MEMORY-DRIVEN TEXT-TO-IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a memory-driven semi-parametric approach to text-to-image generation, which is based on both parametric and non-parametric techniques. The non-parametric component is a memory bank of image features constructed from a training set of images. The parametric component is a generative adversarial network. Given a new text description at inference time, the memory bank is used to selectively retrieve image features that are provided as basic information of target images, which enables the generator to produce realistic synthetic results. We also incorporate the content information into the discriminator, together with semantic features, allowing the discriminator to make a more reliable prediction. Experimental results demonstrate that the proposed memory-driven semi-parametric approach produces more realistic images than purely parametric approaches, in terms of both visual fidelity and text-image semantic consistency.

1 INTRODUCTION

How to effectively produce realistic images from given natural language descriptions with semantic alignment has drawn much attention, because of its tremendous potential applications in art, design, and video games, to name a few. Recently, with the vast development of generative adversarial networks (Goodfellow et al., 2014; Gauthier, 2015; Mirza & Osindero, 2014) in realistic image generation, text-to-image generation has made much progress, where the progress has been mainly driven by parametric models — deep networks use their weights to represent all data concerning realistic appearance (Zhang et al., 2017; 2018; Xu et al., 2018; Li et al., 2019a; Qiao et al., 2019b; Zhu et al., 2019; Hinz et al., 2019; Cheng et al., 2020; Qiao et al., 2019a).

Although these approaches can produce realistic results on well-structured datasets, containing a specific class of objects at the image center with fine-grained descriptions, such as birds (Wah et al., 2011) and flowers (Nilsback & Zisserman, 2008), there is still much room to improve. Besides, they usually fail on more complex datasets, which contain multiple objects with diverse backgrounds, e.g., COCO (Lin et al., 2014). This is likely because, for COCO, the generation process involves a large variety in objects (e.g., pose, shape, and location), backgrounds, and scenery settings. Thus, it is much easier for these approaches to only produce text-semantic-matched appearances instead of capturing difficult geometric structure. As shown in Fig. 1, current approaches are only capable of producing required appearances semantically matching the given descriptions (e.g., white and black stripes for zebra), but objects are unrealistic with distorted shape. Furthermore, these approaches are in contrast to earlier works on image synthesis, which were based on non-parametric techniques that could make use of large datasets of images at inference time (Chen et al., 2009; Hays & Efros, 2007; Isola & Liu, 2013; Zhu et al., 2015; Lalonde et al., 2007). Although parametric approaches can enable the benefits of end-to-end training of highly expressive models, they lose a strength of earlier non-parametric techniques, as they fail to make use of large datasets of images at inference time.

In this paper, we introduce a memory-driven semi-parametric approach to text-to-image generation, where the approach takes the advantage of both parametric and non-parametric techniques. The non-parametric component is a memory bank of disentangled image features constructed from a training set of real images. The parametric component is a generative adversarial network. Given a novel text description at inference time, the memory bank is used to selectively retrieve compatible image features that are provided as basic information, allowing the generator to directly draw clues of target images, and thus to produce realistic synthetic results.

Besides, to further improve the differentiation ability of the discriminator, we incorporate the content information into it. This is because, to make a prediction, the discriminator usually relies on semantic

Figure 1: Examples of text-to-image generation on COCO. Current approaches only generate low-quality images with unrealistic objects. In contrast, our method can produce realistic images, in terms of both visual appearances and geometric structure.

features, extracted from a given image using a series of convolution operators with local receptive fields. However, when the discriminator goes deeper, less content details are preserved, including the exact geometric structure information (Gatys et al., 2016; Johnson et al., 2016). We think that the loss of content details is likely one of the reasons why current approaches fail to produce realistic shapes for objects on difficult datasets, such as COCO. Thus, the adoption of content information allows the model to exploit the capability of content details and then improve the discriminator to make the final prediction more reliable.

Finally, an extensive experimental analysis is performed, which demonstrates that our memory-driven semi-parametric method can generate more realistic images from natural language, compared with purely parametric models, in terms of both visual appearances and geometric structure.

2 Related Work

Text-to-image generation has made much progress because of the success of generative adversarial networks (GANs) (Goodfellow et al., 2014) in realistic image generation. Zhang et al. (2017) proposed a multi-stage architecture to generate realistic images progressively. Then, attention-based methods (Xu et al., 2018; Li et al., 2019a) are proposed to further improve the results. Zhu et al. (2019) introduced a dynamic memory module to refine image contents. Qiao et al. (2019a) proposed text-visual co-embeddings to replace input text with corresponding visual features. Cheng et al. (2020) introduced a rich feature generating text-to-image synthesis. Besides, extra information is adopted on the text-to-image generation process, such as scene graphs (Johnson et al., 2018; Ashual & Wolf, 2019) and layout (e.g., bounding boxes or segmentation masks) (Hong et al., 2018; Li et al., 2019b; Hinz et al., 2019). However, none of the above approaches adopt non-parametric techniques to make use of large datasets of images at inference time, neither feed content information into the discriminator to enable a finer training feedback. Also, our method does not make use of any additional semantic information, e.g., scene graphs and layout.

Text-guided image manipulation is related to our work, where the task also takes natural language descriptions and real images as inputs, but it aims to modify the images using given texts to achieve semantic consistency (Nam et al., 2018; Dong et al., 2017; Li et al., 2020). Differently from it, our work focuses mainly on generating novel images, instead of editing some attributes of the given images. Also, the real images in the text-guided image manipulation task behave as a condition, where the synthetic results should reconstruct all text-irrelevant attributes from the given real images. Differently, the real images in our work are mainly to provide the generator with additional cues of target images, in order to ease the whole generation process.

Memory Bank. Qi et al. (2018) introduced a semi-parametric approach to realistic image generation from semantic layouts. Li et al. (2019c) used the stored image crops to determine the appearance

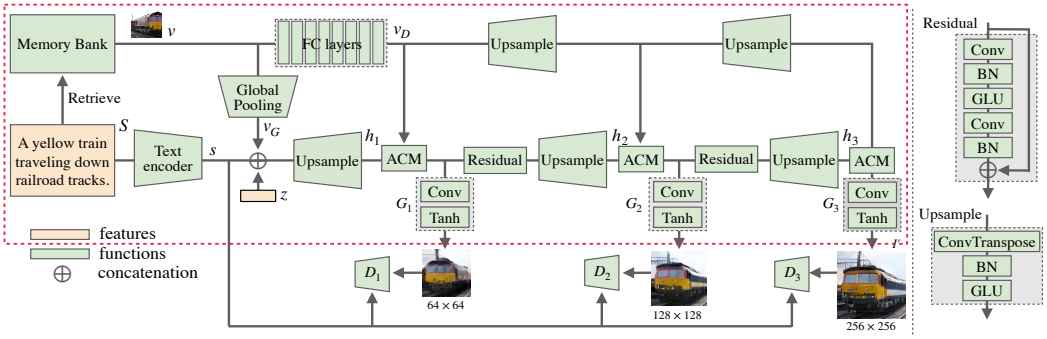

Figure 3: The architecture of our proposed method. The red box indicates the inference pipeline that retrieves image features from a memory bank according to the a given description S, during in training, we directly feed image features from the text-paired training image. z is a random vector drawn from the Gaussian distribution, ACM denotes the text-image affine combination module.

of objects. Tseng et al. (2020) used a differentiable retrieval process to select mutually compatible image patches. Li et al. (2021) studied conditional image extrapolation to synthesize new images guided by the input structured text. Differently, instead of using a concise semantic representation (a scene graph as input), which is less user-friendly and has limited context of general descriptions, we use natural language descriptions as input. Also, Liang et al. (2020) designed a memory structure to parse the textual content. Differently, our method simply uses a deep network to extract image features, instead of involving complex image preprocessing to build a memory bank.

3 Overview

Given a sentence S, we aim to generate a fake image I' that is semantically aligned with the given S. The proposed model is trained on a set of paired text description and corresponding real image features v, denoted by (S, v). This set is also used to generate a memory bank M of disentangled image features v for different categories, where image features are extracted from the training image by using a pretrained VGG-16 network (Simonyan & Zisserman, 2014) (see Fig. 2). Each element in M is an image feature extracted from a training image, associated with corresponding semantically-matched text descriptions from the training datasets.

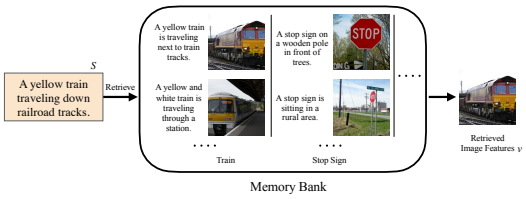

Figure 2: The design of the memory bank to provide image features at inference time. Note that we use the corresponding real training image to represent image features for a better visualization.

