Semantic-shape Adaptive Feature Modulation for Semantic Image Synthesis

Zhengyao Lv[1], Xiaoming Li[2], Zhenxing Niu[3], Bing Cao[4], Wangmeng Zuo[2,5](✉)
[1]Tomorrow Advancing Life [2]Harbin Institute of Technology
[3]Machine Intelligence Lab, Alibaba Group [4]Tianjin University [5]Peng Cheng Laboratory
{cszy98, hit.xmshr}@gmail.com wmzuo@hit.edu.cn

Abstract

Recent years have witnessed substantial progress in semantic image synthesis, it is still challenging in synthesizing photo-realistic images with rich details. Most previous methods focus on exploiting the given semantic map, which just captures an object-level layout for an image. Obviously, a fine-grained part-level semantic layout will benefit object details generation, and it can be roughly inferred from an object's shape. In order to exploit the part-level layouts, we propose a Shape-aware Position Descriptor (SPD) to describe each pixel's positional feature, where object shape is explicitly encoded into the SPD feature. Furthermore, a Semantic-shape Adaptive Feature Modulation (SAFM) block is proposed to combine the given semantic map and our positional features to produce adaptively modulated features. Extensive experiments demonstrate that the proposed SPD and SAFM significantly improve the generation of objects with rich details. Moreover, our method performs favorably against the SOTA methods in terms of quantitative and qualitative evaluation. The source code and model are available at SAFM.

1. Introduction

Semantic image synthesis is a kind of conditional image generation task, which aims to generate semantically aligned and photo-realistic images with the given semantic maps. Compared to unconditional image generation, it has significant flexibility in image generation since we can flexibly control the generated image content by drawing or editing the input semantic maps. Semantic image synthesis has been widely used in many practical scenarios, *e.g.*, content creation and image editing [7, 28, 33, 42].

Recently, Generative Adversarial Networks (GANs) [10] are broadly adopted to solve this problem and achieve impressive results. Most works attempt to model the mapping between different semantic classes and visual appearances. Park *et al*. [28] propose to use spatially-adaptive transformations (SPADE) learned from the input semantic layouts to modulate the activations in the generator.

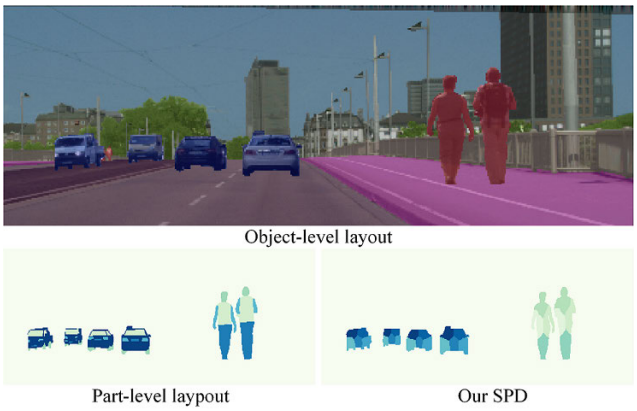


Figure 1. The given semantic map only provides an object-level layout, which is too coarse for generating images with rich details. The part-level semantic layout is implied in the shape/contour of an object instance. By encoding object shape into the proposed SPD feature, we can effectively exploit such part-level layouts for better image details generation.

CC-FPSE [21] subsequently extends SPADE by predicting spatially-varying conditional convolution kernels from the semantic layouts. Most recently, SC-GAN [36] exploits the learned semantic vectors to get spatially-variant and appearance-correlated convolution kernels and normalization parameters for the semantic stylization.

A semantic map has not only semantic labels but also a spatial layout. Such a spatial layout can be used to regularize the semantic image synthesis. Generally, one object instance is composed of some object parts, and pixels from the same object part should have a similar appearance while pixels from distinct object parts should not. For instance, an object 'car' is composed of 'window', 'wheel', *etc*. Thus, the pixels from the 'window' should look different from those from the 'wheel'. In contrast, two pixels both from the 'window' should look similar to each other. By exploiting such a spatial layout, we can suppress artifacts and generate coherent image details.

Semantic layouts have been effectively exploited to improve image synthesis in previous methods. However, the given semantic map just captures an *object-level* layout for an image, which describes whether two pixels belong to the same object instance or not. It is too coarse to capture the

fine-grained structure of an object instance. If we could subtly exploit a **part-level semantic layout**, it will benefit the generation of image high-frequency details.

Obviously, the *shape/contour* of each object instance can be easily identified from the object-level semantic layout. On the other hand, given the shape/contour of an object (*e.g.,* a car), its part-level layout (*e.g.,* the position of 'window' or 'wheel') can be roughly inferred according to the prior knowledge of an object's structure, as shown in Fig. 1. Therefore, there is a strong connection between an object's shape and its part-level layout, *i.e.,* **an object's shape implies its part-level layout**. Thus, the exploitation of an object's part-level layout can be implicitly achieved by modeling its shape.

In this paper, we propose a Shape-aware Position Descriptor (SPD) to describe each pixel's positional feature. Our SPD describes the relative relations (distance and angle) between each pixel inside an object instance and pixels on its contour, as shown in Figure. 2 (a). Thus, the information of object shape has been encoded into each pixel's SPD feature. In other words, the clue of an object's part-level layout has been implicitly encoded into the SPD feature.

