Inferring Semantic Layout for Hierarchical Text-to-Image Synthesis

Seunghoon Hong[†] Dingdong Yang[†] Jongwook Choi[†] Honglak Lee[‡,†]
[†]University of Michigan
[‡]Google Brain
[†]{hongseu,didoyang,jwook,honglak}@umich.edu [‡]honglak@google.com

Abstract

We propose a novel hierarchical approach for text-to-image synthesis by inferring semantic layout. Instead of learning a direct mapping from text to image, our algorithm decomposes the generation process into multiple steps, in which it first constructs a semantic layout from the text by the layout generator and converts the layout to an image by the image generator. The proposed layout generator progressively constructs a semantic layout in a coarse-to-fine manner by generating object bounding boxes and refining each box by estimating object shapes inside the box. The image generator synthesizes an image conditioned on the inferred semantic layout, which provides a useful semantic structure of an image matching with the text description. Our model not only generates semantically more meaningful images, but also allows automatic annotation of generated images and user-controlled generation process by modifying the generated scene layout. We demonstrate the capability of the proposed model on challenging MS-COCO dataset and show that the model can substantially improve the image quality, interpretability of output and semantic alignment to input text over existing approaches.

1. Introduction

Generating images from text description has been an active research topic in computer vision. By allowing users to describe visual concepts in natural language, it provides a natural and flexible interface for conditioning image generation. Recently, approaches based on conditional Generative Adversarial Network (GAN) have shown promising results on text-to-image synthesis task [21, 34, 23]. Conditioning both generator and discriminator on text, these approaches are able to generate realistic images that are both diverse and relevant to input text. Based on conditional GAN framework, recent approaches further improve the prediction quality by generating high-resolution images [34] or augmenting text information [6, 4].

However, the success of existing approaches has been mainly limited to simple datasets such as birds [33] and flowers [17], while generation of complicated, real-world images such as MS-COCO [13] remains an open challenge. As illustrated in Figure 1, generating image from a general sentence *"people riding on elephants that are walking through a river"* requires multiple reasonings on various visual concepts, such as object category (*people* and *elephants*), spatial configurations of objects (*riding*), scene context (*walking through a river*), *etc.*, which is much more complicated than generating a single, large object as in simpler datasets [33, 17]. Existing approaches have not been successful in generating reasonable images for such complex text descriptions, because of the complexity of learning a direct text-to-pixel mapping from general images.

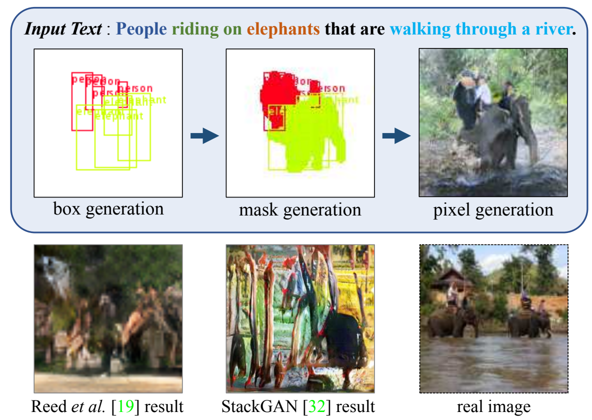


Figure 1. Overall framework of the proposed algorithm. Given a text description, our algorithm sequentially constructs a semantic structure of a scene and generates an image conditioned on the inferred layout and text. Best viewed in color.

Instead of learning a direct mapping from text to image, we propose an alternative approach that constructs *semantic layout* as an intermediate representation between text and image. Semantic layout defines a structure of scene based on object instances and provides fine-grained information of the scene, such as the number of objects, object category, location, size, shape, *etc*. (Figure 1). By introducing a mechanism that explicitly aligns the semantic structure of an image to text, the proposed method can generate complicated images that match complex text descriptions. In addition, conditioning the image generation on semantic structure allows our model to generate semantically more meaningful images that are easy to recognize and interpret.

Our model for hierarchical text-to-image synthesis consists of two parts: the *layout generator* that constructs a semantic label map from a text description, and the *image generator* that converts the estimated layout to an image by taking the text into account. Since learning a direct mapping from text to fine-grained semantic layout is still challenging, we further decompose the task into two manageable subtasks: we first estimate the bounding box layout of an image using the *box generator*, and then refine the shape of each object inside the box by the *shape generator*. The generated layout is then used to guide the image generator for pixel-level synthesis. The box generator, shape generator and image generator are implemented by independent neural networks, and trained in parallel with corresponding supervisions.

Generating semantic layout not only improves quality of text-to-image synthesis, but also provides a number of potential benefits. First, the semantic layout provides instance-wise annotations on generated images, which can be directly exploited for automated scene parsing and object retrieval. Second, it offers an interactive interface for controlling image generation process; users can modify the semantic layout to generate a desired image by removing/adding objects, changing size and location of objects, *etc*.

The contributions of this paper are as follows:

- We propose a novel approach for synthesizing images from complicated text descriptions. Our model explicitly constructs semantic layout from the text description, and guides image generation using the inferred semantic layout.
- By conditioning image generation on explicit layout prediction, our method is able to generate images that are semantically meaningful and well-aligned with input descriptions.
- We conduct extensive quantitative and qualitative evaluations on challenging MS-COCO dataset, and demonstrate substantial improvement on generation quality over existing works.