At inference time, we are given a novel text description S that was not seen during training. Then, S is used to retrieve semantically-aligned image features from the memory bank M, based on designed matching algorithms (more details are shown in Sec. 4.2). Next, the retrieved image features v, together with the given text description S, are fed into the generator to synthesize the output image (see Fig. 3). The generator utilizes the information from the image features, fuses them with hidden features produced from the given text description S, and generate realistic images semantically-aligned with S. The architecture and training of the network are described in Sec. 5.

To incorporate image features into the generation pipeline, we borrow from the text-guided image manipulation literature (Li et al., 2020), and redesign the architecture to make full use of the given image features in text-to-image generation, shown in Fig. 3.

4 Memory Bank

4.1 Representation

The memory bank M is a set of image features v_i extracted from training set images, and each image features v_i is associated with matched text descriptions that are provided in the dataset, e.g., in COCO, each image has five matched text descriptions. These descriptions are used in the matching algorithms, allowing a given text to find the most compatible image features at inference time.

4.2 RETRIEVAL

Given a new text description, in order to effectively retrieve the most compatible image features from the memory bank M, we have designed several matching algorithms and also explored the effectiveness of each algorithms. A detailed comparison between different algorithms is shown in the supplementary material.

4.2.1 SENTENCE-SENTENCE MATCHING

Here, we use image features' associated sentences S'_i as keys, to find the most compatible image features v_i for a given unseen sentence S at inference time. First, we feed both S and S'_i into a pretrained text encoder (Xu et al., 2018) to produce sentence features $s \in \mathbb{R}^{D\times 1}$ and $s'_i \in \mathbb{R}^{D\times 1}$, respectively, where D is the feature dimension. Then, for the given sentence S, we select the most compatible image features v_i in M based on a cosine similarity score:

$$\alpha_i = \frac{(s)^T s'_i}{\|s\| \, \|s'_i\|}. \tag{1}$$

Finally, we fetch the image features v_i using the key S'_i with the highest similarity score α_i.

4.2.2 SENTENCE-IMAGE MATCHING

Instead of using associated sentences as keys, we can calculate the similarity between the sentence feature $s \in \mathbb{R}^{D\times 1}$ and image features $v_i \in \mathbb{R}^{D\times H\times W}$ stored in M, where D is the number of channels, H is the height, and W is the width. To directly calculate the similarity, we first average the image features on the spatial direction to get a global image feature $v_{Gi} \in \mathbb{R}^{D\times 1}$. So, for a given unseen S, we select the most compatible image features v_i in M based on β_i:

$$\beta_i = \frac{(s)^T v_{Gi}}{\|s\| \, \|v_{Gi}\|}. \tag{2}$$

4.2.3 WORDS-WORDS MATCHING

Moreover, we can use a more fine-grained text representation (namely, word embeddings), as keys to find the most compatible image features v_i stored in M for a given unseen sentence S. At inference time, we first feed both S and S'_i into a pretrained text encoder (Xu et al., 2018) to generate word embeddings $w \in \mathbb{R}^{N\times D}$ and $w'_i \in \mathbb{R}^{N\times D}$, respectively, where N is the number of words and D is the feature dimension. Then, we reshape the size of both w and w'_i to $\mathbb{R}^{(D*N)\times 1}$. So, to find the most compatible image features, the cosine similarity score can be defined as follows:

$$\delta_i = \frac{(w)^T w'_i}{\|w\| \, \|w'_i\|}. \tag{3}$$

However, different words in a sentence are not equally important. Thus, if we simply combine all words from a sentence together to calculate the similarity (like above), the similarity score may be less precise. To solve this issue, during training, we reweight each word in a sentence by its importance. We first use convolutional layers to remap word embeddings, and then calculate the importance λ (and λ'_i) for each word in word embeddings $w \in \mathbb{R}^{N\times D}$ (and $w'_i \in \mathbb{R}^{N\times D}$), denoted by: $\lambda = \text{Softmax}(ww^T)$ and $\lambda'_i = \text{Softmax}(w'_i w'^T_i)$, respectively.

Each elements in λ represents the correlation between different words in a sentence. Then, λw (and $\lambda'_i w'_i$) reweight word embeddings for each word based on its correlation with other words. So, using this reweighted word embeddings, we can achieve a more precise similarity calculation between two word embeddings. At inference time, after we reshape the size of both λw and $\lambda'_i w'_i$ to $\mathbb{R}^{(D*N)\times 1}$, the new equation is defined as follows:

$$\delta_i = \frac{(\lambda w)^T \lambda'_i w'_i}{\|\lambda w\| \, \|\lambda'_i w'_i\|}. \tag{4}$$

4.2.4 WORDS-IMAGE MATCHING

Furthermore, we use the word embeddings $w \in \mathbb{R}^{N\times D}$ and image features $v_i \in \mathbb{R}^{D\times H\times W}$ to directly calculate the similarity score between them. To achieve this, we first reshape the size of the

image features to $v_i \in \mathbb{R}^{D\times(H*W)}$. Then, a correlation matrix $c_i \in \mathbb{R}^{N\times(H*W)}$ can be obtained via: $c_i = \text{Softmax}(wv_i)$, where each element in c_i represents the correlation between each word and each image spatial location. Then, a reweighted word embedding $\tilde{w_i} \in \mathbb{R}^{N\times D}$ containing image information can be achieved by $\tilde{w_i} = c_i v_i^T$. So, to find the most compatible image features, we first reshape the size of both w and $\tilde{w_i}$ to $\mathbb{R}^{(D*N)\times 1}$, and the similarity score is defined as follows:

$$\gamma_i = \frac{(w)^T \tilde{w_i}}{\|w\| \, \|\tilde{w_i}\|}. \tag{5}$$

Similarly, we can also reweight word embeddings w and image features v_i based on their importance (see Sec.4.2.3) to achieve a more precise calculation.

5 Generative Adversarial Networks

To generate high-quality synthetic images from natural language descriptions, we propose to incorporate image features v, along with the given sentence S, into the generator. To incorporate image features into the generation pipeline, we borrow from the text-guided image manipulation literature (Li et al., 2020), and redesign the architecture to make full use of the given image features in text-to-image generation, shown in Fig. 3.

5.1 Generator with Image Features

To avoid the identity mapping and also to make full use of image features v in the generator, we first average v on each channel to filter potential content details (e.g., overall spatial structure) contained in v, getting a global image feature v_G, where v_G only keeps basic information of the corresponding real image I, serving as basic image priors. By doing this, the model can effectively avoid copying and pasting from I, and greatly ensure the diversity of output results, especially on the first stage. This is because the following stages focus more on refining basic images produced by the first stage, according to adding more details and improving their resolution, shown in Fig. 3.

However, only feeding the global image feature v_G at the beginning of the network, the model may fail to fully utilize the cues contained in the image features v. Thus, we further incorporate the image features v at each stage of the network. The reason to feed image features v rather than the global feature v_G at the following stages is that v contains more information about the desired output image, such as image contents and geometric structure of objects, where these details can work as candidate information for the main generation pipeline to select. To enable this regional selection effect, we adopt the text-image affine combine module (ACM) (Li et al., 2020), which is able to selectively fuse text-required image information within v into the hidden features h, where h is generated from the given text description S. However, simply fusing image features v into the generation pipeline may introduce constraints on producing diverse and novel synthetic results, because different image information (e.g., objects and visual attributes) in v may be entangled, which means, for example, if the model only wants to generate one object, the corresponding entangled parts (e.g, objects and attributes) may be produced as well. This may cause an additional generation of text-irrelevant objects and attributes. Thus, to avoid these drawbacks, inspired by the study (Karras et al., 2019), we use several fully connected layers to disentangle the image features v, getting disentangled image features v_D, which allows the model to disconnect relations between different objects and also attributes. By doing this, the model is able to prevent the constraints introduced by the image features v, and then selectively choose text-required image information within v_D, where this information is effectively disentangled without a strong connection.

