Image Synthesis via Semantic Composition

Yi Wang[1] Lu Qi[1] Ying-Cong Chen[2] Xiangyu Zhang[3] Jiaya Jia[1,4]
[1]CUHK [2]HKUST [3]MEGVII Technology [4]SmartMore
{yiwang,luqi,leojia}@cse.cuhk.edu.hk yingcong.ian.chen@gmail.com zhangxiangyu@megvii.com

Abstract

In this paper, we present a novel approach to synthesize realistic images based on their semantic layouts. It hypothesizes that for objects with similar appearance, they share similar representation. Our method establishes dependencies between regions according to their appearance correlation, yielding both spatially variant and associated representations. Conditioning on these features, we propose a dynamic weighted network constructed by spatially conditional computation (with both convolution and normalization). More than preserving semantic distinctions, the given dynamic network strengthens semantic relevance, benefiting global structure and detail synthesis. We demonstrate that our method gives the compelling generation performance qualitatively and quantitatively with extensive experiments on benchmarks.

1. Introduction

Semantic image synthesis transforms the abstract semantic layout to realistic images, an inverse task of semantic segmentation (Figure 1). It is widely used in image manipulations and content creation. Recent methods on this task are built upon generative adversarial networks (GAN) [7], modeling image distribution conditioning on segmentation masks.

Despite its substantial achievement [14, 31, 24, 21, 29, 4, 26, 16, 30], this line of research is still challenging due to the high complexity of characterizing object distributions. Recent advance [24, 21] on GAN-based image synthesis concentrates on how to exploit spatial semantic variance in the input for better preserving layout and independent semantics, leading to further generative performance improvement. They both use different parametric operators to handle different objects.

Specifically, SPADE [24] proposes a spatial semantics-dependent denormalization operation for the common normalization, as the feature statistics are highly correlated with semantics. CC-FPSE [21] extends this idea to convolution using dynamic weights, generating spatially-variant convolutions from the semantic layouts. Relationships between objects are implicitly modeled by the weight-sharing convolutional kernels (SPADE) or hierarchical structures brought by stacked convolutions. We believe when performing semantic-aware operations, enhancing object relationship could be further beneficial to final synthesis, since context and long-range dependency have proven effective in several vision tasks [6, 32, 43, 41, 34].

Figure 1. Semantic image synthesis results of our method on face and scene datasets.

To explicitly build connection between objects and stuff in image synthesis, we propose a semantic encoding and stylization method, named semantic-composition generative adversarial network (SC-GAN). We first generate the *semantic-aware* and *appearance-correlated* representations from mapping the discrete semantic layout to their corresponding images.

Our idea is inspired by the following observation. Different semantics are with labels in scene or face parsing datasets. Some of them are highly correlated in appearance, e.g., the left and right eyes in CelebAMask-HQ [19]. We abstract objects in images or feature maps into vectors by encoding the semantic layout, which we call semantic vectors. This intermediate representation is to characterize the relationship between different semantics based on their appearance similarity.

With these semantic vectors, we create *semantic-aware and appearance-consistent local operations in a semantic-stylization fashion*. Consistent with the proposed semantic vectors, these dynamic operators are also variant in se-

mantics and correlated in appearance. Our proposed operators are extended from existing conditional computation [38] to stylize the input conditioned on the semantic vectors. Specifically, we exploit semantic vectors to combine a shared group of learnable weights, parameterizing the convolutions and normalizations used in semantic stylization.

Note the learning of semantic vectors and render candidates is non-trivial. Intuitive designs that encode the semantic layout to semantic vectors for later dynamic computations make semantic vectors stationary, since these semantic vectors are not directly regularized by the appearance information in the image. In this paper, we make the learning of semantic encoding and stylization relatively independent. Semantic encoding is trained by estimating the corresponding natural images from the input semantic layouts by maximum likelihood. Semantic stylization is trained in an adversarial manner.

Our method is validated on several image synthesis benchmark datasets quantitatively and qualitatively. Also, the decoupled design of semantics encoding and stylization makes our method applicable to other tasks, e.g., unpaired image translation [46]. Our contribution is threefold.

- We propose a new generator design for GAN-based semantic image synthesis. We present spatially-variant and appearance-correlated operations (both convolution and normalization) via spatially conditional computation, exploiting both local and global characteristics for semantic-aware processing.
- Our proposed generator with a compact capacity outperforms other popular ones on face and scene synthesis datasets in visual and numerical comparisons.
- The decoupled design in our method finds other applications, *e.g.* unpaired image-to-image translation.

2. Related Work

2.1. Semantic Image Synthesis

This task is to create a realistic image based on the given semantic layout (pixel-level semantic labels). Essentially, it is an ill-posed problem as one semantic layout may correspond to several feasible pictures. It can be dated back to 'Image analogy' in 2001 [10], in which the mentioned mapping uncertainties are resolved by the local matching and constraints from a reference image.

Recent learning-based approaches [14, 31, 35, 24, 21, 29, 36, 4, 26, 16, 30, 33, 47, 28] greatly advance this area, formulating it as an image distribution learning problem conditioned on the given semantic maps. Due to the development of conditional generative adversarial networks (cGAN) [22], pix2pix makes seminal exploration on image synthesis [14]. It gives some components and principles about how to apply cGAN to this problem, including loss design, generator structures (e.g., encoder-decoder and U-Net), and Markovian discriminator (also known as PatchGAN).

