# Example-Guided Scene Image Synthesis using Masked Spatial-Channel Attention and Patch-Based Self-Supervision

Haitian Zheng Haofu Liao Lele Chen Wei Xiong Tianlang Chen Jiebo Luo
University of Rochester
{hzheng15, hliao6, lchen63, wxiong5, tchen45, jluo}@cs.rochester.edu

## Abstract

*Example-guided image synthesis has been recently attempted to synthesize an image from a semantic label map and an exemplary image. In the task, the additional exemplar image serves to provide style guidance that control the appearance of the synthesized output. Despite the controllability advantage, the previous models are designed on datasets with specific and roughly aligned objects. In this paper, we tackle a more challenging and general task, where the exemplar is an arbitrary scene image that is semantically unaligned to the given label map. To this end, we first propose a new Masked Spatial-Channel Attention (MSCA) module which models the correspondence between two unstructured scenes via cross-attention. Next, we propose an end-to-end network for joint global and local feature alignment and synthesis. In addition, we propose a novel patch-based self-supervision scheme to enable training. Experiments on the large-scale CCOO-stuff dataset show significant improvements over existing methods. Moreover, our approach provides interpretability and can be readily extended to other tasks including style and spatial interpolation or extrapolation, as well as other content manipulation.*

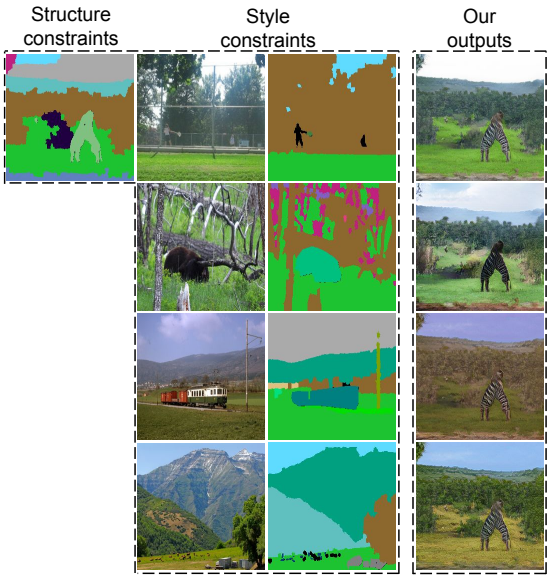


Figure 1. The inputs to style-consistent scene image generation is a *structurally uncorrelated* and *semantically unaligned* segmentation map (column 1) and a reference image (column 2) that constraints the style of the output. The corresponding reference segmentation map is also taken as input. In spite of the complexity of the task, our model can generate high-quality scene images with a consistent style with the reference image.

## 1. Introduction

Conditional generative adversarial network (cGAN) [34] has recently made substantial progresses in realistic image synthesis. In cGAN, a generator $\hat{x} = \mathrm{G}(c)$ aims to output a realistic image $\hat{x}$ with a constraint implicitly encoded by $c$. Conversely, a discriminator $\mathrm{D}(x, c)$ learns such a constraint from ground-truth pairs $\langle x, c\rangle$ by predicting if $\langle \hat{x}, c\rangle$ is real or generated.

The current cGAN models [36, 43, 19] for semantic image synthesis aim to solve the *structural consistency* constraint where the output image $x$ is required to be aligned to a semantic label map $c$. The limitation of the above generative process is that the styles of the image outputs are inherently determined by the model and thus cannot be controlled by users. To provide desired controllability over the generated styles, previous studies [27, 41] impose additional constraints and allow more inputs to the generator: $\hat{x}_{2\to1} = \mathrm{G}(z, c_1, x_2)$, where $x_2$ is an exemplar image that guides the style of $c_1$. However, previous studies are designed on datasets such as face [30, 37], dancing [41] or street view [47], where the input images usually contain similar semantics and the spatial structures of $y$ and $c_x$ are usually similar as well.

Different from the previous studies, we propose to address a more challenging example-guided *scene image* generation task. As shown in Fig. 1, given a semantic label map

$c_1$ (column 1) and an arbitrary scene image $x_2$ (column 2) with its semantic map $c_2$ (column 3) as the input, the task aims to generate a new scene image $\hat{x}_{2\to1}$ (column 4) that matches the semantic structure of $c_1$ and the scene style of $x_2$. The challenge is that scene images have complex semantic structures as well as diversified scene styles, and more importantly, the inputs $c_1$ and $x_2$ are *structurally uncorrelated* and *semantically unaligned*. Therefore, a mechanism is required to better match the structures and semantics for coherent outputs, e.g., the tree styles can be applied to mountains but cannot be applied to sky.

In this paper, we propose a novel Masked Spatial-Channel Attention (MSCA) module (Section 3.2) to propagate features across unstructured scenes. Our module is inspired by a recent work [6] for attention-based object recognition, but we apply a new cross-attention approach to model the semantic correspondence for image synthesis instead. To facilitate example-guided synthesis, we further improve the module by including: i) feature masking for semantic outlier filtering, ii) multi-scaling for global and local feature processing, and iii) resolution extending for image synthesis. As a result, our module provides both clear physical meaning and interpretability for the example-guided synthesis task.

We formulate the proposed approach under an unified synthesis network for joint feature extraction, alignment and image synthesis. We achieve this by applying MSCA modules to the extracted features for multi-scale feature domain alignment. Next, we apply a recent feature normalization technique, SPADE [36] on the aligned features to allow spatially-controllable synthesis. To facilitate the learning of this network, we propose a novel patch-based self-supervision scheme. As opposed to [41], our scheme requires only semantically parsed images for training and does not rely on video data. We show that a model trained with this approach generalizes over scales and across different scene semantics.