Why does the generator with image features work better? Ideally, the generator produces a sample, e.g., an image, from a latent code, and the distribution of these samples should be indistinguishable from the training distribution, where the training distribution is actually drawn from the real samples in the training dataset. Based on this, incorporating image features from real images in training dataest into the generator allows the generator to directly draw cues of the desired distribution that it eventually needs to generate. Besides, the global feature v_G and disentangled image features v_D can provide basic information of target results in advance, and also work as candidate information, allowing the model to selectively choose text-required information without generating it by the model itself, and thus easing the whole generation process. To some extent, the global feature v_G can be seen as the meta-data of target images, which may contain information about what kinds of objects to generate, e.g., zebra or bus, and v_D is able to provides basic information of objects, e.g., the spatial structure like four legs and one head for the zebra and the rectangle shape for the bus.

5.2 DISCRIMINATOR WITH CONTENT INFORMATION

To further improve the discriminator to make a more reliable prediction, with respect to both visual appearances and geometric structure, we propose to incorporate the content information into it. This is mainly because, in a deep convolution neural network, when the network goes deeper, the less content details are preserved, including the exact shape of objects (Gatys et al., 2016; Johnson et al., 2016). We think the loss of content details may prevent the discriminator to provide fine-grained shape-quality-feedback to the generator, which may cause the difficulty for the generator to produce realistic geometric structure. Also, Zhou et al. (2014) showed that the empirical receptive field of a deep convolution neural network is much smaller than the theoretical one especially on deep layers. This means, using convolution operators with a local receptive field only, the network may fail to capture the spatial structure of objects when the size of objects exceeds the receptive field.

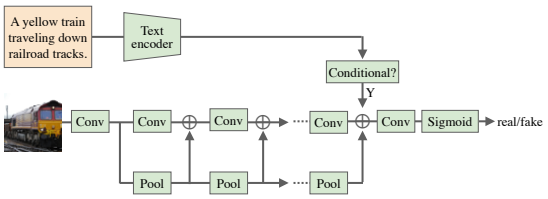

Figure 4: The architecture of the proposed discriminator with the incorporation of content information.

To incorporate the content details, we propose to generate a series of image content features, $\{a_{128}, a_{64}, a_{32}, \ldots, a_4\}$, by aggregating different image regions via applying pooling operators on the given real or fake features. The size of these content features is from $a_{128} \in \mathbb{R}^{C\times128\times128}$ to $a_4 \in \mathbb{R}^{C\times4\times4}$, where C represents the number of channels, and the width and the height of the next image content features are $1/2$ the previous one. Thus, the given image is pooled into representations for different regions, from fine- (a_{128}) to coarse-scale (a_4), which is able to preserve content information of different subregions, such as the spatial structure of objects. Then, these features are concatenated with the corresponding hidden features on the channel-wise direction, incorporating the content information into the discriminator.

The number of different-scale content features can be modified, which is dependent on the size of given images. These features aggregate different image subregions by repetitively adopting fixed-size pooling kernels with a small stride. Thus, these content features maintain a reasonable small gap for image information. For the type of pooling operation between max and average, we perform comparison studies to show the difference in Sec. 6.2.

Why does the discriminator with content information work better? Basically, the discriminator in a generative adversarial network is simply a classifier (Goodfellow et al., 2014). It tries to distinguish real data from the data created by the generator (note that in our method, we implement the Minmax loss in the loss function, instead of the Wasserstein loss (Arjovsky et al., 2017)). Also, the implementation of content information has shown its great effectiveness on classification (Lazebnik et al., 2006; He et al., 2015) and semantic segmentation (Liu et al., 2015; Zhao et al., 2017). Based on this, incorporating the content information into the discriminator is helpful, allowing the discriminator to make a more reliable prediction on complex datasets, especially for the datasets with complex image scenery settings, such as COCO.

5.3 TRAINING

To train the network, we follow (Li et al., 2020) and adopt adversarial training. There are three stages in the model, and each stage has a generator network and a discriminator network. The generator and discriminator are trained alternatively by minimizing the generator loss $\mathcal{L}_G$ and discriminator loss $\mathcal{L}_D$. Please see the supplementary material for more details about training objectives. We only highlight some training differences compared with Li et al. (2020).

Generator objective. The objective functions to train the generator are similar as in (Li et al., 2020), but, differently, the inputs for the generator are a pair of (S, v) and a noise z, denoted by $G_i(z, S, v)$, where i indicates the stage number.

Discriminator objective. To improve the convergence of our GAN-based generation model, the R_1 regularization (Mescheder et al., 2018) is adopted in the discriminator:

$$R_1(\psi) := \frac{\gamma}{2} E_{p_D(x)} \left[\|\bigtriangledown D_\psi(x)\|^2 \right], \tag{6}$$

where ψ represents parameter values of the discriminator.

6 EXPERIMENTS

Table 1: Quantitative comparison on CUB bird: Fréchet inception distance (FID) and R-precision (R-psr) of StackGAN++ (Zhang et al., 2018), AttnGAN (Xu et al., 2018), ControlGAN (Li et al., 2019a), DM-GAN (Zhu et al., 2019), DF-GAN(Tao et al., 2020), and our method. For FID, lower is better, while for R-precision, alignment, and realism, higher is better.

Matrix	StackGAN++	AttnGAN	ControlGAN	DM-GAN	DF-GAN	Ours
FID	15.30	23.98	13.92	16.09	14.81	10.49
R-psr	46.67	67.82	69.33	72.31	-	73.87
Alignment (%)	-	-	-	-	43	57
Realism (%)	-	-	-	-	31	69

Table 2: Quantitative comparison on COCO. Note that we also compare our method with OP-GAN (Hinz et al., 2019), where OP-GAN adopts **bounding box** in their method.

Matrix	StackGAN++	AttnGAN	ControlGAN	DM-GAN	DF-GAN	Ours	OP-GAN
FID	81.59	32.32	33.58	32.64	21.42	19.47	24.70
R-prs	71.88	85.47	82.43	88.56	-	90.32	89.01
Alignment (%)	-	-	-	-	29	71	-
Realism (%)	-	-	-	-	22	78	-

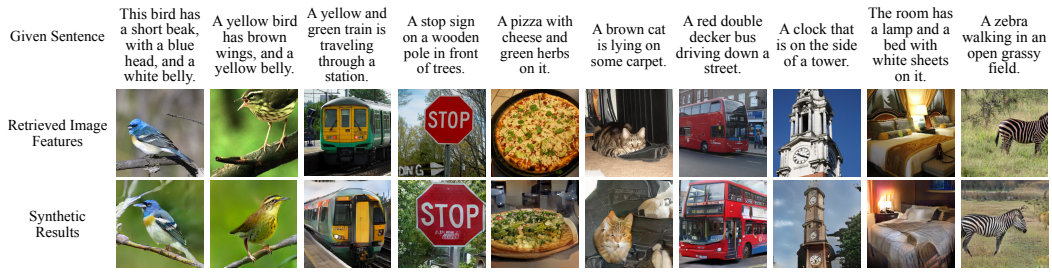

Figure 5: Qualitative results on CUB and COCO: top row is the given unseen sentences; middle row: the image features extracted from the memory bank M (we use corresponding images to represent the image features for a better visualization); bottom row: the synthetic results.

Figure 6: Qualitative comparison between AttnGAN (Xu et al., 2018), DF-GAN (Tao et al., 2020), and our method on COCO.

To verify the effectiveness of our proposed method in realistic image generation from text descriptions, we conduct extensive experiments on the CUB bird (Wah et al., 2011) dataset and more complex COCO (Lin et al., 2014) dataset, where COCO contains multiple objects with diverse backgrounds.

Evaluation metrics. We adopt the Fréchet inception distance (FID) (Heusel et al., 2017) as the primary metric to quantitatively evaluate the image quality and diversity. In our experiments, we use 30K synthetic images vs. 30K real test images to calculate the FID value. However, as FID cannot reflect the relevance between an image and a text description, we use the R-precision (Xu et al., 2018) to measure the correlation between a generated image and its corresponding text.

Human evaluation. To better verify the performance of our proposed method, we conducted a user study between current state-of-the-art method DF-GAN (Tao et al., 2020) and ours on CUB and COCO. We randomly selected 100 text descriptions from the test dataset. Then, we asked 5 workers to compare the results after looking at the output images and given text descriptions based on two criteria: (1) alignment: whether the synthetic image is semantically aligned with the given description, and (2) realism: whether the synthetic image looks realistic, shown in Tables 1 and 2. Please see supplementary material for more details about the human evaluation.

