BachGAN: High-Resolution Image Synthesis from Salient Object Layout

Yandong Li[1*] Yu Cheng[2] Zhe Gan[2] Licheng Yu[2] Liqiang Wang[1] Jingjing Liu[2]
[1]University of Central Florida [2]Microsoft Dynamics 365 AI Research
{liyandong,lwang}@ucf.edu, {yu.cheng,zhe.gan,licheng.yu,jingjl}@microsoft.com

Figure 1: Top row: images synthesized from semantic segmentation maps. Bottom row: high-resolution images synthesized from salient object layouts, which allows users to create an image by drawing only a few bounding boxes.

Abstract

We propose a new task towards more practical application for image generation - high-quality image synthesis from salient object layout. This new setting allows users to provide the layout of salient objects only (i.e., foreground bounding boxes and categories), and lets the model complete the drawing with an invented background and a matching foreground. Two main challenges spring from this new task: (i) how to generate fine-grained details and realistic textures without segmentation map input; and (ii) how to create a background and weave it seamlessly into standalone objects. To tackle this, we propose Background Hallucination Generative Adversarial Network (BachGAN), which first selects a set of segmentation maps from a large candidate pool via a background retrieval module, then encodes these candidate layouts via a background fusion module to hallucinate a suitable background for the given objects. By generating the hallucinated background representation dynamically, our model can synthesize high-resolution images with both photo-realistic foreground and integral background. Experiments on Cityscapes and ADE20K datasets demonstrate the advantage of BachGAN over existing methods, measured on both visual fidelity of generated images and visual alignment between output images and input layouts.[1]

1. Introduction

Pablo Picasso once said "*Every child is an artist. The problem is how to remain an artist once grown up.*" Now with the help of smart image editing assistant, our creative and imaginative nature can well flourish. Recent years have witnessed a wide variety of image generation works conditioned on diverse inputs, such as text [38, 36], scene graph [13], semantic segmentation map [11, 33], and holistic layout [39]. Among them, text-to-image generation provides a flexible interface for users to describe visual concepts via natural language descriptions [38, 36]. The limitation is that one single sentence may not be adequate for describing the details of every object in the intended image.

Scene graph [13], with rich structural representation, can potentially reveal more visual relations of objects in an image. However, pairwise object relation labels are difficult to obtain in real-life applications. The lack of object size, location and background information also limits the quality of synthesized images.

Another line of research is image synthesis conditioned on semantic segmentation map. While previous work [11, 33, 25] has shown promising results, collecting annotations for semantic segmentation maps is time consuming and labor intensive. To save annotation effort, Zhao et al. [39] proposed to take as the input a holistic layout including both foreground objects (*e.g.*, "cat", "person") and background (*e.g.*, "sky", "grass"). In this paper, we push this direction to a further step and explore image synthesis given salient ob-

[1]Project page: https://github.com/Cold-Winter/BachGAN.
* This work was done while the first author was an intern at Microsoft.

ject layout only, with just coarse foreground[3] object bounding boxes and categories. Figure 1 provides a comparison between segmentation-map-based image synthesis (top row) and our setting (bottom row). Our task takes foreground objects as the only input, without any background layout or pixel-wise segmentation map.

The proposed new task presents new challenges for image synthesis: (i) how to generate fine-grained details and realistic textures with only a few foreground object bounding boxes and categories; and (ii) how to invent a realistic background and weave it into the standalone foreground objects seamlessly. Note that no knowledge about the background is provided; while in [39], a holistic layout is provided and only low-resolution (64×64) images are generated. In our task, the goal is to synthesize high-resolution (512×256) images given very limited information (salient layout only).

To tackle these challenges, we propose *Background Hallucination Generative Adversarial Network* (BachGAN).Given a salient object layout, BachGAN generates an image via two steps: (i) a background retrieval module selects from a large candidate pool a set of segmentation maps most relevant to the given object layout; (ii) these candidate layouts are then encoded via a background fusion module to hallucinate a best-matching background. With this retrieval-and-hallucination approach, BachGAN can dynamically provide detailed and realistic background that aligns well with any given foreground layout. In addition, by feeding both foreground objects and background representation into a conditional GAN (via a SPADE normalization layer [25]), BachGAN can generate high-resolution images with visually consistent foreground and background.

Our contributions are summarized as follows:

- We propose a new task - image synthesis from salient object layout, which allows users to draw an image by providing just a few object bounding boxes.
- We present BachGAN, the key components of which are a retrieval module and a fusion module, which can hallucinate a visually consistent background on-the-fly for any foreground object layout.
- Experiments on Cityscapes [4] and ADE20K [40] datasets demonstrate our model's ability to generate high-quality images, outperforming baselines measured on both visual quality and consistency metrics.