Our main contributions include the following:

- A novel masked spatial-channel attention (MSCA) module to propagate features for unstructured scenes.
- An unified synthesis network for joint feature extraction, alignment and image synthesis.
- A novel patch-based self-supervision scheme that requires only annotated images for training.
- Experiments on COCO-stuff [3] dataset that show significant improvements over existing methods. Moreover, our model provides interpretability and can be extended to other tasks of content manipulation.

## 2. Related work

**Generative Adversarial Networks.** Recent years have witnessed the progresses of generative adversarial networks (GANs) [10] for image synthesis. A GAN model consists of a generator and a discriminator where the generator serves to produce realistic images that cannot be distinguished from the real ones by the discriminator. Recent techniques for realistic image synthesis include modified losses [1, 33, 38], model regularization [35], self-attention [48, 2], feature normalization [23] and progressive synthesis [22].

**Image-to-Image translation (I2I).** I2I translation aims to translate images from a source domain to a target domain. The initial work of Isola *et al*. [19] proposes a conditional GAN framework to learn I2I translation with paired images. Wang *et al*. [43] improve the conditional GAN for high-resolution synthesis and content manipulation. To enable I2I translation without using paired data, a few works [50, 29, 17, 25, 4] apply the cycle consistency constraint in training. Recent works on photo-realistic image synthesis take semantic label maps as inputs for image synthesis. Specifically, Wang *et al*. [43] extend the conditional GAN for high-resolution synthesis, Chen *et al*. [5] propose a cascade refine pipeline. More recently, Park *et al*. [36] propose spatial-adaptive normalization for realistic scene image generation.

**Example-Guided Style Transfer and Synthesis.** Example guided style transfer [12, 7] aims to transfer the style of an example image to a target image. More recent works [8, 16, 31, 21, 26, 11, 4, 15, 46] utilize deep neural network features to model and transfer styles. Several frameworks [17, 18, 32] perform style transfer via image domain style and content disentanglement. In addition, domain adaptation [4] applies a cycle consistency loss to cross-domain style transformation.

More recently, example-guided synthesis [27, 41] is proposed to transfer the style of an example image to a target condition, e.g. a semantic label map. Specifically, Lin *et al*. [27] apply dual learning to disentangle the style for guided synthesis, Wang *et al*. [41] extract style-consistent data pairs from videos for model training. In addition, Park *et al*. [36] adopt I2I networks to self-encoding versions for example-guided style transfer. Different from [27, 41, 36], we address spatial alignment of complex scenes for better *style integration in multiple regions of an image*. Furthermore, our patch-based self-supervision learning scheme does not require video data and is a general version of self-encoding.

**Correspondence Matching for Synthesis.** Finding correspondence is critical for many synthesis tasks. For instance, Siarohin *et al*. [39] apply the affine transformation on reference person images to improve pose-guided person image synthesis, Wang *et al*. [42] use optical flow to align frames for coherent video synthesis. However, the affine transformation and optical flow cannot adequately model the correspondences between two arbitrary scenes.

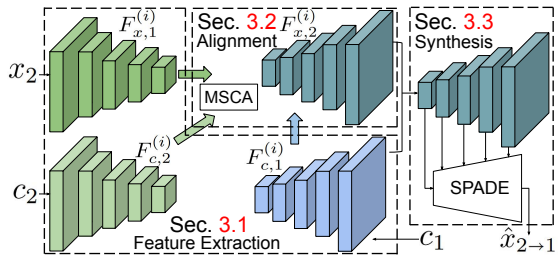


Figure 2. Our generator consists of three steps, namely i) feature extraction, ii) spatial feature alignment, and iii) image synthesis. We elaborate each step in its corresponding section, respectively.

The recent self-attention [44, 48] can capture general pair-wise correspondences. However, self-attention is computationally intensive at high-resolution. Later, Chen *et al*. [6] propose to factorize self-attention for efficient video classification. Inspired by [6], we propose an attention-based module named MSCA. It is worth noting MSCA is based on cross-attention and feature masking for spatial alignment and image synthesis.

## 3. Method

The proposed approach aims to generate scene images that align with given semantic maps. Differ from conventional semantic image synthesis methods [19, 43, 36], our model takes an exemplary scene as an extra input to provide more controllability over the generated scene image. Unlike existing exemplar-base approaches [27, 41], our model addresses the more challenging case where the exemplary inputs are structurally and semantically unaligned with the given semantic map.

Our method takes a semantic map $c_1$, a reference image $x_2$ and its corresponding semantic map $c_2$ as inputs and synthesizes an image $\hat{x}_{2\to1}$ which matches the style of $x_2$ and structure of $x_1$ using a generator G, $\hat{x}_{2\to1} = \mathrm{G}(c_1, x_2, c_2)$. As shown in Fig. 2, the generator G consists of three parts, namely i) feature extraction ii) feature alignment and iii) image synthesis. In Sec. 3.1, we describe the first part that extracts features from inputs of both scenes. In Sec. 3.2, we propose a masked spatial-channel attention (MSCA) module to distill features and discovery relations between two arbitrarily structured scene. Unlike the affine-transformation [20] and flow-base warping [42], MSCA provides a better interpretability to the scene alignment task. In Sec. 3.3, we introduce how to use the aligned features for image synthesis. Finally, in Sec. 3.4, we propose a patch-based self-supervision scheme to facilitate learning.