The rest of the paper is organized as follows. We briefly review related work in Section 2, and provide an overview of the proposed approach in Section 3. Our model for layout and image generation is introduced in Section 4 and 5, respectively. We discuss the experimental results on the MS-COCO dataset in Section 6.

2. Related Work

Generating images from text descriptions has recently drawn a lot of attention from the research community. Formulating the task as a conditional image generation problem, various approaches have been proposed based on Variational Auto-Encoders (VAE) [14], auto-regressive models [22], optimization techniques [16], *etc*. Recently, approaches based on conditional Generative Adversarial Network (GAN) [7] have shown promising results in text-to-image synthesis [21, 23, 34, 6, 4]. Reed *et al*. [21] proposed to learn both generator and discriminator conditioned on text embedding. Zhang *et al*. [34] improved the image quality by increasing image resolution with a two-stage GAN. Other approaches include improving conditional generation by augmenting text data with synthesized captions [6], or adding conditions on class labels [4]. Although these approaches have demonstrated impressive generation results on datasets of specific categories (*e.g*., birds [33] and flowers [17]), the perceptual quality of generation tends to substantially degrade on datasets with complicated images (*e.g*., MS-COCO [13]). We investigate a way to improve text-to-image synthesis on general images, by conditioning generation on the inferred semantic layout.

The problem of generating images from pixel-wise semantic labels has been explored recently [3, 10, 12, 22]. In these approaches, the task of image generation is formulated as translating semantic labels to pixels. Isola *et al*. [10] proposed a pixel-to-pixel translation network that converts dense pixel-wise labels to an image, and Chen *et al*. [3] proposed a cascaded refinement network that generates high-resolution output from dense semantic labels. Karacan *et al*. [12] employed both dense layout and attribute vectors for image generation using conditional GAN. Notably, Reed *et al*. [22] utilized sparse label maps like our method. Unlike previous approaches that require ground-truth layouts for generation, our method *infers* the semantic layout, and thus is more generally applicable to various generation tasks. Note that our main contribution is complementary to these approaches, and we can integrate existing segmentation-to-pixel generation methods to generate an image conditioned on a layout inferred by our method.

The idea of inferring scene structure for image generation is not new, as it has been explored by some recent works in several domains. For example, Wang *et al*. [32] proposed to infer a surface normal map as an intermediate structure to generate indoor scene images, and Villegas *et al*. [29] predicted human joints for future frame prediction. The most relevant work to our method is Reed *et al*. [23], which predicted local key-points of bird or human for text-to-image synthesis. Contrary to the previous approaches that predict such specific types of structure for image generation, our proposed method aims to predict semantic label maps, which is a general representation of natural images.

3. Overview

The overall pipeline of the proposed framework is illustrated in Figure 2. Given a text description, our model progressively constructs a scene by refining semantic structure

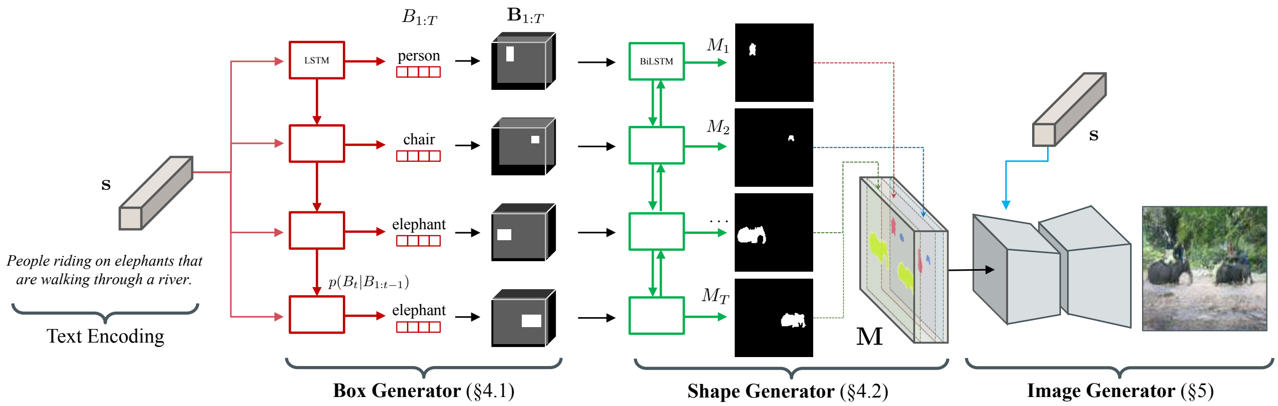


Figure 2. Overall pipeline of the proposed algorithm. Given a text embedding, our algorithm first generates a coarse layout of the image by placing a set of object bounding boxes using the box generator (Section 4.1), and further refines the object shape inside each box using the shape generator (Section 4.2). Combining outputs from the box and the shape generator leads to a semantic label map defining semantic structure of the scene. Conditioned on the inferred semantic layout and the text, a pixel-wise image is finally generated by the image generator (Section 5).

of an image using the following sequence of generators:

- **Box generator** takes a text embedding $\mathbf{s}$ as input, and generates a coarse layout by composing object instances in an image. The output of the box generator is a set of bounding boxes $B_{1:T} = \{B_1, ..., B_T\}$, where each bounding box B_t defines the location, size and category label of the t-th object (Section 4.1).
- **Shape generator** takes a set of bounding boxes generated from box generator, and predicts shapes of the object inside the boxes. The output of the shape generator is a set of binary masks $M_{1:T} = \{M_1, ..., M_T\}$, where each mask M_t defines the foreground shape of the t-th object (Section 4.2).
- **Image generator** takes the semantic label map $\mathbf{M}$ obtained by aggregating instance-wise masks, and the text embedding as inputs, and generates an image by translating a semantic layout to pixels matching the text description (Section 5).

