Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network

Anonymous CVPR submission

Paper ID 2823

Abstract

This paper presents a novel method to deal with the challenging task of generating photographic images conditioned on semantic image descriptions. Our method introduces accompanying hierarchical-nested adversarial objectives inside the network hierarchies, which regularize mid-level representations and assist generator training to capture the complex image statistics. We present an extensile single-stream generator architecture to better adapt the jointed discriminators and push generated images up to high resolutions. We adopt a multi-purpose adversarial loss to encourage more effective image and text information usage in order to improve the semantic consistency and image fidelity simultaneously. Furthermore, we introduce a new visual-semantic similarity measure to evaluate the semantic consistency of generated images, which also alleviates the cost of human evaluation. With extensive experimental validation on three public datasets, our method significantly improves previous state of the arts on all datasets over different evaluation metrics.

1. Introduction

Photographic text-to-image synthesis is a significant problem in generative model research [34], which aims to learn a mapping from a semantic text space to a complex RGB image space. This task requires the generated images to be not only realistic but also *semantically consistent*, i.e., the generated images preserve the object color and sketch described in text and fine-grained descriptive details in image pixels as well.

Recently, generative adversarial networks (GANs) have become the main solution to this task. Reed *et al.* [34] address this task through a GAN based framework. But this method only scale up to 64^2 resolution and can barely generate vivid object details. Based on this method, StackGAN [43] is a successful approach, which stacks another low-to-high resolution GAN to generate high-resolution 256^2 images in a two-stage training strategy. Later on, [8] proposes to bypass the difficult of learning a mapping from text to RGB images and treat it as a pixel-to-pixel translation problem [15]. It works by re-rendering an arbitrary-style 128^2 training image conditioned on a targeting description. However, its high-resolution synthesis capability is still unclear. At present, training a generative model to map from a low-dimensional text space to a high-resolution image space in a fully end-to-end manner still remains unsolved.

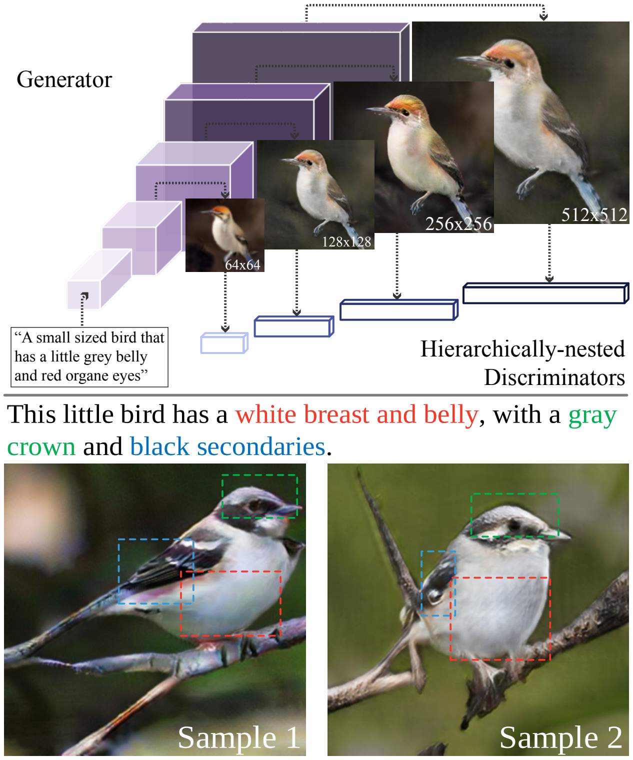

Generator
"A small sized bird that has a little grey belly and red organe eyes"
64x64
128x128
256x256
512x512
Hierarchically-nested Discriminators
This little bird has a white breast and belly, with a gray crown and black secondaries.
Sample 1
Sample 2

Figure 1: Top: Overview of our hierarchically-nested adversarial network. Bottom: Two generated results where fine-grained details are highlighted.

This paper pays attention to two major difficulties for text-to-image synthesis using GANs. The first is balancing the convergence between generators and discriminators

[11, 37], which is actually a long-term problem in GANs. The second is stably modeling the huge pixel space in high-resolution images and guaranteeing semantic consistency [43]. An effective strategy to regularize generators is critical to stabilize the training and help capture the complex image statistics [13].

In this paper, we propose a novel end-to-end method that can directly model high-resolution image statistics and generate semantically consistent and photographic images (see Figure 1). The contributions are described as follows.

Our generator resembles a simple vanilla GAN, without requiring any multiple internal text conditioning like [43] or outside image annotation supervision like [6]. To tackle the problem of the big leap from the text space to the high-resolution image space, our insight is to leverage and regularize hierarchical representations with additional 'deep' adversarial constraints. We introduce accompanying hierarchically-nested discriminators at multi-scale intermediate layers to play adversarial games and jointly encourage the generator approaching the real training data distribution. We also propose a new convolutional neural network (CNN) design for the generator to support accompanied discriminators more effectively in an end-to-end training manner. To guarantee the image diversity and semantic consistency, we enforce discriminators at scale-different generator side outputs to simultaneously differentiate real-and-fake image-text pairs as well as real-and-fake local image patches.

We validate our proposed method on three datasets, CUB birds [39], Oxford-102 flowers [30], and large-scale MSCOCO [23]. In complement of existing evaluation metric (e.g. Inception score [37]) for generative models, we also introduce a new visual-semantic similarity metric to evaluate the alignment between generated images and conditioned text. Extensive experimental results and analysis demonstrate the effectiveness of our method and significantly improved performance compared against previous state of the arts on three metrics.

All source code will be released.

2. Related Work

We discussed related work and further clarify the novelty of our method by comparing with them.