2. Related Work

Conditional Image Generation Conditional image synthesis tasks can facilitate diverse inputs, such as source image [11, 20, 26, 41, 42], sketch [27, 42, 34], scene graph [13, 1], text [21, 38, 36, 18, 19], video clip [17, 6, 29], and dialogue [28, 3]. These approaches fall into three main categories: Generative Adversarial Networks (GANs) [7, 22], Variational Autoencoders (VAEs) [15], and autoregressive models [31, 23]. Our proposed model is a GAN framework aiming for image generation from salient layout only, which is a new task.

In previous studies, the layout is typically treated as an intermediate representation between the input source (*e.g.*, text [9, 18] or scene graph [13]) and the output image. Instead of learning a direct mapping from text/scene graph to an image, the model constructs a semantic layout (including bounding boxes and object shapes), based on which the target image is generated. Well-labeled instance segmentation maps are required to train the object shape generator. There is also prior work that aims to synthesize photo-realistic images directly from semantic segmentation maps [33, 11]. However, obtaining detailed segmentation maps for large-scale datasets is time consuming and labor intensive. In [16], to avoid relying on instance segmentation mask as the key input, additional background layout and object layout are used as the input. [39] proposed the task of image synthesis from object layout; however, both foreground and background object layouts are required, and only low-resolution images are generated. Different from these studies, we propose to synthesize images from salient object layout only, which is more practical in real-life application where the user can simply draw the outlines of intended objects.

High-Resolution Image Synthesis Adversarial learning has been applied to image-to-image translation [11, 32], to convert an input image from one domain to another using image pairs as training data. L_1 loss [12] and adversarial loss [7] are popular choices for many image-to-image translation tasks.

Recently, Chen and Koltun [2] suggest that it might be difficult for conditional GANs to generate high-resolution images due to training instability and optimization issues. To circumvent this, they use a direct regression objective based on a perceptual loss [5] and produce the first model that can synthesize high-quality images. Motivated by this, pix2pix-HD [33] uses a robust adversarial learning objective together with a new multi-scale generator-discriminator architecture to improve high-resolution generation performance. In [32], high-resolution video-to-video synthesis are explored to model temporal dynamics. Park et al. [25] shows that spatially-adaptive normalization (SPADE), a conditional normalization layer that modulates the activations using input semantic layouts, can synthesize images significantly better than state-of-the-art methods.

However, the input of all the aforementioned approaches is still semantic segmentation map. In our work, we adopt the SPADE layer in our generator, but only use a salient object layout as the conditional input. This foreground layout

[3] *Salient* and *foreground* are used interchangeably in this paper.

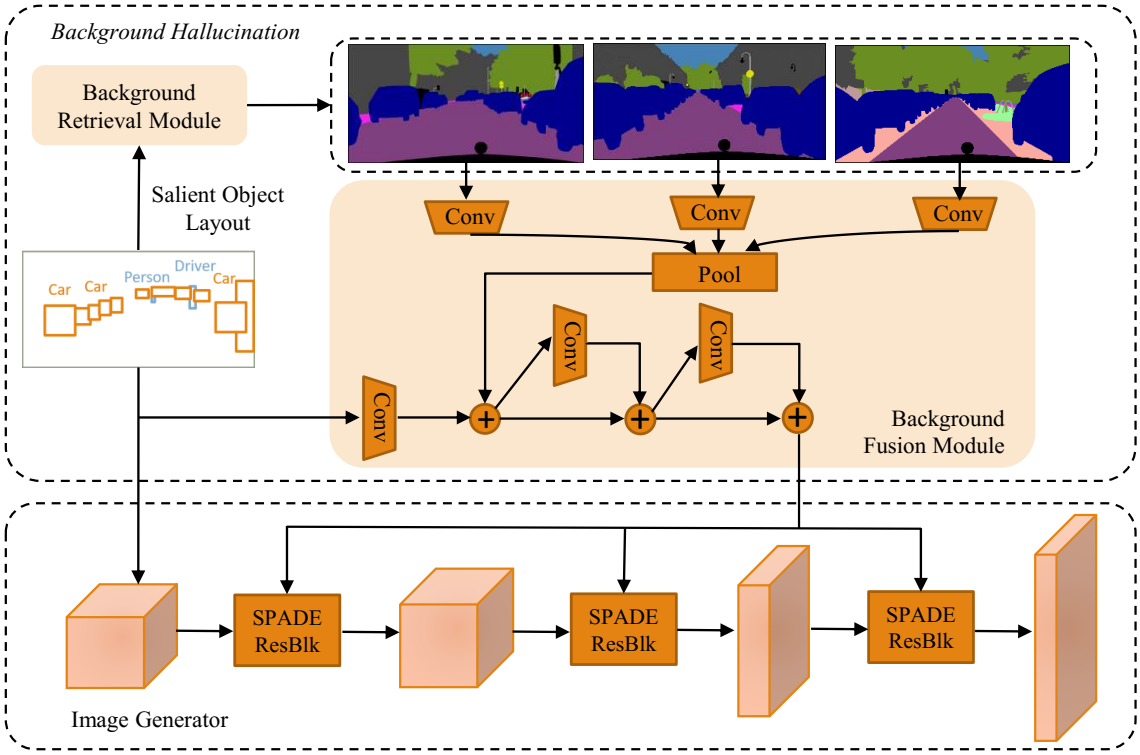

Figure 2: Overview of the proposed BachGAN for image synthesis from salient object layout.

is combined with hallucinated background to obtain a fused representation, which is then fed into the SPADE layer for image generation.

