

Context-Consistent Semantic Image Editing with Style-Preserved Modulation

Wuyang Luo¹, Su Yang¹, Hong Wang¹, Bo Long¹, and Weishan Zhang²

¹ Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University

² School of Computer Science and Technology, China University of Petroleum
{wyluo18,suyang}@fudan.edu.cn
<https://github.com/WuyangLuo/SPMPGAN>

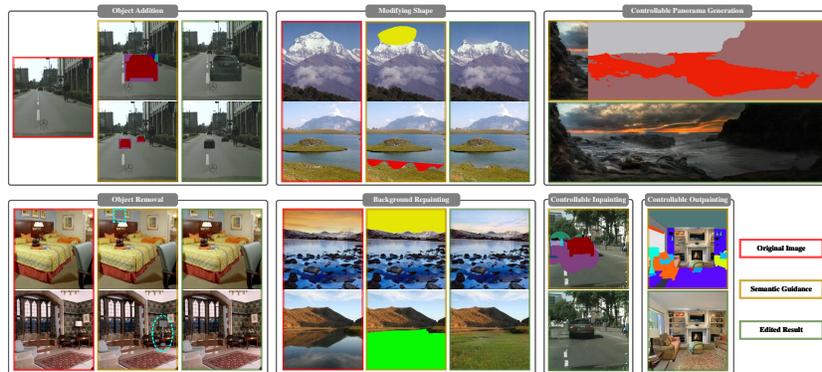


Fig. 1: Applications of the proposed method. Our image editing system is flexible in responding to a wide variety of editing requirements.

Abstract. Semantic image editing utilizes local semantic label maps to generate the desired content in the edited region. A recent work borrows SPADE block to achieve semantic image editing. However, it cannot produce pleasing results due to style discrepancy between the edited region and surrounding pixels. We attribute this to the fact that SPADE only uses an image-independent local semantic layout but ignores the image-specific styles included in the known pixels. To address this issue, we propose a style-preserved modulation (SPM) comprising two modulation processes: The first modulation incorporates the contextual style and semantic layout, and then generates two fused modulation parameters. The second modulation employs the fused parameters to modulate feature maps. By using such two modulations, SPM can inject the given semantic layout while preserving the image-specific context style. Moreover, we design a progressive architecture for generating the edited content in a coarse-to-fine manner. The proposed method can obtain context-consistent results and significantly alleviate the unpleasant boundary between the generated regions and the known pixels.

 Corresponding author

Keywords: Semantic Image Editing, Style-Preserved Modulation

1 Introduction

Image editing aims to generate the desired content in a specific region under users’ control. This task attracts a lot of research enthusiasm due to its wide application in social media, image and video re-creation, and virtual human-object interaction. The well-known commercial software Photoshop has achieved success in this field. However, the use of such software requires many professional skills and much manual effort.

Most image editing methods fall into a few categories. The first category is low-level-guided editing methods [18,3,6,28]. They introduce low-level information such as lines and color. These methods can deal with editing simple contours or shapes but only provide very limited editing control and cannot manipulate the high-level semantics of the image. The second category is classification-based methods [9,12]. They utilize an auxiliary classifier to guide synthesis and edit images. These methods can only control discrete attributes and cannot provide spatial control. The third category methods employ GAN inversion technique [39,4,1,27], which relies on a pre-trained GAN and dissects GANs’ latent spaces, finding disentangled latent codes suitable for editing. They require a powerful well-trained StyleGAN, which is impossible in many cases because training a strong StyleGAN [20,22] model is not easy, especially for complex scenes. Further, such methods lack flexibility, and the editing of each attribute may require independent training. The fourth category methods [11,32] utilize pixel-level semantic label maps, which define the class labels of pixels in edited regions to control edited content. This task is also known as Semantic Image Editing. Following this line of work, our approach can provide users with greater editing flexibility than the other three categories of methods. Our method includes the following editing capabilities: (1) Our method can be applied to complex scene editing. (2) Users can flexibly edit the image via manipulating semantic layout, such as modifying the shape of objects, adding or removing objects. (3) Edited regions can be selected at arbitrary positions, even beyond the original image boundaries. The Figure 1 demonstrates the versatility of our approach.

Semantic image editing is a non-trivial task. Its challenge lies in keeping context style consistent between edited and known regions. Here, "context" refers to the non-edited region of the input image, and "style" is the features of "context" such as color/texture. The previous state-of-the-art method SESAME [32] leverages SPADE block [33] to build their generator. SPADE is remarkably effective in conditional image synthesis. Conditional image synthesis learns a mapping from the semantic map domain to the real image domain, synthesizing the entire image according to the given semantic label map. Therefore, the generator may synthesize simple textures to get visually plausible results. However, since known pixels and fake pixels coexist for the image editing task, our task becomes tougher in that the requirement is synthesizing realistic textures and retaining consistency to the context style. Aside from that, image synthesis requires a