Deep generative models have attracted wide interests recently, including GANs [11], Variational Auto-encoders (VAE) [18], etc [32]. There are substantial existing methods investigating better usage of GANs for different applications, such as image synthesis [33, 38], (unpaired) pixel-to-pixel translation [15, 45], medical applications [5], etc [21, 13].

Text-to-image synthesis is a significant problem in GANs. Reed *et al*. [34] is the first to introduce a method that can generate 64^2 resolution images, which is similar with DCGAN [33]. This method presents a new strategy for image-text matching aware adversarial training. Reed *et al*. [35] propose a generative adversarial what-where network (GAWWN) to enable location and content instructions in text-to-image synthesis. StackGAN *et al*. [43] propose a two-stage stacking GAN training approach that is able to generate 256^2 compelling images. Recently, Dong *et al*. [8] propose to learn a joint embedding of images and text so as to re-render a prototype image conditioned on a targeting description. Cha *et al*. [28] explore the usage of the perceptional loss [16] with a CNN pretrained on ImageNet and Dash *et al*. [6] make use of auxiliary classifiers (similar with [31]) to assist GAN training for text-to-image synthesis.

Learning a continuous mapping from a low-dimensional embedding manifold to a complex real data distribution is a long-standing problem. Although GANs have made significant progress, there are still many unsolved difficulties, e.g. training instability and high-resolution generation. Wide methods have been proposed to address the training instability, such as various training techniques [36, 1, 2, 38, 31], regularization using extra knowledge (e.g. image labels, ImageNet CNNs, etc) [9, 21, 6, 6], or different generator and discriminator combinations [27, 10, 41, 13]. *While our method shows a new way to unite generators and discriminators and does not requires any extra knowledge apart from training paired text and images.* Additionally, it is easy to see the training difficulty increases significantly as the targeting image resolution increases.

To synthesize high-resolution images, cascade networks are effective to decompose originally difficult tasks to multiple subtasks (Figure 2 A). Denton *et al*. [7] train a cascade of GANs in a Laplacian pyramid framework (LAPGAN) and use each to synthesize and refine image details and push up the output resolution stage-by-stage. StackGAN also shares some similar idea with LAPGAN. Inspired by this strategy, Chen *et al*. [4] present a cascaded refinement network to synthesize high-resolution scene from semantic maps. Recently, Karras *et al*. [17] propose a progressive training of GANs. The idea is adding symmetric generator and discriminator layers gradually for high-resolution image generation (Figure 2 C). *Compared with these strategies that train low-to-high resolution GANs stage-by-stage or progressively, our method has the advantage of leveraging mid-level representations to encourage implicit subtask integration, which makes end-to-end high-resolution image synthesis in a single vanilla-like GAN possible.*

Leveraging hierarchical representations of neural networks is an effective way to enhance implicit multi-scaling and ensembling for tasks such as image recognition [22] and pixel or object classification [40, 3, 25]. Particularly, using deep supervision [22] in hierarchical convolutional layers provides short error paths and increases the discriminativeness of feature representations. *Our hierarchically-nested adversarial objective is inspired by these families of deeply-*

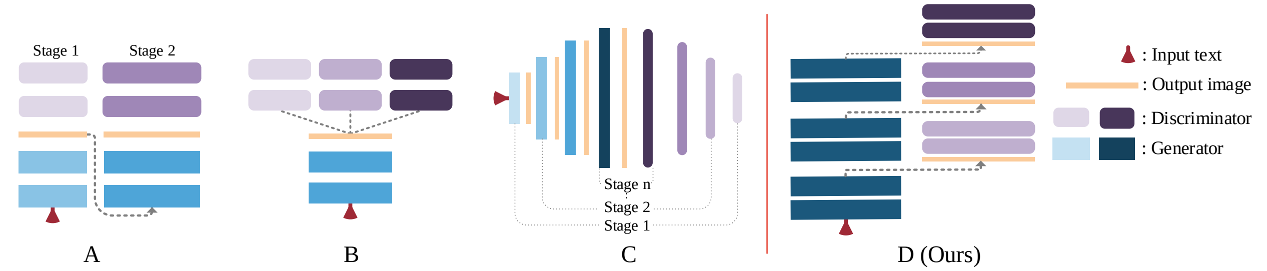


Figure 2: Overviews of some typical GAN frameworks. **A** uses multi-stage GANs [43, 7]. **B** uses multiple discriminators with one generator [10, 29]. **C** progressively trains symmetric discriminators and generators [17, 13]. **A** and **C** can be viewed as decomposing high-resolution tasks to multi-stage low-to-high resolution tasks. **D** is our proposed framework that uses a single-stream generator with hierarchically-nested discriminators trained end-to-end.

supervised CNNs.

3. Method

3.1. Adversarial objective basics

In brief, a GAN [11] consists of a generator G network and a discriminator D network, which are playing an adversarial game to compete with each other. The discriminator is optimized to distinguish synthesized images from real images, meanwhile the generator is optimized to generate fake images to fool the discriminator. Concretely, the overall min-max optimization objective is defined as:

$$G^*, D^* = \arg \min_{G} \max_{D} \mathcal{V}(D, G, Y, \boldsymbol{z}), \tag{1}$$

where Y and $\boldsymbol{z} \sim \mathcal{N}(0, 1)$ denote training images and random noises, respectively. $\mathcal{V}$ is the overall GAN objective, which usually takes the (cross-entropy loss) form $\mathbb{E}_{Y \sim p_{data}}\big[\log D(Y)\big] + \mathbb{E}_{\boldsymbol{z} \sim p_z}\big[\log(1 - D(G(\boldsymbol{z})))\big]$ or other variations [26, 2].