3. BachGAN

We first define the problem formulation and introduce preliminaries in Sec. 3.1, before presenting the proposed Background Hallucination Generative Adversarial Network (BachGAN). As illustrated in Figure 2, BachGAN consists of three components: (*i*) *Background Retrieval Module* (Sec. 3.2), which selects a set of segmentation maps from a large candidate pool given a foreground layout; (*ii*) *Background Fusion Module* (Sec. 3.3), which fuses the salient object layout and the selected candidate into a feature map for background hallucination; and (*iii*) *Image Generator*, which adopts a SPADE layer [25] to generate an image based on the fused representation. Discriminators are omitted in Figure 2 for simplicity.

3.1. Problem Formulation and Preliminaries

Problem Definition Assume we have a set of images $\mathcal{I}$ and their corresponding salient object layouts $\mathcal{L}$. The goal is to train a model that learns a mapping from layouts to images, *i.e.*, $\mathcal{L} \rightarrow \mathcal{I}$. Specifically, given a ground-truth image $\mathbf{I} \in \mathcal{I}$ and its corresponding layout $\mathbf{L} \in \mathcal{L}$, where $\mathbf{L}_i = (x_i, y_i, h_i, w_i)$ denotes the top-left coordinates plus the height and width of the i-th bounding box. Following [25, 33], we first convert $\mathbf{L}$ into a label map $\mathbf{M} \in \{0,1\}^{H \times W \times C_o}$, where C_o denotes the number of categories, and H, W are the height and width of the label map, respectively. Different from the semantic segmentation map used in [33], some pixels $\mathbf{M}(i,j)$ can be assigned to n object instances, *i.e.*, $\mathbf{M}(i,j) \in \{0,1\}^{C_o}$ *s.t.* $\sum_p \mathbf{M}(i,j,p) = n$.

A Naive Solution To draw in the motivation of our framework, we first consider a simple conditional GAN model and discuss its limitations. By considering the label map $\mathbf{M}$ as the input image, image-to-image translation models can be directly applied with the following objective:

$$\min_G \max_D \mathbb{E}_{\mathbf{M},\mathbf{I}}[\log(D(\mathbf{M},\mathbf{I}))] + \mathbb{E}_{\mathbf{M}}[\log(1 - D(\mathbf{M}, G(\mathbf{M})))], \quad (1)$$

where G and D denote the generator and the discriminator, respectively. The generator $G(\cdot)$ takes a label map $\mathbf{M}$ as input to generate a fake image.

State-of-the-art conditional GANs, such as pix2pixHD [33], can be directly applied here. However, some caveats can be readily noticed, as only a coarse foreground layout is provided in our setting, making the generation task much more challenging than when a semantic segmentation map is provided. Thus, we introduce Background Hallucination to address this issue in the following sub-section.

3.2. Background Retrieval Module

The main challenge in this new task is how to generate a proper background to fit the foreground objects. Given an object layout $\mathbf{L}$ that contains k instances: $\mathbf{L}_0^{C_0}, \ldots, \mathbf{L}_k^{C_k}$, where C_i is the category of instance $\mathbf{L}_i$, assume we have a memory bank $\mathbf{B}$ containing pairs of image $\mathbf{I}$ and its fine-grained semantic segmentation map $\mathbf{S}$ with l instances: $\mathbf{S}_0^{C_0}, ..., \mathbf{S}_l^{C_l}$. We first retrieve a pair of $\mathbf{I}$ and $\mathbf{S}$ that contain the most similar layout to $\mathbf{L}$, by using a layout-similarity score, a variant of the Intersect over Union (IoU) metric, to measure the distance between a salient object layout and a fine-grained semantic segmentation map:

$$\text{IoU}_r = \frac{\sum_{j=1}^{C} \mathbf{S}^j \bigcap \mathbf{L}^j}{\sum_{j=1}^{C} \mathbf{S}^j \bigcup \mathbf{L}^j}, \tag{2}$$

where C is the total number of object categories, $\mathbf{S}^j = \bigcup_i \mathbf{S}_i^j$, and $\mathbf{L}^j = \bigcup_i \mathbf{L}_i^j$. $\bigcup$ and $\bigcap$ denote union and intersect, respectively. The proposed metric can preserve the overall location and category information of each object, since the standard IoU score is designed for measuring the quality of object detection. However, instead of calculating the mean IoU scores across all the classes, we use Eqn. (2) to prevent the weights of small objects from growing too high.

