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 images conditioning on semantic image descriptions. We propose an effective end-to-end single-stream network that can generate photographic high-resolution images. Our method leverages deep representations of convolutional layers and introduces accompanying hierarchical-nested adversarial games inside the network hierarchy, which regularize intermediate representations and assist training to capture the complex image statistics. We present an extensile network architecture to cooperate with discriminators and push generated images to high resolutions. We adopt a multi-purpose adversarial training strategy at multiple nested side outputs to encourage more effective image-text alignment 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, thus alleviates the needs 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 (e.g. a 11.86 *Inception score on COCO).*

1. Introduction

Photographic text-to-image synthesis is a significant problem in generative model research [34], which aims to learn a mapping from one semantic text embedding space to one complex RGB image space. This task requires the generated images to be not only realistic but also *semantically consistent*. However, this problem still remains unsolved due to its particular challenges.

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] propose

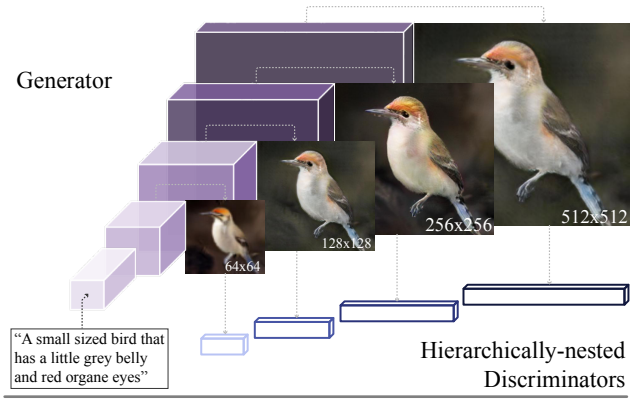


Figure 1: Top: Overview of our hierarchically-nested adversarial network. Bottom: Two fine-grained generated results.

a two-stage training strategy, which stacks another low-to-high resolution GAN to generate high-resolution 256^2 images. Later on, [8] proposes to bypass the difficult of learning mappings from vector embeddings 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 remains unclear.

At present,adversarial training of a generative model mapping a low-dimensional semantic space to high-resolution and diverse images space in a fully end-to-end manner is still an open question.

Text-to-image synthesis using GANs has two main dif-

ficulties. The first is the fundamental problem of balancing the convergence between generators and discriminators [11, 37]. The second is stably modeling the huge pixel space in high-resolution images [43]. An effective strategy to regularize generators is critical to help capture the complex image statistics [13] as well as guarantee semantic consistency. With careful consideration of these problems, 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 image (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 thereby encourage the generator approaching the real training data distribution. We also propose a novel convolutional neural network (CNN) framework for the generator to cooperate with our nested discriminators more effectively. To guarantee the image diversity and semantic consistency, we enforce nested discriminators at different hierarchies to simultaneously differentiate real-and-fake image-text pairs as well as real-and-fake global images or local image patches with certain receptive field. This multi-purpose conditional adversary is mutually beneficial to allow each focus on its respect duty.

To summarize, our contributions are:

We validate our proposed method on three datasets, CUB birds [40], Oxford-102 flowers [30], and large-scale MSCOCO [23]. In complement of existing evaluation metric (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 not only requires diverse and high-quality image generation but also requires precise semantically consistent mapping in the image space. Reed *et al*. [34] is th 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 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 with a CNN pretrained on ImageNet [16] 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 extensions. 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, 42, 13]. *While our method shows a new way to unite generators and discriminators and does not use any extra information apart from training paired text and images.* Additionally, it is easy to see the training difficulty increases significantly as targeting image resolution increases.

To synthesize high-resolution images, cascade networks are useful to decompose original difficult tasks to multiple subtasks (Figure 2 A). Denton *et al*. [7] train a cascade of GANs within a Laplacian pyramid framework (LAPGAN) and use each to generate difference images, conditioned on random noises and the output from the last pyramid level, and thereby push up the output resolution through by-stage refinement. StackGAN also shares similar strategy with LAPGAN. Following 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 for very high-resolution image generation (Figure 2 C). *Compared with these strategies that train low-to-high resolution GANs progressively, our method has advantages 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