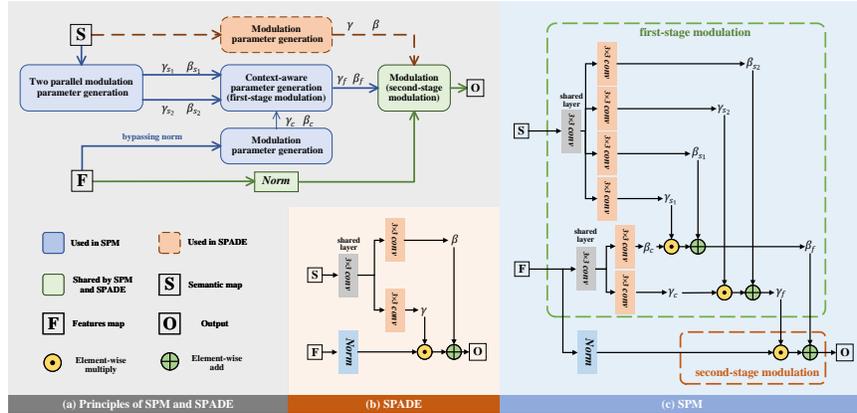


Fig. 2: (a) Principle difference between SPM and SPADE; (b) The structure of SPADE; (c) The structure of the proposed SPM.

full semantic label map, but semantic image editing can only see the semantic layout of the edited region. Thus, if SPADE is employed directly on the editing task, only meaningless modulation parameters would be generated in the known region. Previous work [32] often causes significant style inconsistency and unpleasant boundaries for the above reasons.

To address such limitations of the existing works, we propose a style-preserved modulation module (SPM). Compared with SPADE, which only utilizes one modulation operation, SPM consists of a two-stage modulation process. Inspired by the style transfer [14], which show that non-normalized feature maps contain high-level "style" information, we use non-normalized feature maps for context preserving via "bypassing norm". The principle difference between SPM and SPADE is illustrated in Figure 2(a) and their details are described in the section 3. Specifically, we first generate two parallel pairs of modulation parameters from semantic maps and a pair of modulation parameters from feature maps. Then we fuse them through the first modulation operation to generate two context-aware modulation parameters. The second stage modulation uses the context-aware modulation parameters to modulate feature maps. Through two-stage modulation, SPM can effectively integrate external semantic maps while preserving the image-specific context style.

SPM involves feature maps into the modulation process for preserving contextual style. For image editing tasks, the input is empty in the edited region. The contextual information of the known region is gradually transferred to the edited region through the enlargement of the receptive field of the generator. In order to make the edited region more effectively perceive the contextual style to generate context-aware modulation parameters of SPMs, we build a coarse-to-fine structure to decompose the editing process into multiple scales in a progressive manner. Specifically, we employ multiple generators to receive inputs of different scales. A downsampled version of the input image is fed into the first generator

to produce the coarsest result, which contains the coarse-grained image-specific style of the edited region. Subsequent generators can utilize previous results to effectively preserve the contextual style via SPM and refine the detailed textures. Our contributions are summarized as follows:

- We propose a context style-preserved modulation for the semantic image editing task, which can inject the layout of the external semantic label map while preserving the image-specific context style. The experiment shows the remarkable effect of SPM for alleviating the inconsistency.
- We build the progressive generative adversarial networks with SPMs for coarse-to-fine generation of edited regions.
- Extensive qualitative and quantitative experiments conducted on several benchmark datasets indicate that our model outperforms the state-of-the-art methods, especially in the sense of contextual style consistency.

2 Related Works

2.1 Image-to-Image Translation

Image translation attempts to learn a mapping from a source domain to a target domain. It can be applied to various tasks, such as image synthesis [48,46,47], image editing [11,3,18], style transfer [7,14], image inpainting [34,45,44], image extension [38,43], and image super-resolution [24,25]. Existing works utilize different conditional inputs as source domains such as semantic label maps, scene layouts, key points, and edge maps. Among them, the most relevant subtask is semantic image synthesis, which aims at generating photo-realistic images conditioned on semantic label maps.

Semantic image synthesis has achieved remarkable progress benefitting from GAN [8]. Pix2pix [17] is the seminal work based on cGAN framework [30]. The following work Pix2pixHD [41] is devoted to generating high-resolution images. SPADE [33] proposes a spatially-adaptive normalization that learns transformation parameters from the semantic layout to modulate the activations in normalization layers. CLADE [37] proposes a lightweight class-adaptive normalization to improve the efficiency of SPADE. Semantic image synthesis has been applied to different downstream tasks in recent works, such as semantic image editing [32], semantic view synthesis [13], and portrait editing [53,26].

2.2 Semantic Image Editing

Semantic image editing refers to users providing semantic label maps as a clue to edit the local region of a given image at pixel level. Semantic concepts are more intuitive and fundamental image features than colors, edges, key points, and textures. By manipulating the semantic label map, users can easily edit the image content in many ways, including re-painting, adding, removing, and out-painting semantic objects. Semantic image editing has not been fully developed because

it is challenging. Semantic image editing requires that the edited content not only has high fidelity but also must be consistent with the style of the remaining region. HIM [11] is the earliest attempt at this task. HIM can only operate on one foreground target each time. Furthermore, HIM requires a full semantic label map of the entire image as input, which is inconvenient for users. SESAME [32] only inputs the semantic label map of the edited region, making the image editing tool more practical. SESAME builds its generator with SPADE and uses a new discriminator to process the semantic and image information in separate streams. Although the previous methods can synthesize plausible results, they ignore the consistency of the context between the edited region and the known region. In contrast, our work is dedicated to reducing this inconsistency.