3.2. Hierarchical-nested adversarial objectives

Numerous GAN methods have demonstrated ways to unite generators and discriminators for image synthesis. Figure 2 and Section 2 discuss some typical frameworks. Our method actually explores a new dimension of playing this adversarial game along the depth of a generator (Figure 2 D), which integrates additional hierarchical-nested discriminators at intermediate layers of the generator. The proposed objectives act as regularizers to the hidden space of $\mathcal{G}$, which also offer a short path for better error signal flow and helps reduce the training instability.

The proposed $\mathcal{G}$ is a CNN (defined in Section 3.4) with multiple resolution-growing side outputs:

$$X_1, ..., X_s = \mathcal{G}(\boldsymbol{t}, \boldsymbol{z}), \tag{2}$$

where $\boldsymbol{t} \sim p_{data}$ denotes a sentence embedding vector in the real training data distribution (generated by a pre-trained char RNN encoder). $\{X_1, ..., X_{s-1}\}$ are resolution-growing images and X_s is the final output with the highest resolution.

For each side output X_i from the generator, a distinct discriminator D_i is associated to compete with it. Therefore, our full min-max objective is defined as

$$\mathcal{G}^*, \mathcal{D}^* = \arg \min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{V}(\mathcal{G}, \mathcal{D}, \mathcal{Y}, \boldsymbol{t}, \boldsymbol{z}), \tag{3}$$

where $\mathcal{D} = \{D_1, ..., D_s\}$ and $\mathcal{Y} = \{Y_1, ..., Y_s\}$ is a set of training images at multi-scales, $\{1, ..., s\}$. Compared with Eq. (1), one generator competes with multiple discriminators $\{D_i\}$ at different hierarchies (Figure 2 D), which learn discriminative features in different contextual scales, respectively.

In principle, the lower-resolution side output is used to learn semantic consistent image structures (e.g. object sketches, colors, and background), and the subsequent higher-resolution side outputs are used to render fine-grained details. Since our method is trained in an end-to-end manner, the lower-resolution side outputs can fully enjoy the top-down knowledge from higher resolutions, such that we can observe consistent image structures and styles appear in both low and high resolutions. The experiment demonstrates this advantage compared with StackGAN.

3.3. Multi-purpose adversarial losses

Our generator produces resolution-growing side outputs, which compose an image pyramid. We leverage this hierarchy property and allow adversarial losses to capture different statistics, with a goal to guarantee both semantic consistency and image fidelity.

In order to guarantee semantic consistency, we adopt the matching-aware pair loss proposed by [34]. The discriminator is designed to take image-text pairs as inputs and be trained to differentiate two types of errors: a real image with mismatched text and a fake image with conditioned text.

The pair loss is designed to guarantee the global semantic consistency. However, there lacks an explicit loss that forces the discriminator to differentiate real images from fake images. And combining both tasks (generating realistic images and matching image styles with text) into one loss complicates the already challenging learning tasks. More-

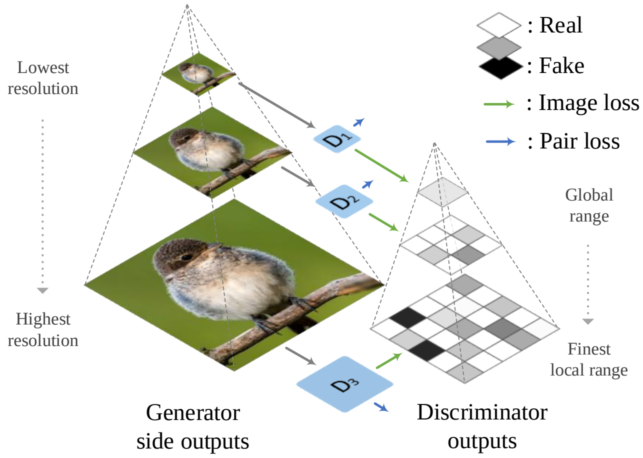


Figure 3: Given a set of images from the side output pyramid of the generator, the corresponding discriminator D_i computes the matching-aware pair loss and local image loss (outputting a $R\times R$ probability map to classify real or fake image patches). R increases as the image size grows.

over, as the image resolution goes higher, it might be challenging for a global pair-loss discriminator to capture the local fine-grained details (results are validated in experiments). In addition, as pointed in [38], a single global discriminator may over-emphasize certain biased local features and lead to artifacts.

To alleviate these issues and guarantee image fidelity, our solution is to add local adversarial image losses. We expect the low-resolution discriminator to focus on global structures, while the high-resolution discriminator to focus on local image details. Each discriminator consists of two branches (see Section 3.4), one computes a single scalar value for the pair loss and another branch computes a $R\times R$ probability map. Each location differentiates whether an overlapping image patch is real or fake. Through the pyramid of the size-growing side outputs, we control R accordingly. Specifically, a larger R value results in a smaller local range and finer focus of discriminators. The local image adversarial loss can be implemented as an fully CNN [38, 45, 15]. Figure 3 illustrates how hierarchically-nested discriminators compute the two losses on the generated images in the pyramid.

