

Dual Pyramid Generative Adversarial Networks for Semantic Image Synthesis

Shijie Li¹

lsj@uni-bonn.de

Ming-Ming Cheng²

cmm@nankai.edu.cn

Juergen Gall¹

gall@iai.uni-bonn.de

¹ Computer Vision Group

University of Bonn

Bonn, Germany

² Media Computing Lab.

Nankai University,

Tianjin, China

Abstract

The goal of semantic image synthesis is to generate photo-realistic images from semantic label maps. It is highly relevant for tasks like content generation and image editing. Current state-of-the-art approaches, however, still struggle to generate realistic objects in images at various scales. In particular, small objects tend to fade away and large objects are often generated as collages of patches. In order to address this issue, we propose a Dual Pyramid Generative Adversarial Network (DP-GAN) that learns the conditioning of spatially-adaptive normalization blocks at all scales jointly, such that scale information is bi-directionally used, and it unifies supervision at different scales. Our qualitative and quantitative results show that the proposed approach generates images where small and large objects look more realistic compared to images generated by state-of-the-art methods.

1 Introduction

Generative adversarial networks [1] have become very successful in generating simple photo-realistic images like faces or animals [2, 3, 4]. However, it is still very challenging to generate images of complex scenes that contain many different objects at various scales. This is also the case when the semantic layout is provided, *e.g.* for content generation or image editing [5, 6, 7]. This task is also known as semantic image synthesis and requires to generate photo-realistic images for given semantic input layouts as illustrated in Fig. 1.

Previous methods for semantic image synthesis usually generate images by an encoder-decoder architecture that takes the semantic label map as input. However, as shown in [8], the flat segmentation maps will fade away due to the usage of common normalization layers, like instance normalization [9]. To address this issue, spatially-adaptive normalization (SPADE) [10] has been proposed, which propagates semantic information throughout the network. This is achieved by modulating the normalized activation in a spatially-adaptive manner, conditioned on the input segmentation mask. SPADE [10], however, uses as many other works a classification network as discriminator that takes the label map as input along with the image and makes a global decision whether an image is real or not. Since a global

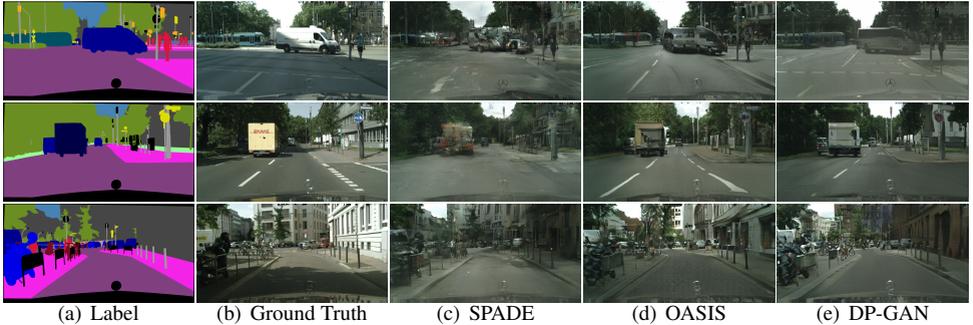


Figure 1: Given a label map (a) as input, current approaches like SPADE (c) or OASIS (d) struggle to generate realistic looking objects. In particular large objects like trucks look highly unrealistic. Instead of generating a consistent appearance of an object, the methods generate a collage of patches. For instance, when zooming in on the truck in the top row of (d), it can be seen that the truck is a combination of patches from the side and back view of a car or van and four wheels are visible. They also struggle to generate other objects like traffic signs (row 2) or bollards (row 3). This is due to the handling of scale in the generator and discriminator, which is addressed by our approach (e).

discriminator does not enforce the generator to learn a precise alignment with the input semantic label maps, OASIS [25, 26] proposed a segmentation-based discriminator to address this issue. While the generated images of OASIS are less fuzzy than the objects generated by SPADE as shown in Fig. 1, there is still a huge difference between the generated and real images. While areas like road, vegetation, or sky are well generated by OASIS, the approach struggles to generate realistic objects at small and large scales. For instance, large objects like trucks, buses, or trams are generated as collages of patches as shown in the first and second row of Fig. 1 and small objects like traffic signs or bollards fade away and are barely recognizable as shown in the second and third row of Fig. 1.