2.3 Modulation Technique

Modulation, also called denormalization, is an effective way to inject external control information. Unlike the unconditional normalization technique, such as BN [16], IN [40], and GN [42], modulation techniques require external data and follow a similar operating flow. First, feature maps are normalized to zero mean and unit deviation using an unconditional normalization layer. Then the normalized feature maps are modulated with scaling and shifting parameters learned from external data. Modulation techniques were initially applied to style transfer tasks, such as AdaIN [14] and later adopted in various vision tasks [21,15,35]. AdaIN only learns global style representation. [33] proposes SPADE for semantic image synthesis to handle external data with spatial dimensions. However, the previous methods only consider the external conditional input and ignore the internal contextual information, which is a fatal disadvantage for our task. This paper proposes a new modulation scheme that can aggregate internal context style and external semantic layout. The experimental results show that the proposed method can effectively preserve the context style and improve consistency for semantic image editing.

3 Approach

We describe our approach from bottom to top. We first analyze the limitations of SPADE for semantic image editing and introduce SPM proposed in this paper. Then, we introduce how to build a progressive architecture based on SPM.

3.1 Rethinking SPADE for Semantic Image Editing

SPADE is a state-of-the-art modulation technology remarkably successful in semantic image synthesis, as shown in Figure 2(b). $F^i \in \mathbb{R}^{N \times C \times H \times W}$ is the input feature maps of the i -th layers. N is the number of samples in one batch. C is the number of channels. H and W represent the height and width, respectively. SPADE learns two modulation parameters, scaling parameters γ and shifting

parameters β , via two convolutional layers from the given semantic label map S . First, F^i is normalized in a channel-wise manner:

$$\bar{F}^i = \frac{F^i - \mu^i}{\sigma^i} \quad (1)$$

where $\mu^i \in \mathbb{R}^{N \times C \times 1 \times 1}$ and $\sigma^i \in \mathbb{R}^{N \times C \times 1 \times 1}$ are the channel-wise means and standard deviations of F^i . Then, we perform the modulation operation:

$$\widetilde{F}^i = (\mathbf{1} + \gamma) \odot \bar{F}^i + \beta \quad (2)$$

Previous work [32] applies SPADE for semantic image editing. However, SPADE is ill-fitted for semantic image editing for the following two reasons: First, SPADE can only generate image-independent modulation parameters from the given external semantic label map. Thus, if two edited images are given the same semantic label map, SPADE will generate the same modulation parameters. This is unreasonable because SPADE ignores image-specific style. Second, for semantic image editing, the generator can only see the semantic layout of the edited region, and the semantic labels of the rest known regions are set to a fixed value. Therefore, SPADE cannot learn effective parameters on the known region. If we naively transfer SPADE to semantic image editing, the above two limitations will cause style inconsistency and unpleasant boundaries.

3.2 Style-Preserved Modulation

To solve the issues mentioned above, we propose a two-stage modulation mechanism for style preserving, as shown in Figure 2(c). The first stage of modulation aims to integrate the context style and the external semantic layout. The second stage of modulation is to inject the fused information into feature maps.

In the first modulation, we generate two kinds of parameters: Four semantic modulation parameters and two context modulation parameters. Semantic modulation parameters include two groups: $(\gamma_{s_1}, \beta_{s_1})$ and $(\gamma_{s_2}, \beta_{s_2})$. The context modulation parameters (γ_c, β_c) are generated from the original feature maps without passing through the normalization layer. The previous style transfer works [14] revealed that the style of the image could be washed away by normalization layers. The non-normalized feature maps can retain the context style more. So, we use the original feature maps to generate two context modulation parameters. Finally, we perform the first modulation to generate the fused modulation parameters γ_f and β_f :

$$\gamma_f = (\mathbf{1} + \gamma_{s_2}) \odot \gamma_c + \beta_{s_2} \quad (3)$$

$$\beta_f = (\mathbf{1} + \gamma_{s_1}) \odot \beta_c + \beta_{s_1} \quad (4)$$

where \odot denotes element-wise multiplication. All modulation parameters have the same shape as the feature maps F^i .

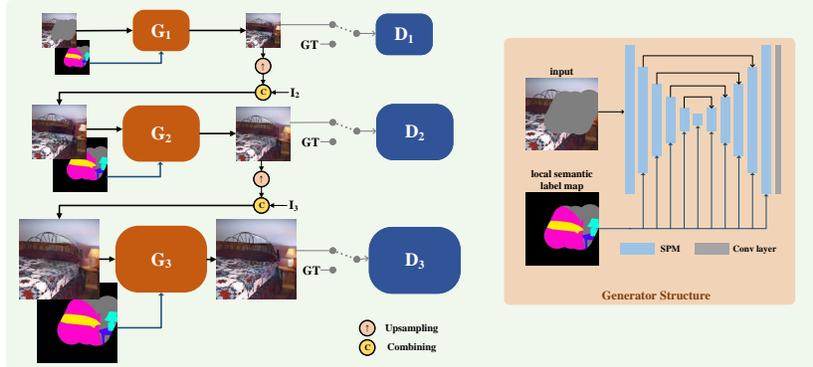


Fig. 3: Overview of the progressive architecture.

In the second modulation, we use fused modulation parameters to modulate the normalized feature maps \bar{F}^i .

$$\tilde{F}^i = (\mathbf{1} + \gamma_f) \odot \bar{F}^i + \beta_f \quad (5)$$

Through two-stage modulation process, SPM overcomes the two shortcomings of SPADE: First, the fused modulation parameters integrate the external semantic layout and retain the internal context style. Second, the fused modulation parameters can generate meaningful modulation parameters for known regions.