Full Objective Overall, our full min-max adversarial objective is defined as

$$\mathcal{V}(\mathcal{G},\mathcal{D},\mathcal{Y},\boldsymbol{t},\boldsymbol{z}) = \sum_{i=1}^{s}\Big(L_2\big(D_i(Y_i)\big) + L_2\big(D_i(Y_i,\boldsymbol{t}_Y)\big) + \overline{L_2}\big(D_i(X_i)\big) + \overline{L_2}\big(D_i(X_i,\boldsymbol{t}_{X_i})\big) + \overline{L_2}\big(D_i(Y_i,\boldsymbol{t}_{\overline{Y}})\big)\Big), \tag{4}$$

where $L_2(x) = \mathbb{E}\big[(x-\mathbb{I})^2\big]$ is the mean-square loss (instead of the conventional cross-entropy loss) and $\overline{L_2}(x) = \mathbb{E}\big[x^2\big]$. This objective is maximized by $\mathcal{D}$. While for $\mathcal{G}$ practically, we minimize $\sum_{i=1}^{s}(L_2(D_i(\mathcal{G}(\boldsymbol{t},\boldsymbol{z})_i)) + L_2\big(D_i(\mathcal{G}(\boldsymbol{t},\boldsymbol{z})_i,\boldsymbol{t}_{X_i}))\big)$. For the local image loss, the shape of $x,\mathbb{I}\in\mathbb{R}^{R\times R}$ vary accordingly (see Figure 3). $R=1$ refers to the (largest local) global range. $D_i(X_i)$ is the image loss branch and $D_i(X_i,\boldsymbol{t}_{X_i})$ is the pair loss branch (conditioned on $\boldsymbol{t}_{X_i}$). $\{Y_i,\boldsymbol{t}_Y\}$ denotes a matched image-text pair and $\{Y_i,\boldsymbol{t}_{\overline{Y}}\}$ denotes a mismatched image-text pair.

In the spirit of variational auto-encoder [19] and the practice of StackGAN [43] (namely conditioning augmentation (CA)), instead of directly using deterministic text embedding, we sample a stochastic vector from a Ganssian distribution $\mathcal{N}(\mu(\boldsymbol{t}),\Sigma(\boldsymbol{t}))$, where μ and Σ are parameterized functions of $\boldsymbol{t}$. We add the Kullback-Leibler divergence regularization term, $D_{KL}(\mathcal{N}(\mu(\boldsymbol{t}),\Sigma(\boldsymbol{t}))||\mathcal{N}(0,\boldsymbol{I}))$ [19], to $\mathcal{G}$ loss to prevent over-fitting and force smooth sampling over the text embedding distribution.

3.4. Architecture Design

Generator The generator is simply composed by three kinds of modules, termed as K-repeat res-blocks, stretching layers, and linear compression layers. A single res-block in the K-repeat res-block is a modified[1] residual block [12], which contains two convolutional (conv) layers (with batch normalization (BN) [14] and ReLU). The stretching layer serves to change the feature map size and dimension. It simply contains a scale-2 nearest up-sampling layer followed by a conv layer with BN+ReLU. The linear compression layer is a single conv layer followed by a Tanh to directly compress feature maps to the RGB space (as side outputs). We prevent any non-linear function in the compression layer that could impede the gradient signals. Starting from a $1024\times4\times4$ embedding, which is computed by CA and a trained embedding matrix, the generator simply uses M K-repeat res-blocks connected by $M-1$ in-between stretching layers until the feature maps reach to the targeting resolution. For example, for 256×256 resolution and K=1, there are M=6 1-repeat res-blocks and 5 stretching layers. At pre-defined side-output positions at scales $\{1,...,s\}$, we apply the compression layer to generate side output images, for the input of the discriminator.

Discriminator The discriminator simply contains consecutive stride-2 conv layers with BN+LeakyReLU. There are two branches are added on the upper layer of the discriminator. One branch is a direct fully convolutional layers to produce a $R\times R$ probability map (see Figure 3) and classify each location as real or fake. Another branch first concatenates a $512\times4\times4$ feature map and a $128\times4\times4$ text embedding (replicated by a reduced 128-d text embedding). Then we use an 1×1 conv to fuse text and image informa-

[1] We remove ReLU after the skip-addition of each residual block, with an intention to reduce sparse gradients.

tion and a 4×4 conv layer to classify the image-text pair to real or fake.

The optimization is similar with the standard alternative training strategy in GANs. Please refer to the supplementary material for more training and network details.

4. Experiments

We denote our method as **HDGAN**, referring as High-Definition results and the idea of Hierarchically-nested Discriminators.

Dataset We test on three widely used datasets. The CUB dataset [39] contains 11,788 bird images belonging to 200 categories. The Oxford-102 dataset [30] contains 8,189 flow images in 102 categories. Each image in both datasets is annotated with 10 descriptions provided by [34]. We pre-process and split the images of CUB and Oxford-102 following the same pipeline in [34, 43]. The COCO dataset [23] contains 82,783 training images and 40,504 validation images. Each image has 5 descriptions. We use the pre-trained text RNN encoder provided in [34] to encode each sentence into a 1024-d text embedding vector.

Evaluation metric We use three kinds of quantitative metrics to evaluate our method. 1) Inception score [37] is a measurement of both objectiveness and diversity of generated images. Evaluating this score needs a pre-trained Inception model on ImageNet. For CUB and Oxford-102, we use the fine-tuned Inception models on the training sets of the two datasets, respectively, provided by StackGAN. 2) Multi-scale structural similarity (MS-SSIM) metric [37] is used for further validation. It tests pair-wise similarity of generated images and can identity mode collapses reliably [31]. Lower score indicates higher diversity of generated images (i.e. less model collapses).

