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

Anonymous CVPR submission

Paper ID ****

Abstract

This paper presents a novel method to deal with the challenging task of generating images conditioned on semantic image descriptions. We propose an effective end-to-end trainable single-stream network that can generate photographic high-resolution images (up to 512^2). Our method leverages the 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 learning strategy at multiple nested hierarchical side outputs to encourage more effective multimodal (i.e. image and text) information alignment in order to guarantee semantic consistency and image fidelity simultaneously. With extensive experimental validation on three major datasets, our method significantly improves previous state of the arts over all datasets on three different metrics.

1. Introduction

Photographic text-to-image synthesis is a significant problem in generative model research [34]. However, insufficient methods have been developed to successfully address this task due to its particular challenges. It is a special case of generative learning tasks, which aim to map a low-dimensional embedding to a complex RGB image space. Differently, the embedding space of the former is induced by semantic descriptions. Therefore, this task requires the generated images to be *semantically consistent*, i.e., the generated images not only preserve sketch concepts of descriptions but also fine-grained descriptive details in image pixels.

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 handles image up to 64^2 resolution and can barely generate vivid object details. Based on this method, Zhang *et al*. [44] present a successful approach (StackGAN) by stacking another low-to-high resolution GAN to generate high-quality 256^2 images, with two separate stage training. Later on, Dong *et al*. [9] propose to bypass the difficult of translate vector embedding to RGB images and treat it as an pixel-to-pixel translation [16], by re-rendering an arbitrary-style training (128^2) image conditioned on a targeting description. But its high-resolution synthesis ability is unclear. At present, how to train from the low-dimensional text space to synthesize high-resolution and diverse images in an end-to-end manner is still an open and attractive question.

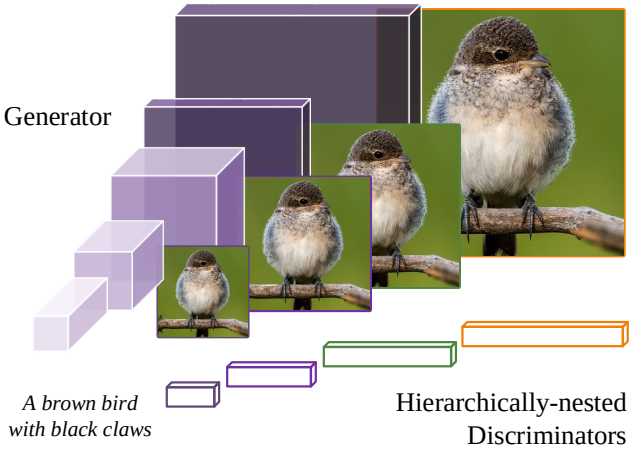


Figure 1: Overview of our method. It contains a single-stream generator and several hierarchically-nested discriminators along the depth dimension of the generator, which are trained end-to-end to generate high-resolution images conditioned on a sentence.

We outline several empirical reasons for the challenges of text-to-image synthesis using GANs. The first is the fundamental difficulty of balancing the convergence between generators and discriminators [12, 37]. The second is the huge pixel space in high-resolution images [44]. An effec-

tive strategy to regularize generators is critical to help capture the complex image statistics [14] as well as guarantee semantic consistency. With careful consideration of these outlined problems, in this paper, we propose a novel fully end-to-end method that can directly model high-resolution image statistics and generate semantic consistent and photographic image up to 512^2 resolution. The contributions are described as follows.

Our generator resembles a simple vanilla-like GAN, without requiring any internal text conditioning like [44] or outside image annotation supervision like [7]. 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 objectives 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 generator convolutional neural network (CNN) framework to cooperate with our nested discriminators more effectively. To better make use of image and text information in GAN training, we enforce nested discriminators at different hierarchies can simultaneously differentiate real-and-fake image-text pairs as well as real-and-fake whole images or image patches. This multi-purpose conditional adversary is mutually beneficial to allow each only focuses on its respect duties.

We validate our proposed method on three datasets, CUB birds [41], Oxford-102 flowers [30], and large-scale MSCOCO [24]. Extensive experimental results and analysis demonstrate the effectiveness of our method and significantly improved performance compared against previous state of the arts. 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 attract wide interests recently, including GANs [12], Variational Auto-encoders (VAE) [19], 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 [16, 46], medical applications [6], etc [22, 14].