3.3 Progressive Editing Architecture

We propose a progressive architecture for image editing based on SPM, called *SPMPGAN*. Our model has three inputs: (1) The input image $I \in \mathbb{R}^{256 \times 256 \times 3}$ which contains only known pixels with masked edited region; (2) the local semantic map S providing the semantic layouts of the edited region; and (3) the corresponding mask map M whose value is 0 in the non-edited region and 1 in the edited region. Our progressive architecture consists of a pyramid of generators $\{G_1, G_2, G_3\}$ and discriminators $\{D_1, D_2, D_3\}$ with an image pyramid of I : $\{I_1, I_2, I_3\}$, where I_n is a downsampled version of I by a factor 2^{3-n} , mask pyramid of M : $\{M_1, M_2, M_3\}$, and semantic map pyramid of S : $\{S_1, S_2, S_3\}$. Each generator G_n is trained with an associated discriminator D_n . G_n learns to generate realistic new content in the edited region and try to fool the corresponding discriminator. D_n attempts to distinguish the edited result and the real image. We adopt an encoder-decoder architecture with skip connections [36] for all generators, as shown in Figure 3. Each generator adds a down-sampling layer in the encoder and an up-sampling layer in the decoder on the previous generator. Inspired by [45], the discriminators are composed of several convolutional layers with 5×5 convolution kernel and spectral normalization [31]. The number of

layers of D_1 , D_2 , and D_3 are 4, 5, and 6, respectively. Thus, each D_n has the receptive field with the size of the input I_n and captures the entire image’s feature. The generation process starts at the coarsest G_1 and sequentially passes through G_2 and G_3 to the original scale. Specifically, the original input I is downsampled to 64×64 to get G_1 ’s input: $I_{G_1} = I_1$, and G_1 ’s output is O_1 . Then, we combine the upsampled O_1 with I_2 as G_2 ’s input: $I_{G_2} = O_1 \odot M_2 + I_2 \odot (1 - M_2)$. All generators and discriminators have independent weights.

3.4 Training

We train our progressive model in an end-to-end manner. The training objective for the n -th generator is comprised of a reconstruction loss and an adversarial loss \mathcal{L}_{adv} . The reconstruction loss consists of L1 distance loss \mathcal{L}_1 and perceptual loss \mathcal{L}_p [19]. We employ the hinge version adversarial loss [2,29]. The overall loss can be written as:

$$\mathcal{L} = \mathcal{L}_1 + 10.0\mathcal{L}_p + \mathcal{L}_{adv} \tag{6}$$

4 Experiments

4.1 Datasets

ADE20K-room ADE20K [51] has over 20,000 images together with detailed semantic labels of 150 classes. We select a subset of the ADE20K comprised of **Bedroom**, **Hotel Room**, and **Living Room**. This subset is called ADE20K-room. We resize all the images with their longer sides no more than 384 and their shorter sides no less than 256. We crop them to 256×256 when training. This dataset has 2246 images for training and 255 for testing.

ADE20K-landscape We also selected the landscape subclass from ADE20K and use the same preprocessing approach. The difference is that this dataset has only background and no foreground objects. The training set and the testing set contain 1689 images and 155 images, respectively.

Cityscapes [5] The dataset collects streetscapes of 50 German cities, which contains 33 semantic categories. The training and testing set has 2975 and 500 images, respectively, with a resolution of 2048×1024 . We downsample all images to 512×256 and crop them to 256×256 patches.

4.2 Baselines

Semantic image editing methods. We employ two existing works [11,32] as baselines. HIM [11] introduces a two-stage method for image editing. They first predict semantic layout from object bounding boxes. Then, they generate new content according to the predicted semantic layout. Because in our setting, the ground truth semantic layout of the edited region is known, we directly input

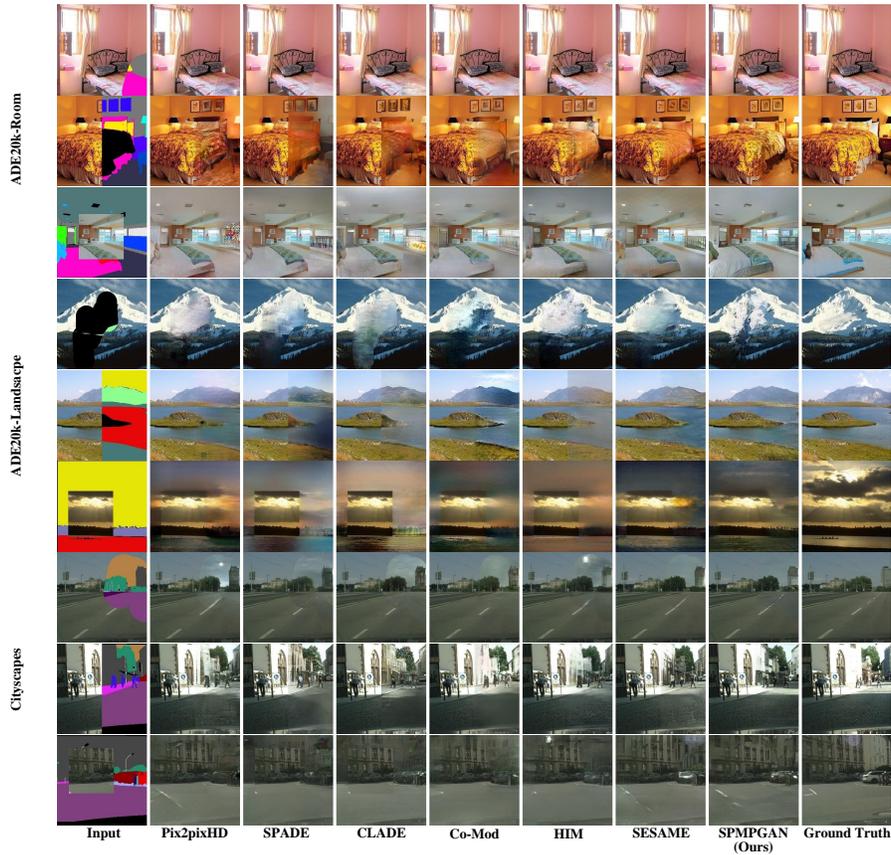


Fig. 4: Visual comparison with other methods.