Implementation. There are three stages in the model, and each stage has a generator network and a discriminator network. The number of stages can be modified, which depends on the resolution of the output image. We utilize a deep neural network layer relu5_3 of a pre-trained VGG-16 to extract image features v, which is able to filter content details in I and keep more semantic information. In the discriminator, the number of different-scale image content features can be modified, which is related to the size of the given image. A same-size pooling kernel with a small stride (stride $= 2$) is repeatedly implemented on the image features, to maximize the preservation of the content information. For the type of pooling operation, average pooling is adopted. For the matching algorithms, word image matching with reweighting based on importance is adopted. The resolution of synthetic results is 256×256. Our method and its variants are trained on a single Quadro RTX 6000 GPU, using the Adam optimizer (Kingma & Ba, 2014) with the learning rate 0.0002. The hyperparameter λ is set to 5. We preprocess datasets according to the method used in (Xu et al., 2018). **No** attention module is implemented in the whole architecture.

6.1 Comparison with Other Approaches

Quantitative comparison. Quantitative results are shown in Tables 1 and 2. As we can see, compared to other approaches, our method achieves better FID and R-precision scores on both datasets, and even has a better performance than OP-GAN, where OP-GAN adopts bounding boxes. This indicates that (1) our method can produce more realistic images from given text descriptions, in terms of image quality and diversity, and (2) synthetic results produced by our method are more semantically aligned with the given text descriptions. Besides, in human evaluation, our method achieves better alignment and realism scores, compared with DF-GAN, which indicates that our results are most preferred by workers, which further verifies the better performance of our method, with respect to semantic alignment and image realism.

Qualitative comparison. In Fig. 5, we present synthetic examples produced by our method at 256×256, along with the corresponding retrieved images that provide image features. As we can see, our method is able to produce high-quality results on CUB and COCO, with respect to realistic appearances and geometric structure, and also semantically matching the given text descriptions. Besides, the synthetic results are different from the retrieved image features, which indicates there is no significant copy-and-paste problem in our method.

Figure 7: Diversity. Top row shows the fixed sentence and image features, where we use the corresponding images to represent image features for a better visualization. The bottom presents diverse synthetic images produced by only changing the input noise z.

Diversity evaluation. To further evaluate the diversity of our method, we fix the given text description and the corresponding retrieved image features, and only change the given noise z to generate output images, shown in Fig. 7. When we fix the sentence and image features and only change the noise, our method can generate obviously different images, but they still semantically match the given sentence and also make use information from the image features. More evaluations are shown in the supplementary material.

6.2 COMPONENT ANALYSIS

Effectiveness of the image features. To better understand the effectiveness of image features in the generator, we conduct an ablation study shown in Table 3. Without image features, the model "Ours w/o Feature" achieves worse quantitative results on both FID and R-precision compared with the baseline, which verifies the effectiveness of image features on high-quality image generation. Interestingly, without image features, even our method becomes a pure text-to-image generation method, similar to other baselines, but the FID of "Ours w/o Feature" is still competitive with other baselines. This indicate that even without the image features fed into our method, our method can still generate better synthetic results, with respect to image quality and diversity. We think this is mainly because with the help of content information, our better discriminator is able to make a more reliable prediction on complex datasets, which in turn encourages the generator to produce better synthetic images.

Table 3: Ablation studies: "Ours w/o Feature" denotes without feeding image features into the generator, "Ours w/o Disen." denotes without using the fully connected layers to disentangle image features v, "Ours w/o Disen.*" is for mismatched pairs, "Ours w/o Content" denotes without incorporating the content information into the discriminator, "Ours w/o Reg." denotes without using the regularization in the discriminator, "Ours w/ Max" denotes using the maximum pooling to extract content information, and "Ours w/ Aver" denotes using the average pooling.

Method	FID	R-psr
Ours w/o Feature	22.20	84.63
Ours w/o Disen.	18.82	92.17
Ours w/o Disen.*	18.80	67.05
Ours w/o Content	20.96	88.95
Ours w/o Reg.	27.12	82.97
Ours w/ Max	26.12	83.11
Ours w/ Aver (baseline)	19.47	90.32

Effectiveness of the disentanglement. Here, we show the effectiveness of the fully connected layers applied on the image features v. Interestingly, from Table 3, the "model w/o Disen." achieves better FID and R-precision compared with the baseline. This is likely because the model may suffer from an identity mapping problem. To verify this identity mapping problem, we conduct another experiment, where we feed mismatched sentence and image pairs into the network without using search algorithms, denoted "model w/o Disen.*". As we can see, on mismatched pairs, although FID is still low, the R-precision degrades significantly.

Effectiveness of the content information. To verify the effectiveness of the content information adopted in the discriminator, we conduct an ablation study, shown in Table 3. As we can see, FID and R-precision degrade when the discriminator without adopting the content information. This may indicate that the content information can effectively strengthens the differentiation abilities of the discriminator. Then, the improved discriminator is able to provide the generator with fine-grained training feedback, regarding to geometric structure, thus facilitating training a better generator to produce higher-quality synthetic results.

Comparison between different pooling types. Here, we conduct a comparison study on different pooling types (i.e., max and average) in Table 3. As we can see, the model with the average pooling works better than max pooling. We think that this is likely because max pooling fails to capture the contextual information between neighboring pixels, because it only picks the maximum value among a region of pixels, while average pooling calculates the average value between them.

Effectiveness of the regularization. We evaluate the effectiveness of the adopted regularization in the discriminator. From Table 3, the model without the regularization has worse quantitative results, compared with the full model. We think that this is because the regularization effectively improves GAN convergence by preventing the generator from training on junk feedback, once the discriminator cannot easily tell the difference between real and fake.

7 CONCLUSION

We have introduced a memory-driven semi-parametric approach to text-to-image generation, which utilizes large datasets of images at inference time. Also, an alternative architecture is proposed for both the generator and the discriminator. Extensive experimental results on two datasets demonstrate the effectiveness of feeding retrieved image features into the generator and incorporating content information into the discriminator.

8 ETHICS STATEMENT

All datasets and baselines used in the paper are public with corresponding citations. Our research mainly explores the interaction between different modal features, and aims to achieve an effective transformation from one domain to the other, which might not have significant potentially harmful insights and potential conflicts of interest and sponsorship.

9 REPRODUCIBILITY STATEMENT

To reproduce our results, we include the details of the datasets we used in our paper (see Sec. D). In the implementation section (see Sec. 6), we show more details on our network, including how to extract image features, and how to generate content information used in the discriminator. We also include the values of hyperparameters, and the kinds of devices that we used to train our network. Sec. 5.3 and Sec. B show objective functions to train our network. Also, all data and baselines used in our paper are public with corresponding citations. We will release our code after the conference.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4561–4569, 2019.

Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2Photo: Internet image montage. *ACM Transactions on Graphics (TOG)*, 28(5):1–10, 2009.

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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10911–10920, 2020.

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

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

Jon Gauthier. Conditional generative adversarial networks for convolutional face generation. *Technical report*, pp. 3, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

James Hays and Alexei A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (ToG)*, 26(3):4–es, 2007.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *arXiv preprint arXiv:1910.13321*, 2019.

Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7986–7994, 2018.

Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369. IEEE, 2010.

Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3048–3055, 2013.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*, pp. 694–711. Springer, 2016.

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*, pp. 1219–1228, 2018.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jean-François Lalonde, Derek Hoiem, Alexei A. Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *ACM Transactions on Graphics (TOG)*, 26(3):3–es, 2007.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 2169–2178. IEEE, 2006.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. In *Advances in Neural Information Processing Systems*, pp. 2063–2073, 2019a.

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

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*, pp. 12174–12182, 2019b.

Yijun Li, Lu Jiang, and Ming-Hsuan Yang. Controllable and progressive image extrapolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2140–2149, 2021.

Yikang Li, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. PasteGAN: A semi-parametric method to generate image from scene graph. *Advances in Neural Information Processing Systems*, 32:3948–3958, 2019c.

Jiadong Liang, Wenjie Pei, and Feng Lu. CPGAN: content-parsing generative adversarial networks for text-to-image synthesis. In *European Conference on Computer Vision*, pp. 491–508. Springer, 2020.

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 the European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

Wei Liu, Andrew Rabinovich, and Alexander C. Berg. ParseNet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? *arXiv preprint arXiv:1801.04406*, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *Advances in Neural Information Processing Systems*, pp. 42–51, 2018.

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

Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8808–8816, 2018.

Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Learn, imagine and create: Text-to-image generation from prior knowledge. *Advances in Neural Information Processing Systems*, 32: 887–897, 2019a.

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*, pp. 1505–1514, 2019b.

Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *arXiv preprint arXiv:1711.09404*, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. DF-GAN: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020.

Hung-Yu Tseng, Hsin-Ying Lee, Lu Jiang, Ming-Hsuan Yang, and Weilong Yang. RetrieveGAN: Image synthesis via differentiable patch retrieval. In *European Conference on Computer Vision*, pp. 242–257. Springer, 2020.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. 2011.