By conditioning the image generation process on the semantic layouts that are explicitly inferred, our method is able to generate images that preserve detailed object shapes and therefore are easier to recognize semantic contents. In our experiments, we show that the images generated by our method are semantically more meaningful and well-aligned with the input text, compared to ones generated by previous approaches [21, 34] (Section 6).

4. Inferring Semantic Layout from Text

4.1. Bounding Box Generation

Given an input text embedding $\mathbf{s}$, we first generate a coarse layout of the image in the form of object bounding boxes. We associate each bounding box B_t with a class label to define which class of object to place and where, which plays a critical role in determining the global layout of the scene. Specifically, we denote the labeled bounding box of the t-th object as $B_t = (\mathbf{b}_t, \boldsymbol{l}_t)$, where $\mathbf{b}_t = [b_{t,x}, b_{t,y}, b_{t,w}, b_{t,h}] \in \mathbb{R}^4$ represents the location and size of the bounding box, and $\boldsymbol{l}_t \in \{0,1\}^{L+1}$ is a one-hot class label over L categories. We reserve the $(L+1)$-th class as a special indicator for the end-of-sequence.

The *box generator* G_{box} defines a stochastic mapping from the input text $\mathbf{s}$ to a set of T object bounding boxes $B_{1:T} = \{B_1, ..., B_T\}$:

$$\widehat{B}_{1:T} \sim G_{\text{box}}(\mathbf{s}). \quad (1)$$

Model. We employ an auto-regressive decoder for the box generator, by decomposing the conditional joint bounding box probability as $p(B_{1:T} \mid \mathbf{s}) = \prod_{t=1}^{T} p(B_t \mid B_{1:t-1}, \mathbf{s})$, where the conditionals are approximated by LSTM [9]. In the generative process, we first sample a class label $\boldsymbol{l}_t$ for the t-th object and then generate the box coordinates $\mathbf{b}_t$ conditioned on $\boldsymbol{l}_t$, *i.e.*, $p(B_t|\cdot) = p(\mathbf{b}_t, \boldsymbol{l}_t|\cdot) = p(\boldsymbol{l}_t|\cdot)\, p(\mathbf{b}_t|\boldsymbol{l}_t, \cdot)$. The two conditionals are modeled by a Gaussian Mixture Model (GMM) and a categorical distribution [8], respectively:

$$p(\boldsymbol{l}_t \mid B_{1:t-1}, \mathbf{s}) = \text{Softmax}(\mathbf{e}_t), \quad (2)$$

$$p(\mathbf{b}_t \mid \boldsymbol{l}_t, B_{1:t-1}, \mathbf{s}) = \sum_{k=1}^{K} \pi_{t,k} \mathcal{N}\left(\mathbf{b}_t; \boldsymbol{\mu}_{t,k}, \boldsymbol{\Sigma}_{t,k}\right), \quad (3)$$

where K is the number of mixture components. The softmax logit $\mathbf{e}_t$ in Eq.(2) and the parameters for the Gaussian mixtures $\pi_{t,k} \in \mathbb{R}$, $\boldsymbol{\mu}_{t,k} \in \mathbb{R}^4$ and $\boldsymbol{\Sigma}_{t,k} \in \mathbb{R}^{4\times 4}$ in Eq.(3) are computed by the outputs from each LSTM step.

Training. We train the box generator by minimizing the negative log-likelihood of ground-truth bounding boxes:

$$\mathcal{L}_{\text{box}} = -\lambda_l \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{l}_t^* \log p(\boldsymbol{l}_t) - \lambda_b \frac{1}{T} \sum_{t=1}^{T} \log p(\mathbf{b}_t^*), \quad (4)$$

where T is the number of objects in an image, and λ_l, λ_b are balancing hyper-parameters. $\mathbf{b}_t^*$ and $\boldsymbol{l}_t^*$ are ground-truth bounding box coordinates and label of the t-th object, respectively, which are ordered based on their bounding box locations from left to right. Note that we drop the conditioning in Eq. (4) for notational brevity. The hyper-parameters are set to $\lambda_l = 4, \lambda_b = 1$ and $K = 20$ in our experiments.

At test time, we generate bounding boxes via ancestral sampling of box coordinates and class label by Eq. (2) and (3), respectively. We terminate the sampling when the sampled class label corresponds to the termination indicator $(L+1)$, thus the number of objects are determined adaptively based on the text.