### 3.1. Feature Extraction

Taking an image $x_2$ and label maps $c_1, c_2$ as inputs, the feature extraction module extracts multi-scale feature maps for each input. Specifically, the feature map $F^{(i)}_{x,2}$ of image $x_2$ at scale $i$ is computed by:

$$F^{(i)}_{x,2} = W^{(i)}_x * F^{(i)}_{\text{vgg}}(x_2), \quad \text{for } i \in \{0, \dots, L\}, \tag{1}$$

where $*$ denotes the convolution operation, $F^{(i)}_{\text{vgg}}$ denotes the feature map extracted by VGG-19 [40] at scale $i$, and $W^{(i)}_x$ denotes a $1 \times 1$ convolutional kernel for feature compression. $L$ is the number scales and we set $L = 4$ in this paper.

For label map $c_1$, its feature $F^{(i)}_{c,1}$ is computed by:

$$F^{(i)}_{c,1} = \begin{cases} \text{LReLU}(W^{(i)}_c * c^{(i)}_1) & \text{for } i = L, \\ \text{LReLU}(W^{(i)}_c * [\Uparrow (F^{(i+1)}_{c,1}), c^{(i)}_1]) & \text{otherwise}, \end{cases} \tag{2}$$

where $\Uparrow(\cdot)$ denotes $\times 2$ bilinear interpolation, $c^{(i)}_1$ denotes the resized label map, $W^{(i)}_c$ denotes a $1 \times 1$ convolutional kernel for feature extraction, and operation $[\cdot, \cdot]$ denotes channel-wise concatenation. Note that as scale $i$ decreases from $L$ down to 0, the feature resolutions in Eq. 2 are progressively increased to match a finer label maps $c^{(i)}_1$.

Similarly, applying Eq. 2 with the same weights to label map $c_2$, we can extract its features $F^{(i)}_{c,2}$:

$$F^{(i)}_{c,2} = \begin{cases} \text{LReLU}(W^{(i)}_c * c^{(i)}_2) & \text{for } i = L \\ \text{LReLU}(W^{(i)}_c * [\Uparrow (F^{(i+1)}_{c,2}), c^{(i)}_2]) & \text{otherwise} \end{cases}. \tag{3}$$

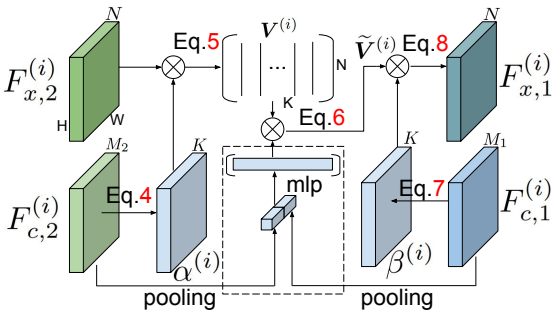


Figure 3. The spatial-channel attention module for feature alignment. Our module takes image features map $F^{(i)}_{x,2}$ and segmentation features map $F^{(i)}_{c,1}$, $F^{(i)}_{c,2}$ as inputs to output a new image feature map $F^{(i)}_{x,1}$ that is aligned to condition $c_1$.

### 3.2. Masked Spatial-channel Attention Module

As shown in Fig. 3, taking the image features $F^{(i)}_{x,2}$ and the label map features $F^{(i)}_{c,1}$, $F^{(i)}_{c,2}$ as inputs[1], the MSCA module generates a new image feature map $F^{(i)}_{x,1}$ that has the content of $F^{(i)}_{x,2}$ but is aligned with $F^{(i)}_{c,1}$. We elaborate the detailed procedures as follows:

[1]We assume spatial resolution at scale $i$ being $H \times W$ and channel size of $F^{(i)}_{x,2}$, $F^{(i)}_{c,1}$, $F^{(i)}_{c,2}$ being $N, M_1, M_2$, respectively.

**Spatial Attention.** Given feature maps $F^{(i)}_{x,2}, F^{(i)}_{c,2}$ of the exemplar scene, the module first computes a spatial attention tensor $\alpha^{(i)} \in [0,1]^{K\cdot H\cdot W}$:

$$\alpha^{(i)} = \mathrm{softmax}_{2,3}(\phi^{(i)} * [F^{(i)}_{x,2}, F^{(i)}_{c,2}]), \tag{4}$$

with $\phi^{(i)} \in \mathbb{R}^{(N+M_2)\cdot K}$ denoting a $1 \times 1$ convolutional filter and $\mathrm{softmax}_{2,3}$ denoting a 2D softmax function on spatial dimensions $\{2,3\}$. The output tensor contains $K$ attention maps of resolution $H \times W$, which serve to attend $K$ different spatial regions on image feature $F^{(i)}_{x,2}$.

**Spatial Aggregation.** Then, the module aggregates $K$ feature vectors from $F^{(i)}_{x,2}$ using the $K$ spatial attention maps of $\alpha^{(i)}$ from Eq. 4. Specifically, a matrix dot product is performed:

$$\boldsymbol{V}^{(i)} = \boldsymbol{F}^{(i)}_{x,2}(\boldsymbol{\alpha}^{(i)})^T, \tag{5}$$

with $\boldsymbol{\alpha}^{(i)} \in [0,1]^{K\cdot HW}$ and $\boldsymbol{F}^{(i)}_{x,2} \in \mathbb{R}^{C\cdot HW}$ denoting the reshaped versions of $\alpha^{(i)}$ and $F^{(i)}_{x,2}$, respectively. The output $\boldsymbol{V}^{(i)} \in \mathbb{R}^{C\cdot K}$ stores feature vectors spatially aggregated from the $K$ independent regions of $F^{(i)}_{x,2}$.