Later, Wang *et al*. propose a new enhanced version pix2pixHD [31]. By introducing a U-Net style generator with a larger capacity and several practical techniques for improving GAN training, including feature matching loss, multi-scale discriminators, and perceptual loss, their method boosts the image synthesis performance on producing vivid details. Further, SPADE is developed on improving the realism of the synthesized images by working on the normalization issue [24]. It shows using the given segmentation masks to explicitly control the affine transformation in the used normalization can better preserve the generated semantics, leading to a noticeable improvement. Such an idea is further extended in CC-FPSE [21]. It dynamically generates convolutional kernels in the generator conditioning on the input semantics.

Besides of GAN-based methods, research explores this task from other perspectives. Chen *et al*. [4] produce images using cascaded refinement networks (CRN) progressively with regression, starting from small-scale semantic layouts. Qi *et al*. [26] proposed a semi-parametric approach to directly utilize object segments in the training data. They retrieve object segments with the same semantics and similar shapes from the training set to fill the given semantic layout, and then regress these assembled results to the final images. Additionally, Li *et al*. [20] employ implicit maximum likelihood estimation to CRN, to pursue more diverse results from a semantic layout.

2.2. Dynamic Computation

In the development of neural network components, dynamic filters [15] or hypernetworks [8] are proposed for their flexibility to input samples. It generates dynamic weights conditioned on the input or input-related features for parameterizing some operators (mostly fully-connected layers or convolutions). This has been applied to many tasks [39, 25, 12, 2, 24, 21, 17].

Conditionally Parameterized Convolutions It is a special case of dynamic filters, which produces dynamic weights by combining the provided candidates conditionally [38]. It uses the input features $\mathbf{X}$ to generate the input-dependent convolution kernels in the neural nets by a learnable linear combination of a group of kernels $\{\mathbf{w}\}_{1,\dots,n}$, as $(\sum_{i=1}^{n} \alpha_i(\mathbf{X})\mathbf{w}_i)$, where $\alpha_i(\mathbf{X})$ computes an input-dependent scalar to choose the given kernel candidates, and n is the expert number. It is equivalent to a linear mixture of experts, with more efficient implementation.

In this paper, our proposed dynamic computation units are extended from conditionally parameterized convolutions. We generalize the scalar condition into a spatial one and also apply these techniques to normalization.

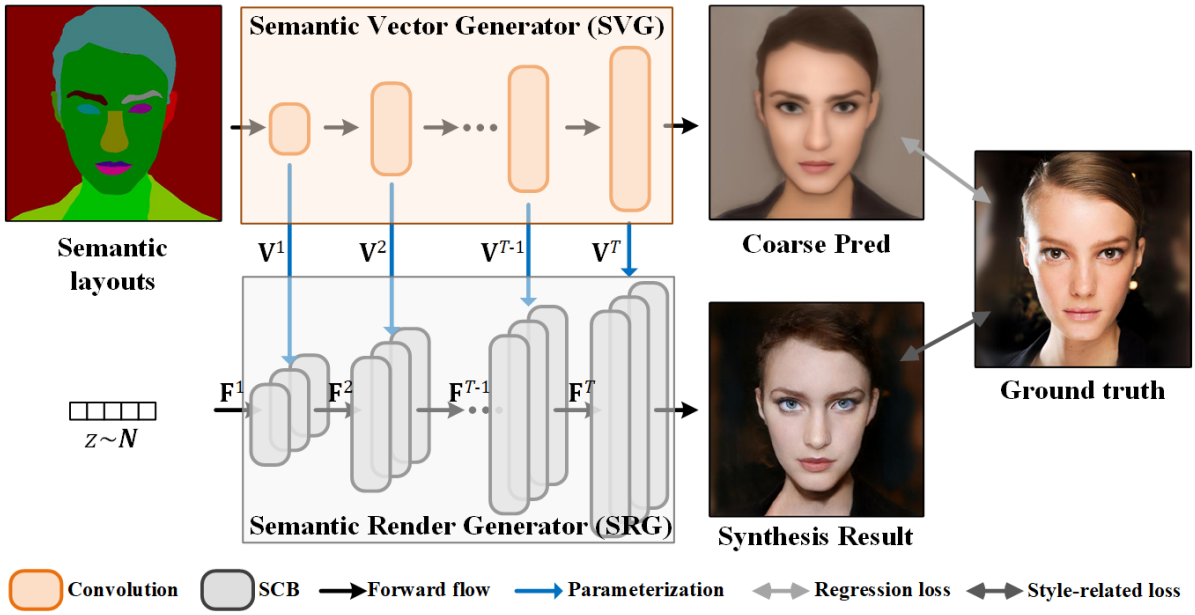


Figure 2. Framework of SC-GAN. SCB denotes the spatially conditional block. It is constructed by spatial conditional convolutions and normalizations, parameterized by semantic vectors $\mathbf{V}$. Its design is given in Figure 4 and Section 3.2.