3) **Visual-semantic similarity** The aforementioned metrics are widely used for general GANs. The problem of them is that they can not measure the alignment between the generated images and conditioned text, i.e., semantic consistency. [43] conducted human evaluation to measure. But this procedure is expensive and difficult to conduct. As inspired by [20], we introduce a new qualitative measurement, namely visual-semantic similarity (VS similarity), that can alleviate the cost. Denote $\boldsymbol{v}$ as the image feature vector extracted by a pre-trained Inception model f_{cnn}. We define a scoring function $c(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{||\boldsymbol{x}||_2\cdot||\boldsymbol{y}||_2}$. Then, we learn two mapping functions f_v and f_t, which map both image and sentence embeddings into a common space in $\mathbb{R}^{512}$, by minimizing the following bi-directional ranking loss:

$$\sum_{\boldsymbol{v}}\sum_{\boldsymbol{t}_{\bar{v}}} \max(0, \delta - c(f_v(\boldsymbol{v}), f_t(\boldsymbol{t}_v)) + c(f_v(\boldsymbol{v}), f_t(\boldsymbol{t}_{\bar{v}})))+ \sum_{\boldsymbol{t}}\sum_{\boldsymbol{v}_{\bar{t}}} \max(0, \delta - c(f_t(\boldsymbol{t}), f_v(\boldsymbol{v}_v)) + c(f_t(\boldsymbol{t}), f_v(\boldsymbol{v}_{\bar{t}}))), \tag{5}$$

Method	Dataset		
	CUB	Oxford-102	COCO
GAN-INT-CLS	2.88±.04	2.66±.03	7.88±.07
GAWWN	3.60±.07	-	-
StackGAN	3.70±.04	3.20±.01	8.45±.03*
StackGAN++	3.84±.06	-	-
TAC-GAN	-	3.45±.05	-
HDGAN	**4.15±.05**	**3.45±.07**	**11.86±.18**

*Recently, it updated to 10.62±.19 in its source code.

Table 1: The Inception-score comparison on the three datasets. HDGAN outperforms others significantly.

Method	Dataset		
	CUB	Oxford-102	COCO
Ground Truth	.302±.151	.336±.138	.426±.157
StackGAN	.228±.162	.278±.134	–
HDGAN	**.246±.157**	**.296±.131**	**.199±.183**

Table 2: The VS similarity evaluation on the three datasets. The higher score represents higher semantic consistency between the generated images and conditioned text. The groundtruth score is shown in the first row.

where δ is the margin, which is set as 0.2. $\{\boldsymbol{v},\boldsymbol{t}\}$ is a ground truth image-text pair, and $\{\boldsymbol{v},\boldsymbol{t}_{\bar{v}}\}$ and $\{\boldsymbol{v}_{\bar{t}},\boldsymbol{t}\}$ denote mismatched image-text pairs. In the testing stage, given an text embedding $\boldsymbol{t}$, and the generated image $\boldsymbol{x}$, the VS score is calculated as $c(f_{cnn}(\boldsymbol{x}),\boldsymbol{t})$. Higher score indicates better semantic consistency.

4.1. Comparative Results

To validate our proposed HDGAN, we compare our results with GAN-INT-CLS [34], GAWWN [35], TAC-GAN [6], StackGAN [43] and also its improved version StackGAN++ [44], and Progressive GAN [17][2]. We especially compare with StackGAN in details (results are obtained from its provided models).

Table 1 compares the Inception score. We follow StackGAN' experiment setting to sample $\sim$30,000 256^2 images for evaluation. HDGAN achieves significantly improvement compared against other methods. For example, it significantly improves StackGAN by .45 and StackGAN++ by .31 on CUB. HDGAN achieves competitive results with TAC-GAN on Oxford-102. TAC-GAN uses image labels to increase the discriminability of generators, while we do not use any extra knowledge. Figure 4 and Figure 5 compare the qualitative results with StackGAN on CUB and Oxford-102, respectively, by demonstrating more, semantic details, natural color, and complex object structures. Moreover, we qualitatively compare the diversity of samples conditioned on the same text (with random input noises) in Figure 7 left.

[2] StackGAN++ and Prog.GAN are two very recently released preprints we noticed. We acknowledge them as they also target at generating high-resolution images.

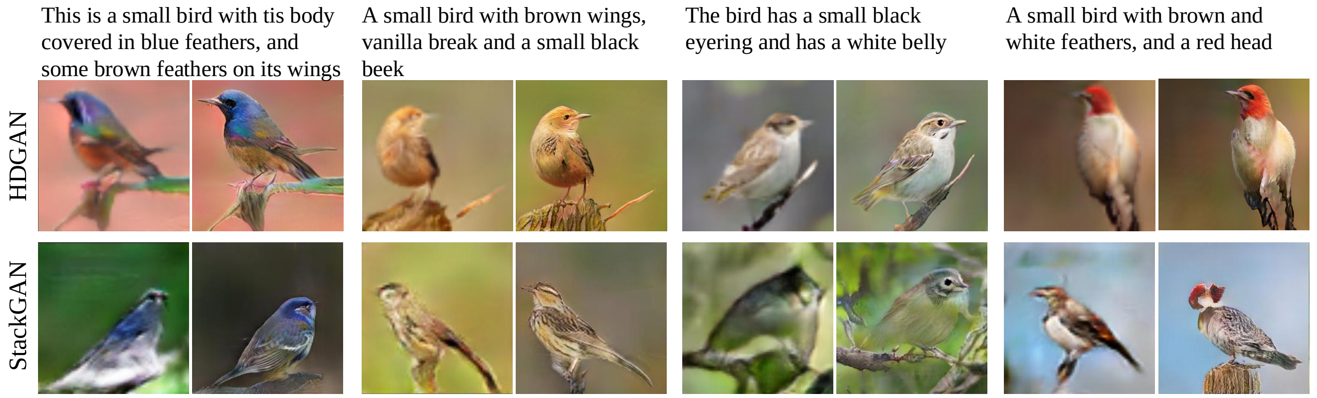

This is a small bird with tis body covered in blue feathers, and some brown feathers on its wings
A small bird with brown wings, vanilla break and a small black beek
The bird has a small black eyering and has a white belly
A small bird with brown and white feathers, and a red head
HDGAN
StackGAN

Figure 4: Generated images on CUB compared with StackGAN. Each sample shows the input text and generated 64^2 (left) and 256^2 (right) images. Our results have significantly higher quality and preserve more semantic details, for example, "the brown and white feathers and read head" in the last column is much better reflected in our images. Moreover, we observed our birds exhibit nicer poses (e.g. the frontal/back views in the second/forth columns). Zoom-in for better observation.