Given a salient object layout $\mathbf{L}_q$ as the query, we rank the pairs of image and semantic segmentation map in the memory bank by the aforementioned layout-similarity score. As a result, we can obtain a retrieved image $\mathbf{I}_r$ with semantic segmentation map $\mathbf{S}_r$, which has a salient object layout most similar to the query $\mathbf{L}_q$. The assumption is that *images with a similar foreground composition may share similar background as well*. Therefore, we treat the retrieved semantic segmentation map, $\mathbf{S}_r$, as the potential background for $\mathbf{L}_q$. Formally, we first convert the background of $\mathbf{S}_r$ into a label map $\mathbf{M}_b$: $\mathbf{M}_b(i,j) \in \{0,1\}^{C_b}$ *s.t.* $\sum_p \mathbf{M}_b(i,j,p) = 1$, where C_b denotes the number of categories in the background. Then, we produce a new label map, encoding both the foreground object layout and the fine-grained background segmentation map, by concatenating $\mathbf{M}_b$ and the foreground label map $\mathbf{M}_q$ of $\mathbf{L}_q$:

$$\hat{\mathbf{M}} = [\mathbf{M}_b; \mathbf{M}_q], \tag{3}$$

where $[;]$ denotes concatenation, and the resulting label map is represented as $\hat{\mathbf{M}} \in \{0,1\}^{H \times W \times (C_o + C_b)}$. Note that the memory bank $\mathbf{B}$ can be much smaller than the scale of training data. Therefore, this approach does not require expensive pairwise segmentation map annotations to generate high-quality images.

A Simple Baseline Based on the obtained new label map $\hat{\mathbf{M}}$, now we describe a simple baseline that motivates our proposed model. We consider the new label map as an input image, and adopt an image-to-image translation model. The following conditional GAN loss can be used for training:

$$\min_G \max_D \mathbb{E}_{\hat{\mathbf{M}},\mathbf{I}_r}[\log(D(\hat{\mathbf{M}}, \mathbf{I}_r))] + \mathbb{E}_{\hat{\mathbf{M}},\mathbf{I}_q}[\log(D(\hat{\mathbf{M}}, \mathbf{I}_q))] + \mathbb{E}_{\hat{\mathbf{M}}}[\log(1 - D(\hat{\mathbf{M}}, G(\hat{\mathbf{M}})))], \tag{4}$$

where $\mathbf{I}_q$ is the ground-truth image corresponding to the query $\mathbf{L}_q$, and $\mathbf{I}_r$ is the retrieved image.[4] We name this baseline method "*BachGAN-r*". Though the ground-truth background cannot be obtained, the use of the retrieved background injects useful information that helps the generator synthesize better images, compared to the objective in Eqn. (1).

3.3. Background Fusion Module

Though the retrieval-based baseline approach can hallucinate a background given a foreground object layout, the relevance of one retrieved semantic segmentation map to the input foreground layout is not guaranteed. One possible solution is to use multiple retrieved segmentation maps in Eqn. (4). However, this renders training unstable as the discriminator becomes unbalanced when several retrieved images are included in the loss function. More importantly, the dimension of the input label map is too high. In order to utilize multiple retrieved segmentation maps for a fuzzy hallucination of the background, we further introduce a Background Fusion Module to encode Top-m retrieved segmentation maps to hallucinate a smoother background.

Assume we obtain m retrieved segmentation maps $\mathbf{S}_{r,0}, ..., \mathbf{S}_{r,m}$ with their corresponding background label maps $\mathbf{M}_{b,0}, ..., \mathbf{M}_{b,m}$. The query salient object layout $\mathbf{L}_q$ has a corresponding label map $\mathbf{M}_q$, where $\mathbf{M}_{b,i} \in \{0,1\}^{H \times W \times C_b}$ and $\mathbf{M}_q \in \{0,1\}^{H \times W \times C_o}$. As illustrated in Figure 2, we first obtain $\hat{\mathbf{M}}_{r,0}, ..., \hat{\mathbf{M}}_{r,m}$ with Eqn. (3), where $\hat{\mathbf{M}}_{r,i} \in \{0,1\}^{H \times W \times (C_o + C_b)}$. $\mathbf{M}_q$ is padded with $\mathbf{0}$ to obtain a query label map $\hat{\mathbf{M}}_q$ with the same shape as $\hat{\mathbf{M}}_{r,i}$. $\hat{\mathbf{M}}_{r,0}, ..., \hat{\mathbf{M}}_{r,m}$ are then concatenated into $\hat{\mathbf{M}}_r \in \{0,1\}^{m \times H \times W \times (C_o + C_b)}$. A convolutional network $\mathcal{F}$ is then used to encode the label maps into feature maps:

$$\mathbf{m}_0 = \mathcal{F}(\hat{\mathbf{M}}_q) \oplus \text{Pool}(\mathcal{F}(\hat{\mathbf{M}}_r)), \tag{5}$$

where Pool represents average pooling, $\oplus$ denotes element-wise addition, and $\mathbf{m}_0 \in \mathbb{R}^{H \times W \times h}$ (h is the number of feature maps). We then use another convolutional network $\mathcal{M}$ to obtain updated feature maps:

$$\mathbf{m}_t = \mathbf{m}_{t-1} \oplus \mathcal{M}(\mathbf{m}_{t-1}). \tag{6}$$