3. Semantic-Composition GAN

We aim to transform a semantic layout $\mathbf{S} \in \{0,1\}^{h\times w\times c}$ to a realistic picture $\hat{\mathbf{I}} \in \mathbb{R}^{h\times w\times 3}$ (where h, w, and c denote the height, width, and the category number in semantic layout, respectively), as $\hat{\mathbf{I}} = f(\mathbf{S})$, in which f is a nonlinear mapping. During training, paired-wise data is available given the corresponding natural image $\mathbf{I}$ of $\mathbf{S}$ provided. We demand synthesized image $\hat{\mathbf{I}}$ to match the given semantic layout. But $\hat{\mathbf{I}}$ and $\mathbf{I}$ are not necessarily identical.

Our semantic-composition GAN (SC-GAN) decouples semantic image synthesis into two parts: semantic encoding and stylization. They are realized by semantic vector generator G_V (SVG) and semantic render generator G_R (SRG), respectively. As shown in Figure 2, SVG takes the semantic layout $\mathbf{S}$ and produces multi-scale semantic vectors in a feature map form (since we treat each feature point as a semantic vector). SRG is to transform a random sampled noise to the final synthesized image with a dynamic network. The key operators (convolution and normalization) in this network are conditionally parameterized by the semantic vectors provided by SVG and a group of weight candidates.

3.1. Semantic Vectors Generation

SVG is to transform the input discrete semantic labels $\mathbf{S}$ into semantic-and-appearance correlated representation of semantic vectors, building the relationship between different semantics according to their appearance similarities. For example, grass and trees are represented by different semantic labels in COCO-stuff [1], sharing similarities in color and texture. Our method represents their corresponding regions with different but similar representations.

Generally, semantic vectors are learned from encoding of the input semantic labels into the corresponding image. SVG takes input of the semantic layout $\mathbf{S}$ and generates the corresponding semantic vectors in feature map form as $\mathbf{V} \in [0,1]^{h'\times w'\times n}$ with different scales. It is expressed as

$$\{\mathbf{V}^t\}_{t=1,\dots,T} = f_{\mathrm{V}}(\mathbf{S}), \tag{1}$$

where t denotes a different spatial scale.

SVG is in a cascaded refinement form, structured as CRN [4]. We feed a small-scale semantic layout into it, encode the input and upsample the features, and concatenate it with a larger-scale semantic layout. We repeat this process until the output reaches the final resolution.

Nonlinear Mapping We regularize the computed semantic vectors using a nonlinear mapping. Directly employing the these unconstrained vectors would lead to performance drop (shown in Section 4.3). Suppose $\mathbf{v} = \mathbf{V}_{i,j} \in \mathbb{R}^n$, where i and j index height and width. We further normalize its values into $[0,1]$ with softmax for better performance and interpretability as

$$g(\tau\mathbf{v})_i = \frac{\exp(\tau\mathbf{v}_i)}{\sum_{j=1}^{n}\exp(\tau\mathbf{v}_j)}, \tag{2}$$

where $\mathbf{v}_i$ denotes the ith scalar in $\mathbf{v}$, and $\tau = 0.05$ is the temperature to control smoothness of the semantic vectors $\mathbf{v}$. The smaller τ is, the smoother $\mathbf{v}$ is. Note $g(\cdot)$ could be sigmoid, tanh, or other nonlinear functions. The performance along with our choices will be empirically compared and analyzed in the Section 4.3.

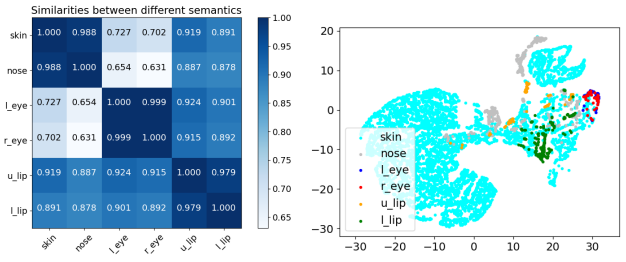


Figure 3. Feature correlation matrix between semantic vectors of different semantics (left) and 2D feature distributions (compressed by t-SNE) of semantic vectors (right) on CelebAMask-HQ.

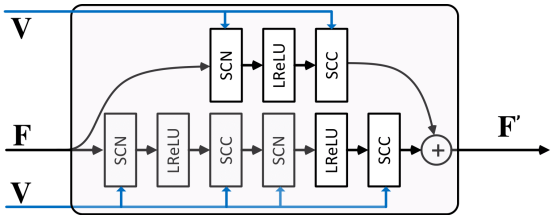


Figure 4. Employed residual block using the proposed spatially conditional convolution (SCC) and normalization (SCN).

Effectiveness of Semantic Vectors We visualize the correlation between different semantic regions using semantic vectors. It is found that these vectors are related by the appearance similarity. Figure 3 (left) shows cosine similarities between mean semantic vectors of different semantics. Note the semantic vectors representing the left and right eyes are almost identical (with cosine similarity 0.999). This is also observed in the relationship between the upper and lower lips. Intriguingly, it also reveals that the semantic vectors for skin are close to those of the nose.