In this work, we address this problem and aim to generate images with realistic looking objects at various scales from semantic image maps. This will be achieved by properly dealing with different object sizes in the generator and discriminator. While previous works [24, 25] condition each spatially-adaptive normalization block statically on the resized label map, our generator adapts the conditioning for each scale by a second pyramid as shown in Fig. 2. In this way, the conditioning is adaptive to the size of the object and artifacts of combining patches of different scales are avoided. This improves the quality of the generated objects both for small objects like signs as well as large objects like trucks as shown in Fig. 1.

For very thin structures like bollards, however, the objects tend to fade away when the objects are getting smaller as shown in Fig. 5. While our generator is able to generate such objects, previously proposed classification-based [24] or segmentation-based [25] discriminators do not handle such small details in the label map, and thus do not encourage the generator to generate small objects. While classification-based discriminators ignore such details completely, the segmentation-based discriminator proposed in [25] does not handle scale variations well and extending the discriminator to different scales even deteriorates the quality of the generated images. To address this issue, we introduce supervision at different scales. While a segmentation-based discriminator provides pixel-wise supervision, we enhance the network by two different types of multi-scale supervision. The first one is a multi-scale feature matching loss [13]. It encourages the generator to generate images that

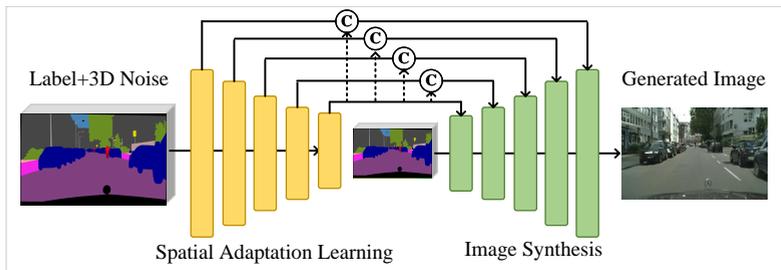


Figure 2: The dual pyramid generator takes a semantic label map concatenated with a 3D noise tensor as input and generates an image. In contrast to previous works that use only one pyramid (green) consisting of spatially-adaptive normalization (SPADE) blocks with upsampling to generate an image from a downsampled version of the input, we propose a network with two pyramids. The yellow pyramid learns the conditioning of the SPADE blocks in the green pyramid, starting from the full resolution. The scale information is in this way used bi-directionally. In addition, the output of the layer with lowest resolution in the yellow pyramid is upsampled and concatenated (denoted by ©) with the features of the previous layers.

align with the input semantic maps at all scales. The second one is multi-scale adversarial supervision at patch level. The loss ensures that important scale information is preserved within the discriminator such that the generator is enforced to generate more realistic objects at various scales. We show that the unification of the three types of supervision is essential to generate realistic objects that align well with the semantic maps.

The contribution of this work is thus two-fold: 1) We propose a dual pyramid generator for semantic image synthesis which adapts the conditioning to the size of the objects. 2) We propose to unify supervision at pixel, patch, and feature level to enforce the generator to generate realistic objects that are well aligned with the semantic maps.

We evaluate our approach on datasets with complex indoor and outdoor scenes where our approach generates realistic images and outperforms the state-of-the-art by a large margin. The source code is available at https://github.com/sj-li/DP_GAN.

2 Related Work

Generative Adversarial Networks. Approaches for semantic image synthesis rely on generative adversarial networks (GAN) [10]. The adversarial training enables the generator to produce photo-realistic images from semantic label maps. While there have been some drawbacks in early works [11] which limited their applications, several works have addressed these problems gradually. Since adversarial training is usually unstable, [10, 18, 24] increase the training stability by regularizers and different loss functions. [23] proposed some design guidelines on the architectural topology of convolutional GANs to increase the robustness during training.