the ground truth layout to the second stage of HIM to get the results. SESAME [32] has similar settings with our work.

Image synthesis methods. Our experiments also include several image generation methods for comparison. These recent works [41,33,37,50] can be directly transferred to our task via only modifying their generators’ input. It is worth mentioning that some recent works cannot be simply adapted for our task. For example, SEAN [53] requires a full segmentation map to calculate their style codes. CoCosNetv2 [52] requires a full segmentation map to perform their domain alignment. However, our task can only see local semantic label maps.

4.3 Implementation Details

To obtain a more flexible model, we employ five types of masks for training: Free-form mask, extension mask, outpainting mask, instance mask, and class mask. The extension mask is the right half of the input. For the outpainting

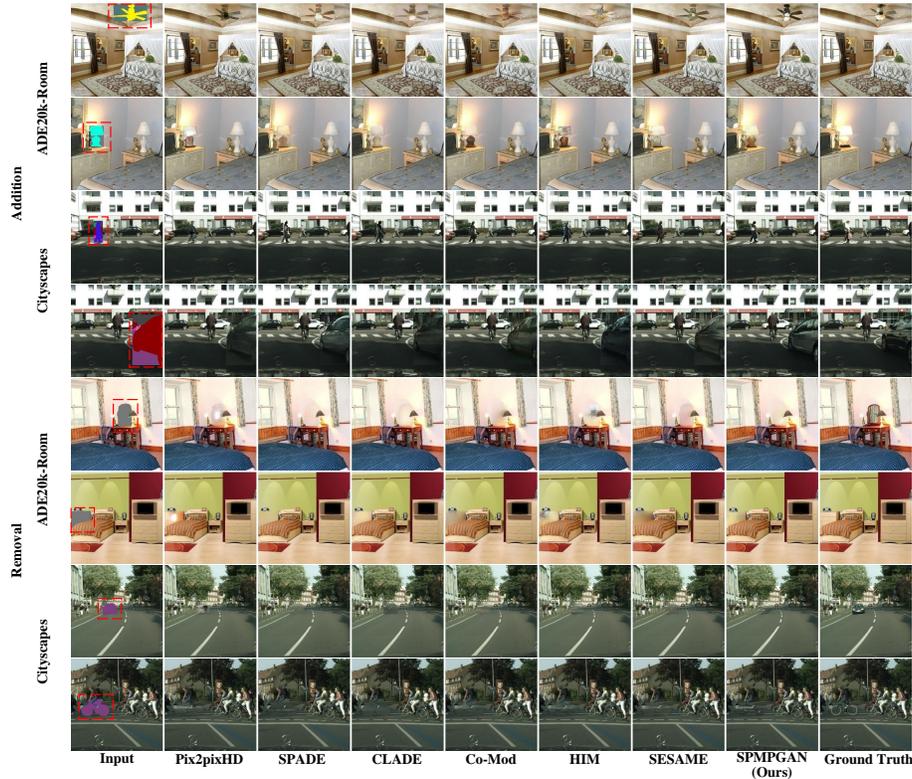


Fig. 5: Visual results of addition and removal objects.

mask, we randomly retain a 128×128 patch as the known region. The instance mask contains only a single foreground target, and the class mask drops all the pixels belonging to a semantic class. During training, each mask is randomly selected and sent to the network at each iteration. We use Adam optimizers [23] for both the generator and the discriminators with momentum $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rates for the generator and the discriminators are set to 0.0001 and 0.0004, respectively. All models are trained for 500 epochs on all datasets. The batch size is set to the maximum value to fit the memory size of a single NVIDIA RTX 3090 GPU.

4.4 Semantic Image Editing

We compare our results with state-of-the-art methods using free-form masks, extension masks, and outpainting masks on the three benchmarks. Figure 4 provides some visual comparisons. Pix2pixHD[41] and HIM[11] only use semantic label maps as conditions in the input layer, and they often generate artifacts. SPADE[33], CLADE[37], and SESAME[32] can synthesize reasonable structures and realistic textures, but they severely suffer from style inconsistencies leading

Table 1: Quantitative comparison with different mask types (\uparrow : Higher is better; \downarrow : Lower is better). In the leftmost column, M, F, E, and O represent Mask Type, Free-Form Mask, Extension Mask, and Outpainting Mask, respectively.