Figure 3 (right) illustrates how these semantic vectors are distributed (compressed to 2D by t-SNE, only 1% points are visualized for clarity). Note that blue points standing for left eyes are much overlapped with those of right eyes. Also, the orange point cluster is close to the green one for upper and lower lips. It validates that semantic vectors can give similar representations to different semantic regions with a similar appearance. These semantic vectors are extracted from 100 random images of CelebAMask-HQ.

3.2. Semantic Render Generation

SRG is a generator in a progressive manner, built with the residual block [9] (Figure 4) formed by our proposed spatial conditional convolution (SCC, left of Figure 5) and spatial conditional normalization (SCN, right of Figure 5). It transforms random noise z into the target image $\hat{\mathbf{I}}$, conditioned on $\{\mathbf{V}^t\}$ of $\mathbf{S}$. It is formulated as

$$\hat{\mathbf{I}} = f_{\mathrm{R}}(z|\{\mathbf{V}^t\}), \tag{3}$$

where $z \sim \mathcal{N}$ and $\mathcal{N}$ is a standard normal distribution.

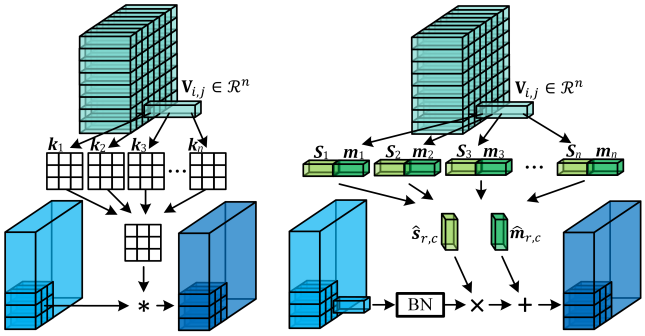


Figure 5. Conceptual illustration of spatial conditional computation. Left: spatially conditional convolution (SCC), right: spatially conditional normalization (SCN).

Both SCC and SCN are spatially conditional, parameterized from semantic vectors and a shared group of weights, making them produce semantic-aware and appearance-correlated operators. Their designs are detailed below.

Spatially Conditional Convolution For input feature maps $\mathbf{F} \in \mathbb{R}^{h\times w\times c}$ (intermediate representation of z in SRG), assuming its learnable kernel candidates are $\{\mathbf{k}_1, \mathbf{k}_2, ..., \mathbf{k}_n\}$, with semantic vectors $\mathbf{V} \in \mathbb{R}^{h\times w\times n}$, we compute the corresponding output as

$$\mathrm{scc}(\mathbf{F}, \mathbf{V}; \{\mathbf{k}_i\}_{i=1,...,n})_{r,c} = \sum_{i=1}^{n} (\mathbf{V}_{r,c,i}\mathbf{k}_i) * \mathbf{F}_{r,c}, \tag{4}$$

where r and c indicate the row and column indexes of the given feature maps, and i indexes both the channel of $\mathbf{V}$ and the weight candidate.

Spatially Conditional Normalization Similarly, the spatially conditional normalization employs linearly combined mean and variance for the affine transformation after normalization. Still for $\mathbf{F}$ and $\mathbf{V}$, suppose its learnable mean and variance candidates are $\{\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_n\}$ and $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n\}$, we yield the normalized output as

$$\begin{aligned} &\mathrm{scn}(\mathbf{F}, \mathbf{V}; \{\mathbf{m}_i, \mathbf{s}_i\})_{r,c} = \frac{\mathbf{F}_{r,c} - \mu(\mathbf{F})}{\sigma(\mathbf{F})} \times \hat{\mathbf{s}}_{r,c} + \hat{\mathbf{m}}_{r,c}, \\ &\text{where} \quad \hat{\mathbf{s}}_{r,c} = \sum_{i=1}^{n} (\mathbf{V}_{r,c,i}\mathbf{s}_i), \ \hat{\mathbf{m}}_{r,c} = \sum_{i=1}^{n} (\mathbf{V}_{r,c,i}\mathbf{m}_i), \end{aligned} \tag{5}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are to compute the mean and standard variance of their input, respectively. Mean and variance candidates are initialized to 0 and 1, respectively.

Analysis Inheriting from the spatially adaptive processing idea from SPADE [24] in semantic image synthesis, our model generates semantic-aware convolutions and normalization to handle different semantic regions indicated by the input. This idea is also explored in CC-FPSE [21], which employs segmentation masks to parameterize the conditional convolution directly by generating weights. Although

it improves the generation quality of SPADE, the computation is expensive, since its independent spatially variant convolutional operations can only be implemented using local linear projection instead of standard convolution.

In contrast, our operators are correlated when they cope with regions with similar appearance and yet with different labels, achieved by the semantic vectors and the shared group of weights. Besides, our SCC utilizes standard convolutions for efficient training and evaluation. As validated in our experiments, our design is beneficial to long-range dependency modeling, consistently improving generation performance.