4.2. Shape Generation

Given a set of bounding boxes obtained by the box generator, the shape generator predicts more detailed image structure in the form of object masks. Specifically, for each object bounding box B_t obtained by Eq. (1), we generate a binary mask $M_t \in \mathbb{R}^{H\times W}$ that defines the shape of the object inside the box. To this end, we first convert the discrete bounding box outputs $\{B_t\}$ to a binary tensor $\mathbf{B}_t \in \{0,1\}^{H\times W\times L}$, whose element is 1 if and only if it is contained in the corresponding class-labeled box. Using the notation $M_{1:T} = \{M_1, ..., M_T\}$, we define the *shape generator* G_{mask} as

$$\widehat{M}_{1:T} = G_{\text{mask}}(\mathbf{B}_{1:T}, \mathbf{z}_{1:T}), \tag{5}$$

where $\mathbf{z}_t \sim \mathcal{N}(0, I)$ is a random noise vector.

Generating an accurate object shape should meet two requirements: (i) First, each instance-wise mask M_t should match the location and class information of $\mathbf{B}_t$, and be recognizable as an individual instance (instance-wise constraints). (ii) Second, each object shape must be aligned with its surrounding context (global constraints). To satisfy both, we design the shape generator as a recurrent neural network, which is trained with two conditional adversarial losses as described below.

Model. We build the shape generator G_{mask} using a convolutional recurrent neural network [25], as illustrated in Figure 2. At each step t, the model takes $\mathbf{B}_t$ through encoder CNN, and encodes information of all object instances by bi-directional convolutional LSTM (Bi-convLSTM). On top of the convLSTM output at t-th step, we add noise $\mathbf{z}_t$ by spatial tiling and concatenation, and generate a mask M_t by forwarding it through a decoder CNN.

Training. Training of the shape generator is based on the GAN framework [7], in which generator and discriminator are alternately trained. To enforce both the global and the instance-wise constraints discussed earlier, we employ two conditional adversarial losses [15] with the instance-wise discriminator D_{inst} and the global discriminator D_{global}.

First, we encourage each object mask to be compatible with class and location information encoded by object bounding box. We train an instance-wise discriminator D_{inst} by optimizing the following *instance-wise adversarial loss*:

$$\begin{aligned}\mathcal{L}_{\text{inst}}^{(t)} = \mathbb{E}_{(\mathbf{B}_t, M_t)}\Big[\log D_{\text{inst}}\big(\mathbf{B}_t, M_t\big)\Big] \\ + \mathbb{E}_{\mathbf{B}_t, \mathbf{z}_t}\Big[\log\Big(1 - D_{\text{inst}}\big(\mathbf{B}_t, G_{\text{mask}}^{(t)}(\mathbf{B}_{1:T}, \mathbf{z}_{1:T})\big)\Big)\Big],\end{aligned} \tag{6}$$

where $G_{\text{mask}}^{(t)}(\mathbf{B}_{1:T}, \mathbf{z}_{1:T})$ indicates the t-th output from mask generator. The instance-wise loss is applied for each of T instance-wise masks, and aggregated over all instances as $\mathcal{L}_{\text{inst}} = (1/T)\sum_t \mathcal{L}_{\text{inst}}^{(t)}$.

On the other hand, the global loss encourages all the instance-wise masks form a globally coherent context. To consider relation between different objects, we aggregate them into a global mask[1] $G_{\text{global}}(\mathbf{B}_{1:T}, \mathbf{z}_{1:T}) = \sum_t G_{\text{mask}}^{(t)}(\mathbf{B}_{1:t}, \mathbf{z}_{1:t})$, and compute an global adversarial loss analogous to Eq. (6) as

$$\begin{aligned}\mathcal{L}_{\text{global}} = \mathbb{E}_{(\mathbf{B}_{1:T}, M_{1:T})}\Big[\log D_{\text{global}}\big(\mathbf{B}_{\text{global}}, M_{\text{global}}\big)\Big] \\ + \mathbb{E}_{\mathbf{B}_{1:T}, \mathbf{z}_{1:T}}\Big[\log\Big(1 - D_{\text{global}}\big(\mathbf{B}_{\text{global}}, G_{\text{global}}(\mathbf{B}_{1:T}, \mathbf{z}_{1:T})\big)\Big)\Big],\end{aligned} \tag{7}$$

where $M_{\text{global}} \in \mathbb{R}^{H\times W}$ is an aggregated mask obtained by taking element-wise addition over $M_{1:T}$, and $\mathbf{B}_{\text{global}} \in \mathbb{R}^{H\times W\times L}$ is an aggregated bounding box tensor obtained by taking element-wise maximum over $\mathbf{B}_{1:T}$.

Finally, we additionally impose a reconstruction loss $\mathcal{L}_{\text{rec}}$ that encourages the predicted instance masks to be similar to the ground-truths. We implement this idea using perceptual loss [11, 3, 31, 2], which measures the distance of real and fake images in the feature space of a pre-trained CNN by

$$\mathcal{L}_{\text{rec}} = \sum_l \big\|\Phi_l(G_{\text{global}}) - \Phi_l(M_{\text{global}})\big\|, \tag{8}$$

where Φ_l is the feature extracted from the l-th layer of a CNN. We use the VGG-19 network [26] pre-trained on ImageNet [5] in our experiments. Since our input to the pre-trained network is a binary mask, we replicate masks to channel dimension and use the converted mask to compute Eq. (8). We found that using the perceptual loss improves the stability of GAN training and the quality of object shapes, as discussed in [3, 31, 2].