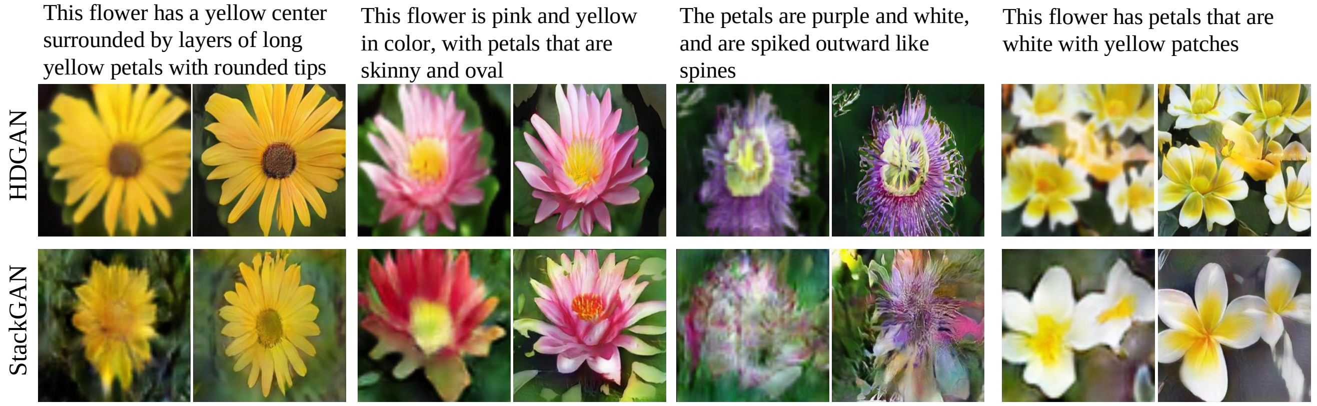

This flower has a yellow center surrounded by layers of long yellow petals with rounded tips
This flower is pink and yellow in color, with petals that are skinny and oval
The petals are purple and white, and are spiked outward like spines
This flower has petals that are white with yellow patches
HDGAN
StackGAN

Figure 5: Generated images on Oxford-102 compared with StackGAN. Our generated images perform more natural satisfiability and higher contrast and can generate complex flower structures (e.g. spiked petals in the third example).

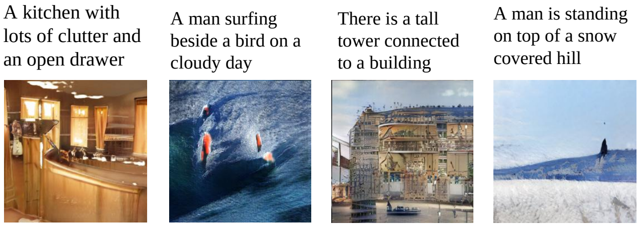
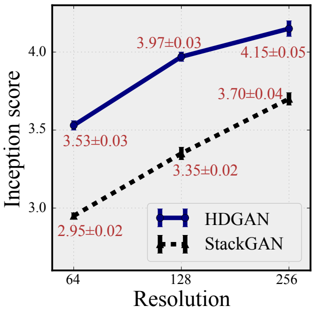

A kitchen with lots of clutter and an open drawer
A man surfing beside a bird on a cloudy day
There is a tall tower connected to a building
A man is standing on top of a snow covered hill

Figure 6: Samples on the COCO validation set, which contain descriptions across different scenes.

HDGAN can generate obviously more successful samples. See figure captions for explanations. The demonstrated results strongly prove the superiority of HDGAN.

Different from CUB and Oxford-102, COCO is much more challenging and contains largely diverse natural scenes. Our method obtains significantly improvement on COCO (Table 1). Figure 6 also shows some generated samples in several different scenes. Please refer to the supplementary material for more results.

Table 2 compares the proposed VS results on the three datasets. The score of the groundtruth image-text pair are also shown for reference. HDGAN achieves consistently better performance on both CUB and Oxford-102. These results demonstrate that HDGAN can better capture the visual semantic information in generated images.



The right figure compares the multi-resolution Inception score on CUB. Our results are from the side outputs of a single model. As can be observed, our 64^2 results outperform the 128^2 results of StackGAN and our 128^2 results also outperform 256^2 results of StackGAN substantially. It strongly demonstrates our HDGAN better preserves semantically consistent information in all resolutions (as stated in Section 3.2). Figure 7 right validate this property qualitatively. However, we observed that, in StackGAN, the low-resolution images and high-resolution images sometimes are visually inconsistent with respect to colors and structures (see examples in Figure 4 and Figure 5).