3.3. Learning Objective

The optimization goal of our method consists of two parts with style-related loss and regression loss. The former contains perceptual loss, GAN loss, and feature matching loss. Regression loss on SVG is to learn semantic-aware and appearance-consistent vectors, named semantic vector generation loss. In general, our optimization target is

$$\mathcal{L} = \lambda_p \mathcal{L}_p + \lambda_{gan} \mathcal{L}^G_{gan} + \lambda_{fm} \mathcal{L}_{fm} + \lambda_s \mathcal{L}_s, \quad (6)$$

where λ_p, λ_{gan}, λ_{fm}, and λ_{rwg} are trade-off regularization parameters, set to 10, 1, 10, and 2, respectively.

Perceptual Loss We employ the pretrained natural image manifold to constrain the generative space of our model. Specifically, we minimize the discrepancy between the produced images and their corresponding ground truth in the feature space of VGG19 [27] as

$$\mathcal{L}_p(\hat{\mathbf{I}}, \mathbf{I}) = \sum_{i=1}^{5} ||\Phi_{i\_1}(\hat{\mathbf{I}}) - \Phi_{i\_1}(\mathbf{I})||_1, \quad (7)$$

where Φ_{i_1} indicates extracting the feature maps from the layer ReLU i_1 of VGG19.

GAN Loss We train our generator and discriminator using hinge loss. Its learning goal on generator is

$$\mathcal{L}^G_{gan} = -\mathrm{E}_{z\sim\mathbb{P}_z, \mathbf{S}\sim\mathbb{P}_{\text{data}}} D(G(\mathbf{S}|z)), \quad (8)$$

where G and D denote the generator (including SVG and SRG) and discriminator of our SC-GAN, respectively. The corresponding optimization goal for the discriminator is

$$\begin{aligned} \mathcal{L}^D_{gan} =& \mathrm{E}_{\mathbf{I}\sim\mathbb{P}_{\text{data}}}[\max(0, 1 - D(\mathbf{I}))] + \\ & \mathrm{E}_{z\sim\mathbb{P}_z, \mathbf{S}\sim\mathbb{P}_{\text{data}}}[\max(0, 1 + D(G(\mathbf{S}|z)))]. \end{aligned} \quad (9)$$

For the discriminator, we directly employ the feature pyramid semantics-embedding discriminator from [21].

Semantic Vector Generation Loss To prevent SVG generates trivial semantic vectors for later generation, we regularize its learning by predicting the corresponding information of the fed semantic layout as

$$\mathcal{L}_s = ||f^{\text{out}}_{\text{V}}(\mathbf{S}) - \mathbf{I}||_p, \quad (10)$$

where $f^{\text{out}}_{\text{V}}(\mathbf{S})$ denotes the predicted image from SVG. $p = 1$ or 2. Note that we can also conduct such measure in the perceptual space of a pretrained classification network like Eq. (7). This is studied in Section 4.3.

3.4. Implementation

We apply spectral normalization (SN) [23] both on the generator and discriminator. Also, a two time-scale update rule [11] is used during training. The learning rates for the generator and discriminator are $1e-4$ and $4e-4$, respectively. The training is conducted with Adam [18] optimizer with $\beta_1 = 0$ and $\beta_2 = 0.999$. For the used batch normalization, all statistics are synchronized across GPUs. Unless otherwise specified, the used candidate number n from Eqs. (4) and (5) of our method is set to 3 in experiments.

4. Experiments

Our experiments are conducted on four face and scene parsing datasets: CelebAMask-HQ [19], Cityscapes [5], ADE20K [45], and COCO-Stuff [1]. In our experiments, images and their corresponding semantic layouts in CelebAMask-HQ, Cityscapes, ADE20K, and COCO-Stuff are resized and cropped into 512×512, 256×512, 256×256, 256×256, respectively. The train/val splits follow the setting of these datasets.

For training of our method, we take 100, 200, 200, and 100 epochs on CelebAMask-HQ, Cityscapes, ADE20K, and COCO-Stuff, respectively. The first half of epochs on the first three datasets are with full learning rates and the remaining half linearly decays with learning rates approaching 0. Our computational platform is with 8 NVIDIA 2080Ti GPUs.

Baselines We take CRN [4], SIMS [26], Pix2pixHD [31], SPADE [24], Mask-GAN [19], and CC-FPSE [21] as baselines. The following evaluation is all with their original implementation and pretrained models from their official release. Some new results on CelebAMask-HQ from SPADE are trained from scratch with their default training setting. CC-FPSE is not applicable here since it consumes more GPU memory on the face dataset than we can afford. The reason is that it realizes conditional convolution with spatially independent local linear processing, leading to considerable computational overhead. Note that Mask-GAN is developed for facial image manipulation in style transfer, which exploits both the semantic layout and a reference image. Its qualitative and quantitative evaluation is only for reference because it uses extra input information.

About the model complexity, our SC-GAN has the minimal capacity of 66.2M parameters, compared to referred numbers of parameters of 183.4M for Pix2pixHD, 93.0M for SPADE, and 138.6M for CC-FPSE.

Table 1. Quantitative results on the validation sets from different methods.