Next, we design the Semantic-shape Adaptive Feature Modulation (SAFM) block to combine the given semantic map and our SPD features together, and modulate the input features adaptively. Specifically, our SAFM block first conditionally produces semantic-specific convolution kernels, and then performs semantic-specific convolution on the SPD features. At last, the SAFM block accepts input feature maps, adaptively modulates them, and forwards them to the next block, as shown in Figure. 2 (b).

Note that our SPD is inspired by the shape context descriptor [2] which describes the relations of pixels just on the shape contour, but our SPD describes the relations between pixels inside an object and pixels on the contour.

Our main contributions can be summarized as follows:

- We propose a Shape-aware Position Descriptor (SPD) to describe the pixel's positional feature, where the object's part-level layout can be exploited and leveraged.
- We design a Semantic-shape Adaptive Feature Modulation (SAFM) block, which combines the semantic maps and SPD features to produce adaptively modulated feature maps.
- Experimental results show that our method performs favorably on Cityscapes, COCO-stuff, and ADE20K datasets against SOTA methods and can generate more photo-realistic results with rich details.

2. Related Work

2.1. Semantic Image Synthesis

Generative Adversarial Networks (GANs) [10] have achieved impressive results on unconditional image generation related tasks [4, 13, 14]. Subsequently, by introducing external information, such as class labels [22, 26, 27], natural language descriptions [17, 18, 40], or semantic maps [28, 35], many kinds of conditional GANs are proposed to improve the controllability of image generation.

Semantic image synthesis is a task that takes semantic segmentation maps as input, which provides pixel-level class labels. Pix2Pix [12] is first proposed to use an encoder-decoder generator and PatchGAN discriminator to conduct semantic image generation. Pix2PixHD [35] improved it by adopting a coarse-to-fine generator and multi-scale discriminators to generate vivid details at high-resolution space. Particularly, Pix2PixHD introduced the instance-level boundary map as extra input to separate different instances for sharper boundaries. Further, panoptic aware convolutions and upsampling layers [9] are utilized to differentiate occluded instances.

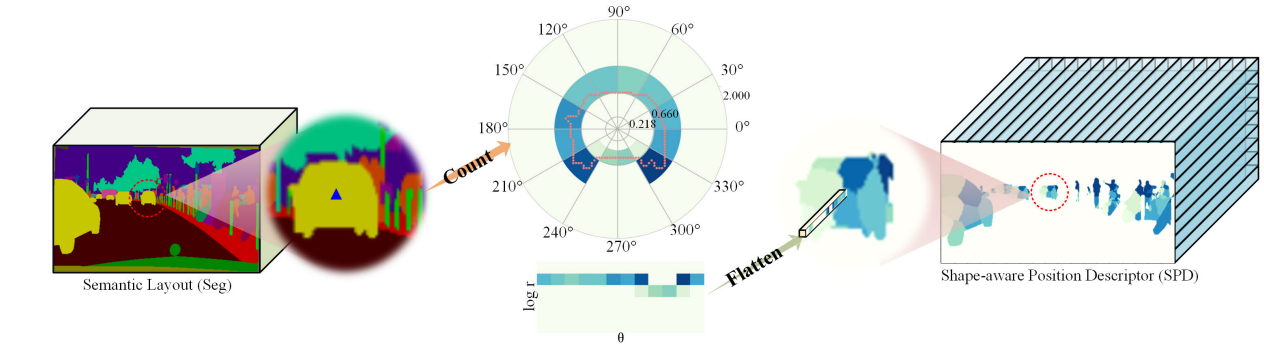
Recently, most works focus on how to sufficiently leverage the given semantic layouts. SPADE [28] proposed to modulate the activations with spatially-adaptive transformations learned from semantic layouts. CC-FPSE [21] learned to predict conditional convolution kernels based on the given semantic layouts. Additionally, a feature pyramid semantic-embedding discriminator is employed to enable the generator to synthesize semantically aligned images with high-quality details. Similarly, Ntavelis *et al*. [25] proposed a two-stream discriminator by using semantic features to guide the scores of the discriminator. LGGAN [33] proposed a local class-specific and global image-level generative adversarial network to separately learn the global appearance distribution and generation of different object classes. EdgeGAN [32] generated edges from semantic layouts to introduce detailed structure information for image synthesis. Most recently, SCGAN [36] learned semantic vectors to parameterize spatially conditional convolution and normalization. Besides, OASIS [31] re-designed the discriminator with a segmentation-based network for synthesizing semantically aligned images with higher fidelity.

Except for these GAN-based methods, CRN [7] adopted a cascaded refinement network for semantic image synthesis. Qi *et al*. [29] proposed a semi-parametric approach, which retrieved compatible fragments and composited them to assist the semantic image synthesis.