Semantic Image Synthesis. Pix2pix [19] has been the first work that uses a conditional GAN [19] for semantic image synthesis. Based on it, Pix2pixHD [30] aims at generating high-resolution images. This is achieved by a multi-resolution approach. These works, however, suffer from the problem that flat segmentation maps fade away due to the usage of common normalization layers. This is solved by SPADE [21] which modulates the activa-

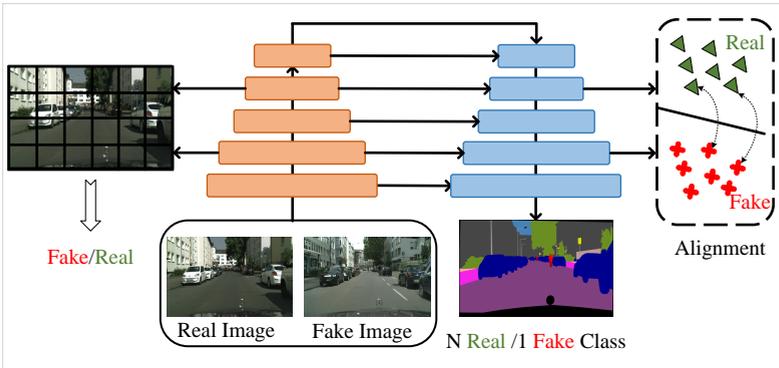


Figure 3: The proposed discriminator exploits supervision at different scales including a loss at patch level (left), feature level (top right), and pixel level (bottom right).

tions of the normalization layers by a learned affine transformation. In contrast to SPADE, CC-FPSE [17] utilizes the recent progress of dynamic filter networks and learns the convolution weight in the generator directly. A semantics-embedding discriminator is further utilized to improve fine details and boost the semantic alignment. While previous approaches used a classification-based discriminator, OASIS [25] proposed a segmentation-based discriminator, which substantially improved the alignment between generated images and semantic label maps. Apart from GAN-based methods, some other methods have been also proposed for semantic image synthesis [8, 10, 12, 16, 22, 30].

3 Methodology

In this section, we discuss the generator and discriminator, which are shown in Fig. 2 and Fig. 3, respectively.

3.1 Dual-Pyramid Generator

The proposed dual pyramid generator is shown in Fig. 2. It consists of two pyramids, one for image synthesis and another one for spatial adaptation learning. The input to the generator is a channel-wise concatenation of a label map \mathbf{I} and a 3D noise tensor \mathbf{z} as in [25]. The noise tensor can be used to generate different plausible images for the same label map as we show in Fig. 4.

The image synthesis pyramid is built from ResNet blocks [8] with spatially-adaptive normalization (SPADE) [21]. At layer i , SPADE performs a pixel-wise normalization conditioned on the input \mathbf{I} :

$$\gamma_{x,y,c}^i(\mathbf{I}) \frac{h_{x,y,c,n}^i - \mu_c^i}{\sigma_c^i} + \beta_{x,y,c}^i(\mathbf{I}) \quad (1)$$

where $h_{x,y,c,n}^i$ is the input to the normalization layer, and n denotes the training sample in the batch, c the channel, and x and y the pixel location for the resolution at layer i . μ_c^i and σ_c^i are the mean and standard deviation of $h_{x,y,c,n}^i$ in channel c , *i.e.* computed over n , x , and y .

In [21, 25], the tensors $\gamma_{x,y,c}^i(\mathbf{I})$ and $\beta_{x,y,c}^i(\mathbf{I})$ depend on the input \mathbf{I} that is downsampled to the resolution of the layer i . In this way, γ^i and β^i do not contain any information of small

objects in the early layers of the generator for image synthesis due to the low resolution of \mathbf{I}^i . As shown in Fig. 1, small objects like traffic signs thus do not look realistic and fade. Large objects like trucks also do not look realistic since \mathbf{I}^i just provides redundant information at different scales, which results in collages of patches instead of consistent objects.

We address this issue by learning γ^i and β^i for each scale jointly instead of just scaling \mathbf{I} . Since γ^i and β^i are learned, any relevant information of objects including scale information can already be encoded at early layers where the resolution is still low. This is achieved by the spatial adaptation learning pyramid shown in Fig. 2, which comprises blocks consisting of a convolutional layer, ReLU, and a batch normalization layer. We denote the output of each layer in the spatial adaptation learning pyramid as α^i . To match the layer index of the image synthesis pyramid, we use an inverse index i for the spatial adaptation learning pyramid, *i.e.* α^0 is the output of the last layer of the spatial adaptation learning pyramid, which has the lowest resolution. For $i > 0$, we concatenate α^i with an upsampled version of α^0 . The upsampling (Up) is conducted once to match the resolution of α^0 to the resolution of α^i . Finally, we apply a convolution to obtain γ^i and β^i :

$$\begin{aligned}\gamma^i &= \text{Conv}(\alpha^i \oplus \text{Up}(\alpha^0)) \\ \beta^i &= \text{Conv}(\alpha^i \oplus \text{Up}(\alpha^0)).\end{aligned}\quad (2)$$

Since the learned modulation, *i.e.* γ^i and β^i , goes from the high resolution to the low resolution, but the image generator goes in the other direction from the low resolution to the high resolution, we term it bi-directional. By learning the spatially-adaptive normalization for all scales jointly, our network generates not only a realistic background, but also realistic objects as shown in Fig. 1.