After T steps, we obtain the final feature map $\hat{\mathbf{m}} = \mathbf{m}_T$, which contains information from both salient object layout and hallucinated background.

[4] Empirically, we observe that adding the retrieved image to the GAN loss improves the performance.

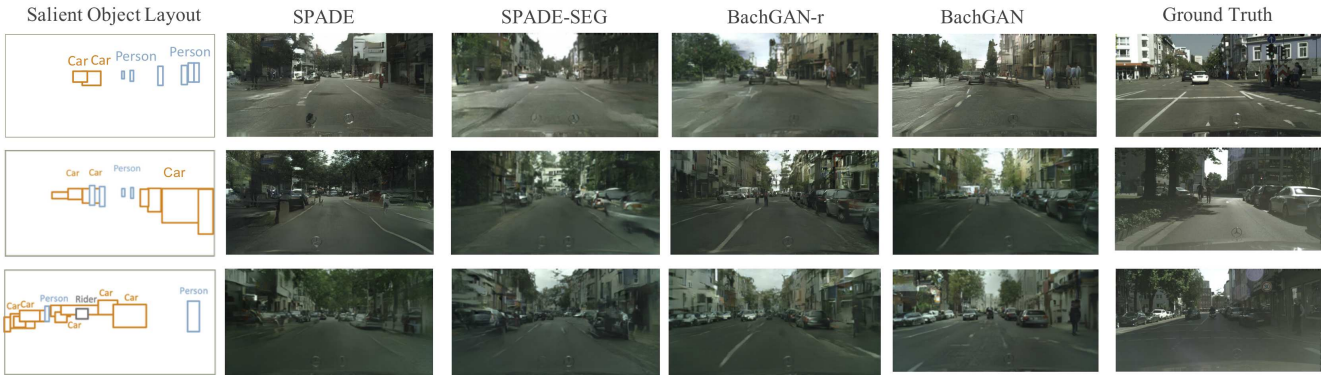

Figure 3: Examples of image synthesis results from different models on the Cityscapes dataset. Results from Layout2im are not included due to low resolution of the generated images (provided in Appendix).

Training Objective Based on the feature map $\hat{\mathbf{m}}$, BachGAN uses the following conditional GAN loss for training:

$$\min_G \max_D \mathbb{E}_{\hat{\mathbf{m}},\mathbf{I}_q}[\log(D(\hat{\mathbf{m}},\mathbf{I}_q))] + \mathbb{E}_{\hat{\mathbf{m}}}[\log(1-D(\hat{\mathbf{m}},G(\hat{\mathbf{m}})))]\,, \quad (7)$$

where $\mathbf{I}_q$ is the ground-truth image corresponding to the query $\mathbf{L}_q$. Compared with Eqn. (4), multiple retrieved segmentation maps are used to hallucinate the background, which leads to better performance in practice.

Image Generator Now, we describe how the generator $G(\cdot)$ takes $\hat{\mathbf{m}}$ as input to generate a high-quality image. In order to generate photo-realistic images, we utilize the spatially-adaptive normalization (SPADE) layer [25] in our generator. Let $\mathbf{h}_i$ denote the activation feature map of the i-th layer of the generator G. Similar to batch normalization [10], SPADE [25] first normalizes $\mathbf{h}_i$, then produces the modulation parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ to denormalize it, both of which are functions of $\hat{\mathbf{m}}$:

$$\hat{\mathbf{h}}_i = \text{norm}(\mathbf{h}_i) \otimes \boldsymbol{\gamma}(\hat{\mathbf{m}}) \oplus \boldsymbol{\beta}(\hat{\mathbf{m}})\,, \quad (8)$$

where $\hat{\mathbf{h}}_i$ denotes the output of a SPADE layer, norm $(\cdot)$ is a normalization operation, and $\otimes$ and $\oplus$ are element-wise production and addition, respectively. An illustration of the generator is provided in the bottom part of Figure 2. More details about SPADE can be found in [25].

4. Experiments

In this section, we describe experiments comparing BachGAN with state-of-the-art approaches on the new task, as well as detailed analysis that validates the effectiveness of our proposed model.