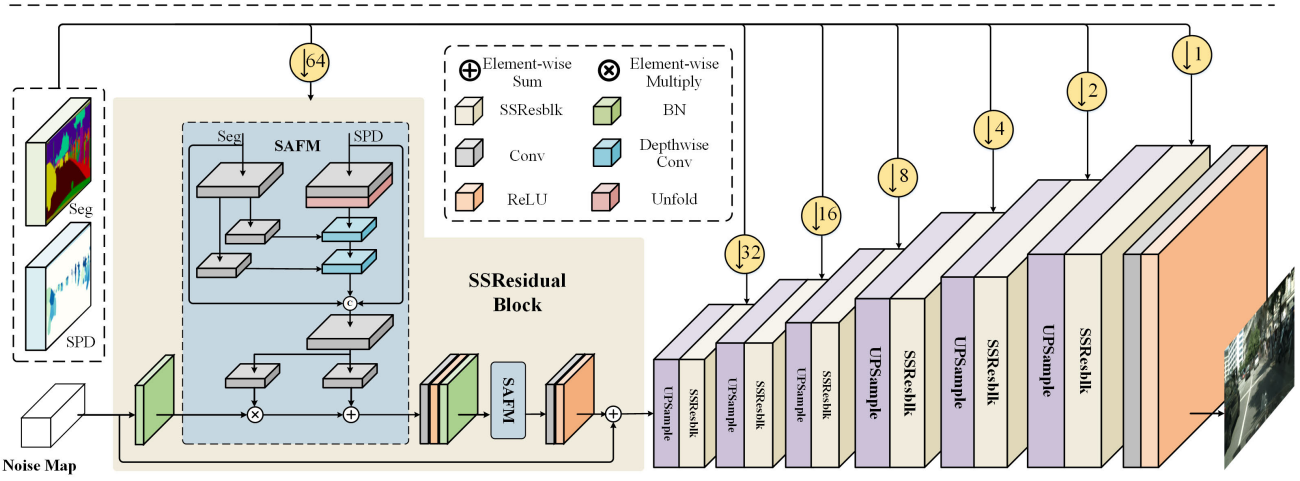
Most of those methods just exploit object-level semantic layouts, which are too coarse to capture the part-level structure of object instances. Although the part-level layouts are unknown, they can be roughly inferred from objects' shapes. In our work, such part-level layouts are effectively exploited by encoding objects' shapes into our SPD descriptors.

2.2. Shape Context Descriptor

Shape context descriptor was first proposed by Belongie *et al*. [2] for category-level shape matching and object recognition. Through counting the histogram of the relative position distribution of other shape points, a rich



(a) The calculation process of shape-aware position descriptor



(b) Architecture of our network

Figure 2. Overview of our proposed method. (a) shows the calculation process of the SPD features of a certain point in a car instance (denoted by a blue ▲). After calculating all points inside the instances, we get a SPD map, as shown in (a) (right). (b) illustrates the architecture of our generator network, where SAFM is mainly constructed by conditional convolutions.

local descriptor that implies the global shape points can be obtained for each point. After that, Thayananthan *et al*. [34] propose an efficient dynamic programming scheme to constrain the figural continuity of shape context matching. Instead of Euclidean distance, Ling *et al*. [20] adopt inner-distance to measure the spatial relation between shape points, which can better capture the structure of complex shapes with articulations.

The shape context descriptor can bring sufficient information that captures the relative locations within the whole instance beyond the point itself. Due to the robustness and discrimination in reducing the ambiguity in class matching, these types of descriptors have been widely employed for different object recognition problems [3,24], but are seldom exploited in semantic image synthesis tasks. In this work, we extend the shape context descriptor [2] to characterize the position of each point inside an object instance, where object shapes are explicitly exploited and leveraged.

3. Method

Given a semantic layout $S \in \mathbb{R}^{H\times W\times C}$ with C class labels, our goal is to synthesize a photo-realistic image $I_s \in \mathbb{R}^{H\times W\times 3}$, which is semantically aligned with S. Following [35], we adopt the instance-level segmentation map as supplementary input to obtain each instance region.

In the following, we first introduce the Shape-aware Position Descriptor (SPD), where object shapes are exploited and leveraged. Next, we design the Semantic-shape Adaptive Feature Modulation (SAFM) block to combine the semantic maps and SPD features to adaptively modulate the input feature maps.

3.1. Shape-aware Position Descriptor

The shape of an object instance implies its part-level layout, as shown in Fig. 1. In our approach, we propose the Shape-aware Position Descriptor (SPD) to describe each pixel's positional feature, where the object shape is explicitly considered. In this way, the clue of an object's part-level layout could be effectively exploited and leveraged.

To balance the computation cost and the robustness of the descriptor, we only use the contour point set of an object instance to describe its shape information, instead of using all the points inside the segmentation region.

Calculation process of the SPD. Figure. 2 (a) illustrates

the calculation process of our proposed SPD. Taking the rear-view car in the semantic map as an example, we can easily get its contour shape $T \in \{0,1\}^{H\times W}$ according to its segmentation map, where the points on the contour are denoted by label 1 and the rest are set to 0. Further, the contour shape of the car can be discretely represented as a point set $P = \{(x,y)|T(x,y) = 1\}$.