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*, pp. 1316–1324, 2018.

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

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, 2018.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, 2017.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene CNNs. *arXiv preprint arXiv:1412.6856*, 2014.

Jun-Yan Zhu, Philipp Krahenbuhl, Eli Shechtman, and Alexei A. Efros. Learning a discriminative model for the perception of realism in composite images. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3943–3951, 2015.

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*, pp. 5802–5810, 2019.

A ARCHITECTURE

Here we show details about the network architectures for the components of our model.

A.1 TEXT ENCODER

The text encoder used in our method is a pretrained bidirectional LSTM (Xu et al., 2018), which is trained together with an image encoder Inception-v3 (Szegedy et al., 2016), maximizing the cosine similarity between text features and the corresponding image features. The text features are encoded from a given text description using the text encoder, and the image features are extracted from the corresponding matched image.

A.2 IMAGE ENCODER

The image encoder used in our main architecture is a VGG-16 (Simonyan & Zisserman, 2014) network, pretrained on ImageNet (Russakovsky et al., 2015). A deep neural network layer relu5_3 is adopted to extract image features. Thus, the image features are able to contain more semantic information than content details.

A.3 TEXT-IMAGE AFFINE COMBINATION MODULE

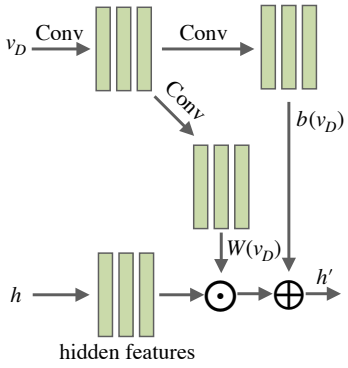

Figure 8: The architecture of the affine combination module.

To better fuse different-modal text and image features, and also to enable a regional selection effect, we adopt the text-image affine combination module (Li et al., 2020), shown in Fig. 8. The affine combination module takes two inputs: (1) the hidden features $h \in \mathbb{R}^{C\times H\times W}$ from the given text description or intermediate hidden representation between two stages, where C is the number of channels, H is the height, and W is the width of the feature map, and (2) the corresponding disentangled image features $v_D \in \mathbb{R}^{C\times H\times W}$, achieved by applying fully connected layers on the image features.

According to applying two convolutional layers, the disentangled image features v_D are converted into trainable weights $W(v_D) \in \mathbb{R}^{C\times H\times W}$ and trainable biases $b(v_D) \in \mathbb{R}^{C\times H\times W}$. Then, the fused feature $h' \in \mathbb{R}^{C\times H\times W}$ is generated by

$$h' = h \odot W(v_D) + b(v_D), \tag{7}$$

where W and b represent the functions that convert the image features v_D into weights $W(v_D)$ and biases $b(v_D)$, and $\odot$ denotes the Hadamard element-wise product.

A.4 REWEIGHTING IMAGE FEATURES BASED ON IMPORTANCE

Here, we show how to reweight image features based on its importance, mentioned in Sec. 4.2.4. First, during the training, we use convolutional layers to remap image features, and then reshape image features into $v \in \mathbb{R}^{D\times(H*W)}$. Thus, to calculate the importance λ for each spatial locations in

image features, we apply the following equation: $\lambda = \text{Softmax}(v^T v)$, where $\lambda \in \mathbb{R}^{(H*W)\times(H*W)}$, and each element in λ represents the correlation between different spatial locations. Finally, we reweight image features based on importance by adopting $v\lambda$.

B OBJECTIVE FUNCTIONS

Here we show the complete objective functions for training our method. The discriminator and generator in our model are trained alternatively by minimizing both the generator loss $\mathcal{L}_G$ and the discriminator loss $\mathcal{L}_D$.

B.1 GENERATOR OBJECTIVE

The generator objective for training a generator at stage i contains an unconditional adversarial loss, a conditional adversarial loss, and a text-image matching loss $\mathcal{L}_{\text{DAMSM}}$ (Xu et al., 2018).

$$\begin{aligned}\mathcal{L}_{G_i} = &\underbrace{-\frac{1}{2}E_{z\sim P_z, v\sim P_{\text{data}}}\left[\log(D_i(G_i(z,S,v)))\right]}_{\text{unconditional adversarial loss}} \\ &\underbrace{-\frac{1}{2}E_{z\sim P_z, v\sim P_{\text{data}}}\left[\log(D_i(G_i(z,S,v),S))\right]}_{\text{conditional adversarial loss}} + \lambda\mathcal{L}_{\text{DAMSM}},\end{aligned} \tag{8}$$

where G_i and D_i represent the corresponding generator network and discriminator network at stage i, respectively, S is the text description, v is the image features that are extracted from the corresponding real image I that correctly semantically matches S, where the I is sampled from the true distribution P_{data}, z is a noise vector drawn from the Gaussian distribution P_z.

Thus, the complete objective function for training the generator networks is:

$$\mathcal{L}_G = \sum_{k=1}^{K}(\mathcal{L}_{G_i}), \tag{9}$$

where K is the total number of stages in the network.

B.2 DISCRIMINATOR OBJECTIVE

The discriminator objective for training a discriminator at stage i contains an unconditional adversarial loss and a conditional adversarial loss.

$$\begin{aligned}\mathcal{L}_{D_i} = &\underbrace{-\frac{1}{2}E_{I_i\sim P_{\text{data}}}\left[\log(D_i(I_i))\right] - \frac{1}{2}E_{z\sim P_z}\left[\log(1-D_i(G_i(z,S,v)))\right]}_{\text{unconditional adversarial loss}} \\ &\underbrace{-\frac{1}{2}E_{I_i\sim P_{\text{data}}}\left[\log(D_i(I_i,S))\right] - \frac{1}{2}E_{z\sim P_z}\left[\log(1-D_i(G_i(z,S,v),S))\right]}_{\text{conditional adversarial loss}},\end{aligned} \tag{10}$$

where I_i denotes the real image sampled from the true image distribution P_{data} at stage i. Thus, the complete objective function for training the discriminator networks is:

$$\mathcal{L}_D = \sum_{k=1}^{K}(\mathcal{L}_{D_i}) + R_1(\psi), \tag{11}$$

where $R_1(\psi)$ is a regularization term described in the paper. This regularization term is derived from zero-centered gradient penalties (Ross & Doshi-Velez, 2017) on local stability, which penalizes the discriminator for deviating from the Nash-equilibrium. This ensures that when a GAN-based model converges (i.e., the generator produces the true data distribution), the discriminator cannot create a non-zero gradient orthogonal to the data manifold without suffering a loss in the GAN game.

C EVALUATION METRICS

In this section, we show more details about the evaluation metrics used in the paper.

C.1 FRÉCHET INCEPTION DISTANCE

The Fréchet inception distance (FID) (Heusel et al., 2017) measures the Fréchet distance between generated image features and real image features, where both features are extracted by an Inception-v3 network (Szegedy et al., 2016) pretrained on ImageNet (Russakovsky et al., 2015). Consequently, a lower FID implies a closer distance between the synthetic image distribution and the real image distribution.

C.2 R-PRECISION

To measure the semantic alignment between the synthetic image and the given text description, the R-precision (Xu et al., 2018) is adopted. The R-precision is calculated by retrieving relevant text descriptions given an image query. To measure the relevance between the text and the image, the cosine similarity between text and image features is adopted. Thus, we compute a global image vector and 100 candidate sentence vectors, where the 100 candidate sentence vectors contain R number of ground-truth text descriptions that correctly describe the image, and $100 - R$ randomly chosen mismatched descriptions. For each image query, if a results in the top R ranked retrieval text descriptions are relevant, then the R-precision is a/R. In the paper, we measure the top-1 R-precision (i.e., $R = 1$).

D MORE EXPERIMENTS

In this section, we show additional experimental results to further evaluate and verify the performance of our proposed method.

D.1 DATASETS

CUB bird (Wah et al., 2011) contains 8,855 training images and 2,933 test images, and each image has 10 corresponding text descriptions. COCO (Lin et al., 2014) contains 82,783 training images and 40,504 validation images. Each image has 5 descriptions.

D.2 QUANTITATIVE COMPARISON BETWEEN DIFFERENT ALGORITHMS

Table 4: Quantitative comparison between different matching algorithms on CUB. Sent. represents sentence, and ReW represents we reweighting word embeddings and (or) image features based on importance. For FID, lower is better, while for R-precision, higher is better.