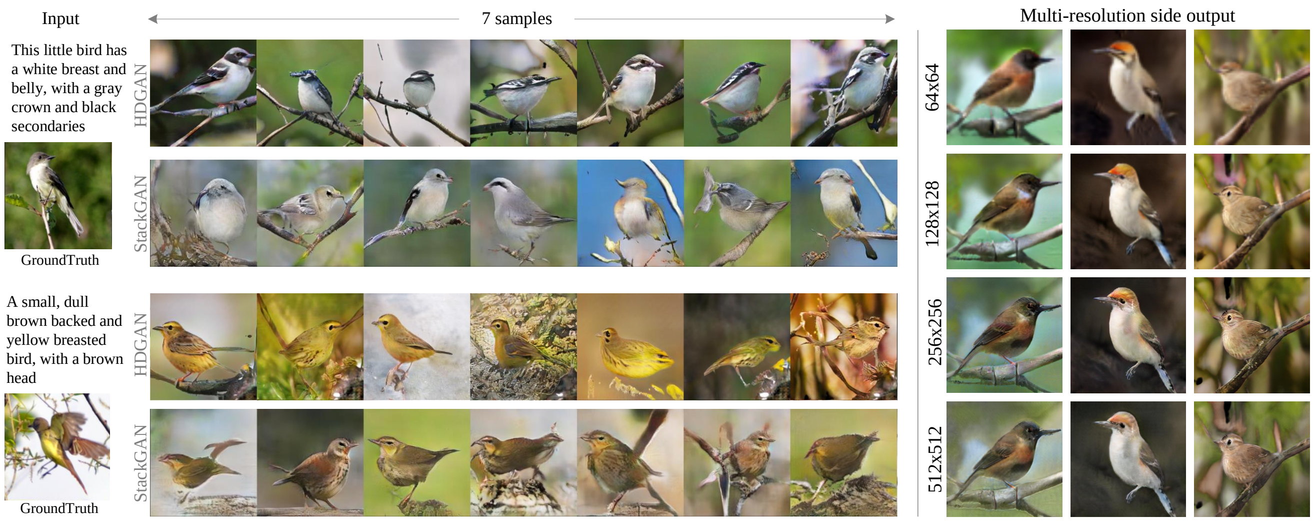
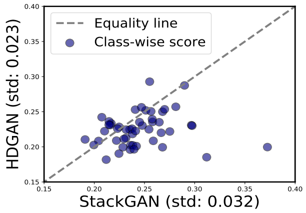

Input
7 samples
Multi-resolution side output
This little bird has a white breast and belly, with a gray crown and black secondaries
GroundTruth
A small, dull brown backed and yellow breasted bird, with a brown head
GroundTruth
HDGAN
StackGAN
HDGAN
StackGAN
64x64
128x128
256x256
512x512

Figure 7: Left: Diverse samples are shown given a single input text. The proposed HDGAN (top) show obviously more fine-grained details. Right: Side outputs of HDGAN with increasing resolutions. Different resolutions are semantically consistent and semantic details appear as the resolution increases.



Method	MS-SSMI
StackGAN	0.234
Prog.GAN	0.225
HDGAN	**0.215**

Table 3: Left: Class-wise MS-SSMI evaluation. Lower score indicates higher intraclass dissimilarity. The points below the equality line represent classes our HDGAN wins. The inter-class std is shown in axis text. Right: Overall (not class-wised) MS-SSMI evaluation.

Table 3 compares the MS-SSMI score with StackGAN and Prog.GAN for bird image generation. StackGAN and our HDGAN use text as input so the generated images are separable in class. We randomly sample $\sim$20,000 image pairs (400 per class) and compare the class-wise score in the left figure. HDGAN outperforms StackGAN in majority of classes and also has a lower standard deviation (.023 vs. .032). Note that Prog.GAN uses noise input rather than text. We can compare with it for the general measure of the image diversity. Following the procedure of Prog.GAN, we randomly sample $\sim 10,000$ image pairs from all generated samples[3] and show the results in Table 3 right. HDGAN outperforms both methods.

4.2. Style Transfer Using Sentence Interpolation

Ideally, a well trained model should learn a smooth linear latent manifold of the images. To demonstrate this ability of HDGAN, we generate images using the linearly interpolated embeddings between a source sentence and a target sentence. As shown in Figure 8, the generated images show smooth style transformation and well reflect the semantic details in sentences. For example, in the second row, two sentences with complicated appearance descriptions are used. We can see that our model successfully captures these subtle features and tune the bird appearance gradually.

4.3. Ablation Study and Discussion

Hierarchically-nested adversarial training Our hierarchically-nested discriminators play a role of regularizing the layer representations (at scale $\{64, 128, 256\}$). In Table 4, we compare the performance of HDGAN by removing part of discriminators on both CUB and COCO datasets. As can be seen, increasing the usage of discriminators at different scales have positive effects. And using discriminators at 64^2 is critical (by comparing 64-256 and 128-256 cases). For now, it is uncertain if adding more discriminators and even on lower resolutions would be helpful. Further validation will be conducted. StackGAN emphasizes the importance of using text embeddings not only at the input but also with intermediate features of the generator, by showing a large drop from 3.7 to 3.45 without doing so. While our method only uses text embeddings at the input. Our results strongly demonstrate the effectiveness of our hierarchically-nested adversarial training to maintain such semantic information and a high Inception score.

[3] We use 256^2 bird images provided by Prog.GAN at https://github.com/tkarras/progressive_growing_of_gans. Note that Prog.GAN is trained on the LSUN [42] bird set, which contains $\sim$2 million bird images.

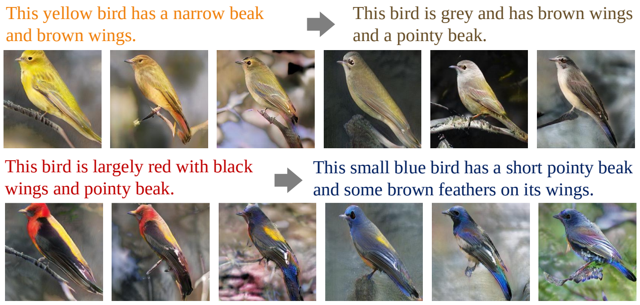

This yellow bird has a narrow beak and brown wings.
This bird is grey and has brown wings and a pointy beak.
This bird is largely red with black wings and pointy beak.
This small blue bird has a short pointy beak and some brown feathers on its wings.

Figure 8: Text embedding interpolation from the source to target sentence results in smooth image style changes to match the targeting sentence.