4.1. Experimental Setup

Datasets We conduct experiments on two public datasets: Cityscapes [4] and ADE20K [40]. Cityscapes contains images with street scene in cities. The size of training and the validation set is 3,000 and 500, respectively. We exclude 23 background classes and use the remaining 10 foreground objects in the salient object layout. With provided instance-level annotations, we can readily transform a semantic segmentation instance to its bounding box, by taking the max and min of the coordinates of each pixel in an instance. ADE20K consists of 20,210 training and 2,000 validation images. The dataset contains challenging scenes with 150 semantic classes. We exclude the 35 background classes and utilize the remaining 115 foreground objects. There are no instance-level annotations for ADE20k, thus, we use a simple approach to find contours [30] from a semantic segmentation map and then obtain the bounding box for each contour. A separate memory bank is used for each dataset. We train all the image synthesis methods on the same training set and report their results on the same validation set.

Baselines We include several strong baselines that can generate images with object layout as input:

- **SPADE**: We adopt SPADE [25] as our first baseline, taking as input the salient object layout instead of semantic segmentation map used in the original paper.
- **SPADE with Segmentation (SPADE-SEG)**: We obtain the second baseline by exploiting the pairs of segmentation mask and image from the memory bank. Besides GAN loss, the model is trained with an additional loss. It minimizes the segmentation loss between the real image and the output from the generator based on the memory bank.
- **Layout2im**: We use the code from Layout2im [39], which generates images from holistic layouts and supports the generation of 64×64 images only.

Performance metrics Following [2, 33], we run a semantic segmentation model on the synthesized images and measure the segmentation accuracy. We use state-of-the-art segmentation networks: DRN-D-105 [37] for Cityscapes,

Figure 4: Examples of image synthesis results from different models on the ADE20K dataset.

Figure 5: Generated images by adding bounding boxes sequentially to previous layout (Cityscapes).

Model	Cityscapes		ADE20K	
	Acc	FID	Acc	FID
Layout2im [39]	-	99.1	-	-
SPADE	57.6	86.7	55.3	59.4
SPADE-SEG	60.2	81.2	60.9	57.2
BachGAN-r	67.3	74.4	64.5	53.2
BachGAN	**70.4**	**73.3**	**66.8**	**49.8**
SPADE-v [25]	81.9†	71.8†	79.9†	33.9†

Table 1: Results on Cityscapes and ADE20K w.r.t. FID and the pixel accuracy (Acc). Results with (†) are reported in [25], serving as the upper bound of our model performance.

and UperNet101 [35] for ADE20K. Pixel accuracy (Acc) is compared across different models. This is done using real objects cropped and resized from ground-truth images in the training set of each dataset. In addition to classification accuracy, we use the Frechet Inception Distance (FID) [8] to measure the distance between the distribution of synthesized results and the distribution of real images.

Implementation Details All the experiments are conducted on an NVIDIA DGX1 with 8 V100 GPUs. We use Adam [14] as the optimizer, and learning rates for the generator and discriminator are both set to 0.0002. For Cityscapes, we train 60 epoches to obtain a good generator, and ADE20k needs 150 epoches to converge. m is set to 3 for both datasets.

4.2. Quantitative Evaluation

Table 1 summarizes the results of all the models w.r.t. the FID score and classification accuracy. We also report the scores on images generated from vanilla SPADE (SPADE-v) using segmentation map as input (upper bound). Measured by FID, BachGAN outperforms all the baselines in both datasets with a relatively large margin. For Cityscapes, BackGAN achieves a FID score of 73.3, which is close to the upper bound. In ADE20K, the improved gain over

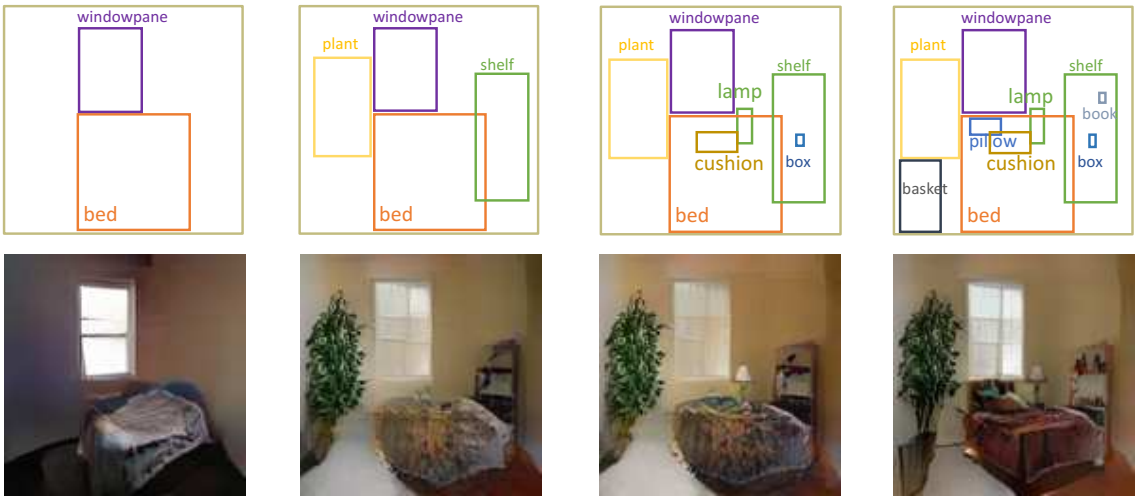

Figure 6: Generated images by adding bounding boxes sequentially to previous layout (ADE20k).