3.2 Scale-Enhancement Discriminator

Semantic image synthesis requires that the generated images are well aligned with the given semantic label map. Since commonly used classification-based discriminators [24] do not provide strong supervision for the alignment, [25] proposed a segmentation-based discriminator, which provides semantic supervision at the pixel level. Using a loss only at the pixel level, however, does not ensure realistic structures at all scales, as Fig. 1 shows. To solve this issue, we propose a discriminator that provides supervision at the pixel, patch, and feature level as illustrated in Fig. 3.

As in [25], we use an encoder-decoder architecture consisting of ResNet blocks [8]. The network predicts for each pixel the probabilities of $N + 1$ classes where N is the number of semantic classes and an additional ‘fake’ class is added. During training, the ground-truth label for each pixel is defined by the input label map for a real image and by the ‘fake’ class for a generated image. Since the classes are imbalanced, a weighted $N + 1$ -class cross-entropy loss is used:

$$\mathcal{L}_{\text{pixel}} = -\mathbf{E}_{(\mathbf{x}, \mathbf{l})} \left[\sum_{c=1}^N \alpha_c \sum_{x,y} \mathbf{1}_{x,y,c} \log D(\mathbf{x})_{x,y,c} \right] - \mathbf{E}_{(\mathbf{z}, \mathbf{l})} \left[\sum_{x,y} \log D(G(\mathbf{z}, \mathbf{l}))_{x,y,c=N+1} \right] \quad (3)$$

where \mathbf{x} is a real image, \mathbf{l} is the semantic label map, and (\mathbf{z}, \mathbf{l}) is the concatenation of the 3D noise tensor and the label map. The weight α_c is modeled as inverse of the per-pixel class

frequency to give rare classes more weight:

$$\alpha_c = E_{\mathbf{I}} \left[\frac{H \times W}{\sum_{x,y} \mathbf{I}_{x,y,c}} \right]. \quad (4)$$

Since the pixel-wise loss is insufficient as shown in Fig. 1, we add two additional loss functions. The first one is a multi-scale adversarial patch-based loss [20], which is applied to feature maps of lower resolution:

$$\mathcal{L}_{ms}^i = -\mathbf{E}_{\mathbf{x}} [\min(-1 + D_p^i(\psi^i(\mathbf{x})), 0)] - \mathbf{E}_{(\mathbf{z}, \mathbf{I})} [\min(-1 - D_p^i(\psi^i(G(\mathbf{z}, \mathbf{I}))), 0)] \quad (5)$$

where $\psi^i(\mathbf{x})$ and $\psi^i(G(\mathbf{z}, \mathbf{I}))$ are the feature maps at the layer i of the encoder of the discriminator for a real or generated image. D_p^i is a convolutional network consisting of blocks with convolutional layer, ReLU, and a batch normalization layer. It makes the prediction for each feature which corresponds to patches of different scales in the original image. The loss \mathcal{L}_{ms} is then averaged over all layers. In our implementation, we apply it after the fourth and sixth block.

The second loss is based on a multi-scale feature matching loss [13] that computes the L2 distance between the features of a real image and a corresponding generated image. While the features are extracted at the layers of the decoder of the discriminator, the loss is used for training the generator $G(\mathbf{z}, \mathbf{I})$ and added in (7). The purpose of the loss is to encourage the generator to generate images that are close to the corresponding real images in the feature space of the decoder of the discriminator. For a single layer i , the loss is defined by

$$\mathcal{L}_{fm}^i = \mathbf{E}_{(\mathbf{x}, \mathbf{I}, \mathbf{z})} \left[\frac{\sum_{x,y}^{H^i \times W^i} \|\phi^i(\mathbf{x})_{x,y} - \phi^i(G(\mathbf{z}, \mathbf{I}))_{x,y}\|_2^2}{C^i \times H^i \times W^i} \right], \quad (6)$$

where ϕ^i is the feature vector of size C^i and $H^i \times W^i$ corresponds to the resolution at layer i . As \mathcal{L}_{ms} , the loss \mathcal{L}_{fm} is averaged over all layers. In our implementation, we apply it after all blocks except for the first one.