Matrix	Sent. & Sent.	Sent. & Image	Word & Word	Word & Word & ReW	Word & Image	Word & Image & ReW
FID	11.34	11.41	10.98	10.88	11.03	10.49
R-psr	69.98	68.47	71.32	72.88	71.36	73.87

Table 5: Quantitative comparison between different matching algorithms on COCO. For FID, lower is better, while for R-precision, higher is better.

Matrix	Sent. & Sent.	Sent. & Image	Word & Word	Word & Word & ReW	Word & Image	Word & Image & ReW
FID	20.87	20.76	19.98	20.03	20.12	19.47
R-psr	86.76	86.34	89.02	89.23	88.98	90.32

Here, we show the quantitative comparison between different matching algorithms, shown in Tables 4 and 5. As we can see, the algorithm word image matching with reweighting based on importance achieves the best FID and R-psr scores on CUB and COCO datasets. Therefore, the algorithm word image matching with reweighting is adopted in our method.

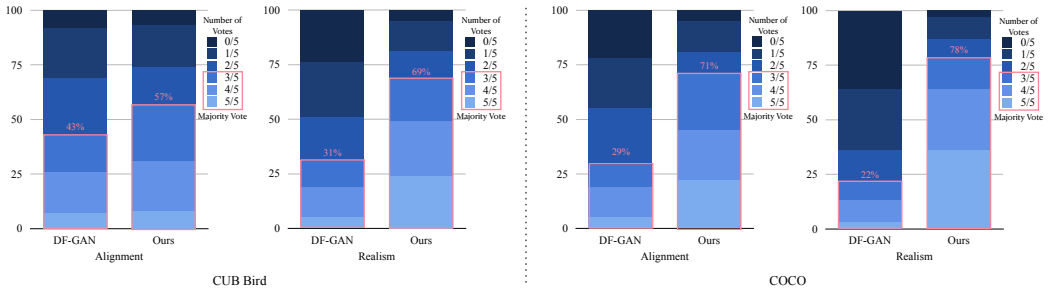

Figure 9: Human evaluation between DF-GAN and ours on CUB and COCO datasets. Of the decisions with 5/5 majority voting, the results produced by our method are most preferred by workers on both alignment and realism.

D.3 Details of Human Evaluation

Because the automatic metric cannot comprehensively evaluate the improvement of our proposed method, we conducted a side-by-side human evaluation study to analyze the improvement. The study compares synthetic images from our method and current state-of-the-art text-to-image generation method DF-GAN (Tao et al., 2020) on both CUB and COCO, according to (1) alignment, and (2) realism. We presented synthetic images from different methods along with the given text descriptions. We randomly switch our method and the baseline and also anonymized them. Then, we asked workers to choose the best images based on above two criteria. In this study, we randomly choose 100 text descriptions sampled from the test dataset, and then assign corresponding synthetic images generated by different methods to 5 workers to reduce variance.

D.4 Qualitative Results

In Fig. 10, we show more qualitative results generated by our method on the CUB bird dataset, along with the corresponding retrieved images that provide image features. As we can see, our method is able to produce high-quality results on CUB, semantically matching the given text descriptions. Also, the synthetic results look obviously different from the retrieved images, but our method can selectively choose information from the retrieved image to generate better synthetic results.

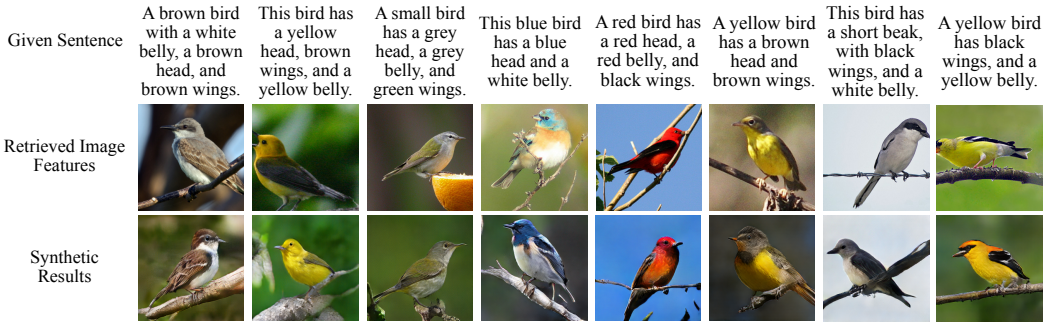

Figure 10: Qualitative results on CUB: top row is the given unseen sentences; middle row: the image features extracted from the memory bank M (we use corresponding images to represent the image features for a better visualization); bottom row: the synthetic results.

D.5 Diversity

D.5.1 SSIM

We also compare the Structural Similarity Index (SSIM) score (Hore & Ziou, 2010) between the generated images and corresponding ground-truth images to evaluate the diversity of our method. SSIM is originally used to measure the recovery result from distorted images. In our case, higher SSIM means synthetic and real images are more similar, which indicates that there may exist a

Table 6: Quantitative comparison: Structural Similarity Index (SSIM) of StackGAN++ (Zhang et al., 2018), AttnGAN (Xu et al., 2018), ControlGAN (Li et al., 2019a), DM-GAN (Zhu et al., 2019), and our method on the CUB and COCO datasets. For SSIM, higher means synthetic and ground-truth images are more similar, which indicates that there may exist a copy-and-paste problem and the network has a worse diversity.

Dataset	StackGAN++	AttnGAN	ControlGAN	DM-GAN	Ours
CUB	0.2727	0.2656	0.2505	0.2196	0.2371
COCO	0.2065	0.1810	0.1599	0.1690	0.1791

copy-and-paste problem and the network has a worse diversity. Based on this, for SSIM, lower is better, which means a better diversity.

To calculate the SSIM, for other baseline methods, we evaluate them on the test dataset by calculating the SSIM between each synthetic and ground-truth image pairs, and then get the average of all scores; for our method, we calculate the SSIM between the synthetic image and the image that provide image features. As shown in Table 6, our method achieves competitive SSIM scores on both CUB and COCO, compared with other baselines. This indicates that (1) even if our method has image features as image priors, it can still produce diverse synthetic results that are different from the corresponding real images, (2) there is no significant copy-and-paste problem in our method, and (3) our method can effectively disentangle objects and attributes in the given image features, which then can work as candidate information for the main generation pipeline to choose.

D.5.2 Semantic Information Exploration

Here, we further verify whether our method suffers from an copy-and-paste problem by exploring whether our method can make use of semantic information contained in the retrieved image features. To verify this, instead of extracting image features from RGB images, we use segmentation masks to provide semantic image features, shown in Fig. 11. As we can see, although there is no any content information provided in the given segmentation masks, our method is still able to generate realistic images, which indicate that our method can make use of semantic information contained in the image features, instead of simply copying and pasting the retrieved image features to produce output images. Furthermore, discussed in the following Sec. D.7, given a partially matched text and image features, our method is able to pick the semantic information (e.g., structure of train, cat, and bus) and filter detailed content color information (e.g., yellow and green, brown, and yellow) to generate text-required output images, as shown in Fig. 12.

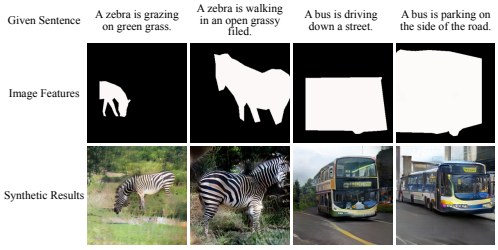

Figure 11: Semantic information exploration. Top row: given sentences; middle row: image features, where we use corresponding segmentation masks to represent the image features for a better visualization; bottom row: synthetic images.

D.6 Effectiveness of Image Features

Actually, when there are no image features that are fed into our method, our method becomes a traditional text-to-image generation model, where the inputs for the model are only the natural language descriptions and random noise. As shown in Table 7, "Ours w/o Feature" still has a competitive performance, compared with other baselines, which means that our method can still generate images with good quality and diversity. We think this is mainly because of the powerful discriminator with content information, which is able to provide fine-grained training feedback to the generator, in terms of realistic appearance and geometric structure. Note that the way to block image features to build the model "Ours w/o Feature" is to remove image features and ACM components in the network, and only keep the new discriminator with content information.