**Feature Masking.** The exemplar scene $x_2$ may contain irrelevant semantics to the label map $c_1$, and conversely, $c_1$ may contain semantics that are unrelated to $x_2$. To address this issue, we apply feature masking on the output of Eq. 5 by multiplying $\boldsymbol{V}^{(i)}$ with a length-$K$ gating vector at each row:

$$\widetilde{\boldsymbol{V}}^{(i)} = (\boldsymbol{V}^{(i)})^T \circ \mathrm{mlp}([\mathrm{gap}(F^{(i)}_{c,1}), \mathrm{gap}(F^{(i)}_{c,2})]), \tag{6}$$

where $\mathrm{mlp}(\cdot)$ denotes a 2-layer MLP followed by a sigmoid function, gap denotes a global average pooling layer, $\circ$ denotes broadcast element-wise multiplication, and $\widetilde{\boldsymbol{V}}^{(i)}$ denotes the masked features. The design of feature masking in Eq. 6 resembles to Squeeze-and-Excitation [14]. Using the integration of global information from label maps $c_1$ and $c_2$, features are filtered.

**Channel Attention.** Given feature $F^{(i)}_{c,1}$ of label map $c_1$, a channel attention tensor $\beta^{(i)} \in [0,1]^{K\cdot H\cdot W}$ is generated as follows:

$$\beta^{(i)} = \mathrm{softmax}_1(\psi^{(i)} * F^{(i)}_{c,1}), \tag{7}$$

with $\psi^{(i)} \in \mathbb{R}^{M_1\cdot K}$ denoting a $1\times 1$ convolutional filter and $\mathrm{softmax}_1$ denoting a softmax function on channel dimension. The output $\beta^{(i)}$ serves to dynamically reuse features from $\widetilde{\boldsymbol{V}}^{(i)}$.

**Channel Aggregation.** With channel attention $\beta^{(i)}$ computed in Eq. 7, feature vectors at $HW$ spatial locations are aggregated again from $\widetilde{\boldsymbol{V}}^{(i)}$ via matrix dot product:

$$\boldsymbol{F}^{(i)}_{x,1} = \widetilde{\boldsymbol{V}}^{(i)}(\boldsymbol{\beta}^{(i)})^T, \tag{8}$$

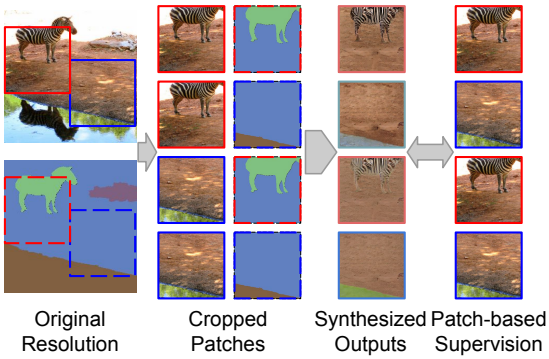


Figure 4. Our patch-based self-supervision scheme performs self-reconstruction and cross-reconstruction for two-patches from the same image.

where $\boldsymbol{\beta}^{(i)} \in \mathbb{R}^{K\cdot HW}$ denotes the reshaped version of $\beta^{(i)}$. The output $\boldsymbol{F}^{(i)}_{x,1} \in \mathbb{R}^{N\cdot HW}$ represents the aggregated features at $HW$ locations. The output feature map $F^{(i)}_{x,1}$ is generated by reshaping $\boldsymbol{F}^{(i)}_{x,1}$ to size $N \times H \times W$.

**Remarks.** Spatial attention (Eq. 4) and aggregation (Eq. 5) attend to $K$ independent regions from feature $F^{(i)}_{x,2}$, then store the $K$ features into $\boldsymbol{V}^{(i)}$. After feature masking, given a new label map $c_1$, channel attention (Eq. 4) and aggregation (Eq. 8) combine $\widetilde{\boldsymbol{V}}^{(i)}$ at each location to compute a output feature map. As results, each output location finds its correspondent regional features or ignored via feature masking. In this way, the feature of example scene is aligned. Note that when $K = 1$ and $\alpha^{(i)}$ is constant, the above operations is essentially a global average pooling. We show in experiment that $K = 8$ is sufficient to dynamically capture visually significant scene regions for alignment.

**Multi-scaling.** Both global color tone and local appearances are informative for the style-constraint synthesis. Therefore, we apply MSCA modules at all scales $i \in \{0, \ldots, L\}$ to generate global and local features $F^{(i)}_{x,1}$.

### 3.3. Image Synthesis

The extracted features $F^{(i)}_{c,1}$ in Sec. 3.1 capture the semantic structure of $c_1$, whereas the aligned features $F^{(i)}_{x,1}$ in Sec. 3.2 capture the appearance style of the example scene. In this section, we leverage $F^{(i)}_{c,1}$ and $F^{(i)}_{x,1}$ as control signals to generate output images with desired structures and styles.

Specifically, we adopt a recent synthesis model, SPADE [36], and feed the concatenation of $F^{(i)}_{x,1}$ and $F^{(i)}_{c,1}$ to the spatially-adaptive denormalization layer of SPADE at each scale. By taking the style and structure signal as inputs, spatially-controllable image synthesis is achieved. We refer readers to appendix for more network details of the synthesis module.