For any point $o = (x_o, y_o)$ inside the instance, we calculate its positional descriptor through the following steps. Firstly, we take the point o as the pole to construct a polar coordinate space around it. And then we divide this coordinate into $m \times n$ bins B with m polar radius intervals and n polar angle intervals (in this work $m = 12$ and $n = 6$). Each bin $B_{i,j}$ should satisfy the following condition:

$$B_{i,j} = \{(r,\theta)|r_{i-1} <= r < r_i, \theta_{j-1} <= \theta < \theta_j\}. \quad (1)$$

In order to make the descriptor more sensitive to the nearby points relative to the farther ones, we use bins that are uniform in log-polar space.

After that, the distance and angle distribution of each point in the contour point set P relative to the pole o can be formulated as $P' = \{(r_i,\theta_i)|_{i=1}^{|P|}\}$. Finally, we count the number of the points in the contour point set P' that fall in each bin $B_{i,j}$, denoted as $H_{i,j}$:

$$H_{i,j} = |\{p|p \in P', p \in B_{i,j}\}|, \quad (2)$$

where $|\cdot|$ denotes the quantization operation. By integrating the number of contour points in all $m\times n$ bins and flattening it, we can get a vector $v_o \in \mathbb{R}^{m\times n}$ about point o that stores the contour point distribution. The final SPD $\hat{v}_o$ for point o can be obtained through the normalization:

$$\hat{v}_o = \frac{v_o}{|P'|}. \quad (3)$$

After calculating the descriptor for all the points inside each instance, we can get a SPD map that explicitly represents the detailed position for each point.

Discussion about the SPD. For example, there are three rear-view cars in Fig. 3 (a), which have similar shape contours but different spatial locations and scales. Intuitively, the corresponding points at the same object part should have consistent SPD features. In contrast, the points from different object parts should have different SPD features.

Taking the points on the left wheel as an example (denoted by blue ▲), the SPDs of the two points are shown in Fig. 3 (b) and (c). Another example is the points at the center of car (denoted by blue ⋆), their SPD features are shown in Fig. 3 (d) and (e).

We can observe that: (i) For the corresponding positions of different instances, their SPD features look similar to each other (b *vs* c and d *vs* e). (ii) For different positions of the same instance, there are obvious differences between their SPD features (c *vs* d). (iii) Even if the absolute location or scale of an instance changes, we can still get similar SPD features, which indicates our SPD is only dependent on object shape (d *vs* e). In other words, object shapes are robust and discriminatively encoded into our SPD features.

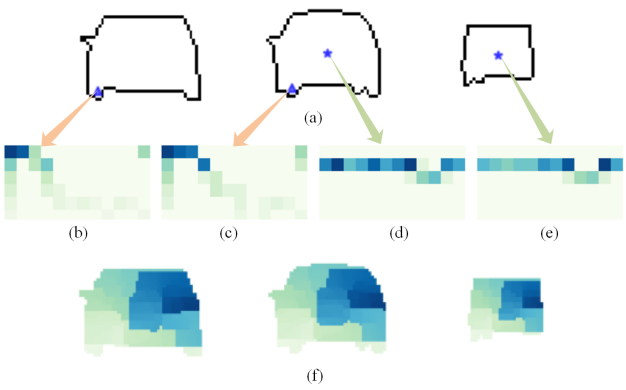


Figure 3. Visualization of the SPD features of the car instances. There are three different rear-view cars with similar shapes in (a). (b-e) show the descriptor of blue ▲ and ⋆ points in (a). (f) illustrates the descriptor compressed to 1D by t-SNE.

Furthermore, we also jointly consider the SPD features of all pixels inside an object instance. Specifically, we use t-SNE to compress each pixel's SPD feature as a scalar, and hence we obtain a compact 2D map corresponding to all pixels inside an object instance, as shown in (f). We can see that all the three cars share similar patterns in the compact 2D map. More importantly, for each instance, **the compact 2D map could well describe the part-level layout of a car**. Thus, we claim that our SPD features could implicitly exploit an object's part-level layout.

3.2. Semantic-Shape Adaptive Feature Modulation

Another thing should be noticed is that different classes of object instances may have similar shapes. For example, the painting and washer shown in Fig. 4 (a) are both rectangular, and the patterns of their SPD features are quite similar (compact 2D map), but their appearance and structure are different, which will confuse the image synthesis.

To circumvent this issue, we designed the Semantic-shape Adaptive Feature Modulation (SAFM) block (Fig. 2 (b)), which combines the semantic information and our SPD features to compensate each other. For instance, the proposed SPD features can bring more detailed descriptions about the point positions to the semantic layouts, while the semantic layouts can bring complementary semantic information to the SPD features. And then the SAFM yields semantic-shape adaptive modulation parameters for different positions of different classes, so as to subtly guide the semantic image synthesis.