M	Method	ADE20k-Room			ADE20k-Landscape			Cityscapes		
		FID \downarrow	LPIPS \downarrow	mIoU \uparrow	FID \downarrow	LPIPS \downarrow	mIoU \uparrow	FID \downarrow	LPIPS \downarrow	mIoU \uparrow
M	pix2pixHD	23.72	0.107	27.49	33.90	0.120	28.30	15.28	0.090	58.69
	SPADE	27.65	0.124	27.47	41.92	0.134	28.41	15.83	0.099	59.10
	CLADE	30.77	0.126	25.91	46.59	0.139	26.39	17.06	0.103	57.72
	F	Co-Mod	27.37	0.111	27.52	32.35	0.124	28.60	15.88	0.097
	HIM	28.64	0.133	28.04	35.89	0.116	28.43	15.58	0.093	58.99
	SESAME	21.73	0.101	27.50	30.30	0.116	28.28	12.89	0.082	58.88
	SPMPGAN	18.83	0.090	28.22	23.11	0.105	28.73	11.90	0.084	58.80
F	pix2pixHD	38.08	0.223	27.32	56.15	0.242	28.10	26.14	0.176	58.55
	SPADE	36.43	0.211	27.62	68.96	0.277	28.44	25.78	0.194	59.01
	CLADE	41.77	0.242	25.67	65.33	0.267	26.39	25.29	0.195	58.09
	E	Co-Mod	38.61	0.231	27.13	53.96	0.249	28.09	29.27	0.188
	HIM	40.69	0.239	27.61	52.14	0.234	28.42	25.20	0.180	58.91
	SESAME	36.43	0.211	27.62	48.16	0.232	28.31	20.30	0.168	59.08
	SPMPGAN	32.61	0.199	27.73	45.10	0.217	28.48	19.46	0.167	59.10
E	pix2pixHD	52.14	0.323	27.49	82.56	0.360	28.30	39.50	0.253	58.72
	SPADE	47.72	0.305	27.40	88.79	0.389	28.30	33.97	0.268	59.07
	CLADE	52.45	0.346	25.47	86.77	0.388	24.49	34.19	0.276	57.49
	O	Co-Mod	51.45	0.325	26.54	79.77	0.360	26.70	50.29	0.264
	HIM	54.51	0.337	28.19	77.18	0.352	28.57	36.27	0.252	58.99
	SESAME	47.72	0.305	27.40	72.28	0.344	28.13	28.27	0.237	58.75
	SPMPGAN	41.52	0.288	27.85	63.32	0.328	27.56	27.63	0.233	58.53

to unpleasant boundaries. Because they only use the image-independent external semantic map when injecting the semantic label map and completely ignoring the context information. Co-Mod[50] also has the apparent texture inconsistency as it lacked a specific design for the image editing tasks. The proposed method can effectively integrate the contextual style and the semantic layout to produce realistic textures while preserving the contextual style. Table 1 also shows the quantitative comparison results. FID [10] has been widely demonstrated that it is consistent with human visual perception. A lower FID value indicates that results have higher fidelity. LPIPS[49] evaluates the similarity between the generated image and the corresponding ground truth in a pairwise manner. A lower LPIPS indicates that the generated image is closer to the ground truth. mIoU is employed in the semantic synthesis task [33] to evaluate the alignment between the semantic label map and the generated result. Our method outperforms the other methods in most evaluation metrics.

4.5 Addition and Removal of Objects

Our work is capable of adding or removing individual objects by modifying the semantic label maps. Visual results are demonstrated in Figure 5. For the object

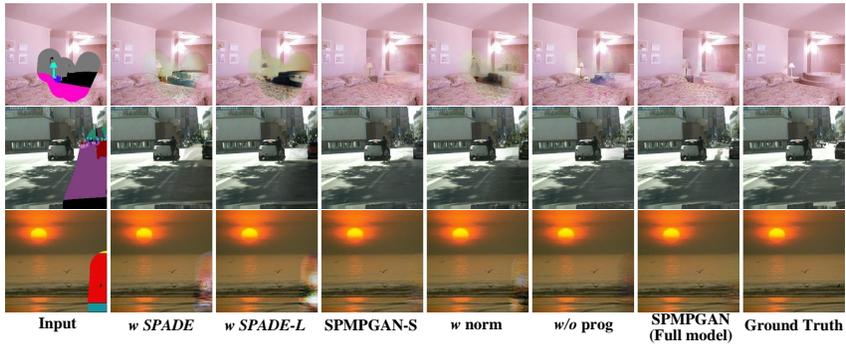


Fig. 6: Visual comparison of ablation studies.

Table 2: Addition and removal results for Cityscapes and ADE20k-Room.

Manipulation	Method	ADE20k-Room			Cityscapes		
		FID↓	LPIPS↓	mIoU↑	FID↓	LPIPS↓	mIoU↑
Addition	pix2pixHD	6.29	0.027	27.09	11.77	0.030	58.28
	SPADE	5.66	0.027	27.17	10.48	0.031	58.66
	CLADE	6.21	0.028	27.16	11.03	0.031	57.55
	Co-Mod	5.75	0.026	27.23	11.28	0.031	56.40
	HIM	9.80	0.046	27.22	11.41	0.030	58.75
	SESAME	5.50	0.024	27.14	9.70	0.027	58.56
	SPMPGAN	5.14	0.022	27.43	9.04	0.026	58.68
Removal	pix2pixHD	4.52	0.019	28.32	15.01	0.039	55.02
	SPADE	3.96	0.019	28.35	15.48	0.040	55.04
	CLADE	4.12	0.019	28.34	16.18	0.040	54.22
	Co-Mod	4.03	0.019	28.33	15.05	0.041	55.10
	HIM	7.44	0.035	28.33	15.10	0.040	55.11
	SESAME	4.02	0.018	28.34	15.52	0.041	55.08
	SPMPGAN	3.68	0.016	28.35	14.63	0.039	55.01

addition, we randomly select an instance of input and extract the boundary boxes to generate its local semantic label map. For object removal, we delete a instance and fill it with nearby background semantic class. Quantitative results shown in Table 2 indicate that our method achieves the best results in style preservation and fidelity.