Combining Eq.(6), (7) and (8), the overall training objective for the shape generator becomes

$$\mathcal{L}_{\text{shape}} = \lambda_i\mathcal{L}_{\text{inst}} + \lambda_g\mathcal{L}_{\text{global}} + \lambda_r\mathcal{L}_{\text{rec}}, \tag{9}$$

where λ_i, λ_g and λ_r are hyper-parameters that balance different losses, which are set to 1, 1 and 10 in the experiment, respectively.

[1] G_{global} is computed by summation to model overlaps between objects.

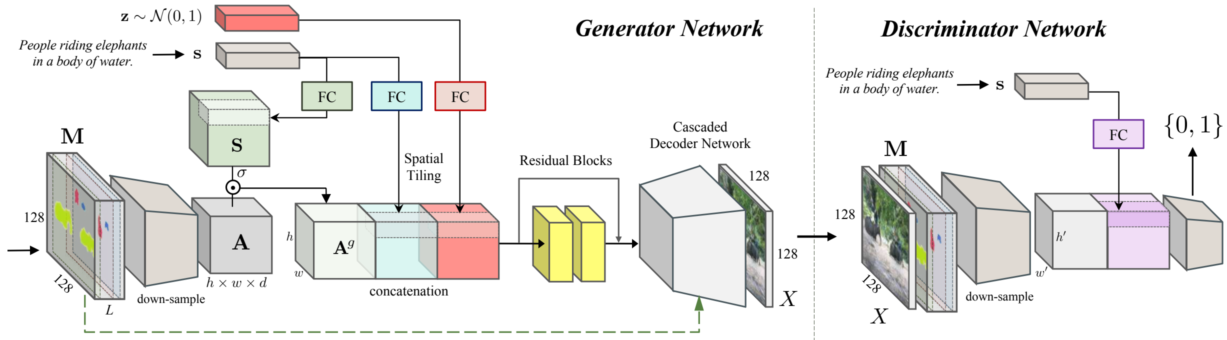


Figure 3. Architecture of the image generator. Conditioned on the text description and the semantic layout generated by the layout generator, it generates an image that matches both inputs.

5. Synthesizing Images from Text and Layout

The outputs from the layout generator define location, size, shape and class information of objects, which provide semantic structure of a scene relevant to the text. Given the semantic structure and text, the objective of the image generator is to generate an image that conforms to both conditions. To this end, we first aggregate binary object masks $M_{1:T}$ to a semantic label map $\mathbf{M} \in \{0,1\}^{H\times W\times L}$, such that $\mathbf{M}_{ijk} = 1$ if and only if there exists an object of class k whose mask M_t covers the pixel (i,j). Then, given the semantic layout $\mathbf{M}$ and the text $\mathbf{s}$, the image generator is defined by

$$\widehat{X} = G_{\text{img}}(\mathbf{M}, \mathbf{s}, \mathbf{z}), \tag{10}$$

where $\mathbf{z} \sim \mathcal{N}(0, I)$ is a random noise. In the following, we describe the network architecture and training procedures of the image generator.

Model. Figure 3 illustrates the overall architecture of the image generator. Our generator network is based on a convolutional encoder-decoder network [10] with several modifications. It first encodes the semantic layout $\mathbf{M}$ through several down-sampling layers to construct a layout feature $\mathbf{A} \in \mathbb{R}^{h\times w\times d}$. We consider that the layout feature encodes various context information of the input layout along the channel dimension. To adaptively select a context relevant to the text, we apply attention to the layout feature. Specifically, we compute a d-dimensional vector from the text embedding, and spatially replicate it to construct $\mathbf{S} \in \mathbb{R}^{h\times w\times d}$. Then we apply gating on the layout feature by $\mathbf{A}^g = \mathbf{A} \odot \sigma(\mathbf{S})$, where σ is the sigmoid nonlinearity, and $\odot$ denotes element-wise multiplication. To further encode text information on background, we compute another text embedding with separate fully-connected layers and spatially replicate it to size $h \times w$. The gated layout feature $\mathbf{A}^g$, the text embedding and noises are then combined by concatenation along channel dimension, and subsequently fed into several residual blocks and decoder to be mapped to an image. We employ a cascaded network [3] for the decoder, which takes the semantic layout $\mathbf{M}$ as an additional input to every upsampling layer. We found that the cascaded network enhances conditioning on layout structure and produces better object boundary.

For the discriminator network D_{img}, we first concatenate the generated image X and the semantic layout $\mathbf{M}$. It is fed through a series of down-sampling blocks, resulting in a feature map of size $h' \times w'$. We concatenate it with a spatially tiled text embedding, from which we compute a decision score of the discriminator.