Text-to-image synthesis not only requires diverse and high-quality 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, which uses extra information to help generate 128^2 resolution images. StackGAN *et al*. [44] propose a two-stage stacking GAN training approach that is able to generate 256^2 compelling results. Recently, Dong *et al*. [9] propose to learn a joint embedding of images and text so as to re-render a prototype image conditioned on targeting descriptions. Cha *et al*. [28] explore the usage of the perceptional loss with a CNN pretrained on ImageNet [17] and Dash *et al*. [7] make use of auxiliary classifiers (similar with [31]) to assist GAN training for text-to-image synthesis.

Learning a continuous mapping from low-dimensional embeddings 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 those tasks, through various stabilization training techniques [36, 1, 3, 38, 31], regularization using outside knowledge (e.g. image labels, ImageNet CNNs, etc) [10, 22, 7, 7], or multiple discriminators [27, 11, 43]. *While our method does not use any extra information apart from training paired text and images.* Moreover, 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*. [8] 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 last level of the pyramid, and push up the output resolution through by-stage refinement. StackGAN also shares similar strategy with LAPGAN. Following this strategy, Chen *et al*. [5] present a cascaded refinement network to synthesize high-resolution scene from semantic maps. Huang *et al*. [14] propose a top-down stacked GAN to leverage mid-level representations, which shares some similarities with StackGAN and our method. However, this method needs multiple symmetric bottom-up pre-trained discriminators and the usage for high-resolution image is unclear. Recently, Karras *et al*. [18] propose a progressive training of GANs for high-resolution image generation (Figure 2 C). *Compared with these strategies that train low-to-high resolution GANs progressively, our method has advantages of utilizing multi-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 [23] and pixel or object classification [42, 4, 26]. DSN [23] proposes deep supervision in hierarchical convolutional layers to increase the discriminativeness of feature representations. *Our hierarchically-nested adversarial objective is inspired by CNNs with deep supervision [23, 42].*

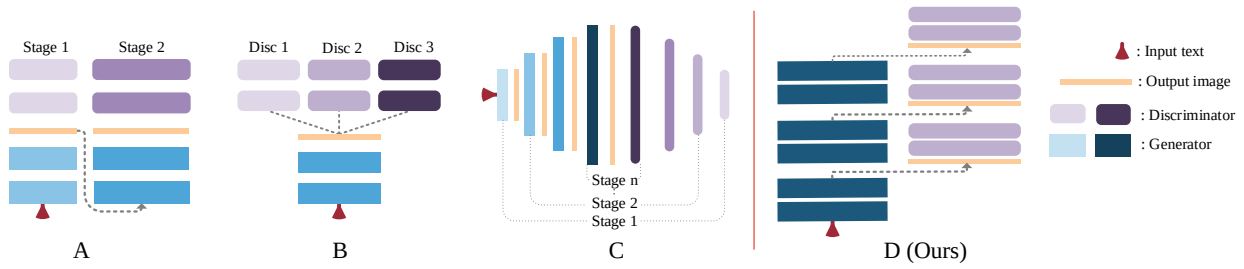


Figure 2: Overviews of some typical GAN frameworks. **A** decomposes tasks multi-stage GANs [44, 8]. **B** uses multiple discriminators with one generator [11, 29]. **C** progressively trains symmetric discriminators and generators [18, 14]. **A** and **C** can be viewed as decomposing high-resolution 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.

3. Method

3.1. Adversarial objective basics

In brief, the two-player game in GANs [12] is played by a generator G network and a discriminator D network, which are alternatively trained to compete with each other and thereby improve each other. The discriminator is optimized to distinguish synthesized images from real images, meanwhile, the generator is trained to generate realistic images to fool the discriminator. Concretely, the overall min-max optimization objective is defined as:

$$G^*, D^* = \arg \min_G \max_D \mathcal{L}(D, G) \tag{1}$$

where the loss $\mathcal{L}$ aims to minimize the Jensen-Shannon divergence between the data and model distributions.

3.2. Hierarchical-nested adversarial objectives

Numerous methods have demonstrated a variety of ways to build GANs for image synthesis. Figure 2 and Section 2 discuss some typical frameworks. Our method 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 to learn a stronger generator $\mathcal{G}$. 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 signals.