Figure 7: Top row: synthesized images based on salient object layouts from the test set. Bottom row: synthesized images based on salient layouts with flipping objects, modified from the top-row layouts.

baselines is not significant. This is because most images in ADE20K are dominated by salient foregrounds, with relatively less space for background, which limits the effect of our hallucination module. The pixel accuracy of our method is also higher than other baselines. BachGAN-r also achieves reasonable performance on both datasets.

4.3. Qualitative Analysis

In Figure 3 and 4, we provide qualitative comparison of all the methods. Our model produces images with much higher visual quality compared to the baselines. Particularly, in Cityscapes, our method can generate images with detailed/sharp backgrounds while the other approaches fail to. In ADE20K, though the background region is relatively smaller than Cityscapes, BachGAN still produces synthesized images with better visual quality.

Figure 5 and 6 demonstrate that BachGAN is able to manipulate a series of complex images progressively, by starting from a simple layout and adding new bounding boxes sequentially. The generated samples are visually appealing, with new objects depicted at the desired locations in the images, and existing objects remain consistent to the layout in previous rounds. These examples demonstrate our model's ability to perform controllable image synthesis based on layout.

Figure 7 further illustrates that BachGAN also works well when objects are positioned in an unconventional way. In the bottom row, we flip some objects in the object layouts (*e.g.*, windowpane in the top-left image), and generate images with the manipulated layouts. BachGAN is still able to generate high-quality images with a reasonable background, proving the robustness of BachGAN.

In Figure 8, we sample some retrieved images from Cityscapes. The top-3 results are consistent and similar with the original background. More synthesized and retrieval results (for ADE20K) are provided in Appendix.

4.4. Human Evaluation

We use Amazon Mechanical Turk (AMT) to evaluate the generation quality of all the approaches. AMT turkers are provided with one input layout and two synthesized outputs from different methods, and are asked to choose the image that looks more realistic and more consistent with the input layout. The user interface of the evaluation tool also provides a neutral option, which can be selected if the turker thinks both outputs are equally good. We randomly

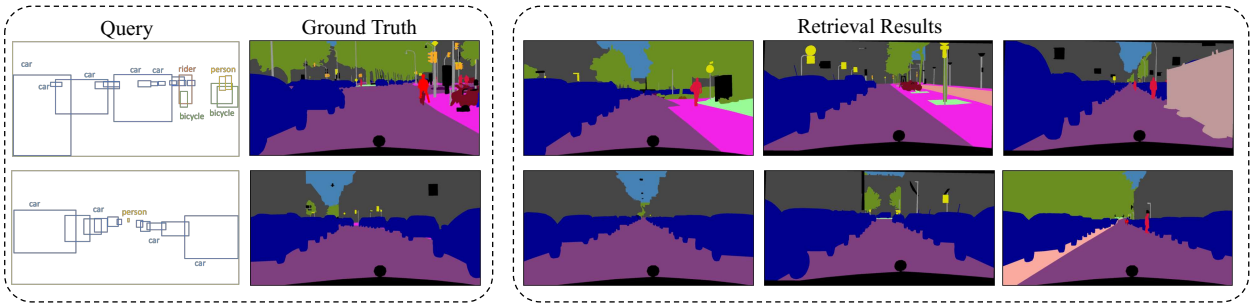

Figure 8: Examples of top-3 retrieved images from the memory bank of Cityscapes.

Dataset	BachGAN vs. SPADE			BachGAN vs. SPADE-Seg			BachGAN vs. BachGAN-r			BachGAN vs. Layout2im		
	win	loss	tie	win	loss	tie	win	loss	tie	win	loss	tie
Cityscapes	85.5	3.4	11.1	71.7	12.4	15.9	61.6	24.1	14.3	96.0	0.2	3.8
ADE20K	75.9	12.8	11.3	66.8	17.4	15.8	57.2	18.7	24.1	-	-	-

Table 2: User preference study. Win/lose/tie indicates the percentage of images generated by BachGAN are better/worse/equal to the compared model.

Method	BachGAN-3	BachGAN-4	BachGAN-5
FID	73.31	73.03	72.95

Table 3: FID scores of BachGAN with different numbers of retrieved segmentation maps (Cityscapes).