Components			Inception score	
64	128	256	CUB	COCO
		✓	3.52±.04	-
	✓	✓	3.99±.04	-
✓		✓	4.14±.03	11.29±.18
✓	✓	✓	4.15±.05	11.86±.18

Table 4: Ablation study of hierarchically-nested adversarial discriminators on CUB and COO. ✓ indicates whether a discriminator at a certain scale is used. See text for detailed explanation.

The local image loss We analyze the effectiveness of the proposed local adversarial image loss. Table 4 compares the case without using it (denoted as 'w/o local image loss'). The local image loss helps improve the visual-semantic matching evidenced by a higher VS score. We hypothesis that it is because adding the separate local image loss can offer the pair loss more "focus" on learning the semantic consistency. Furthermore, the local image loss helps generate fine-grained image details. As demonstrated in Figure 9, although both models can successfully capture the semantic details in text, the 'w/ local' model generate complex object structures described in the conditioned text more precisely.

Design principles StackGAN claims the failure of directly training a vanilla 256×256 GAN to generate meaningful images. We test this extreme case using our method by removing all nested discriminators. Our method still generates fairly meaningful results (the first row of Table 4), which demonstrate the effectiveness of our proposed framework (see Section 3.4).

Initially, we tried to share the top layers of the hierarchical-nested discriminators of HDGAN inspired by [24]. The intuition is that all discriminators have a common goal to differentiate real and fake despite difficult scales and such sharing would reduce their inter-variances. However, we did not observe benefits from this mechanism and our independent discriminators can be trained very stably.

	Inc. score	VS
w/o local image loss	3.12±.02	.263±.130
w/ local image loss	3.45±.07	.296±.130

Table 5: Ablation study on the local image loss on Oxford-102. See text for detailed explanation.

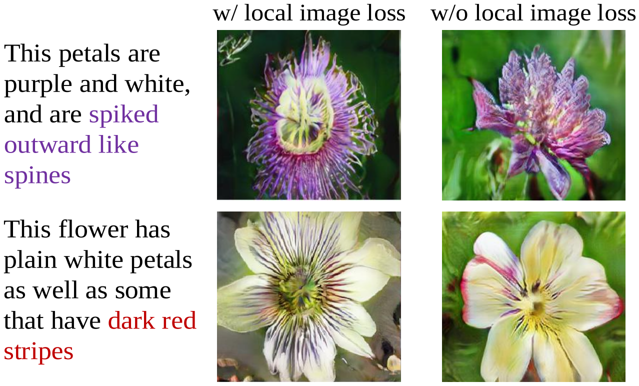

w/ local image loss
w/o local image loss
This petals are purple and white, and are spiked outward like spines
This flower has plain white petals as well as some that have dark red stripes

Figure 9: Qualitative evaluation of the local image loss. The two images w/ the local image loss more precisely exhibit the complex flower petal structures described in the (colored) text.

5. Conclusion

In this paper, we present a novel and effective method to tackle the problem of generating images conditioned on text descriptions. We present a new dimension of playing adversarial games along the depth of the generator using the hierarchical-nested adversarial objectives. A multi-purpose adversarial loss is adopted to help render fine-grained image details. We also introduce a new evaluation metric to evaluate the semantic consistency between generated images and conditioned text. Extensive experiment results demonstrate that our method, namely HDGAN, can generate high-resolution photographic images and performs significantly better than existing state of the arts on three public datasets.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2

[2] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 2, 3

[3] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, pages 354–370. Springer, 2016. 2

[4] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. *ICCV*, 2017. 2

[5] P. Costa, A. Galdran, M. I. Meyer, M. D. Abràmoff, M. Niemeijer, A. M. Mendonça, and A. Campilho. Towards adversarial retinal image synthesis. *arXiv preprint arXiv:1701.08974*, 2017. 2

[6] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Z. Afzal, and M. Liwicki. Tac-gan-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017. 2, 5

[7] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pages 1486–1494, 2015. 2, 3

[8] H. Dong, S. Yu, C. Wu, and Y. Guo. Semantic image synthesis via adversarial learning. *ICCV*, 2017. 1, 2

[9] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016. 2

[10] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016. 2, 3

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2, 3

[12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 4

[13] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. *CVPR*, 2017. 2, 3

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 1, 2, 4

[16] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *arXiv preprint arXiv:1710.10196*, 2016. 2, 3, 5

[18] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[19] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, 2013. 4

[20] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 5

[21] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *CVPR*, 2017. 2

[22] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AIS*, 2015. 2

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 5

[24] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. 8

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[26] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016. 3

[27] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016. 2

[28] H. T. K. Miriam Cha, Youngjune Gwon. Adversarial nets with perceptual losses for text-to-image synthesis. *arXiv preprint arXiv:1708.09321*, 2017. 2

[29] T. D. Nguyen, T. Le, H. Vu, and D. Phung. Dual discriminator generative adversarial nets. In *NIPS*, 2017. 3

[30] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICCVGIP*, 2008. 2, 5

[31] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016. 2, 5

[32] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *ICML*, 2016. 2

[33] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[34] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *ICML*, 2016. 1, 2, 3, 5

[35] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *NIPS*, 2016. 2, 5

[36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016. 2

[37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In *NIPS*. 2016. 2, 5

[38] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *CVPR*, 2017. 2, 4

[39] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. 2010. 2, 5

[40] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 2

[41] J. Yang, A. Kannan, D. Batra, and D. Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation. *arXiv preprint arXiv:1703.01560*, 2017. 2

[42] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 7

[43] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 1, 2, 3, 4, 5

[44] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan++: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *arXiv preprint arXiv:1710.10916*, 2017. 5

[45] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017. 2, 4