The proposed $\mathcal{G}$ is a network (see Section 3.4), with $s-1$ side outputs at the intermediate layers:

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

where $\boldsymbol{z} \sim \mathcal{N}(0, 1)$ denotes a random noise and $\boldsymbol{t} \sim p(data)$ denotes a sentence embedding vector in the real training data distribution. $\mathcal{G}$ produces $s-1$ side outputs (size-growing images) and a final output X_s with the highest resolution.

For each side output X_i, we propose to apply a discriminator D_i to perform independently adversarial games. Therefore, our full min-max objective can be defined as follows:

$$\begin{aligned} &\mathcal{G}^*, \mathcal{D}^* = \arg \min_G \max_{\mathcal{D}} \mathcal{L}(\mathcal{G}, \mathcal{D}, \mathcal{Y}, \boldsymbol{t}, \boldsymbol{z}) \\ &\text{where } \mathcal{D} \in \{D_1, ..., D_s\},\ \mathcal{Y} = \{Y_1, ..., Y_s\} \end{aligned} \tag{3}$$

where $\mathcal{Y}$ denotes training images at multi-scales, $\{1, ..., s\}$. Compared with Eq. (1), one generator competes with multiple discriminators at different hierarchies. Discriminators do not share weights with each other, which offers them full flexibility to focus on learning diverse discriminative features in different contextual scales. In principle, the lower-resolution side output is forced to learn semantic consistent image structures (e.g., object sketch, color, and background), and the subsequent higher-resolution side outputs are used to render details to the images. Since the model is trained in an end-to-end fashion, we can observe more consistent information exhibited in both lower and higher-resolution outputs than StackGAN [44]. More results can be founded in experiments. According to this goal, we present the following adversarial losses to support more effective multi-purpose discriminators.

3.3. Functional adaptive adversarial losses

Our generator produces size-growing side outputs which compose an image pyramid. We leverage the hierarchy of it and allow adversarial losses to be multi-purpose to guarantee both semantic consistency and image fidelity.

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 mismatch conditional text information and fake image with real text.

The pair loss is designed to capture the global semantic context. However, this loss has two major weaknesses. First, there is no explicit loss that force the discriminator to differentiate real images from fake images. Second, as the image resolution goes higher, it might be challenging for a global pair-loss discriminator to capture local fine-grained

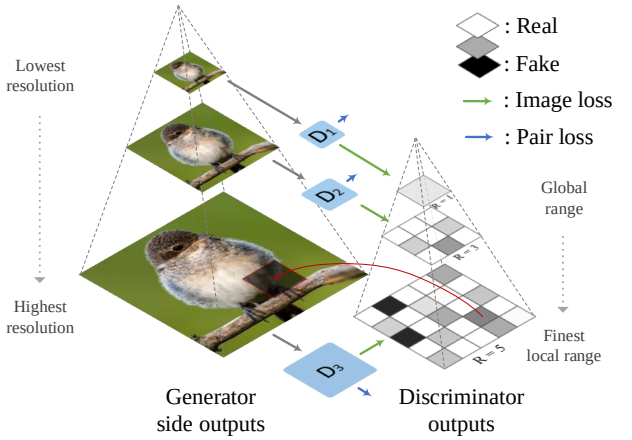


Figure 3: Given an image in the side output pyramid of the generator, the corresponding discriminator D_i computes the matching-aware pair loss and adaptive local image loss (outputting a $R\times R$ probability map to indicate real or fake patches). The focal range increases as image resize grows. Lower range results in finer results.

details. In addition, as pointed in [38], a single global discriminator may over-emphasize certain unexpected image features and lead to artifacts.

To guarantee image fidelity and overcome these two weaknesses, our solution is to add adaptive local adversarial image losses on image patches. The insights are two folds:

- Further forcing the generated samples to have only only global but also local similar statistics to real image patches, which is beneficial for rendering fine-grained details.
- Image loss without using paired text can be beneficial in the sense that it focus more on image information. The richness of discriminator supervision usually has advantages for generalization.

We expect the low-resolution discriminator to focus on global structures, while high-resolution outputs focus more on local image details. Therefore, through the pyramid of the size-growing side outputs, we configure the focal range (the size of each non-overlapping image patch) to be also growing. The adaptive local image adversarial loss is implemented as a fully CNN [38, 46]. Figure 3 illustrates how hierarchically-nested discriminators measure the two losses on the generated images.