In the SAFM block, semantic layouts and SPD features are first scaled to the same size. Then semantic layouts are fed into two convolution layers to predict two sets of semantic-adaptive 3×3 convolution kernels, which vary with the class label of spatial position. After that, through depthwise convolution layers, the semantics information of

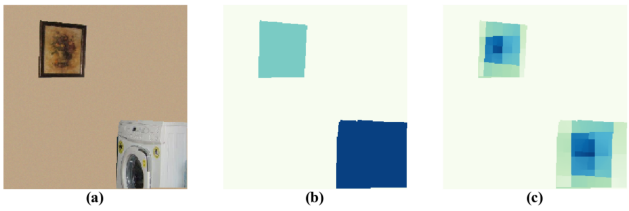


Figure 4. The (a) appearance, (b) shape and (c) the compact 2D map of the painting and washer instances.

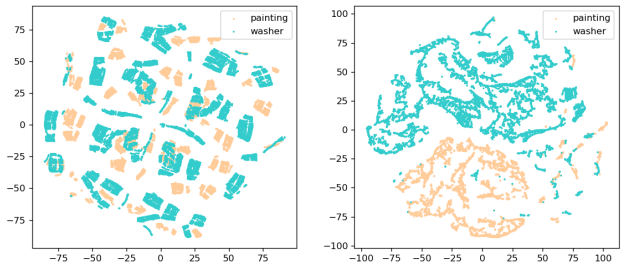


Figure 5. The distribution of the SPD of points in the painting and washer before and after using SAFM block.(mapping by t-SNE).

each spatial position is fused into the corresponding position in the SPD features. Finally, it yields semantic-shape adaptive modulation parameters with the fused features for feature modulation.

With the SAFM block, semantic information and spatial position information can be integrated together. Fig. 5 shows the distribution of the SPD features without and with the SAFM block. Note that the light green dots represent the washer, while the orange dots stand for the painting. One can see that by incorporating the SAFM block, the washer and painting points tend to be better separated, which shows the effectiveness of the SAFM block in combining the SPD features and semantic features.

3.3. Learning Objective

In our approach, we adopt adversarial loss $\mathcal{L}_{adv}$, feature matching loss $\mathcal{L}_{fm}$, and perceptual loss $\mathcal{L}_{perc}$ to achieve high fidelity and realness of generation. In addition, $\mathcal{L}_{seg}$ is suggested from the pre-trained segmentation model for constraining the semantic alignment.

Adversarial Loss. Adversarial learning can effectively keep the generated images staying at the real image manifold and has been widely used in many image generation tasks [4, 13, 22]. In this work, we adopt the hinge-based adversarial loss [19, 23, 39] and the optimization of generator G and discriminator D can be formulated as:

$$\begin{aligned}\mathcal{L}_{adv}^{D} = &- \mathbb{E}_{(I,S)}[min(0, -1 + D(I,S))] \\ &-\mathbb{E}_{(z,S)}[min(0, -1 - D(G(z,S),S))],\end{aligned} \tag{4}$$

$$\mathcal{L}_{adv}^{G} = -\mathbb{E}_{(z,S)} D(G(z,S),S), \tag{5}$$

where I is the real image, S is the corresponding semantic layouts, and z is the noise map fed into the generator.

Feature Matching Loss. Following [35], we adopt the feature matching loss $\mathcal{L}_{fm}$ to enhance the supervision for stabilizing the training process which constrains the features of synthesized images to be close to the real one in different feature spaces of discriminator D. This can be defined as:

$$\mathcal{L}_{fm} = \sum_{i=1}^{n} \frac{1}{N_i} ||D_i(I,S) - D_i(G(z,S),S)||_1, \tag{6}$$

where N_i is the number of elements in feature $D_i(I,S)$.

Perceptual Loss. We adopt the pre-trained VGG19 model Φ [30] to separately extract the features from the real image I and the generated images $\hat{I}$. The perceptual loss L_{perc} is computed in multi-scale feature space and is formulated as:

$$\mathcal{L}_{perc} = \sum_{k=1}^{K} ||\Phi_k(\hat{I}) - \Phi_k(I)||_1, \tag{7}$$

where ϕ_k denotes the k-th feature map extracted from the VGG19 model Φ. In our implementation, we set $K = 5$.

Semantic Alignment Loss. In order to explicitly constrain the semantic consistency between the generated image and the given semantic layout, we further introduce the semantic alignment loss $\mathcal{L}_{seg}$ to optimize the learning process:

$$\mathcal{L}_{seg} = -\sum_{i=1}^{C} w_i \sum_{j=1}^{H} \sum_{k=1}^{W} S_{i,j,k} [\log Seg(I)_{i,j,k} + \log Seg(\hat{I})_{i,j,k}], \tag{8}$$

$$w_i = \frac{H \times W}{\sum_{j=1}^{H} \sum_{k=1}^{W} S_{i,j,k}}, \tag{9}$$

where Seg is a pre-trained segmentation model [1].

The overall learning objective can be summarized as:

$$\mathcal{L} = \lambda_{adv}\mathcal{L}_{adv}^{G} + \lambda_{fm}\mathcal{L}_{fm} + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{seg}\mathcal{L}_{seg}, \tag{10}$$

where λ_{adv}, λ_{fm}, λ_{perc}, and λ_{seg} are trade-off parameters.