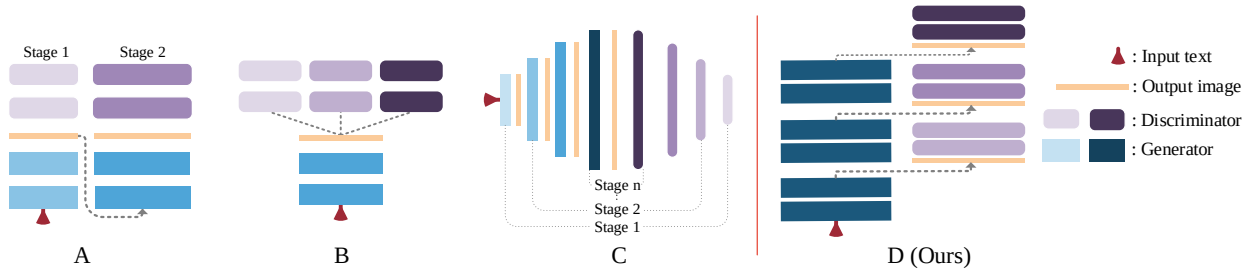


Figure 2: Overviews of some typical GAN frameworks. **A** uses multi-stage GANs [43, 7]. **B** uses multiple discriminators with one generator [10, 29]. **C** progressively trains symmetric discriminators and generators [17, 13]. **A** and **C** can be viewed as decomposing high-resolution tasks to multi-stage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

and ensembling for tasks such as image recognition [22] and pixel or object classification [41, 3, 25]. Particularly, using deep supervision [22] in hierarchical convolutional layers is shown to be ablue to increase the discriminativeness of feature representations. *Our hierarchically-nested adversarial objective is inspired by these families of CNNs with deep supervision.*

3. Method

3.1. Adversarial objective basics

Generative adversary networks (GANs) consists of a generator G and a discriminator D, which are alternatively trained to compete with each other in a two-player minimax game. D is optimized to distinguish synthesized images from real images, meanwhile, G is trained to fool D by generating fake images. Concretely, the optimal G and D can be obtained by playing the following two-player min-max game,

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

where Y and $\boldsymbol{z}$ denotes training images and random noises, respectively. $\mathcal{V}$ is the overall GAN loss, which usually takes the 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].

3.2. Hierarchical-nested adversarial objectives

Numerous 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 CNN (Figure 2 D), which integrates additional discriminators at mid-level features of the generator. The hierarchically-nested objectives act as regularizers to the hidden space of $\mathcal{G}$, which can reduce the training instability and also offer a short path for better error signal flows.

The proposed $\mathcal{G}$ is a CNN (defined in Section 3.4) with multiple 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[34]). $\mathcal{G}$ produces $s-1$ side outputs $\{X_1, ..., X_{s-1}\}$ (size-growing images) and a final output X_s with the highest resolution.

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

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

where $\mathcal{D} = \{D_1, ..., D_s\}$, and $\mathcal{Y} = \{Y_1, ..., Y_s\}$ denotes a set of training images at multi-scales, $\{1, ..., s\}$. Compared with Eq. (1), our generator competes with multiple discriminators $\{D_i\}$ at different hierarchies (Figure 2 D), where each focus on learning diverse discriminative features in different contextual scales. In principle, the lower-resolution side output can learn semantic consistent image structures (e.g., object sketch, color, and background), and the subsequent higher-resolution side outputs are used to render more details to images. Since our method is trained in an end-to-end fashion, the lower-resolution outputs can also fully utilize top-down knowledge from discriminators for higher resolutions. We can observe consistent image structures, color and styles in outputs of both low and high resolutions without using addition supervising term[44]. As an evidence, we can obtain significantly more consistent results exhibited in both low- and high-resolution outputs compared with StackGAN (see experiments).

3.3. Functional adversarial losses

Our generator produces size-growing side outputs which compose an image pyramid. We leverage this hierarchy property and allow adversarial losses to be multi-purpose to capture different (conditioned and unconditioned) statistics, with a goal to guarantee both semantic consistency and image fidelity.