4.6 Controllable Panorama Generation

A well-trained model can be used recursively to obtain panoramas. Specifically, we employ the generated region of the previous step as the known region of the next step in a sliding window manner. Thus, the input is extended to the right by 128 pixels in each step so that images with arbitrary width can be controllably synthesized. Figure 1 shows a recursive generated result.

Table 3: Ablation study with different mask types.

M	Method	ADE20k-Room			ADE20k-Landscape			Cityscapes		
		FID↓	LPIPS↓	mIoU↑	FID↓	LPIPS↓	mIoU↑	FID↓	LPIPS↓	mIoU↑
F	<i>w SPADE</i>	23.27	0.098	27.60	34.71	0.118	28.39	14.20	0.091	58.80
	<i>w norm</i>	20.51	0.098	27.58	29.87	0.111	28.31	12.64	0.085	58.78
	<i>w/o prog</i>	20.47	0.096	27.42	25.87	0.109	28.42	13.07	0.089	58.73
	<i>w SPADE-L</i>	24.11	0.098	27.61	34.68	0.116	28.43	14.40	0.090	58.79
	<i>SPMPGAN-S</i>	18.93	0.090	28.24	23.21	0.106	28.70	11.89	0.084	58.82
	<i>SPMPGAN</i>	18.83	0.090	28.22	23.11	0.105	28.73	11.90	0.084	58.80
E	<i>w SPADE</i>	36.84	0.220	27.51	53.02	0.239	28.92	21.99	0.173	58.88
	<i>w norm</i>	32.76	0.205	27.56	48.43	0.228	28.92	20.50	0.176	59.01
	<i>w/o prog</i>	33.87	0.205	27.44	45.96	0.222	28.86	21.00	0.170	59.09
	<i>w SPADE-L</i>	36.14	0.218	27.48	53.13	0.240	28.93	21.86	0.174	58.81
	<i>SPMPGAN-S</i>	31.92	0.200	27.74	45.17	0.218	28.47	19.12	0.167	59.12
	<i>SPMPGAN</i>	32.61	0.199	27.73	45.10	0.217	28.48	19.46	0.167	59.10
O	<i>w SPADE</i>	47.37	0.321	28.38	71.52	0.357	28.84	31.33	0.244	58.95
	<i>w norm</i>	42.31	0.300	28.52	66.52	0.337	28.82	27.74	0.235	57.98
	<i>w/o prog</i>	43.98	0.297	28.05	66.32	0.329	27.39	29.54	0.238	58.53
	<i>w SPADE-L</i>	47.16	0.318	28.39	70.33	0.354	28.83	31.43	0.243	58.95
	<i>SPMPGAN-S</i>	41.49	0.289	27.80	62.43	0.330	27.63	27.39	0.228	58.59
	<i>SPMPGAN</i>	41.52	0.288	27.85	63.32	0.328	27.56	27.63	0.233	58.53

Table 4: Comparison of the number of parameters.

	<i>w SPADE</i>	<i>w SPADE-L</i>	<i>SPMPGAN</i>	<i>SPMPGAN-S</i>
ADE20k-Room	63.4 M	90.0 M	118.4 M	76.9 M
Cityscapes	57.8 M	81.5 M	112.7 M	74.0 M

4.7 Ablation Study

Style-preserved modulation

We study the importance of SPM for style preserving. We replace all SPMs with SPADE blocks ("*w SPADE*"). The visual results are shown in Figure 6. It can be observed that SPADE leads to unpleasant boundaries. This is because SPADE completely ignores the image-specific context style and only uses local semantic label maps to modulate feature maps. As a comparison, SPM can relieve the inconsistency. The two-stage modulation can integrate the context style and the external semantic label map. In addition, SPM can also help the generator to synthesize more realistic texture details. We also study the influence of "bypassing norm" for style preserving. Specifically, for the generation of γ_c and β_c in SPM, we replace original feature maps by normalized feature maps ("*w norm*"). The experimental results show that the style preserving is significantly weakened. It proves that the normalization operation washes away context style. Therefore, we use the original feature maps without normalization in SPM. Quantitative results are also demonstrated in Table 3.

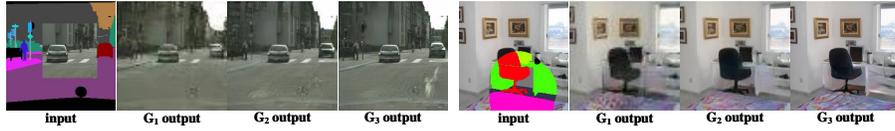


Fig. 7: Outputs of all generators.

Effectiveness of progressive architecture

We conduct an ablation study to demonstrate the effectiveness of the progressive design for synthesizing high-quality results. We only use the last level generator as the baseline (*"w/o prog"*). Figure 6(c) shows that without the progressive generation, the model will produce style inconsistency and unrealistic textures. The outputs of the generators of all scales are shown in the Figure 7. It can be seen, G_1 synthesizes the global structure, and G_2 and G_3 produce the sharper detail. Quantitative results are given in Table 3, which indicates that progressive architecture contributes to performance improvement.