Method	CelebAMask-HQ		Cityscapes			ADE20K			COCO-Stuff		
	FID ↓	LPIPS ↓	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓
CRN [4]	N/A	N/A	52.4	77.1	104.7	22.4	68.8	73.3	23.7	40.4	70.4
SIMS [26]	N/A	N/A	47.2	75.5	49.7	N/A	N/A	N/A	N/A	N/A	N/A
pix2pixHD [31]	54.7	0.529	58.3	81.4	95.0	20.3	69.2	81.8	14.6	45.7	111.5
SPADE [24]	42.2	0.487	62.3	81.9	71.8	38.5	79.9	33.9	37.4	67.9	22.6
CC-FPSE [21]	N/A	N/A	65.6	82.3	54.3	43.7	82.9	31.7	41.6	70.7	19.2
Ours	**19.2**	**0.395**	**66.9**	**82.5**	**49.5**	**45.2**	**83.8**	**29.3**	**42.0**	**72.0**	**18.1**

Table 2. User study. Each entry gives the percentage of cases where our results are favored.

Methods	CelebAHQ	Cityscapes	ADE20K	COCO
Ours > SPADE	76.00%	59.50%	66.62%	57.78%
Ours > CC	N/A	54.12%	60.28%	53.60%

Table 3. Impact of different operators (Op) on the generative performance on Cityscapes.

Op	Conv+BN	Conv+SCN	SCC+BN	SCC+SCN
mIoU↑	62.6	63.1	66.3	**66.9**
FID↓	69.7	60.9	54.0	**49.5**

Evaluation Metrics Following the testing protocol in the semantic image synthesis task [31, 24], we evaluate our methods along with baselines in the following perspectives: quantitative performance in Fréchet Inception Score (FID) [11] and learned perceptual image patch similarity (LPIPS) [44], semantic segmentation, and user study. In semantic segmentation, like the existing work [31, 24], we use mIoU and mAcc (mean pixel accuracy) performed on the synthesized results from the trained segmentation models to assess the result quality. DeepLabV2 [3], UperUnet101 [37], and DRN-D-105 [40] are employed for COCO-Stuff, ADE20K, and Cityscapes, respectively.

4.1. Quantitative Comparison

Table 1 indicates that our method yields decent performance, manifesting the effectiveness of our design of generating layout-aligned and appearance-related operators. On CelebAMask-HQ, our proposed SC-GAN improves face synthesis in terms of FID to 19.2, compared to FID of Spade 42.2. Our score is even lower than that computed from MaskGAN (21.4), which specializes in face manipulation utilizing *ground truth* face images for style reference. In scene-related datasets, *e.g.* Cityscapes, ADE20K, and COCO-Stuff, our method works nicely regarding segmentation and generation evaluation, giving non-trivial improvements compared with baselines, especially on FID.

In the comparison shown in Table 1 among CRN, CC-FPSE, and our SC-GAN, SC-GAN also shows better results in terms of mIoU, Acc, and FID. It proves the necessity to learn the semantic vectors from the segmentation masks in our method, since CRN concatenates segmentation masks in every input stage and CC-FPSE parameterizes convolution by generating weights from segmentation masks.

User Study Adhering to the protocol in SPADE [24], we list our user study results in Table 2 to compare our method with SPADE and CC-FPSE. Specifically, the subject judges which synthesizing result looks more natural corresponding to the input semantic layout. In all conditions, results from our model are more preferred by users compared with those from others, especially on CelebAMask-HQ.

4.2. Qualitative Results

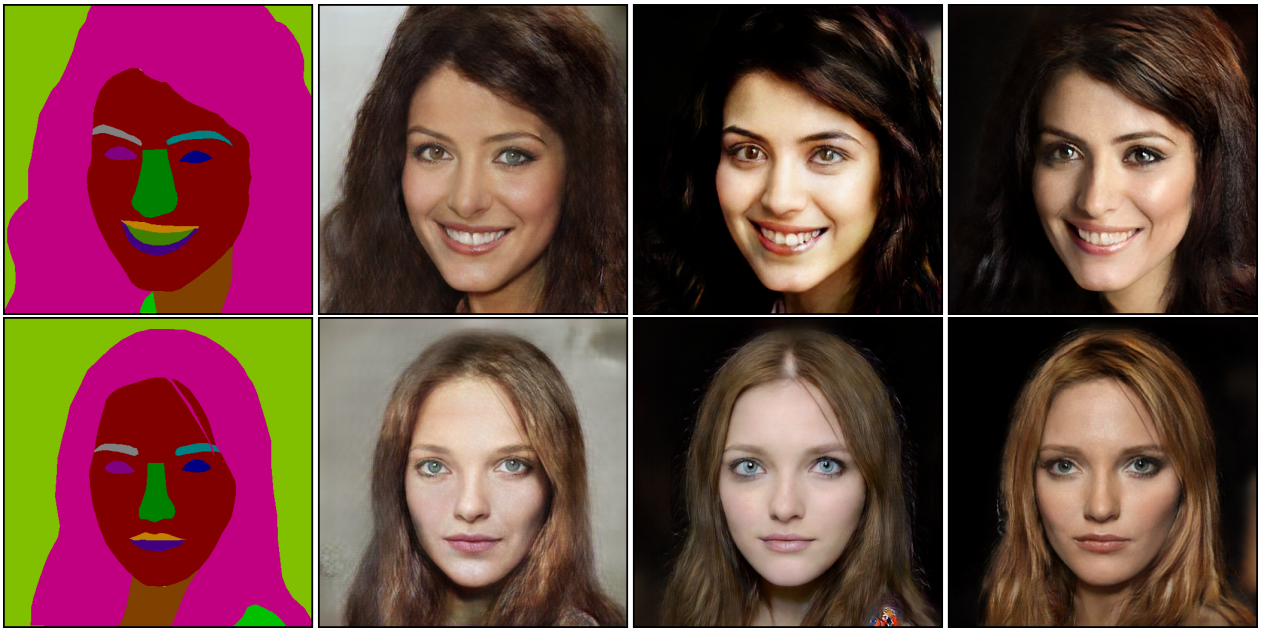
Figures 6 and 7 give the visual comparison of our method and other baselines. Our method generates natural results with less noticeable visual artifacts and more appearing details. Note that the skin of persons by our method is more photorealistic. Moreover, due to the effectiveness of semantic vectors, our method yields more consistent eye regions in Figure 6, and even creates intriguing reflections on the water in the bottom row of Figure 7. More visual results are given in the supplementary material.