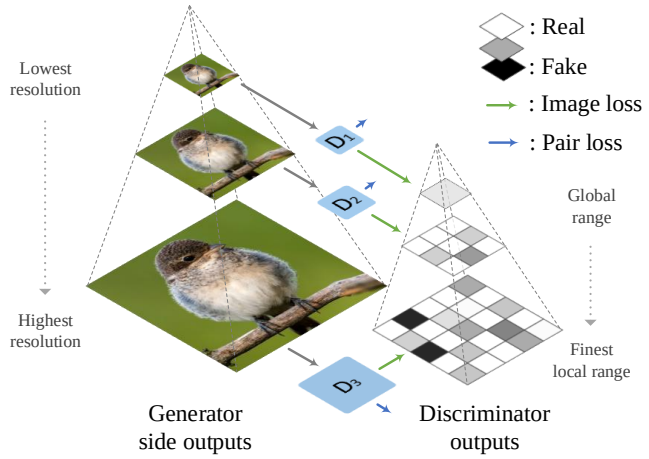


Figure 3: Given a set of images from the side output pyramid of the generator, the corresponding discriminator D_i computes the matching-aware pair loss and adaptive local image loss (outputting a $R\times R$ probability map to classify real or fake patches). The focal range decreases as the image size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

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 trained to identify two types of errors: real image with mismatched text and fake image with conditioned text.

The pair loss is designed to guarantee the global semantic consistency. However, there is no explicit loss that forces the discriminator to differentiate pure real images from fake images. Combining both tasks (generating realistic images and matching image with text) into one output further complicates the learning tasks, which is already challenging. Second, as the image resolution goes higher, it might be challenging for a global pair-loss discriminator to capture local fine-grained details. In addition, as pointed in [38], a single global discriminator may over-emphasize certain unexpected image features and lead to artifacts.

To guarantee the image fidelity, our solution is to add local adversarial image losses on image patches. We expect the low-resolution discriminator to focus on global structures, while high-resolution outputs focus more on local image details. We validate this analysis in the experiments. Through the pyramid of the size-growing side outputs, we configure the focal range (the size of each image patch need to classify) to be adaptive. The local image adversarial loss is implemented as a fully CNN [38, 45], with an extra branch of the discriminator for the pair loss (see Section 3.4). Figure 3 illustrates how hierarchically-nested discriminators compute the two losses on generated images.

Full Objective Overall, our hierarchically-nested discriminators $\{D_i\}$ minimize the following loss[1]:

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

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 $\overline{L_2}(x) = \mathbb{E}\big[x^2\big]$. For local image loss with varying focal ranges, the shape of $x, \mathbb{I} \in \mathbb{R}^{R\times R}$ varies accordingly (see Figure 3). $R = 1$ refers to the (largest local) global range. For the matching-aware pair loss, $\{Y_i, \boldsymbol{t}_Y\}$ denotes a matched image-text pair and $\{Y_i, \boldsymbol{t}_{\overline{Y}}\}$ denotes a mismatched image-text pair.

Furthermore, this separation of the image loss from the matching loss in our method opens further possibilities, such as utilizing the unlabeled image to improve both the generator and discriminator.

In the spirit of variational auto-encoder [19] and the practice of StackGAN [43], 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 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[2] residual block [12], which contains two convolutional (conv) layers (with batch normalization (BN) [14] and ReLU). The stretching layer serves to reduce feature map size and dimension. It simply contains an scale-2 nearest up-sampling layer followed by a convolutional layer with BN+ReLU. The linear compression layer is one conv layer followed by a Tanh to directly compress features map to the RGB space (as side outputs) linearly. We prevent any non-linear functions that could impede the gradient signals. Starting from a $1024\times4\times4$ embedding, which are computed by the CA output followed by a trainable embedding matrix, the generator simples use M K-repeat res-blocks connected by $M-1$ in-between stretching layers until the feature maps reach to the targeting resolution. So for 256×256 resolution with $K=1$, there are $M=6$ 1-repeat res-blocks and 5 stretching layers. With a predefined side-output scales $\{1, ..., s\}$, we apply the compression layer at those scales to produce synthetic images (i.e., side outputs).

[1] The objective of the generator is omitted as it can be easily inferred.

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

Discriminator The discriminator simply contains consecutive stride-2 conv layers with BN and LeakyReLU. There are two branches are added on the upper layer for the proposed functional discriminator design (see next section). 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 $128 \times 4 \times 4$ text embedding (replicated by a reduced 128-d text embedding). Then we use an 1×1 conv to fuse text and image information and a 4×4 conv layer to classify the image-text pair is real or fake.

The training is similar with the standard GAN alternative training strategy. Please refer to the supplementary material for training details.

4. Experiments

This section evaluates the proposed method both qualitatively and quantitatively on three public datasets. 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 [40] contains 11,788 bird images belonging to 200 different categories. We pre-process and split the images following the same pipeline in [34, 43] The Oxford-102 dataset [30] contains 8189 flow images in 102 different categories. Each image in both datasets is associated with 10 text descriptions. In the COCO dataset, [23] there are 80k training images and a validation 40k images. Each image has 5 text descriptions. We use the pre-trained text encoder model provided in [34] to encode each text description into a 1024 embedding vector.