D.7 Image Generation with Partial Text-Image Matching

Interestingly, when the retrieved image features have a good quality (e.g., desired objects in image features can provide enough information), but are not perfectly aligned with the given text descriptions,

Table 7: Quantitative comparison: Fréchet inception distance (FID) and R-precision (R-psr) of StackGAN++ (Zhang et al., 2018), AttnGAN (Xu et al., 2018), ControlGAN (Li et al., 2019a), DM-GAN (Zhu et al., 2019), OP-GAN (Hinz et al., 2019), and our method on the COCO dataset. "Ours w/o Feature" denotes that our model does not have any image features and just has a similar generation pipeline as other traditional text-to-image generation methods. For FID, lower is better, while for R-psr, higher is better.

Matrix	StackGAN++	AttnGAN	ControlGAN	DM-GAN	OP-GAN	Ours w/o Feature
FID	81.59	32.32	33.58	32.64	24.70	22.20
R-prs (%)	71.88	85.47	82.43	88.56	89.01	84.63

Figure 12: Our method can produce realistic images even if image features partially match the given text description. To observe this situation, we manually feed partially matched pairs into the network.

which means that the given text description and corresponding retrieved image features only partially match on the semantic meaning, our method is still able to produce realistic images, shown in Fig. 12. As we can see, our method is able to generate the desired objects with required attributes, even if image features only partially match the given text description. For example, in the provided "train" image features, there is a yellow and green train, but the given description requires a red train. However, our method is still able to generate a realistic train with a red color. Besides, our method can even produce a novel composition, e.g., the sign is flying in the sky. We think that this is mainly because the generator can selectively make use of the information provided by the image features, instead of directly copying and pasting information from it. Also, features and attributes are disentangled in the provided image features, which enable this independent selection without additional generation.

D.8 Regional Selection Effect

Figure 13: Effectiveness of regional selection effect. "Original Text" denotes the corresponding description in the dataset matching the given image prior, "Given Text" denotes the description fed into the network along with the image features.

In Fig. 12, we can observe the regional selection effect involved in the generation process. For the train example, our full model is able to selectively keep the relevant information (e.g., train) and filter the irrelevant contents (e.g., yellow and green color) to avoid a wrong object generation (e.g., red color). This effect can be magnified when the given image has multiple objects, and the given text only partially describes it, shown in Fig. 13. There are multiple objects (e.g., vase, flowers, chairs, and window for the top example; three zebras, enclosure, and grass for the bottom one) in the given image features. However, our method only selectively makes use of some information (e.g., shape and texture of flowers and zebra) and generates text-required objects without keeping irrelevant contents in the image features (e.g., chair, window, and multiple zebras).

E Limitations and Future Work

Here, we discuss some limitations of the proposed method and also the future work. We have observed that our method may fail to produce realistic images when the retrieved image features can only provide limited information, e.g., the target object is too small in the corresponding real image, or there are no desired objects in the retrieved image features. As shown in Fig. 14 left, the stop sign, zebra, bus, and train in the corresponding image are too small, which means that the extracted image

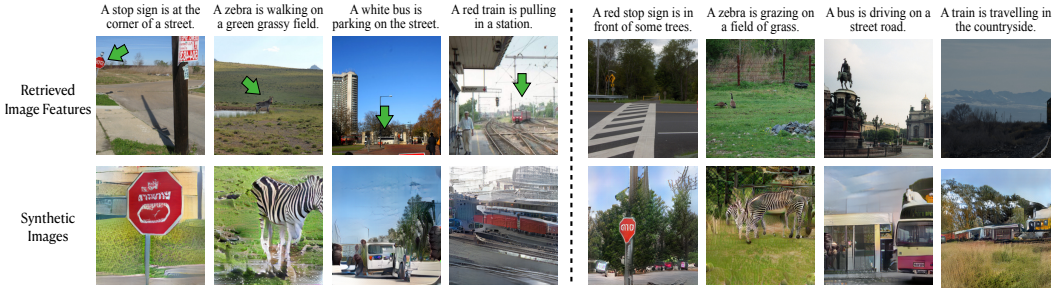

Figure 14: Failure cases, we use the corresponding real image to represent image features for a better visualization. Left: the retrieved image features can only provide limited information. Right: no desired objects exist in the retrieved image features.

features can only provide very limited information about the desired object zebra, stop sign, bus, and train to the generation pipeline. Furthermore, when the retrieved image features have no desired objects, shown in Fig. 14 right, our proposed method may fail to generate high-quality images as well. No desired objects presented in the retrieved image features are mainly caused by the image preprocessing (e.g., crop) and also the limitation of matching algorithms. In such cases, our method is more similar to a pure text-to-image generation method, like other baselines, because the provided image features cannot provide any useful information. To solve these problems, we suggest to build a better memory bank with higher-quality image features, and also improve the matching algorithms to find the most compatible image features for a given text description.

Besides, our method is a semi-parametric approach, which needs to retrieve image features from the memory bank. So, it might slow down the inference time, compared with other purely parametric methods. To solve this problem, we suggest to (1) run matching algorithms parallel to speed up the whole inference time, and (2) encourage users to provide the category of the main object in their text descriptions, and then we can use this category as a key to narrow down the retrieval regions.

F Additional Qualitative Comparison

Here, we show an additional qualitative comparison between the different text-to-image generation approaches StackGAN++ (Zhang et al., 2018), AttnGAN (Xu et al., 2018), and DF-GAN (Tao et al., 2020) with our method on the COCO dataset (Lin et al., 2014).

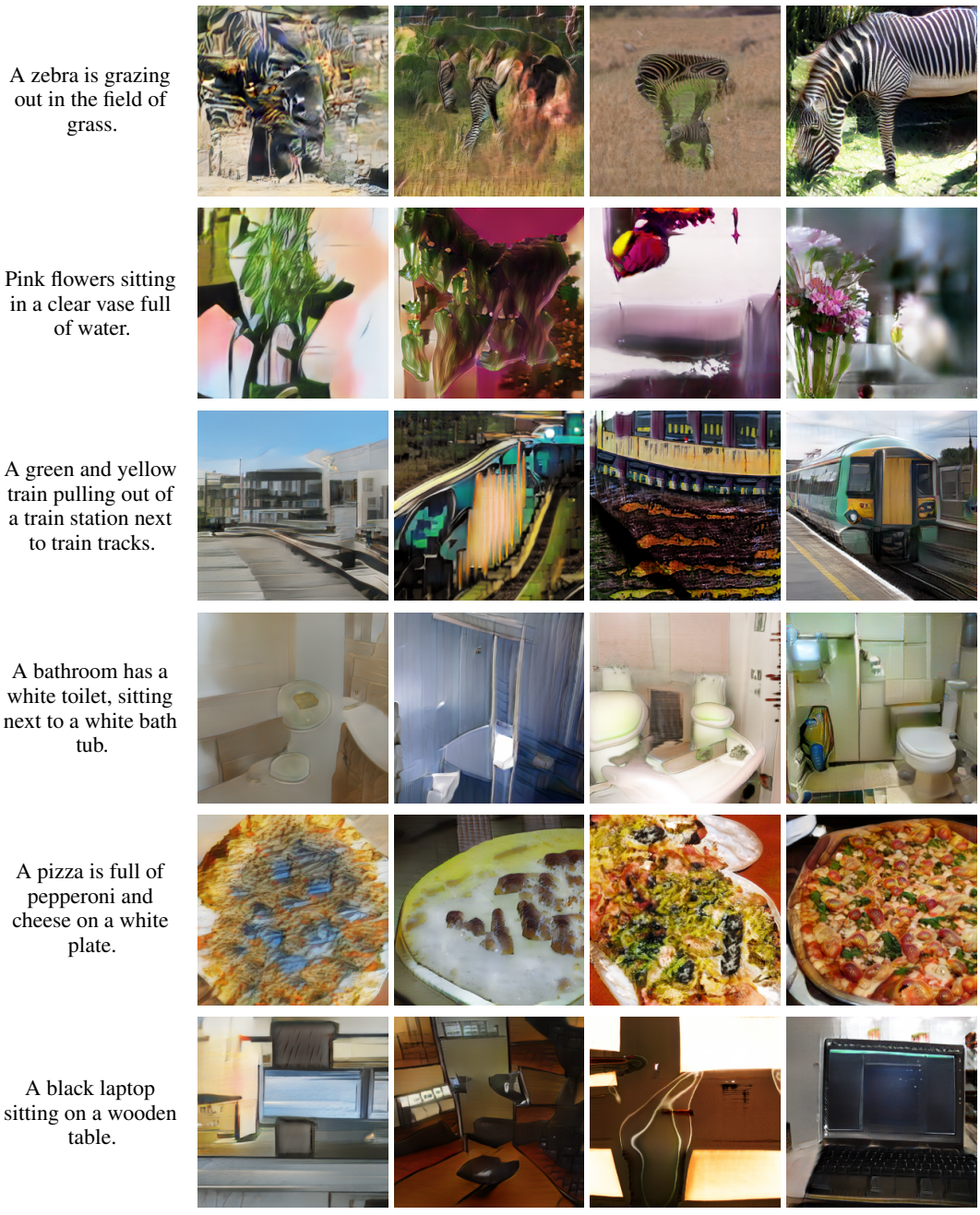

Figure 15: Additional comparison results between StackGAN++, AttnGAN, DF-GAN, and Ours on the COCO dataset.