4. Experiments

Extensive experiments are conducted to evaluate the effectiveness of our proposed SPD features and SAFM block. We report the quantitative and qualitative results in comparison with the competing methods, including CRN [7], SIMS [29], Pix2PixHD [35], SPADE [28], CC-FPSE [21], OASIS [31], LGGAN [33] and SC-GAN [36]. Besides, the ablation study is further conducted to explore the benefits of each component of our method that bring to the results.

4.1. Dataset and Experimental Details

Dataset. Our experiments are conducted on three challenging datasets, *i.e.*, Cityscapes [8], ADE20K [41], and COCO-Stuff [5]. The Cityscapes dataset contains images of urban street scenarios, which has 3,000 images for training and 500 for validation. The ADE20K dataset has 20,210 images for training and 2,000 for validation each of which has 150 semantic classes, covering indoor and outdoor scenarios. Similarly, The COCO-Stuff contains 182 classes

Methods	Cityscapes			ADE20K			COCO-Stuff		
	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓
CRN [7]	52.4	77.1	104.7	22.4	68.8	73.3	23.7	40.4	70.4
SIMS [29]	47.2	75.5	49.7	N/A	N/A	N/A	N/A	N/A	N/A
pix2pixHD [35]	58.3	81.4	95.0	20.3	69.2	81.8	14.6	45.7	111.5
SPADE [28]	62.3	81.9	71.8	38.5	79.9	33.9	37.4	67.9	22.6
CC-FPSE [21]	65.6	82.3	54.3	43.7	82.9	31.7	41.6	70.7	19.2
LGGAN [33]	68.4	83.0	57.7	41.6	81.8	N/A	N/A	N/A	N/A
OASIS [31]	69.3	N/A	**47.7**	48.3	N/A	**28.3**	**44.1**	N/A	**17.0**
SC-GAN [36]	66.9	82.5	49.5	45.2	83.8	29.3	42.0	72.0	18.1
Ours	**70.4**	**83.1**	49.5	**50.1**	**86.6**	32.8	43.3	**73.4**	24.6

Table 1. The quantitative comparison with the competing methods on different datasets. ↑ (↓) indicates higher (lower) is better.

that cover diverse scenarios and provides 118,000 images for training and 5,000 for validation. The real images and their corresponding semantic layouts in ADE20K and COCO-Stuff are resized and cropped to 256×256 while those in Cityscapes are processed to 256×512.

Experimental Details. We adopt the generator of SPADE [28] and SESAME [25] discriminator as baseline model. Following SPADE, Spectral Norm [23] is incorporated in all convolutional layers in our model. We adopt the ADAM [15] optimizer with $\beta_1 = 0, \beta_2 = 0.999$, and the learning rate is set to 1×10^{-4} and 4×10^{-4} for generator and discriminator, respectively. Our model is trained on ADE20K and Cityscapes for 200 epochs, and 100 epochs on COCO-Stuff. The trade-off parameters λ_{adv}, λ_{fm}, λ_{perc}, and λ_{seg} are set to 1, 10, 10, 1, respectively. The experiments are carried out on a server with 4 2080Ti GPUs.

Evaluation Metrics. Following previous semantic synthesis works [21, 28], we use three metrics to quantitatively evaluate the performance, *i.e.*, Fréchet Inception Distance (FID) [11], mean Intersection-over-Union (mIoU), and pixel accuracy (Acc). Among these metrics, FID is introduced to assess the realism of the synthesized images by computing the Wasserstein-2 distance between the distributions of the synthesized and real images. Acc and mIoU are proposed to measure the differences of semantic labels between the synthesized images and the input semantic layouts. Following [28], we use the pre-trained semantic segmentation models DRN-D-105 [38], UperUnet101 [37] and DeepLabV2 [6] for the semantic evaluation of Cityscapes, ADE20K and COCO-Stuff, respectively. In addition, we demonstrate the generated results for visual comparison with other competing methods. Finally, a user study is reported to further evaluate the effectiveness of our method.

4.2. Quantitative and Qualitative Results

Quantitative comparisons. Table 1 lists the semantic segmentation and FID performance of our model and the competing methods on the Cityscapes, ADE20k, and COCO-Stuff datasets. In terms of semantic alignment, the mIoU of our method achieves 70.4 and 50.1 on Cityscapes and ADE20K, respectively (at least 1.1 and 1.8 higher than the second-best one). In addition, our method obtains the comparable FID scores, which ensures the distribution consistency between the generated results and the real images. The best semantic segmentation performance of our method indicates that the results of our method are not only more consistent with the target layout, but also photo-realistic in appearance, both of which can be attributed to the introduction of the SPD feature and the SAFM block.

Note that OASIS achieves nearly the best performance on the COCO-Stuff dataset, but performs inferior to ours on the Cityscapes and ADE20K datasets, we analyze that the COCO-stuff dataset has more stuff classes without part-level semantics (91 stuff classes that cover about 66% of the pixels), which makes the superiority of our SPD features for object instances not obvious in the quantitative results.