Training. Conditioned on both the semantic layout $\mathbf{M}$ and the text embedding $\mathbf{s}$ extracted by [20], the image generator G_{img} is jointly trained with the discriminator D_{img}. We define the objective function by $\mathcal{L}_{\text{img}} = \lambda_a \mathcal{L}_{\text{adv}} + \lambda_r \mathcal{L}_{\text{rec}}$, where

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{(\mathbf{M},\mathbf{s},X)}\Big[\log D_{\text{img}}\big(\mathbf{M},\mathbf{s},X\big)\Big] \tag{11}$$
$$+ \mathbb{E}_{(\mathbf{M},\mathbf{s}),\mathbf{z}}\Big[\log\Big(1 - D_{\text{img}}\big(\mathbf{M},\mathbf{s},G_{\text{img}}(\mathbf{M},\mathbf{s},\mathbf{z})\big)\Big)\Big],$$

$$\mathcal{L}_{\text{rec}} = \sum_l \|\Phi_l(G_{\text{img}}(\mathbf{M},\mathbf{s},\mathbf{z})) - \Phi_l(X)\|, \tag{12}$$

where X is a ground-truth image associated with semantic layout $\mathbf{M}$. As in the mask generator, we apply the same perceptual loss $\mathcal{L}_{\text{rec}}$, which is found to be effective. We set the hyper-parameters $\lambda_a = 1$, $\lambda_r = 10$ in our experiment.

6. Experiments

6.1. Experimental Setup

Dataset. We use the MS-COCO dataset [13] to evaluate our model. It contains 164,000 training images over 80 semantic classes, where each image is associated with instance-wise annotations (*i.e.*, object bounding boxes and segmentation masks) and 5 text descriptions. The dataset has complex scenes with many objects in a diverse context, which makes generation very challenging. We use the official train and validation splits from MS-COCO 2014 for training and evaluating our model, respectively.

Evaluation metrics. We evaluate text-conditional image generation performance using various metrics: Inception score, caption generation, and human evaluation.

Method	Box	Mask	Caption generation						Inception [24]
			BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	
Reed *et al.* [21]	-	-	0.470	0.253	0.136	0.077	0.122	0.160	7.88 ± 0.07
StackGAN [34]	-	-	0.492	0.272	0.152	0.089	0.128	0.195	8.45 ± 0.03
Ours	Pred.	Pred.	**0.541**	**0.332**	**0.199**	**0.122**	**0.154**	**0.367**	**11.46 ± 0.09**
Ours (control experiment)	GT	Pred.	0.556	0.353	0.219	0.139	0.162	0.400	11.94 ± 0.09
	GT	GT	0.573	0.373	0.239	0.156	0.169	0.440	12.40 ± 0.08
Real images (upper bound)	-	-	0.678	0.496	0.349	0.243	0.228	0.802	-

Table 1. Quantitative evaluation results. Two evaluation metrics based on caption generation and the Inception score are presented. The second and third columns indicate types of bounding box or mask layout used in image generation, where "GT" indicates ground-truth and "Pred." indicates predicted one by our model. The last row presents the caption generation performance on real images, which corresponds to upper-bound of caption generation metric. Higher is better in all columns.

Figure 4. Qualitative examples of generated images conditioned on text descriptions on the MS-COCO validation set, using our method and baselines (StackGAN [34] and Reed *et al*. [21]). The input text and ground-truth image are shown in the first row. For each method, we provide a reconstructed caption conditioned on the generated image.

Method	ratio of ranking 1st	*vs.* Ours
StackGAN [34]	18.4 %	29.5 %
Reed *et al*. [21]	23.3 %	32.3 %
Ours	**58.3** %	-

Table 2. Human evaluation results.

Inception score — We compute the Inception score [24] by applying pre-trained classifier on synthesized images and investigating statistics of their score distributions. It measures recognizability and diversity of generated images, and has been known to be correlated with human perceptions on visual quality [18]. We use the Inception-v3 [27] network pre-trained on ImageNet [5] for evaluation, and measure the score for all validation images.

Caption generation — In addition to the Inception score, assessing the performance of text-conditional image generation necessitates measuring the relevance of the generated images to the input texts. To this end, we generate sentences from the synthesized image and measure the similarity between input text and predicted sentence. The underlying intuition is that if the generated image is relevant to input text and its contents are recognizable, one should be able to guess the original text from the synthesized image. We employ an image caption generator [30] trained on MS-COCO to generate sentences, where one sentence is generated per image by greedy decoding. We report three standard language similarity metrics: BLEU [19], METEOR [1] and CIDEr [28].

Human evaluation — Evaluation based on caption generation is beneficial for large-scale evaluation but may introduce unintended bias by the caption generator. To verify the effectiveness of caption-based evaluation, we conduct human evaluation using Amazon Mechanical Turk. For each text randomly selected from MS-COCO validation set, we presented 5 images generated by different methods, and asked users to rank the methods based on the relevance of generated images to text. We collected results for 1000 sentences, each of which is annotated by 5 users. We report results based on the ratio of each method ranked as the best, and one-to-one comparison between ours and the baselines.

6.2. Quantitative Analysis

We compare our method with two state-of-the-art approaches [21, 34] based on conditional GANs. Table 1 and Table 2 summarizes the quantitative evaluation results.