Figure 16: Additional comparison results between StackGAN++, AttnGAN, DF-GAN, and Ours on the COCO dataset.

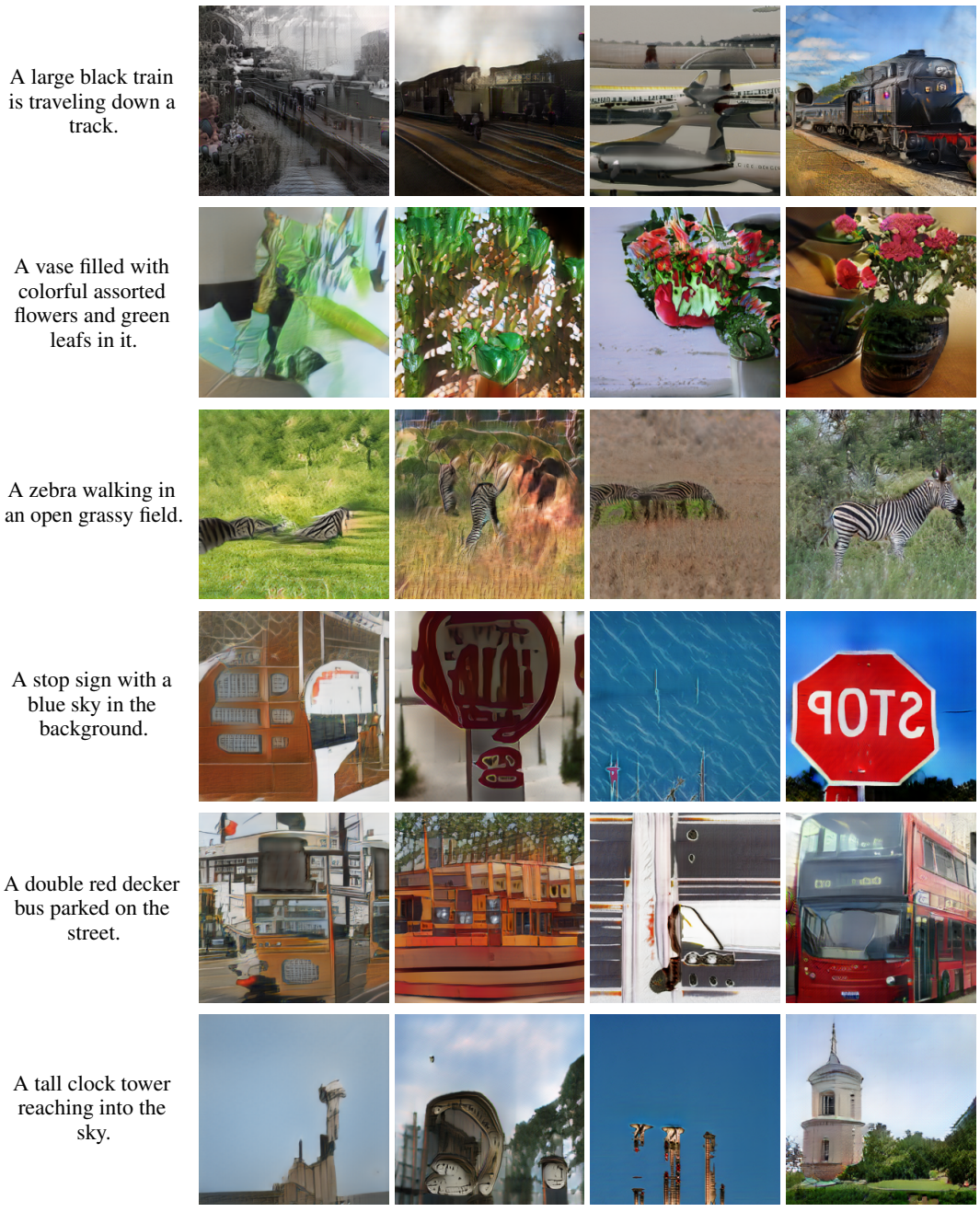

Figure 17: Additional comparison results between StackGAN++, AttnGAN, DF-GAN, and Ours on the COCO dataset.

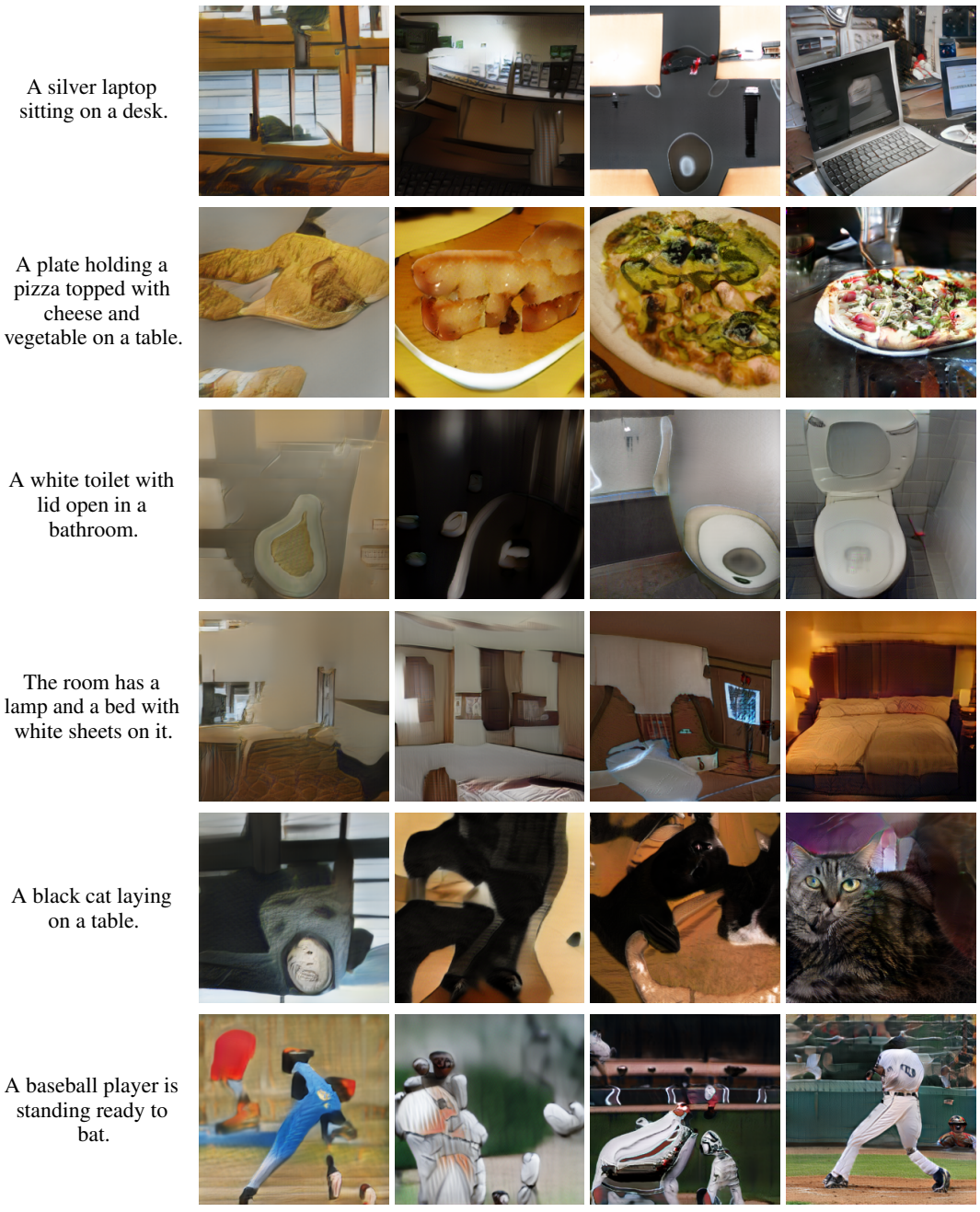

Figure 18: Additional comparison results between StackGAN++, AttnGAN, DF-GAN, and Ours on the COCO dataset.