In our experiments, we show that applying \mathcal{L}_{ms} to the encoder of the discriminator and \mathcal{L}_{fm} to the decoder performs best. In this way, \mathcal{L}_{ms} ensures that scale information is not lost in the encoder, which improves the quality of generated objects at all scales, and \mathcal{L}_{fm} ensures a better alignment of the generated images with real images.

3.3 Training

The generator and discriminator are trained jointly. For the generator, we use the loss:

$$\begin{aligned} \mathcal{L}_G = & -\mathbf{E}_{(\mathbf{z}, \mathbf{I})} \left[\sum_{c=1}^N \alpha_c \sum_{x,y}^{H \times W} \mathbf{I}_{x,y,c} \log D(G(\mathbf{z}, \mathbf{I}))_{x,y,c} \right] \\ & - \frac{1}{L} \sum_{i=1}^L \mathbf{E}_{(\mathbf{z}, \mathbf{I})} [\min(-1 + D_p^i(\psi^i(G(\mathbf{z}, \mathbf{I}))), 0)] + \mathcal{L}_{fm} \end{aligned} \quad (7)$$

where each term corresponds to supervision at pixel, patch, and feature level. Furthermore, we include the LabelMix regularization proposed in [25] for a fair comparison:

$$\mathcal{L}_{LM} = \|D_{logits}(\text{LabelMix}(\mathbf{x}, \hat{\mathbf{x}}, M)) - \text{LabelMix}(D_{logits}(\mathbf{x}), D_{logits}(\hat{\mathbf{x}}), M)\|_2^2 \quad (8)$$

Methods	Cityscapes		ADE20K		ADE20K-Outdoor	
	FID ↓	mIoU ↑	FID ↓	mIoU ↑	FID ↓	mIoU ↑
CRN [9]	104.7	52.4	73.3	22.4	99.0	16.5
pix2pixHD [80]	95.0	58.3	81.8	20.3	97.8	17.4
SPADE [21]	71.8	62.3	33.9	38.5	63.3	30.8
DAGAN [22]	60.3	66.1	31.9	40.5	N/A	N/A
LGGAN [23]	57.7	68.4	31.6	41.6	N/A	N/A
CC-FPSE [10]	54.3	65.5	31.7	43.7	N/A	N/A
SIMS [24]	49.7	47.2	N/A	N/A	67.7	13.1
OASIS [25]	47.7	69.3	28.3	48.8	48.6	40.4
DP-GAN	44.1	73.6	26.1	52.7	45.8	40.4

Table 1: Comparison to state-of-the-art methods on different datasets. Lower FID and higher mIoU are better. Bold denotes the best performance.

	road	swalk	build	wall	fence	pole	<u>flight</u>	<u>sign</u>	veg	terrain	sky	<u>person</u>	<u>rider</u>	<u>car</u>	<u>truck</u>	<u>bus</u>	<u>train</u>	<u>mbike</u>	<u>bike</u>	obj-mIoU
SPADE [10]	97.5	80.8	88.5	54.3	50.6	40.4	39.0	41.9	88.7	69.1	92.0	66.2	41.5	89.1	64.6	73.2	42.1	29.7	61.5	53.6
DAGAN [22]	97.4	80.0	89.0	60.1	53.7	41.2	39.4	46.5	88.9	65.9	92.5	66.8	45.8	89.9	71.2	75.4	57.0	25.8	60.9	56.4
CC-FPSE [10]	97.7	82.8	89.8	56.1	61.3	42.3	41.8	50.4	89.6	69.3	92.5	68.5	48.3	90.2	69.7	74.3	45.4	43.4	65.0	58.1
LGGAN [23]	97.8	83.1	89.7	59.8	56.0	42.5	42.8	50.5	89.5	70.0	92.7	69.0	48.6	90.6	72.2	80.2	52.4	38.8	64.0	59.2
OASIS [25]	96.9	79.2	85.1	70.3	64.2	41.6	50.7	49.9	85.0	74.8	92.0	64.9	54.0	88.4	65.6	79.9	63.4	53.9	63.7	61.5
DP-GAN	97.5	81.9	87.2	71.4	72.7	46.9	55.5	60.3	87.3	72.9	92.4	67.4	55.5	89.9	81.5	83.1	73.9	55.3	66.9	66.9

