yu, Liqiang Wang, and Jingjing Liu. Bachgan: High-resolution image synthesis from salient object layout. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8365–8374, 2020. 2

[16] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6329–6338, 2019. 1

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of European Conference on Computer Vision, ECCV*, pages 740–755, 2014. 6

[18] Zhenhuan Liu, Jincan Deng, Liang Li, Shaofei Cai, Qianqian Xu, Shuhui Wang, and Qingming Huang. Ir-gan: Image manipulation with linguistic instruction by increment reasoning. In *Proceedings of the 28th ACM International Conference on Multimedia, ACM MM*, pages 322–330, 2020. 3

[19] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, pages 42–51, 2018. 2

[20] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *Proceedings of Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 6

[21] Dario Pavllo, Aurelien Lucchi, and Thomas Hofmann. Controlling style and semantics in weakly-supervised image generation. In *Proceedings of European Conference on Computer Vision, ECCV*, pages 482–499, 2020. 2

[22] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1505–1514, 2019. 2

[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. 2

[24] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139, pages 8821–8831, 2021. 2, 3, 5, 6

[25] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of International Conference on Machine Learning, ICML*, pages 1681–1690, 2016. 1, 2

[26] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, pages 2234–2242, 2016. 6

[27] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. Chatpainter: Improving text to image generation using dialogue. In *Proceedings of International Conference on Learning Representations Workshop*, 2018. 1

[28] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 10531–10540, 2019. 2

[29] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, AI for Content Creation Workshop*, 2020. 2

[30] Fuwen Tan, Song Feng, and Vicente Ordonez. Text2scene: Generating compositional scenes from textual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6710–6719, 2019. 1

[31] Hongchen Tan, Xiuping Liu, Meng Liu, Baocai Yin, and Xin Li. Kt-gan: Knowledge-transfer generative adversarial network for text-to-image synthesis. *IEEE Transactions on Image Processing*, 30:1275–1290, 2021. 2

[32] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the International Conference on Neural Information Processing Systems, NeurIPS*, pages 6309–6318, 2017. 2, 3, 4

[33] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of International Conference on Machine Learning, ICML*, pages 1747–1756, 2016. 3

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, pages 5998–6008, 2017. 2

[35] Fuxiang Wu, Jun Cheng, Xinchao Wang, Lei Wang, and Dapeng Tao. Image hallucination from attribute pairs. *IEEE Transactions on Cybernetics*, 52(1):568–581, 2022. 2

[36] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1316–1324, 2018. 1, 2, 5, 6

[37] Mingkuan Yuan and Yuxin Peng. Text-to-image synthesis via symmetrical distillation networks. In *Proceedings of ACM Multimedia Conference, ACM MM*, pages 1047–1415, 2018. 1

[38] Mingkuan Yuan and Yuxin Peng. Bridge-gan: Interpretable representation learning for text-to-image synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11):4258–4268, 2020. 2

[39] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1947–1962, 2019. 1, 2, 6

[40] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6199–6208, 2018. 2

[41] Xingran Zhou, Siyu Huang, Bin Li, Yingming Li, Jiachen Li, and Zhongfei Zhang. Text guided person image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3663–3672, 2019. 3

[42] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. DM-GAN: dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5802–5810, 2019. 1, 2, 5, 6