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

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

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

where $X_i = \mathcal{G}(\boldsymbol{t}_X, \boldsymbol{z})$. $L_2(x) = (x-\mathbb{I})^2$ is the mean-square loss (we do not use the original cross-entropy loss) and $\overline{L_2}(x) = (x)^2$. For adaptive local image loss with varying focal range, the shape of $x, \mathbb{I} \in \mathbb{R}^{R\times R}$ varies accordingly (see Figure 3) for the local image loss. $R=1$ refers to the (largest local) global range, e.g., $R\equiv 1$ for the matching aware pair loss. 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 [20] and the practice of [44] (namely conditional augmentation), we sample the input text variable $\boldsymbol{t}$ from a Ganssian distribution $\mathcal{N}(\mu(\boldsymbol{t}), \Sigma(\boldsymbol{t}))$, where μ and Σ are functions of $\boldsymbol{t}$. We add the Kullback-Leibler divergence regularization term, $D_{KL}(\mathcal{N}(\mu(\boldsymbol{t}), \Sigma(\boldsymbol{t}))||\mathcal{N}(0, \boldsymbol{I}))$, to the G loss to 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 [13], which contains two convolutional (conv) layers (with batch normalization (BN) [15] 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 $1024\times4\times4$ embeddings replicated by a 1024-d text embedding, 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).

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 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 one first concatenate $128\times4\times4$ text embedding (replicated from a 128-d text embedding). Then

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

we use an 1×1 conv to fuse text and image information and a 4×4 conv layer to classify the the pair is real or fake.

All intermediate conv layers for both generators and discriminators use 3×3 kernels (with reflection padding and no bias). We also experimented other normalization (i.e. instance normalization [40] and layer normalization [2]) used by recent advances [46, 5]. Both are not satisfactory.

3.5. Training Details

The Adam optimizer is used for all experiments. The initial learning rate is set as 0.0002 and decreased by half for every 100 epochs (30 for COCO). The CUB and Oxford-102 datasets are trained for 500 epochs in total (250 epochs for COCO with 4 GPUs). For the adaptive local image loss, we set $R = 1$ for 64^2 side output, $R = 3$ for 128^2 side output and $R = 5$ for 256^2 and 512^2 outputs. We use 1-repeat residual block for the generator till 256^2 resolution. To generate 512^2 images, we pre-train the generator to 256^2 for the consideration of memory saving and faster convergence. We use 3-repeat res-block followed by the stretching and linear compression layer. Since the 256^2 image already decides the overall semantics and details, to encourage the 512^2 maintain these information, we also use a reconstruction loss to self-regularize the generator in case the discriminator confuses the expected output.

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 as well as the idea of Hierarchically-nested Discriminators.

Dataset We evaluate our method extensively using three widely used datasets. The CUB dataset [41] contains 11,788 bird images belonging to 200 different categories. We pre-process and split the images using the same pipeline in [34, 44] 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, [24] 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 of the three datasets into a 1024 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 the variation of generated outputs and can find mode collapses reliably [31]. Lower score indicates higher variance.

Method	Dataset		
	CUB	Oxford	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**	-

*Recently, it updated to 10.62±.19 in its latest github.

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.

Scale	StackGAN	HDGAN
64×64	2.95±.02	**3.46±.04**
128×128	3.35±.02	**3.85±.04**
256×256	3.70±.04	**4.15±.05**

Table 2: Multi-scale inception score comparison. See text for explanation.

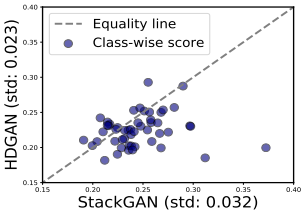
Figure 4: Results on CUB

Visual-semantic similarity Since the aforementioned evaluation methods can not measure alignment between the generated images and the text description. We introduce a new qualitative measurement, namely neural visual-semantic similarity (VS similarity) [21]. Denote $\boldsymbol{v}$ as the feature vector extract by a pre-trained CNN model f_{cnn} (inception v2[39] pre-trained on Imagenet), and $\boldsymbol{t}$ denote the text embedding. We define a scoring function $s(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{||\boldsymbol{x}||\cdot||\boldsymbol{y}||}$. We learn two mapping functions f_v and f_t, which map the image and sentence embeddings into one semantic space, using the following bi-directional ranking loss:

$$\begin{aligned}\mathcal{L}_{vs} = &\sum_{\boldsymbol{v}}\sum_{\boldsymbol{t}^-} \max(0, \delta - s(f_v(\boldsymbol{v}), f_t(\boldsymbol{t}^-))) \\ &+ \sum_{\boldsymbol{t}}\sum_{\boldsymbol{v}-} \max(0, \delta - s(f_t(\boldsymbol{t}), f_v(\boldsymbol{v}^-)))\end{aligned} \tag{5}$$

where δ is the margin, which is set as 0.2, every $(\boldsymbol{v}, \boldsymbol{t})$ is a ground truth image text pair, $\boldsymbol{t}^-$ denotes a mismatching text description for the image corresponding to $\boldsymbol{v}$, and vice-versa for $\boldsymbol{v}^-$. In the testing stage, given an text embedding $\boldsymbol{t}$, and the generated images o, the corresponding vs-similarity will be calculated as $s(f_{cnn}(o), \boldsymbol{t})$.

Figure 5: Results on Oxford



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 invariance. The points below the equality line represent classes our HDGAN wins. Right: Overall (no class-wise) score evaluation.

4.1. Comparative Results

To validate our method, we compare our results with GAN-INT-CLS [34], GAWWN [35], TAC-GAN [7], StackGAN [44] and also its improved version StackGAN++ [45], and Progressive GAN [18] [3]

Table 4 shows the quantitative inception-score evaluation. We follow StackGAN to sample $30,000$ images for evaluation. HDGAN achieves significantly improvement compared to other methods. For example, HDGAN improves GAN-INT-CLS by $\sim$30.1% and $\sim$8.4% on CUB. It also significantly outperforms StackGAN StackGAN++. HDGAN achieves competitive results with TAC-GAN on the Oxford flower dataset. TAC-GAN uses image labels to increase the discriminability of generators, while we do not use any extra knowledge.

Table 2 quantitatively compares the the multi-resolution inception score on CUB. Our results are from the side outputs of a single model. Usually higher resolution images gives rise to better inception score since they contain details. 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.

Table 3 shows the MS-SSMI image quality evaluation score compared with StackGAN and ProgressiveGAN on CUB. StackGAN and our HDGAN use the semantic text as input so the generated images retain class information. We compare the score with StackGAN using the provided pretrained model). We randomly sample $20,000$ image pairs (400 per class) and show the class-wise score in the left figure. HDGAN shows better performance in majority of classes and also has lower inter-class standard divition (0.023 vs. 0.032). Prog.GAN takes noises rather than text. Since MS-SSMI evaluates the image quality without text or class involvement, we can compare with it. Following the procedure of Prog.GAN, we randomly sample $10,000$ image pairs for each comparing method (For Prog.GAN, we use generated 256^2 images provided by the author) and show the results in the right table. Our method outperforms both StackGAN and Prog.GAN.

[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, in spite of their differences in motivations and network and training designs from ours.

Method	Dataset		
	CUB	Oxford	COCO
StackGAN	0.228±.162	0.278±.134	−±−
Ground Truth	0.302±.151	0.336±.138	-
HDGAN	**0.259±.159**	−±−	-

Table 4: The visual-semantic similarity evaluation on three datasets. The higher score represents higher degree of matching between the generated images and the text information.

4.1.1 Style Transfer Using Sentence Interpolation

Ideally, a trained model should learn a smooth linear latent manifold of the images. To demonstrate our model's generalization capability, we generate images using the linearly interpolated embedding using two source sentences. During this experiment, we fix the all the random noises to make the object and background consistent. As is shown in Figure **??**, the generated images show gradual and smooth changes reflecting the semantic changing in sentences, while still maintaining plausible object pose and shape. In the first row, the generated birds gradually turn the color from red to yellow, in the second row, more complicated sentences containing more detailed appearance information (e.g., blue peaks, and yellow wing) are used, we can see that our model is still be able to successfully capture these subtle information and tune the bird appearance gradually.