Table 2: Per-class IoU for Cityscapes. Classes corresponding to objects, *i.e.* the object, human, and vehicle group of Cityscapes, are underlined. obj-mIoU is mIoU only for object classes. mIoU for all classes is reported in Table 1.

where D_{logits} are the logits before the last softmax and M is a binary mask for mixing real and generated images $(\mathbf{x}, \hat{\mathbf{x}})$ by LabelMix $(\mathbf{x}, \hat{\mathbf{x}}, M) = M \odot \mathbf{x} + (1 - M) \odot \hat{\mathbf{x}}$. The entire loss is thus given by

$$\mathcal{L} = \mathcal{L}_G + \mathcal{L}_{pixel} + \mathcal{L}_{ms} + \lambda_{LM} \mathcal{L}_{LM}. \quad (9)$$

More details regarding the network are provided in the supplemental material.

4 Experiments

Datasets. We evaluate our method on the two challenging datasets Cityscapes [9] and ADE20K [53], including the ADE20K-outdoor subset. We follow the protocol used in [21, 25]. The Cityscapes dataset comprises complex street images recorded by a driving car. It contains 2975 training images and 500 validation images with 35 classes, which are mapped to 19 classes for evaluating the semantic alignment. While the resolution of the images is 1024×2048 , we resized them to 256×512 as in previous works for a fair comparison. The ADE20K dataset is a large-scale dataset aiming at scene understanding. It contains 20210 training images and 2000 validation images with 150 classes. The images are resized to 256×256 . The ADE20K-outdoor dataset is a subset of the ADE20K dataset and contains only outdoor scenes. It contains 9649 training images and 943 validation images.

Evaluation Metrics. We use Fréchet Inception Distance (FID) [9] and mean Intersection-over-Union (mIoU) for evaluation as previous works [21, 25]. mIoU is used to measure the semantic consistency between generated and real images via a pre-trained semantic segmentation network. As in [21, 25], UperNet101 [6, 61] is used for ADE20K and DRN [6, 62] for Cityscapes. FID [9] is widely used to measure the quality of generated images and we use the implementation from [25].

	OASIS [25]	DP-GAN
Cityscapes	48.1±0.18	44.6±0.17
ADE20K	29.0±0.03	26.7±0.09

Table 3: The mean and variance of FID computed over 5 runs. In this experiment, we change the noise tensor z to generate diverse images as shown in Fig. 4.

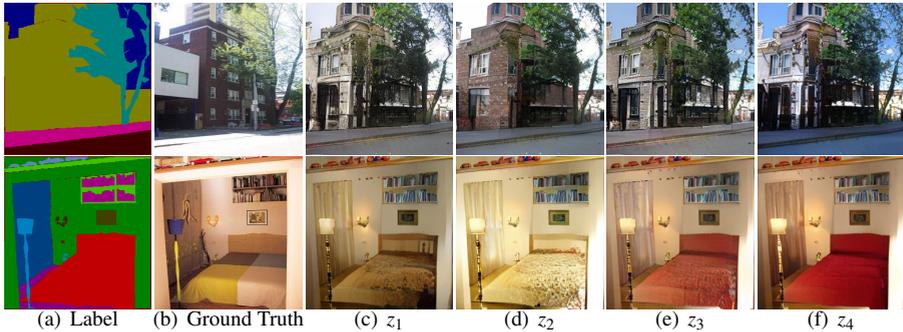


Figure 4: Images generated with four different random noise tensors z .

4.1 Comparison to State-of-the-Art

We compare our method to state-of-the-art methods in Table 1. Our method (DP-GAN) achieves the best performance, measured by FID and mIoU, on all datasets. On Cityscapes, our method outperforms previous state-of-the-art methods by -3.6 FID and +4.3 mIoU. On the ADE20K dataset, our approach achieves an improvement by -2.2 FID and +3.9 mIoU. Only on the ADE20K-outdoor dataset, the mIoU of our approach is the same as OASIS [25]. This is not surprising since the outdoor images contain less objects as shown in Fig. 6 and our goal is to generate photo-realistic objects in complex and cluttered scenes. Nevertheless, our approach achieves -2.8 FID compared to OASIS and thus generates more realistic images. In order to analyze the effect on object classes and other semantic classes more in detail, we report the per-class IoU for Cityscapes in Table 2. Except for the class fence, the IoU for non-object classes is similar to other approaches, but our approach achieves a much higher IoU for object classes. In particular for the small object classes pole, traffic light, and sign and the large object classes truck, bus, and train, the improvements in IoU are by a very large margin. For all object classes, the mIoU is improved by +5.4, which is higher than the improvement for all classes.