Evaluation metric We use three different quantitative metrics to evaluate our method. 1) Inception score [37] is a measurement of both objectiveness and diversity of generated images, it is closely correlated with human judgment on the image quality. For CUB and Oxford-102, we use the fine-tuned inception model provided by StackGAN. For MS COCO dataset, we directly use the model pre-trained on ImageNet. 2) We also adopt the multi-scale structural similarity (MS-SSIM) metric [37] for further validation. It tests pair-wise variation of generated outputs and can find 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. We introduce a new qualitative measurement, namely visual-semantic similarity (VS similarity) inspired by [20]. Denote $\boldsymbol{v}$ as the image feature vector extracted by a pre-trained Inception model f_{cnn}[39]. We define a scoring function $c(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||}$. 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:

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

where δ is the margin, which is set as 0.2. $\{\boldsymbol{v}, \boldsymbol{t}\}$ is a ground truth image text pair, and $\{\boldsymbol{v}, \boldsymbol{t}_{\overline{v}}\}$ and $\{\boldsymbol{v}_{\overline{t}}, \boldsymbol{t}\}$ 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, i.e. stronger alignment between generated images and conditioned text.

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 evaluation on three datasets. The higher score reflects more meaningful synthetic images and higher diversity. The proposed 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 the text information. The groundtruth score is shown in the first row.

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][3]. We especially compare with the current state of the art, StackGAN (results are obtained from its provided models).

Table 1 shows the quantitative inception-score evaluation. We follow StackGAN' experiment setting to sample

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

Figure 4: The 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 have more diverse poses (e.g. the frontal view in the the second and the back view forth columns).

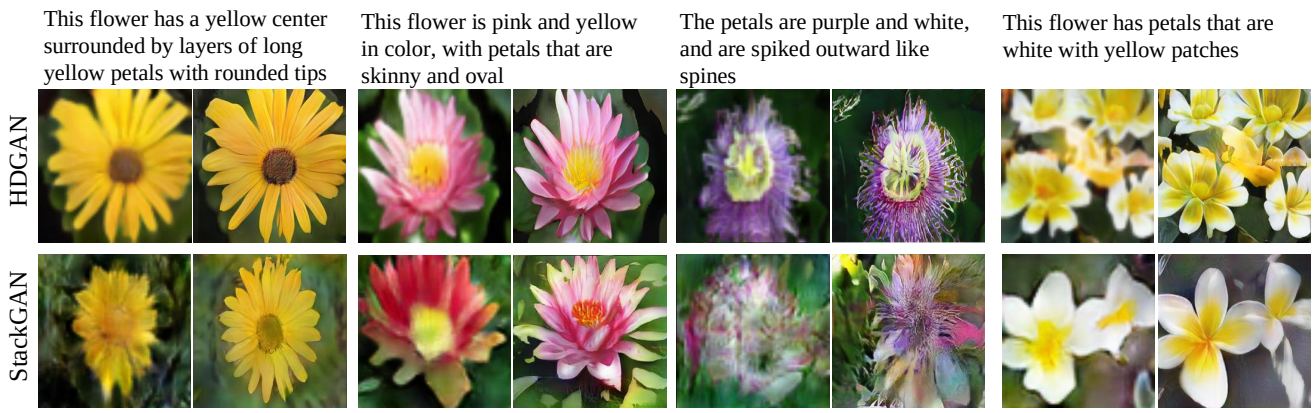
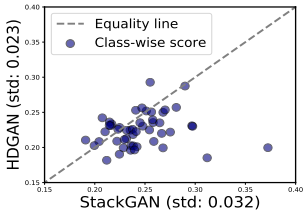


Figure 5: The generated images on Oxford-102 compared with StackGAN. Our generated images perform more natural satisfiability and higher contrast and can generate complex flower structures (e.g. spiked petals) like the third example.

$\sim 30,000$ 256^2 images for evaluation. HDGAN achieves significantly improvement compared against other methods. For example, it 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 show the results compared with StackGAN and strongly demonstrate the superiority of HDGAN. Moreover, we qualitatively evaluate the diversity of samples conditioned on the same text (with different input noises) in Figure 6 left. HDGAN can generate obviously more successful samples than StackGAN.