Qualitative comparisons. Fig. 6 gives the qualitative comparisons on the Cityscapes, ADE20K, and COCO-Stuff datasets, from which we can observe that (i) with the SPD, our method can generate more realistic details (*e.g.* the washing machine in the 3-rd row), which is benefited from the discriminative and effective spatial position characterization. (ii) From the 4-th to 6-th rows, our method can well handle the instances of the same class with different shapes, while others fail to generate plausible results, indicating the effectiveness of our SPD features and SAFM block. (iii) With the constraints of semantic alignment, our method also performs well in unstructured textures, contributing to better visual quality, which can be seen from the 1-st row.

User Study. Following [28], we conduct a user study to further compare our method with SPADE, CC-FPSE, LGGAN, and OASIS on the Cityscapes and ADE20K datasets. For each set of experiments, participants with computer vision backgrounds[1] are required to select the image that has better performance in semantic alignment and photo-realistic appearance. From Table 2 we can see that users are more likely to favor our results, especially on the Cityscapes.

Multi-modal synthesis. Following SPADE [28], we train an additional encoder for multi-modal synthesis or style-guided image with the KL Divergence loss in the way of VAE [16]. By controlling the mean and variance vector to

[1]The participants have been informed that the collected data will be only used for academic purposes, and their identities will not be recorded.

Figure 6. Visual comparisons on the COCO-Stuff (1-st $\sim$ 2-nd rows), ADE20K (3-rd row) and Cityscapes (4-th $\sim$ 6-th rows) datasets.

Methods	Cityscapes	ADE20K
Ours > SPADE	74.76%	63.32%
Ours > CC-FPSE	63.20%	58.24%
Ours > LGGAN	68.48%	58.96%
Ours > OASIS	65.24%	56.76%

Table 2. User study. The numbers represent the percentage of our method favored by users relative to competing methods.

sample different random noises, our generator can also synthesize images with diverse and photo-realistic appearances for the given input segmentation mask, as shown in Fig. 8.

Results with segmentation-based discriminator. Noting the success of the segmentation-based discriminator in OASIS, we verified the effectiveness of SPD with the discriminator and training tricks of OASIS. Specifically, we use SPD to replace the 3D noise in the OASIS generator, resulting in improved results for the Cityscapes dataset (FID: 43.81 and mIoU: 71.8). More qualitative results are shown in the supplementary materials.

4.3. Ablation Study

We conduct the ablation study on the Cityscapes dataset to evaluate the effectiveness of our SPD feature and SAFM block, which contains the following variants. (1) *Baseline*: by adopting SPADE [28] generator and SESAME [25] discriminator as the baseline model. (2) *Baseline+*$\mathcal{L}_{seg}$: by adding the semantic alignment loss upon the *Baseline* model. (3) *Baseline+SPD*: by concatenating the SPD features with semantic layouts and feeding them into the SPADE block to generate spatially adaptive modulation parameters. (4) *Baseline+SPD+SAFM*: by introducing the SAFM block to the generator to exploit the SPD features instead of directly concatenating them. (5) *Ours-Full*: by incorporating the Baseline, $\mathcal{L}_{seg}$, SPD features and SAFM block together. The quantitative results and visual comparisons are shown in Table 3 and Fig. 7, respectively.

We can see that (i) although $\mathcal{L}_{seg}$ equally promotes the realistic texture generation of object classes (*e.g.* car, washer classes) and stuff classes (*e.g.* sky, earth classes), the mIoU of object classes (mO) and stuff classes (mS) increase by 2.9 and 3.7, respectively, which greatly improves the performance of the synthesized images on mIoU and Acc metrics (3.4 and 0.5 higher than Baseline), it still can not promote the generator to synthesize rich structural details. Intuitively, its FID score has been slightly improved. (ii) The generator by introducing SPD as an additional condition can synthesize richer details, such as realistic car windows and lights, thereby greatly improving the mIoU, Acc,

Figure 7. Visual comparisons of different variants.

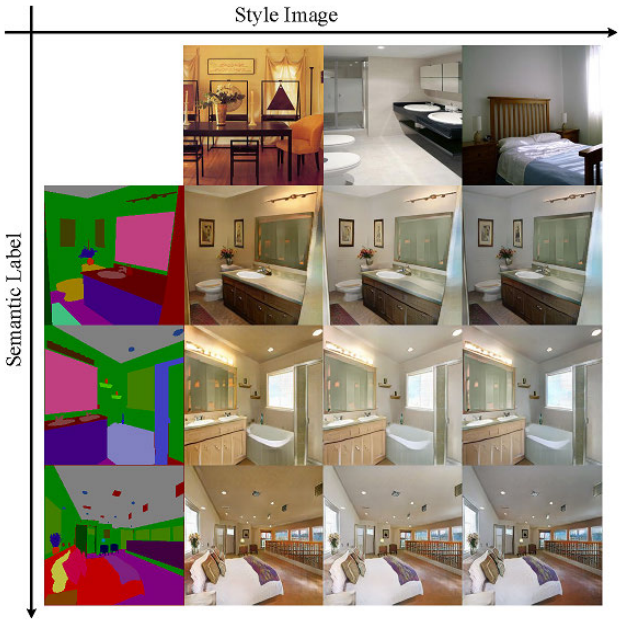


Figure 8. Visual results of multi-modal synthesis.

and FID scores of the generated results (see *Baseline VS. Baseline+SPD*). From Table 3 we can see that the mIoU of the object classes of the *Baseline+SPD* model increased by 3.3 compared to the *Baseline* model, and the FID reduced from 54.2 to 50.1, which indicates the effectiveness of our proposed SPD. (iii) The SAFM block enables the generator to better model the mapping between the SPD features and the shape appearance, which further improves the performance of our model, especially on the mIoU of object classes. (iv) By incorporating the SPD features, SAFM block, and $\mathcal{L}_{seg}$, the performance of *Ours-Full* achieved the best performance, indicating the effectiveness of each component of our method in the synthesizing process.