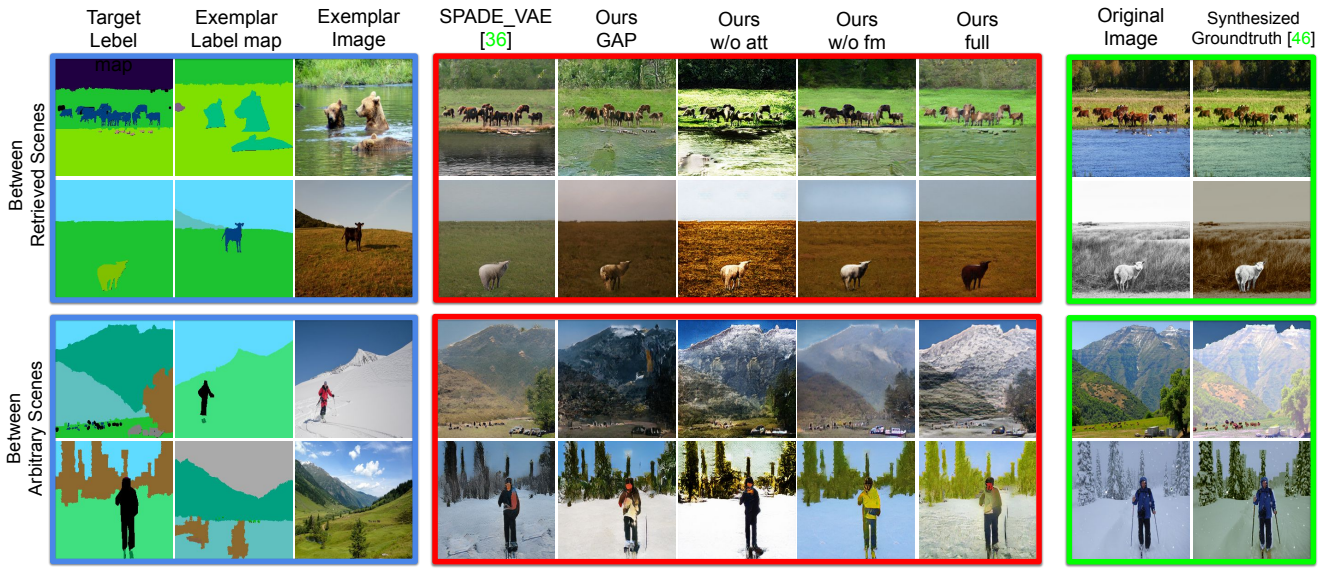


Figure 5. Visual comparisons with SPADE_VAE, and ours ablation models. Example-guided scene synthesis is performed between two **retrieved scenes** (rows 1,2) and two **arbitrary scenes** (rows 3,4). Columns 1 to 3 (blue) depict depict the target label maps, exemplar label maps and the associated images, respectively. Columns 4 to 8 in (red) depict different methods and our model (Columns 8). Columns 9 and 10 (green) respectively depict original images from target label map and synthesized ground truth using [46] in the *retrieved* dataset (top of Table 1). In comparison, our method clearly produces the most style-consistent (**with the exemplar!**) and visually plausible images.

### 3.4. Patch-Based Self-Supervision

Training a synthesis model requires style-consistent scene pairs. However, paired scenes are hard to acquire. To overcome the issue, we propose a patch-based self-supervision scheme which enables training.

Our basic assumption is that if patches $x_p$ and $x_q$ come from the same scene, they share the same style. Consequently, using patch $x_p$ as exemplar, both $x_p$ and the other patch $x_q$ can be reconstructed, i.e. self-reconstruction and cross-reconstruction. More formally, we sample non-overlapping patches $\langle x_p, c_p \rangle$ and $\langle x_q, c_q \rangle$ at locations $p$ and $q$ from a same scene $\langle x, c \rangle$. To enable training, four images are synthesized in one training step:

$$\begin{aligned} \hat{x}_{p\to p} &= \mathrm{G}(c_p, x_p, c_p), \\ \hat{x}_{p\to q} &= \mathrm{G}(c_q, x_p, c_p), \\ \hat{x}_{q\to p} &= \mathrm{G}(c_p, x_q, c_q), \\ \hat{x}_{q\to q} &= \mathrm{G}(c_p, x_p, c_p), \end{aligned} \tag{9}$$

and compared against groundtruths $x_p, x_q, x_p, x_q$. An illustrative example is shown in Fig. 4. Note that patches $x_p, x_q$ do not necessary share the same semantics and our model is required to complete example-missing regions with reasonable content through learning. Our training objective is adopted from to [36]. However, we apply pixel domain $\ell_1$ loss to encourage color consistency. In our implementation, the generation processes in Eq. 9 share the same feature extraction, spatial attention, channel attention computation to reduce memory footprint during training.

## 4. Experiments

**Dataset** Our model is trained on the *COCO-stuff* dataset [3]. It contains densely annotated images captured from various scenes. We remove indoor images and images of random objects from the training/validation set, resulting in $21,648/499$ scene images for training/testing.

During training, we resize images to $512 \times 512$ then crop two non-overlapping $256 \times 256$ patches to facilitate patch-based self-supervision. The two patches are cropped either in the left and right halves of the $512 \times 512$ image, or alternatively in the top and bottom halves.

The COCO-stuff dataset does not provide ground-truth for example-guided scene synthesis, i.e. two scene images with the exact same styles. To qualitatively evaluate model performances, we require a model to transfer the style from $x_2$ to $x_1$, where $x_2$ is the test image and $x_1$ is the generated image, in three ways: i) *duplicating*: we use the test image itself to test self-reconstruction, ii) *mirroring*: $x_1$ is generated by horizontally mirroring $x_2$, iii) *retrieving*: $x_1$ is generated by finding the best match from the larger image pool. Specifically, we generate 20 candidate images from the training set with the smallest label histogram intersections. Out of the 20 images, the best-matching image $x_1$ is generated using SIFT Flow [28]. Finally, since the color of $x_1$ and $x_2$ are not the same, we apply [46] on image $x_1$ for color correction. Examples of the *retrieving* pairs are shown in Fig. 5, in columns 3 and 10.