Multi-modal Outputs and Interpolation Our method synthesizes multi-modal results from the same semantic layout using an additional encoder trained in the VAE manner. With different sampled random noise, our method gives diverse output in terms of appearance (top row in Figure 8). Also, we can apply linear interpolation of these random vectors, achieving a smooth transition from having a beard to not with it (bottom row in Figure 8). It validates the effectiveness of the learned manifold in our model.

4.3. Ablation Studies

We ablate the key design of our method on Cityscapes.

SCC vs. Standard Conv We construct three baselines, where SRG employs conventional convolutions (Conv) and Batch normalizations (BN), Conv and SCN, SCC and BN, respectively, and SVG remains intact. For baselines using Conv, we triple its Conv number in SRG compared to SCC for fair comparisons, as $n = 3$ in Eq. (4). We use extra input (concatenating $\mathbf{V}$ to the input feature maps $\mathbf{F}$) for every Conv, to see how the usage of $\mathbf{V}$ affects the performance.



(1) Input (2) SPADE (3) Mask-GAN (4) Ours

Figure 6. Visual comparison on CelebAMask-HQ.

(1) Input (2) SPADE (3) CC-FPSE (4) Ours

Figure 7. Visual comparison on COCO-stuff and ADE20K.

From Table 3, we notice SCC boosts Cityscapes synthesis from Conv in terms of mIoU and FID (mIoU: 62.6 $\rightarrow$ 66.3, FID: 69.7 $\rightarrow$ 54.0 with BN, and mIoU: 63.1 $\rightarrow$ 66.9, FID: 60.9 $\rightarrow$ 49.5 with SCN). It manifests the effectiveness of SCC. Exploiting spatially variant features ($\mathbf{V}$) by the proposed dynamic operators is more beneficial to yielded results than by static ones in this task, with a slimmer capacity (66.2M vs. 67.3M (Conv+BN)).

SCN vs. BN vs. SPADE The design of SCN is also validated in Table 3. SCN improves synthesis more on the generation quality (FID: 69.7 $\rightarrow$ 60.9 with Conv, and FID: 54.0 $\rightarrow$ 49.5 with SCC). Its influence on the semantic alignment (mIoU) is relatively small. Also, Conv+SCN gives better mIoU and FID compared SPADE (Table 1), further

Figure 8. Multi-modal predictions (top row) and interpolation (bottom row) of our method on CelebAMask-HQ.

Figure 9. Unpaired image-to-image translation results from our model on summer→winter. The source images, their reference ones (target images), and their corresponding translated results are marked by blue, purple, and black rectangles, respectively.

Table 4. Impact of semantic vector generation loss $\mathbf{L}_s$ in Eq. (10) about the generative performance on Cityscapes.

	w/o $\mathcal{L}_s$	w/ $\mathcal{L}_s$ (ℓ_1)	w/ $\mathcal{L}_s$ (ℓ_2)	w/ $\mathcal{L}_s$ (vgg)
mIoU↑	60.3	**66.9**	63.4	64.2
FID↓	82.5	**49.5**	53.3	51.8

Table 5. Generative performance on Cityscapes regarding the number of the shared weight candidates in SCC.

	$n=8$	$n=4$	$n=2$	$n=1$
mIoU↑	**67.9**	67.3	64.8	61.2

validating the importance of SCN.

Semantic Vector Generation Loss As mentioned in Section 3.1, semantic vector generation loss is to improve the relationship between different semantics according to their appearance. As shown in Table 4, without it in Eq. (10), the corresponding quantitative performance degrades notably both on object alignment (mIoU: 66.9 → 60.3) and generation (FID: 49.5 → 82.5). Also, using ℓ_1 norm in Eq. (10) is better than using ℓ_2 norm or perceptual loss considering both alignment and generation.

Number of Shared Weight Candidates in SRG We reduce the number of conv weight candidates in SRG while preserving that of norm candidates (fixed to 4). The corresponding segmentation results are shown in Table 5. With the increase of n, the semantic alignment becomes better.