4.4. Limitation and Impact

Limitation. Since the part-level semantic layouts for each object class are learning from data, the performance of our approach heavily depends on the quantity of training data. Thus, the rare object classes or the rare shape patterns cannot be well modeled. For instance, non-rigid human bodies sometimes have uncommon posture and shape, from which it is hard to infer the implied part-level layout with insufficient training samples. Nevertheless, our method cloud significantly improve the quality of image synthesis for common object classes and common shape patterns.

Impact. This paper proposes a method for semantic image synthesis which can synthesize or edit images based on semantic maps. Malicious usage of semantic image synthesis models may have adverse social repercussions, such as the synthesis of images for the purpose of spreading fake news.

Methods	mIoU ↑	mS ↑	mO ↑	Acc ↑	FID ↓
Baseline	66.0	70.0	60.7	82.5	54.2
+$\mathcal{L}_{seg}$	69.4	73.7	63.6	83.0	53.2
+SPD	68.5	71.8	64.0	82.7	50.1
+SPD+SAFM	69.4	71.5	**66.4**	82.8	50.6
Ours-Full	**70.4**	**74.2**	65.3	**83.1**	**49.5**

Table 3. Quantitative comparison of five variants on Cityscapes. Here, mS (mO) represents the mIoU of Stuff (object) classes.

5. Conclusion

In this paper, the shape of object instances is explicitly encoded into the proposed SPD features. Thus, the object's part-level layouts could be exploited to improve the generation of images with rich details. Furthermore, the SAFM block is proposed to combine the semantic map and SPD features through conditional convolution operation, which could adaptively modulate the input features. The quantitative and qualitative results demonstrate the superior performance of our method in synthesizing semantically aligned images with rich as well as photo-realistic details.

Acknowledgments. This work was supported in part by National Key R&D Program of China under Grant No. 2020AAA0104500, and by the National Natural Science Foundation of China (NSFC) under Grant No.s U19A2073 and 62006064.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 5

[2] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. *Advances in neural information processing systems*, 13:831–837, 2000. 2, 3

[3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002. 3

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2, 5

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 5

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 6

[7] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017. 1, 2, 5, 6

[8] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5

[9] Aysegul Dundar, Karan Sapra, Guilin Liu, Andrew Tao, and Bryan Catanzaro. Panoptic-based image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8070–8079, 2020. 2

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1, 2

[11] 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. *Advances in neural information processing systems*, 30, 2017. 6

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2, 5

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 2

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[17] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Controllable text-to-image generation. *arXiv preprint arXiv:1909.07083*, 2019. 2

[18] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7880–7889, 2020. 2

[19] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 5

[20] Haibin Ling and David W Jacobs. Shape classification using the inner-distance. *IEEE transactions on pattern analysis and machine intelligence*, 29(2):286–299, 2007. 3

[21] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and Hongsheng Li. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. *arXiv preprint arXiv:1910.06809*, 2019. 1, 2, 5, 6

[22] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2, 5

[23] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5, 6

[24] Greg Mori and Jitendra Malik. Estimating human body configurations using shape context matching. In *European conference on computer vision*, pages 666–680. Springer, 2002. 3

[25] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. Sesame: semantic editing of scenes by adding, manipulating or erasing objects. In *European Conference on Computer Vision*, pages 394–411. Springer, 2020. 2, 6, 7

[26] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016. 2

[27] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017. 2

[28] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 1, 2, 5, 6, 7

[29] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8808–8816, 2018. 2, 5, 6

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[31] Vadim Sushko, Edgar Schönfeld, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. *arXiv preprint arXiv:2012.04781*, 2020. 2, 5, 6

[32] Hao Tang, Xiaojuan Qi, Dan Xu, Philip HS Torr, and Nicu Sebe. Edge guided gans with semantic preserving for semantic image synthesis. *arXiv preprint arXiv:2003.13898*, 2020. 2

[33] Hao Tang, Dan Xu, Yan Yan, Philip HS Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7870–7879, 2020. 1, 2, 5, 6

[34] Arasanathan Thayananthan, Bjoern Stenger, Philip HS Torr, and Roberto Cipolla. Shape context and chamfer matching in cluttered scenes. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003. 3

[35] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 2, 3, 5, 6

[36] Yi Wang, Lu Qi, Ying-Cong Chen, Xiangyu Zhang, and Jiaya Jia. Image synthesis via semantic composition. *arXiv preprint arXiv:2109.07053*, 2021. 1, 2, 5, 6

[37] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 6

[38] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017. 6

[39] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019. 5

[40] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017. 2

[41] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5

[42] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020. 1