Bank size	BachGAN	BachGAN-r
$\|\mathbf{B}\|$	73.31	74.44
$2 \times \|\mathbf{B}\|$	72.50	73.95

Table 4: FID scores of BachGAN and BachGAN-r trained using memory bank of different sizes (Cityscapes).

sampled 300 image pairs, each pair judged by a different group of three people. Only workers with a task approval rate greater than 98% can participate in the study.

Table 2 reports the pairwise comparison between our method and the other four baselines. Based on human judgment, the quality of images generated by BachGAN is significantly higher than SPADE. Comparing with two strong baselines (SPADE-SEG and BachGAN-r), BachGAN achieves the best performance. As expected, Layout2im receives the lowest acceptance by human judges, due to its low resolution.

4.5. Ablation Study

Effect of segmentation map retrieval First, we train three BachGANs with different numbers of retrieved segmentation maps, setting m to 3, 4 and 5, and evaluate them on Cityscapes. The FID scores of different models are summarized in Table 3. The model using Top-5 retrieved segmentation maps (BachGAN-5) achieves the best performance, compared to models with Top-3 and Top-4. This analysis demonstrates that increasing the number of selected segmentation maps can slightly improve the scores. Due to the small performance gain, we keep $m = 3$ in our experiments.

Effect of memory bank We also compare models trained with memory banks of different sizes. Specifically, we compare the performance of BachGAN and BachGAN-r with memory bank size $|\mathbf{B}|$ (used in our experiments) and $2 \times |\mathbf{B}|$. Results are summarized in Table 4. With a larger memory bank, both models are able to improve the evaluation scores. Interestingly, the gain of BachGAN is larger than that of BachGAN-r, showing that BachGAN enjoys more benefit from the memory bank. More analysis about the size of memory bank is provided in Appendix.

5. Conclusion

In this paper, we introduce a novel framework, BachGAN, to generate high-quality images conditioned on salient object layout. By hallucinating the background based on given object layout, the proposed model can generate high-resolution images with photo-realistic foreground and integral background. Comprehensive experiments on both Cityscapes and ADE20K datasets demonstrate the effectiveness of our proposed model, which can also perform controllable image synthesis by progressively adding salient objects in the layout. For future work, we will investigate the generation of more complicated objects such as people, cars and animals [24]. Disentangling the learned representations for foreground and background is another direction.

Acknowledgements This work was also supported in part by NSF-1704309.

References

[1] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *ICCV*, 2019. 2

[2] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 2, 5

[3] Yu Cheng, Zhe Gan, Yitong Li, Jingjing Liu, and Jianfeng Gao. Sequential attention GAN for interactive image editing via dialogue. *arXiv preprint arXiv:1812.08352*, 2018. 2

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 5

[5] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *NeurIPS*, 2016. 2

[6] Lijie Fan, Wenbing Huang, Chuang Gan, Junzhou Huang, and Boqing Gong. Controllable image-to-video translation: A case study on facial expression generation. In *AAAI*, 2019. 2

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 6

[9] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *CVPR*, 2018. 2

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1, 2

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2

[13] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 1, 2

[14] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[16] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *NeurIPS*, 2018. 2

[17] Donghoon Lee, Tomas Pfister, and Ming-Hsuan Yang. Inserting videos into videos. In *CVPR*, 2019. 2

[18] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *CVPR*, 2019. 2

[19] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. In *CVPR*, 2019. 2

[20] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017. 2

[21] Elman Mansimov, Emilio Parisotto, Jimmy Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. In *ICLR*, 2016. 2

[22] Youssef Mroueh, Chun-Liang Li, Tom Sercu, Anant Raj, and Yu Cheng. Sobolev GAN. In *ICLR*, 2018. 2

[23] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *NeurIPS*, 2016. 2

[24] Xi Ouyang, Yu Cheng, Yifan Jiang, Chun-Liang Li, and Pan Zhou. Pedestrian-synthesis-gan: Generating pedestrian data in real scene and beyond. *arXiv preprint arXiv:1804.02047*, 2018. 8

[25] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 1, 2, 3, 5, 6

[26] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[27] Patsorn Sangkloy, Jingwan Lu, Chen Fang, FIsher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*, 2017. 2

[28] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. Chatpainter: Improving text to image generation using dialogue. In *ICLR Workshop*, 2018. 2

[29] Guangyao Shen, Wenbing Huang, Chuang Gan, Mingkui Tan, Junzhou Huang, Wenwu Zhu, and Boqing Gong. Facial image-to-video translation by a hidden affine transformation. In *Proceedings of the 27th ACM international conference on Multimedia*, 2019. 2

[30] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 1985. 5

[31] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 2

[32] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 2

[33] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 1, 2, 3, 5

[34] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Texturegan: Controlling deep image synthesis with texture patches. *arXiv preprint arXiv:1706.02823*, 2017. 2

[35] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 6

[36] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018. 1, 2

[37] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017. 5

[38] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 1, 2

[39] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019. 1, 2, 5, 6

[40] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 5

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2

[42] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017. 2