Comparisons to other methods. We first present systemic evaluation results based on Inception score and cap-

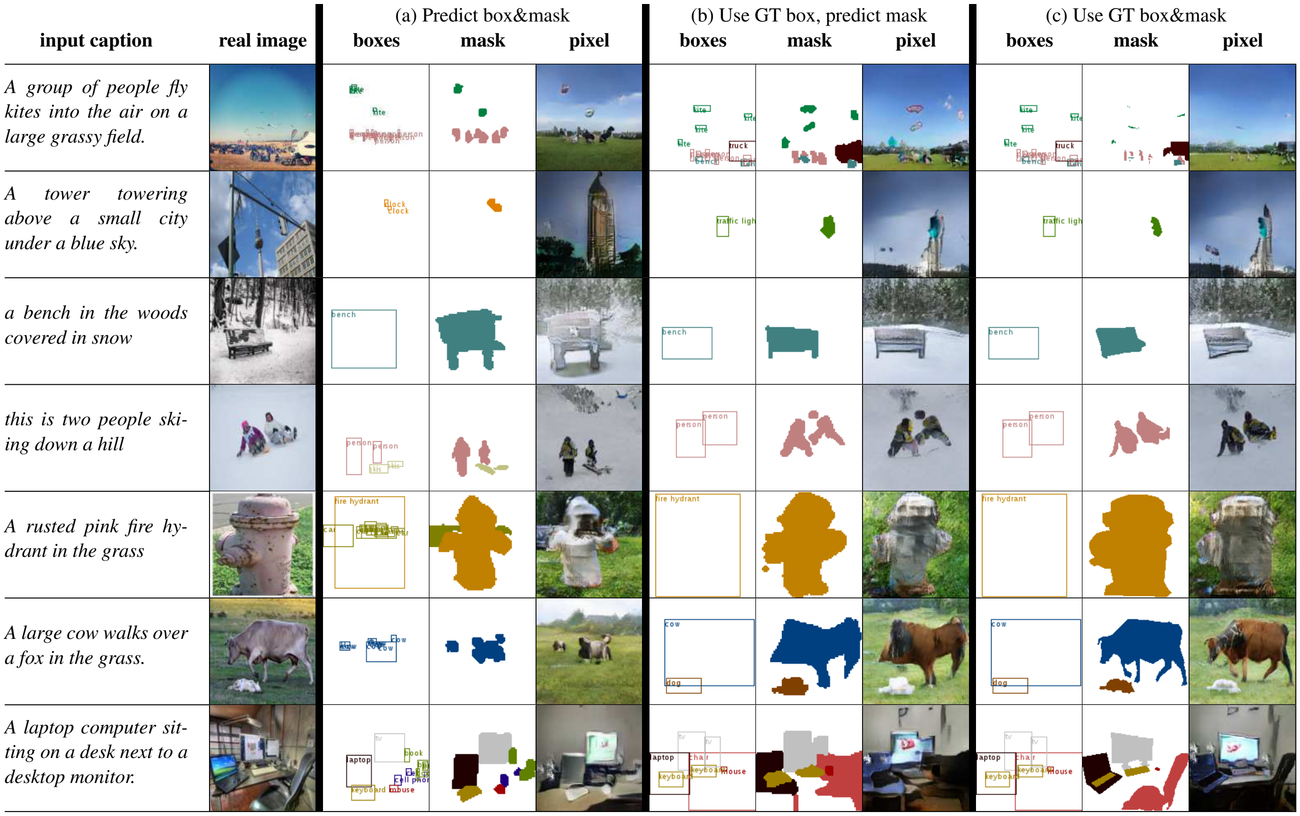


Figure 5. Image generation results of our method. Each column corresponds to generation results conditioned on (a) predicted box and mask layouts, (b) ground-truth box and predicted mask layout and (c) ground-truth box and mask layouts. Classes are color-coded for illustration purpose. Best viewed in color.

tion generation performance. The results are summarized in Table 1. The proposed method substantially outperforms existing approaches based on both evaluation metrics. In terms of Inception score, our method outperforms the existing approaches with a substantial margin, presumably because our method generates more recognizable objects. Caption generation performance shows that captions generated from our synthesized images are more strongly correlated with the input text than the baselines. This shows that images generated by our method are better aligned with descriptions and are easier to recognize semantic contents.

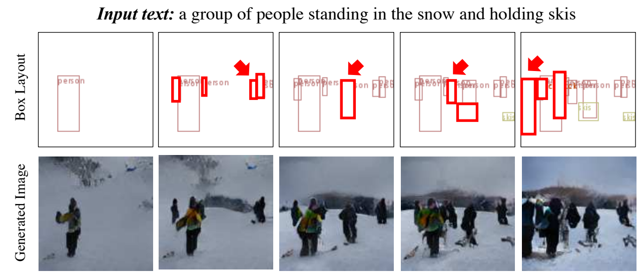
Table 2 summarizes comparison results based on human evaluation. When users are asked to rank images based on their relevance to input text, they choose images generated by our method as the best in about 60% of all presented sentences, which is substantially higher than baselines (about 20%). This is consistent with the caption generation results in Table 1, in which our method substantially outperforms the baselines while their performances are comparable.

Figure 4 illustrates qualitative comparisons. Due to adversarial training, images generated by the other methods, especially StackGAN [34], tend to be clear and exhibits high-frequency details. However, it is difficult to recognize contents from the images, since they often fail to predict important semantic structure of object and scene. As a result, the reconstructed captions from the generated images are usually not relevant to the input text. Compared to them, our method generates much more recognizable and semantically meaningful images by conditioning the generation with the inferred semantic layout, and is able to reconstruct descriptions that better align with the input sentences.