We show some qualitative results in Fig. 1 and 6. We observe that objects are often not well generated by SPADE and the clear boundaries of semantic classes often disappear in particular for Cityscapes. The quality of the images that are generated by OASIS is much better, but a common artifact is the merging of multiple instances like parking cars or two beds that are close. Our approach does not suffer from this issue and generates realistic looking objects.

While in the previous experiments we used the same noise tensor z in all experiments, our approach can generate diverse images for the same input label map by generating different random noise tensors z as shown in Fig. 4. For a quantitative comparison with OASIS, which generates diverse images in the same way, we report mean and variance of FID over 5 runs with different noise tensors. The results are reported in Table 3.

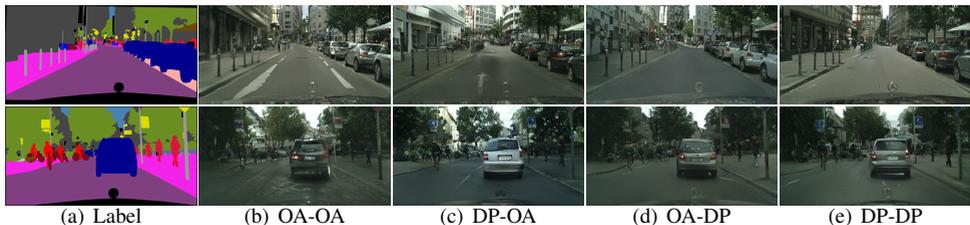


Figure 5: Impact of the proposed generator and discriminator. DP or OA denote if the generator or discriminator from OASIS (OA) or our approach (DP) are used, *e.g.* DP-OA means that our generator is combined with the discriminator from OASIS. First row: Given the input label map (a), the image generated by OASIS (b) looks reasonable but when zooming in we observe that the parking cars on the right hand side contain strong artifacts and the five bollards on the left hand side fade away. Using the proposed generator (c) improves the quality of the image and the parking cars look realistic. Small objects like the bollards, however, still fade away. If we use only our discriminator (d), the bollards are clearly visible but the building on the left hand side looks unrealistic. Only our full model (e) generates an image of high fidelity. The parking cars on the right hand side and the building and bollards on the left hand side look realistic. Second row: Although there is only one large car in the label map (a), the generator of OASIS overlays a small car over a large car (d), which becomes visible when zooming in. Whereas the proposed generator generates only one car (c) and the quality of the car is further improved by the proposed discriminator (e).

(a) Gen / Dis					(b) \mathcal{L}_{ms} in (5)					(c) \mathcal{L}_{fm} in (6)				
Gen	Dis	FID	mIoU	obj-mIoU	Enc	Dec	FID	mIoU	obj-mIoU	Enc	Dec	FID	mIoU	obj-mIoU
OA	OA	47.7	69.3	61.5			49.2	67.9	59.5			44.1	69.9	62.1
OA	DP	47.9	74.0	67.4		✓	44.5	72.1	64.4	✓		44.4	69.2	60.8
DP	OA	45.4	69.9	62.0	✓	✓	44.3	72.8	66.4	✓	✓	45.0	73.8	66.8
DP	DP	44.1	73.6	66.9	✓		44.1	73.6	66.9		✓	44.1	73.6	66.9

Table 4: Ablation study. ‘OA’ denotes OASIS [25] whereas ‘DP’ denotes our method (DP-GAN). Using ‘OA’ as generator corresponds to a generator with a single pyramid.

4.2 Ablation Study

We analyze the impact of the proposed generator and discriminator in Table 4(a). To this end, we combine the proposed generator and discriminator with the generator and discriminator of OASIS [25]. Replacing the discriminator of OASIS by our discriminator substantially improves mIoU by +4.7. For the object classes, mIoU even improves by +5.9. FID, however, does not significantly change. In contrast, replacing the generator of OASIS with our generator substantially decreases FID by -2.3, but increases mIoU only slightly. This shows that the combination of the proposed generator and discriminator is essential to achieve a low FID and a high mIoU. Fig. 5 illustrates the effect of the generator and discriminator on the generated images.