**Implementation Details** The number of attention maps

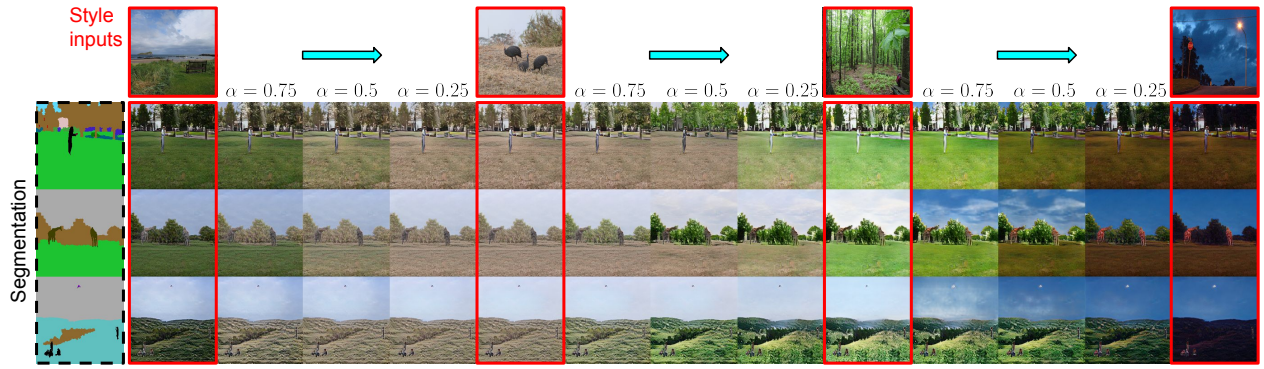


Figure 6. Style interpolation and traverse at test stage. We perform style traverse along path `grass` → `dessert` → `forest` → `night`. Please refer to the Interpolation part in Sec. 4 for details.

| Methods | PSNR↑ | SSIM↑ | LPIPS↓ | FID↓ |
|---|---|---|---|---|
| *retrieving* | | | | |
| SPADE_VAE [36] | 15.62 | 0.39 | 0.480 | 89.77 |
| ours GAP | 15.77 | 0.39 | 0.456 | 89.55 |
| ours MSCA w/o att | 11.76 | 0.27 | 0.524 | 98.35 |
| ours MSCA w/o fm | 15.64 | **0.40** | 0.455 | 89.58 |
| our full | **15.98** | **0.40** | **0.449** | **85.87** |
| *mirroring* | | | | |
| SPADE_VAE [36] | 15.72 | 0.39 | 0.478 | 89.58 |
| ours GAP | 16.06 | 0.39 | 0.446 | 89.54 |
| ours MSCA w/o att | 12.13 | 0.28 | 0.512 | 98.02 |
| ours MSCA w/o fm | 16.52 | **0.42** | 0.442 | 88.40 |
| our full | **16.95** | **0.42** | **0.425** | **83.20** |
| *duplicating* | | | | |
| SPADE_VAE [36] | 15.35 | 0.38 | 0.476 | 90.69 |
| ours GAP | 15.70 | 0.38 | 0.438 | 88.51 |
| ours MSCA w/o att | 11.92 | 0.28 | 0.508 | 102.24 |
| ours MSCA w/o fm | 15.91 | **0.40** | 0.437 | 89.44 |
| our full | **16.50** | **0.40** | **0.420** | **84.93** |

Table 1. Quantitative comparisons of different methods in terms of PSNR, SSIM, LPIPS [49] and Frchet Inception Distance (FID) [13]. Higher scores are better for metrics with uparrow (↑), and vice versa.

$K$ for MSCA modules are set to $8, 16, 16, 16, 16$ from scale $0$ to $4$. The learning rate is set to $0.0002$ for the generator and the discriminator. The weights of generator are updated every $5$ iterations. We adopt the Adam [24] optimizer ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) in all experiments. Our synthesis model and all comparative models are trained for $20$ epochs to generate the results in the experiments.

During implementation, we pretrain the spatial-channel attention with a lightweight feature decoder to avoid the ineffective but extremely slow updating of SPADE parameters. Specifically, at each scale, the concatenation of $F^{(i)}_{x,1}$ and $F^{(i)}_{c,1}$ in Sec. 3.3 at each scale is fed into a $1 \times 1$ convolutional layer to reconstruct the ground-truth VGG feature at the corresponding scale. The pretraining takes around $4\%$ of the total training time to converge. More details of the pretraining procedure is provided in the appendix.

**Quantitative Evaluation** We compare our approach with an example-guided synthesis approach: variational autoencoding SPADE (SPADE_VAE) [36] which is based on a self-reconstruction loss for training. Therefore, we directly use the resized $256 \times 256$ images to train the model. We also attempt to train two example-guided synthesis models [27] and [41] ([41] is trained using patch-based self-supervision) but cannot achieve visually good results. We leave the result of [27, 41] in the appendix. In addition, three ablation models are evaluated (see Ablation Study).

For quantitative evaluation, we apply low-level metrics including PSNR and SSIM [45], and perceptual-level metrics including Perceptual Image Patch Similarity Distance (LPIPS) [49] and Frchet Inception Distance (FID) [13] on different models. For LPIPS, we use the linearly calibrated VGG model (see [49] for details).

As shown in Table 1, our method clearly outperforms the remaining methods. Improvements in low-level and perceptual-level measurements suggest that our model better preserves color and texture appearances. We observe that the performances of various approaches on the *retrieving* dataset are worse and less differentiated than their counterparts on the *mirroring* and *duplicating* datasets. It suggests that the *retrieving* dataset is harder and noisier, as one cannot retrieve images that have the exact same styles. On *retrieving* dataset, our approach achieves a moderate +0.36 PSNR gain over SPADE_VAE (from 15.62 to 15.98). By contrast, our approach achieves visually superior results over SPADE_VAE on *duplicating* and *mirroring*, e.g. +1.15 PSNR gain (from 15.35 to 16.50) on *duplicating* and +1.23 PSNR gain (from 15.72 to 16.95) in PSNR on *mirroring*.