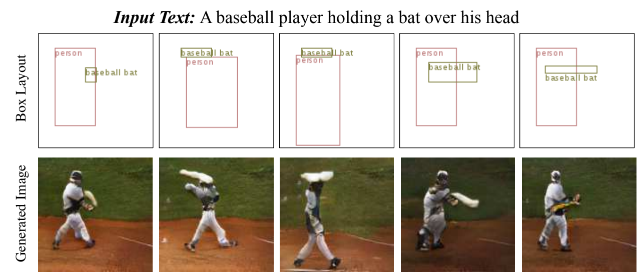
Figure 6. Multiple samples generated from a text description.

Ablative Analysis. To understand quality and the impact of the predicted semantic layout, we conduct an ab-

Figure 7. Generation results by manipulating captions. The manipulated parts of texts are highlighted in **bold** characters, where the types of manipulation is indicated by different colors. **Blue**: scene context, **Magenta**: spatial location, **Red**: the number of objects, **Green**: object category.



(a) Generation results by adding new objects.



(b) Generation results by changing spatial configuration of objects.

Figure 8. Examples of controllable image generation.

lation study by gradually replacing the bounding box and mask layouts predicted by layout generator with the ground-truths. Table 1 summarizes quantitative evaluation results. As it shows, replacing the predicted layouts to ground-truths leads with gradual performance improvements, which shows predictions errors in both bounding box and mask layouts.

6.3. Qualitative Analysis

Figure 5 shows qualitative results of our method. For each text, we present the generated images alongside the predicted semantic layouts. As in the previous section, we also present our results conditioned on ground-truth layouts. As it shows, our method generates reasonable semantic layout and image matching the input text; it generates bounding boxes corresponding to fine-grained scene structure implied in texts (*i.e*. object categories, the number of objects), and object masks capturing class-specific visual attributes as well as relation to other objects. Given the inferred layouts, our image generator produces correct object appearances and background compatible with text. Replacing the predicted layouts with ground-truths makes the generated images to have a similar context to original images.

Diversity of samples. To assess the diversity in the generation, we sample multiple images while fixing the input text. Figure 6 illustrates the example images generated by our method. Our method generates diverse semantic structures given the same text description, while preserving semantic details such as the number of objects and object categories.

Text-conditional generation. To see how our model incorporates text description in generation process, we generate images while modifying parts of the descriptions. Figure 7 illustrates the example results. When we change the context of descriptions such as object class, number of objects, spatial composition of objects and background patterns, our method correctly adapts semantic structure and images based on the modified part of the text.

Controllable image generation. We demonstrate controllable image generation by modifying bounding box layout. Figure 8 illustrates the example results. Our method updates object shapes and context based on the modified semantic layout (*e.g*. adding new objects, changing spatial configuration of objects) and generates reasonable images.

7. Conclusion

We proposed an approach for text-to-image synthesis which explicitly infers and exploits a semantic layout as an intermediate representation from text to image. Our model hierarchically constructs a semantic layout in a coarse-to-fine manner by a series of generators. By conditioning image generation on explicit layout prediction, our method generates complicated images that preserve semantic details and highly relevant to the text description. We also showed that the predicted layout can be used to control generation process. We believe that end-to-end training of layout and image generation would be interesting future work.

Acknowledgments This work was supported in part by ONR N00014-13-1-0762, NSF CAREER IIS-1453651, DARPA Explainable AI (XAI) program #313498, and Sloan Research Fellowship.

References

[1] S. Banerjee and A. Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL*, 2005. 6

[2] M. Cha, Y. Gwon, and H. T. Kung. Adversarial nets with perceptual losses for text-to-image synthesis. *arXiv preprint arXiv:1708.09321*, 2017. 4

[3] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 2, 4, 5

[4] 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. 1, 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4, 6

[6] H. Dong, J. Zhang, D. McIlwraith, and Y. Guo. I2t2i: Learning text to image synthesis with textual data augmentation. *ICIP*, 2017. 1, 2

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. In *NIPS*, 2014. 2, 4

[8] D. Ha and D. Eck. A Neural Representation of Sketch Drawings. *arXiv preprint arXiv:1704.03477*, 2017. 3

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 3

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 5

[11] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*, 2016. 4

[12] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *CoRR*, 2016. 2

[13] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. . L. Zitnick, and P. Dollár. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 1, 2, 5

[14] E. Mansimov, E. Parisotto, and J. Ba. Generating images from captions with attention. In *ICLR*, 2016. 2

[15] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 4

[16] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. In *CVPR*, 2017. 2

[17] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 1, 2

[18] A. Odena, C. Olah, and J. Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *ICML*, 2017. 6

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*, 2002. 6

[20] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 5

[21] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 1, 2, 3, 6

[22] S. Reed, A. Oord, N. Kalchbrenner, S. Gómez, Z. Wang, D. Belov, and N. Freitas. Parallel multiscale autoregressive density estimation. In *ICML*, 2017. 2

[23] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *NIPS*, 2016. 1, 2

[24] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. In *NIPS*, 2016. 6

[25] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *NIPS*, 2015. 4

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 6

[28] R. Vedantam, C. L. Zitnick, and D. Parikh. CIDEr: Consensus-based Image Description Evaluation. In *CVPR*, 2015. 6

[29] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017. 2

[30] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 6

[31] C. Wang, C. Xu, C. Wang, and D. Too. Perceptual adversarial networks for image-to-image transformation. *arXiv preprint arXiv:1706.09138*, 2017. 4

[32] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016. 2

[33] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 1, 2

[34] 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, 6, 7