4.8 Study of Model Scale

This study demonstrates that our performance improvement stems from the novel design of SPM rather than increasing parameters. As shown in Table 4, our model follows SPADE to set the number of output channels C^h of the shared layer to 128. We reduce C^h of all SPMs to 64 and keep the structure unchanged (*"SPMGAN-S"*). We do not observe the performance drop. In addition, we insert more SPADE blocks into *"w SPADE"* to obtain a new baseline *"w SPADE-L"*. The experimental results are shown in the Table 3, *"w SPADE-L"* does not obtain performance gain by simply increasing the network scale and computational consumption. The performance of *"SPMGAN-S"* still significantly outperforms *"w SPADE-L"* with fewer parameters.

5 Conclusion

This paper is dedicated to solving style inconsistency for the semantic editing task. We propose a style-preserved modulation and a progressive architecture that effectively injects the structure from semantic label maps while preserving the context style. The key of SPM lies in effectively integrating contextual information and semantic label maps. We also demonstrate the ability of our method for various applications.

Acknowledgement This work is supported by State Grid Corporation of China (Grant No. 5500-202011091A-0-0-00).

References

1. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.H.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing. arXiv preprint arXiv:2111.15666 (2021)
2. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
3. Chen, S.Y., Liu, F.L., Lai, Y.K., Rosin, P.L., Li, C., Fu, H., Gao, L.: Deepfaceediting: Deep face generation and editing with disentangled geometry and appearance control. arXiv preprint arXiv:2105.08935 (2021)
4. Chong, M.J., Lee, H.Y., Forsyth, D.: Stylegan of all trades: Image manipulation with only pretrained stylegan. arXiv preprint arXiv:2111.01619 (2021)
5. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
6. Dong, H., Liang, X., Zhang, Y., Zhang, X., Shen, X., Xie, Z., Wu, B., Yin, J.: Fashion editing with adversarial parsing learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8120–8128 (2020)
7. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
9. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing* **28**(11), 5464–5478 (2019)
10. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems. pp. 6626–6637 (2017)
11. Hong, S., Yan, X., Huang, T., Lee, H.: Learning hierarchical semantic image manipulation through structured representations. arXiv preprint arXiv:1808.07535 (2018)
12. Hou, X., Zhang, X., Liang, H., Shen, L., Lai, Z., Wan, J.: Guidedstyle: Attribute knowledge guided style manipulation for semantic face editing. *Neural Networks* **145**, 209–220 (2022)
13. Huang, H.P., Tseng, H.Y., Lee, H.Y., Huang, J.B.: Semantic view synthesis. In: European Conference on Computer Vision. pp. 592–608. Springer (2020)
14. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
15. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European conference on computer vision (ECCV). pp. 172–189 (2018)
16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)

18. Jo, Y., Park, J.: Sc-fegan: Face editing generative adversarial network with user's sketch and color. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1745–1753 (2019)
19. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
21. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
22. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020)
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
24. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 624–632 (2017)
25. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
26. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5549–5558 (2020)
27. Ling, H., Kreis, K., Li, D., Kim, S.W., Torralba, A., Fidler, S.: Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems* **34** (2021)
28. Liu, H., Wan, Z., Huang, W., Song, Y., Han, X., Liao, J., Jiang, B., Liu, W.: Deflocnet: Deep image editing via flexible low-level controls. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10765–10774 (2021)
29. Liu, M.Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10551–10560 (2019)
30. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
31. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
32. Ntavelis, E., Romero, A., Kastanis, I., Gool, L.V., Timofte, R.: Sesame: Semantic editing of scenes by adding, manipulating or erasing objects. In: European Conference on Computer Vision. pp. 394–411. Springer (2020)
33. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2337–2346 (2019)
34. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)

35. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
36. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
37. Tan, Z., Chen, D., Chu, Q., Chai, M., Liao, J., He, M., Yuan, L., Hua, G., Yu, N.: Efficient semantic image synthesis via class-adaptive normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021)
38. Teterwak, P., Sarna, A., Krishnan, D., Maschinot, A., Belanger, D., Liu, C., Freeman, W.T.: Boundless: Generative adversarial networks for image extension. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 10521–10530 (2019)
39. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)* **40**(4), 1–14 (2021)
40. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016)
41. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
42. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
43. Yang, Z., Dong, J., Liu, P., Yang, Y., Yan, S.: Very long natural scenery image prediction by outpainting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 10561–10570 (2019)
44. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018)
45. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4471–4480 (2019)
46. Zhan, F., Lu, S.: Esir: End-to-end scene text recognition via iterative image rectification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2059–2068 (2019)
47. Zhan, F., Lu, S., Xue, C.: Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 249–266 (2018)
48. Zhan, F., Zhu, H., Lu, S.: Spatial fusion gan for image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3653–3662 (2019)
49. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
50. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Eric, I., Chang, C., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. In: International Conference on Learning Representations (2020)
51. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)

52. Zhou, X., Zhang, B., Zhang, T., Zhang, P., Bao, J., Chen, D., Zhang, Z., Wen, F.: Cocosnet v2: Full-resolution correspondence learning for image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11465–11475 (2021)
53. Zhu, P., Abdal, R., Qin, Y., Wonka, P.: Sean: Image synthesis with semantic region-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5104–5113 (2020)