**Qualitative Evaluation** Fig. 5 qualitatively compares our model against the remaining models on two **retrieved** scenes (rows 1-2) and two **arbitrary** scenes (rows 3-4). Our model achieves better style-consistent example-guided synthesis. Remarkably, in rows 3-4, even though the two scenes have very different semantics (indicated by the different col-

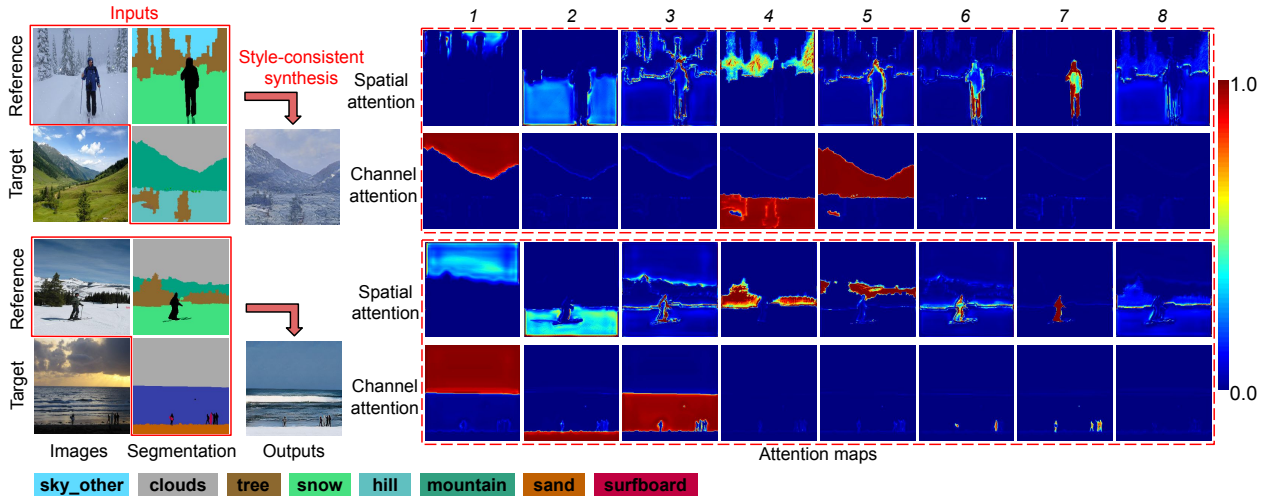


Figure 7. Right: the learned spatial attention and channel attention. Left: inputs and outputs of our model. Each spatial and channel attention attends to a specific region in the reference image and segmentation, respectively. By examining the segmentation semantics, we observe the following transformation patterns: `sky_other` → `clouds`, `tree` → {`tree, hill`} for row 1, and `clouds` → `clouds`, `snow` → `sand`, `other` → {`surfboard, other`} for row 2.

ors of the corresponding label maps), our model can still maintain the styles of the exemplars while maintaining the correct semantics of the target label maps, e.g. generating "snow" rather than "grass" in row 4.

Also notice that sometimes our results are more style-consistent than the synthesized ground truths (last column). This further shows that the existing style transfer approach [9, 46, 31] cannot be directly applied to exemplar-guided scene synthesis for satisfactory results.

**Ablation Study** To evaluate the effectiveness of our design, we separately train three variants of our model: i) *our GAP* that replaces the MSCA module with global average pooling, ii) *our MSCA w/o att* that keeps MSCA modules but replaces spatial and channel attention of MSCA by one-hot label maps from source and target domains, respectively. In such way, alignment is performed on regions with the same semantic labeling, and iii) *our MSCA w/o fm* that keeps MSCA modules but removes the feature masking procedures. In Table 1 and Fig. 5, our model clearly achieves the best quantitative and qualitative results. In comparison, in Fig. 5, *our GAP* produces similar appearances in each region, as GAP cannot distinguish local appearances. *Our w/o att* is less stable in training and cannot generate plausible results. We hypothesize that the label-level alignment will generate more misaligned and noisier feature maps, thus hurts training. *our MSCA w/o fm* cannot perform correct appearance transformation, for instance, transferring "sky" to "snow" (Fig. 5, last row).

**The Effect of Attention** To understand the effect of spatial-channel attention, we visualize the learned spatial and channel attention in Fig. 7. We observe that: a) spatial attention can attend to multiple regions of the reference image. For each reference region, channel attention finds the corresponding target region. b) spatial-channel attention can detect and utilize the semantic similarities between segments to transfer visual features. In the top row of Fig. 7, attention in channels $1, 4$ respectively perform transformations: `sky_other` → `clouds`, `tree` → {`tree, hill`}. In the bottom row, attention in channels $1, 2, 7$ respectively perform transformations: `clouds` → `clouds`, `snow` → `sand` and `other` → {`surfboard, other`}.

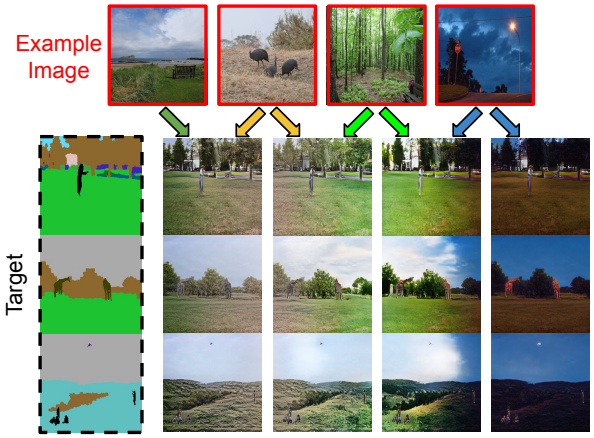


Figure 8. Spatially interpolate two styles in a single image at test stage. The styles of the synthesized images are deliberately changed from left to right.