Next, we evaluate the impact of multi-scale adversarial supervision (\mathcal{L}_{ms}) in Table 4(b). The first row shows the result without the loss \mathcal{L}_{ms} . For the other rows, the loss is either added to the encoder or decoder of the discriminator or to both. Adding \mathcal{L}_{ms} improves FID and mIoU. This shows that adding \mathcal{L}_{ms} to a segmentation-based discriminator improves the quality of the generated images substantially. The differences of adding the loss to the encoder, decoder, or both are smaller, but using it only for the encoder performs best.



Figure 6: Qualitative results for ADE20K. Our approach performs well for indoor and outdoor images. SPADE and OASIS struggle to generate multiple instances of objects, like one of the chairs in row 2 is either of poor quality or in the wrong direction, and the two beds in row 3 are connected. In contrast, our approach generates in all cases realistic images.

Table 4(c) evaluates the impact of using feature alignment (\mathcal{L}_{fm}). While it does not improve the FID, the mIoU improves by +3.7 if the feature matching loss is added to the decoder of the discriminator. In contrast, adding the feature matching loss to the encoder decreases the performance. This result is interesting since it has been reported in [25] that adding a feature matching loss to a segmentation-based discriminator decreases the performance. Our results, however, show that it is important where the feature matching loss is added. For the decoder, it actually increases the performance. For more qualitative results and ablation studies, we refer to the supplemental material.

5 Conclusion

In this paper, we have addressed the problem that previous methods for semantic image synthesis struggle to generate photo-realistic objects at different scales in complex scenes. To this end, we proposed a dual pyramid architecture for the generator that adapts the conditioning depending on the object size. As a second contribution, we proposed to unify multi-scale supervision within a single framework. The proposed approach outperforms the state-of-the-art for semantic image synthesis by a large margin. While the quality of the generated images has been substantially improved compared to the state-of-the-art, the qualitative results contain still some artifacts. Although semantic image synthesis is important for image generation and editing tasks and applications like autonomous driving in order to generate additional training data for rare situations, such technology can be potentially misused for disinformation and propaganda.

Acknowledgement The work has been supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2070 - 390732324 and GA 1927/5-2 (FOR 2535 Anticipating Human Behavior).

References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.
- [2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019.
- [3] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, pages 1511–1520, 2017.
- [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*, pages 3213–3223, 2016.
- [5] CSAILVision. Pytorch implementation for semantic segmentation/scene parsing on mit ade20k dataset. In <https://github.com/CSAILVision/semantic-segmentation-pytorch.git>, 2019.
- [6] fyu. Pytorch implementation for dilated residual network. In <https://github.com/fyu/drn.git>, 2019.
- [7] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [9] 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. *arXiv preprint arXiv:1706.08500*, 2017.
- [10] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, pages 172–189, 2018.
- [11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.
- [12] Liming Jiang, Changxu Zhang, Mingyang Huang, Chunxiao Liu, Jianping Shi, and Chen Change Loy. Tsit: A simple and versatile framework for image-to-image translation. In *ECCV*, pages 206–222. Springer, 2020.
- [13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.
- [14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.
- [15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

- [16] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, pages 35–51, 2018.
- [17] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and Hongsheng Li. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. *NeurIPS*, 2019.
- [18] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2794–2802, 2017.
- [19] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [20] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.
- [21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019.
- [22] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *CVPR*, pages 8808–8816, 2018.
- [23] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.
- [25] Edgar Schönfeld, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *ICLR*, 2021.
- [26] Vadim Sushko, Edgar Schönfeld, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. Oasis: Only adversarial supervision for semantic image synthesis. *IJCV*, 2022.
- [27] Hao Tang, Song Bai, and Nicu Sebe. Dual attention gans for semantic image synthesis. In *ACM MM*, pages 1994–2002, 2020.
- [28] 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 *CVPR*, pages 7870–7879, 2020.
- [29] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.
- [30] 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*, pages 8798–8807, 2018.

-
- [31] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018.
 - [32] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, pages 472–480, 2017.
 - [33] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017.