Table 2 compares the VS results on the three datasets. We also add the evaluation results on the ground truth image and sentence pair for reference. HDGAN achieves consistent better performance on both CUB and Oxford-102 datasets. This results demonstrate that HDGAN can better capture the visual semantic information in generated images and results in higher visual similarity.



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 titles. Right: Overall (no class-wise) MS-SSMI evaluation.

Table 3 compares the MS-SSMI image quality evaluation score with StackGAN and Prog.GAN for bird image generation. StackGAN and our HDGAN use the semantic text as input so the generated images are separable in class. We randomly sample $20,000$ image pairs (400 per class) and show the class-wise score in the left figure. HDGAN outperforms StackGAN in majority of classes and

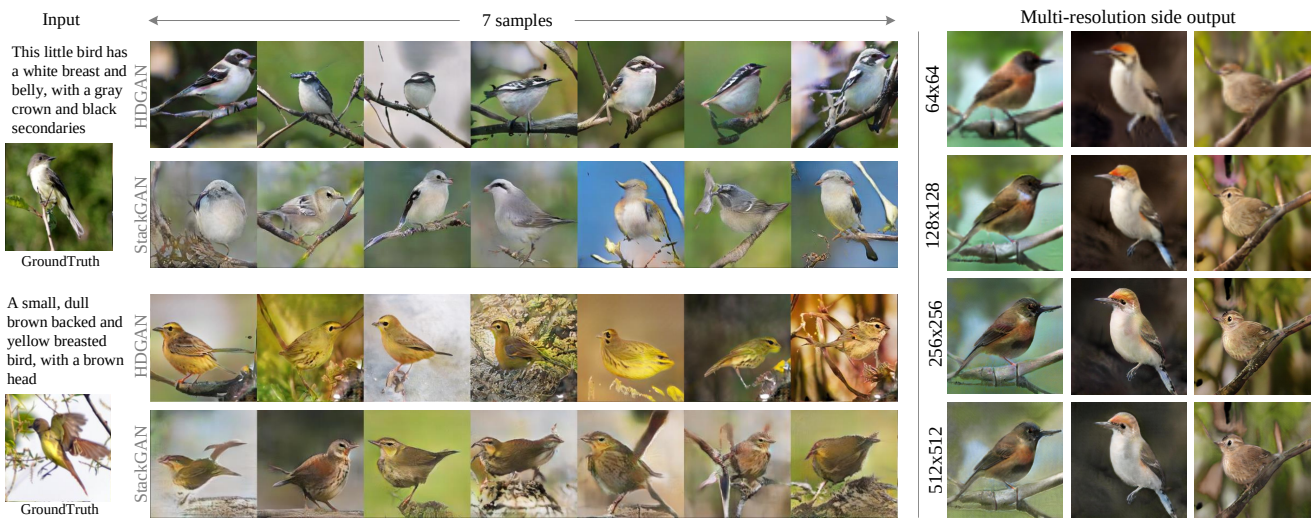
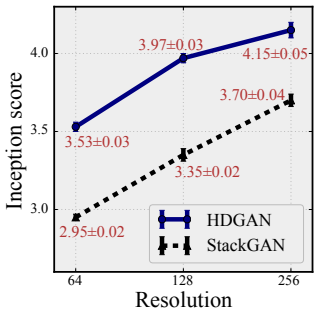


Figure 6: Left: Diversity test given a single input text. The proposed HDGAN (top) show obviously more details and higher diversity. Right: Side outputs of HDGAN with increasing resolutions. Different resolutions are semantically consistent and semantic details emerges as resolution increases.

also has lower inter-class standard deviation (.023 vs. .032). Prog.GAN uses noise input rather than text. We can compare with it for general measure of synthesis diversity. Following the procedure of Prog.GAN, we randomly sample $10,000$ image pairs from all generated samples (We use 256^2 images provided by the Prog.GAN online) and show the results in Table 3 right. Our HDGAN outperforms both methods.

The right figure compares the 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 images outperforms 128^2 images of StackGAN and our 128^2 images outperforms 256^2 images of StackGAN substantially. It strongly demonstrates our HDGAN better preserves semantically consistent information in all resolutions. Figure 6 right qualitatively validate this property. However, in StackGAN, the low resolution and high resolution image sometimes are visually inconsistent (see examples in Figure 4 and Figure 5).