Transformation of Semantic Vector Computation We evaluate different nonlinear functions for semantic vector computation, as given in Table 6. Note incorporating functions on $\mathbf{V}$ is vital as mIoU and FID scores become worse without it (mIoU: 60.2, FID: 73.9), and using softmax yields the best quantitative generation performance. In our model, bounded activation functions work better than unbounded (sigmoid, tanh, softmax vs. ReLU) ones, and the normalized one performs better than the unnormalized setting (softmax vs. sigmoid and tanh).

Table 6. Different nonlinear functions affect the generative performance on Cityscapes.

Nonlinear	Sigmoid	Tanh	ReLU	None	Softmax
mIoU↑	62.6	63.1	61.7	60.2	**66.9**
FID↓	57.6	56.8	66.5	73.9	**49.5**

Table 7. Unpaired image-to-image translation evaluation on summer-to-winter dataset.

Methods	MUNIT [13]	DMIT [42]	Ours
FID↓	118.225	87.969	**82.882**
IS↑	2.537	2.884	**3.183**

With A New Discriminator and Training Tricks We note new efforts [28] were made to enhance synthesis results by using a more effective discriminator and training approaches. Incorporating these techniques into our method could further boost our generation performance, *e.g.*, mIoU: 69.9, FID: 47.2 on Cityscapes, and mIoU: 49.1, FID: 27.6 on ADE20K, as given in Section 2.1 of the supp. material.

4.4. Unpaired Image-to-Image Translation

Due to our semantic encoding and stylization design, our model can also be applied to unpaired image-to-image (i2i) translation with minor modification. This modified framework is given in the supp. material. Table 7 shows the quantitative evaluation about our model along with unpaired i2i baselines MUNIT [13] and DMIT [42] on Yosemite summer-to-winter dataset. The superiority of FID and IS from our model demonstrates its generality. Visual results are given in Figure 9.

5. Concluding Remarks

In this paper, we have presented a new method to represent visual content in a semantic encoding and stylization manner for generative image synthesis. Our method introduces appearance similarity to semantic-aware operations, proposing a novel spatially conditional processing for both convolution and normalization. It outperforms the compared popular synthesis baselines on several benchmark datasets both qualitatively and quantitatively.

This new representation is also beneficial to unpaired image-to-image translation. We will study its applicability to other generation tasks in the future.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 3, 5

[2] Jin Chen, Xijun Wang, Zichao Guo, Xiangyu Zhang, and Jian Sun. Dynamic region-aware convolution. *arXiv preprint arXiv:2003.12243*, 2020. 2

[3] 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. *TPAMI*, 40(4):834–848, 2017. 6

[4] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, pages 1511–1520, 2017. 1, 2, 3, 5, 6

[5] 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

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 1

[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*, pages 2672–2680, 2014. 1

[8] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[10] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, 2001. 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. In *NeurIPS*, pages 6626–6637, 2017. 5, 6

[12] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *CVPR*, pages 1575–1584, 2019. 2

[13] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 8

[14] 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. 1, 2

[15] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, pages 667–675, 2016. 2

[16] 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*, 2020. 1, 2

[17] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018. 2

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, pages 5549–5558, 2020. 1, 5

[20] Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. In *ICCV*, pages 4220–4229, 2019. 2

[21] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *NeurIPS*, pages 570–580, 2019. 1, 2, 4, 5, 6

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

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

[24] 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. 1, 2, 4, 5, 6

[25] Lu Qi, Xiangyu Zhang, Yingcong Chen, Yukang Chen, Jian Sun, and Jiaya Jia. Pointins: Point-based instance segmentation. *arXiv preprint arXiv:2003.06148*, 2020. 2

[26] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *CVPR*, pages 8808–8816, 2018. 1, 2, 5, 6

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

[28] 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, 8

[29] Hao Tang, Dan Xu, Nicu Sebe, Yanzhi Wang, Jason J Corso, and Yan Yan. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In *CVPR*, pages 2417–2426, 2019. 1, 2

[30] 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. 1, 2

[31] 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. 1, 2, 5, 6

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 1

[33] Yi Wang, Ying-Cong Chen, Xin Tao, and Jiaya Jia. Vcnet: A robust approach to blind image inpainting. In *ECCV*, pages 752–768, 2020. 2

[34] Yi Wang, Ying-Cong Chen, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Attentive normalization for conditional image generation. In *CVPR*, pages 5094–5103, 2020. 1

[35] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *NeurIPS*, 2018. 2

[36] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Wide-context semantic image extrapolation. In *CVPR*, 2019. 2

[37] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 6

[38] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *NeurIPS*, pages 1307–1318, 2019. 2

[39] Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. In *NeurIPS*, pages 320–330, 2018. 2

[40] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, pages 472–480, 2017. 6

[41] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018. 1

[42] Xiaoming Yu, Yuanqi Chen, Shan Liu, Thomas Li, and Ge Li. Multi-mapping image-to-image translation via learning disentanglement. In *NeurIPS*, 2019. 8

[43] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 1

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6

[45] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 5

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 2

[47] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *CVPR*, pages 5104–5113, 2020. 2