We also evaluate some sentences that generated by ourself, the resulting images are shown in Figure **??**, as we can see, our model are robust and generalize well to human descriptions that never seen before.

4.1.2 What role does noise play

In our model, there are two kinds of randomness injected to the input of the Generator, ie, epsilon ϵ and Gaussian noise vector z. In this experiment, we fix the sentence embedding and one noise, and change the other random variable. For each random noise, we generate 4 different values. So In total 16 images are generated using all noise combinations, the resulting images are shown in Figure **??**, ϵ and z corresponds to row and column dimension, respectively. As is shown in the image, we can see that, ϵ can change the pose of the image, while z is mainly responsible for the background variations. This clearly demonstrate that, our

Metric	No Local	With Local
MS-SSMI	–	0.215
VS similarity	–	0.259±.159
Inception score	–	4.15±.05

Table 5: Results using three quantitative evaluation metrics on birds dataset.

Figure 6

model learns to disentangle the semantic information from the pose and background information.

5. On the effects of Individual Components

5.1. Hierarchically-nested adversary

Recall that the proposed hierarchically-nested adversarial supervision plays a role of regularizing the layer representations. In practice, we apply it on feature maps at $\{64, 128, 256\}$. In Table 6, we show the inception score by removing partial of supervision components. As can be observed, ...

StackGAN emphasizes the importance of using text embeddings in the (stage-II) mid-level features of the 256×256 generator by showing an large drop from 3.7 to 3.45 without doing so. The text embedding plays an important role in maintaining the diversity and semantic consistency in StackGAN. While in our method, we only use text embeddings at the input. Our improved results demonstrate that our hierarchically-nested adversarial supervision regularizes the generator to achieve the same goal.

5.2. On the Local Image Loss

We also study the effectiveness of the proposed local image loss. We conduct experiments with using architecture with side output 64×64 and 256 × 256. As can be seen in Table**??**, we can see that by removing local image loss (denoted as "No Local"), the inception score drop from 4.27 to 3.8. To quantitatively compare the results, we also show generated results using two model in Figure **??**, We can see that, also both models can successfully capture the semantic correspondence between image and sentence, 256× 256 results generated by model trained with local image loss provide more details, thus improving the visual quality.

5.3. Design principles

Note that for text-to-image synthesis, StackGAN is the only successful method to generate 256×256 images. At the same time, it also shows the difficulty (impossibility) of directly training a vanilla 256×256 GAN, which fails to generate meaningful images. We test this extreme case using our method by removing all nested supervisions (the first row of Table 6). Our method still generates fairly good results. Figure 6 shows the qualitatively results. This strongly proves the effectiveness of our designs in generators and discriminator losses.

Components			Score
64	128	256	
		✓	3.52±.04
	✓	✓	3.99±.04
✓		✓	4.14±.03
✓	✓	✓	4.15±.05

Table 6: Ablation study for hierarchically-nested adversarial supervision.

Initially, we tried to share top layers of the hierarchical discriminators of HDGAN inspired by [25] 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 any benefits from this and our independent discriminators can be well-trained by themselves.

Advantages The advantages of our hierarchical-nested adversary are clearer from different views, compared with the popular stacking GAN strategy [].

First, it is memory efficient. When viewing it as ensembled multi-scale GAN (Figure **??**), each GAN fully uses the parameters of its lower resolution GAN. Usually, a deeper generator is necessary to compact a discriminator. Since the depth of discriminator increases as image resolution increases, if stacking separate GANs based on low resolution GAN outputs, much more layers are needed in generators as wells. While in our method, for instance, from a 64^2 GAN to a 128^2 GAN, only a minimum two conv layers (1-repeat res-blocks) are needed.

Second, it fully enjoys top-down knowledge. The low resolution can not use high resolution information in the stacking GAN strategy. While in our method, it is easy to see the low resolution GAN enjoys knowledge from higher resolution. As an evidence, our 64^2 synthetic images have higher inception score than 64^2 and even 128^2 images of StackGAN (see Section 5).

6. Conclusion

...

References

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

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

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

[10] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, pages 658–666, 2016.

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

[12] 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.

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

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

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

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

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[18] 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.

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

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

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

[22] 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.

[23] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, 2015.

[24] 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

[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.

[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.

[40] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

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

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

[43] 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.

[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 *ICCV*, 2017.

[45] 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.

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