4.2. Style Transfer Using Sentence Interpolation

Ideally, a well trained model should learn a smooth linear latent manifold of the images. To demonstrate the generalization capability of HDGAN, we generate images using the linearly interpolated embeddings between two source sentences. As shown in Figure 7, the generated images show smooth style transformation and well reflect the semantic details in sentences. In the first row, the generated birds gradually turn the color from yellow to brown. In the second row, more complicated sentences with detailed appearance descriptions (e.g., blue peaks and yellow wing) are used, we can see that our model is still be able to successfully capture these subtle features and tune the bird appearance gradually.

4.3. On the effects of Individual Components

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 effectives. And using discriminators on 64^2 is critical (by comparing 64-256 and 128-256 cases). For now, we are unsure if adding more discriminators and even on lower resolution would be helpful. Further investigation will be conducted to validate it. StackGAN emphasizes the importance of combining text embeddings with intermediate features of the 256^2 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 significantly improved results demonstrate the effectiveness of our strategy to maintain such semantic information and high inception score.

On the Local Image Loss We also study the effectiveness of the proposed local adversarial image loss. Table 4 compares the case by removing the local loss (denoted as 'w/o local'). As can be seen, the inception score drop from 3.45 to 3.12. Besides, the local image loss also helps improve the visual-semantic matching evidenced by the higher VS score of w/local. We hypothesis that it is because adding the separate local image loss can offer the pair loss more

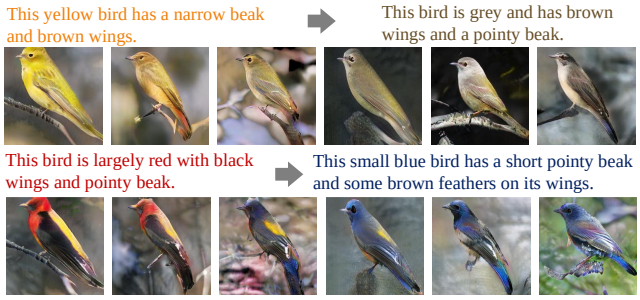


Figure 7: Text embedding interpolation from the source to target sentence results in smooth image style changes to match the targeting sentence.

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.

	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 adversarial loss. See text for detailed explanation.

focuses on learning the semantic consistency. Furthermore, we also quantitatively compare the generated results in Figure 8. Although both models can successfully capture the semantic details in text. The model with the local loss successfully generate more fine-grained structures described in text and thus improving the visual quality.

Design principles StackGAN shows the difficulty of directly training a vanilla 256×256 GAN to generate meaningful images. We test this extreme case using our method by removing all nested intermediate discriminators (the first row of Table 4). Our method still generates fairly good results. Based on our experience, the success is because of our effective generator designs that helps gradient flows.

Initially, we tried to share top layers of the hierarchical discriminators of HDGAN inspired by [24] with an intuition to reduce their variances and unify their common goal (i.e. differentiates real and fake despite difficult scales). However, we did not find benefits from this mechanism and our independent discriminators can be trained very stably.

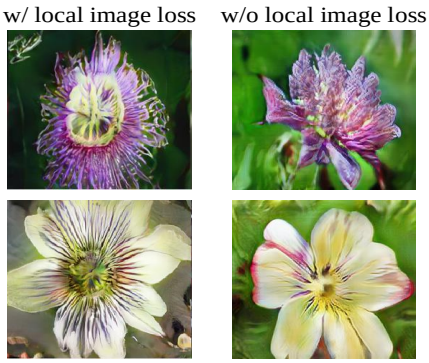


Figure 8: Qualitative evaluation of the local adversarial loss. We can see the two images w/ the local image loss more accurately exhibit complex flower petal structures described in the (colored) text.

5. Conclusion

In this paper, we present an effective and end-to-end method to tackle the problem of generating high-resolution images conditioned on text descriptions. We introduce a novel hierarchical-nested adversarial discriminators to regularize the mid-level representations of a generator. To improve the model's capability to render fine-grained local details, we propose a multi-purpose discriminator loss to emphasize local patch realism as well as matching aware pair. We also introduce a new evaluation metric for text-to-image synthesis to evaluate the semantic consistency between generated images and conditioned text. Extensive experiment results demonstrate that our method, namely HDGAN, performs significantly better than existing state of the arts.

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] 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. 3

[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

[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

[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. 3

[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. 3

[26] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/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, 4, 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] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, June 2015. 5

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

[41] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 3

[42] 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

[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. 3, 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