**Interpolation** We can easily control the synthesized styles in the test stage by manipulating attentions. Here, we show how to interpolate between two styles using our trained model: given two example images $x_2$ and $x_3$, we

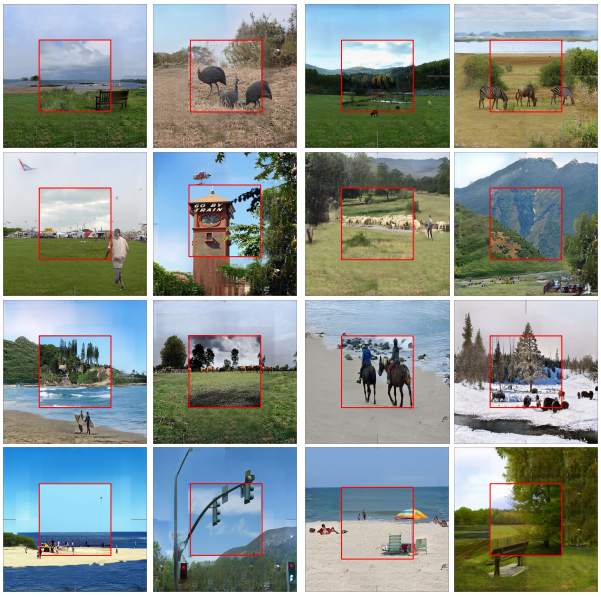

Figure 9. Given a scene patch at center, our model can generate Scene extrapolation based on patch.

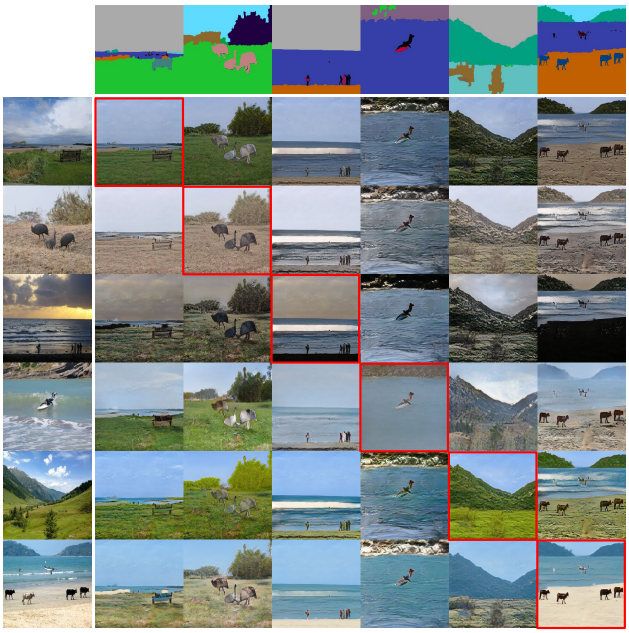

Figure 10. Style-structure swapping on 6 semantically unaligned arbitrary scenes at resolution $256 \times 256$. Our model can generalize across very different scene semantics and synthesize images with reasonable and consistent styles. Note that the images along the diagonal (red boxes) are *self-reconstruction*, which are generally quite good. Please zoom in for details.

first compute their image features $F^{(i)}_{x,2}, F^{(i)}_{x,3}$ and the spatial-attention maps $\alpha^{(i)}_2, \alpha^{(i)}_3$. Given an interpolating factor $\alpha \in [0, 1]$ where $\alpha = 1$ means ignoring the example scene $x_3$, the spatial attention map of the first scene is modified by $\alpha^{(i)}_2 := \alpha^{(i)}_2 + \log(\frac{\alpha^{(i)}_2}{1-\alpha^{(i)}_2})$. Afterwards, both feature maps $F^{(i)}_{x,2}, F^{(i)}_{x,3}$ and spatial attention $\alpha^{(i)}_2, \alpha^{(i)}_3$ are concatenated along the horizontal axis. In addition, the masking score (output of the 2-layer MLP in Eq. 6) is also interpolated. With the remaining procedures unchanged, i.e., same spatial aggregation, feature masking, channel aggregation and synthesis, interpolation results are readily generated. As shown in Fig. 6, with slight modifications, our model can perform effective style interpolation. Specifically, the style traverses along the path `grass` $\rightarrow$ `dessert` $\rightarrow$ `forest` $\rightarrow$ `night` is achieved in Fig. 6.

Likewise, by manipulating the channel attention at each spatial location, it is possible to adaptively mix style to synthesize an output image, i.e. spatial styles interpolation. As shown in Figure 8, using the previous input, we interpolate between styles from left to right in a single image.

**Extrapolation** Given a scene patch at the center, our model can achieve scene extrapolation, i.e. generating beyond-the-border image content according to the semantic map guidance. A $512 \times 512$ extrapolated images is generated by weighted combining synthesized $256 \times 256$ patches at 4 corners and 10 other random locations. As shown in Fig. 9, our model generates visually plausible extrapolated images, showing the promise of our proposed framework for guided scene panorama generation.

**Swapping Style** Fig. 10 shows reference-guided style swapping on six distinctively different scenes. For the same segmentation mask, we generate multiple outputs using different reference images. Our approach can reasonably transfer styles among multiple scenes, including grassland, dessert, ocean view, ice land, etc. More results are included in the appendix.

## 5. Conclusion

We propose to address a challenging example-guided scene image synthesis task. To propagate information between structurally uncorrelated and semantically unaligned scenes, we propose an MSCA module that leverages decoupled cross-attention for adaptive correspondence modeling. With MSCA, we propose a unified model for joint global-local alignment and image synthesis. We further propose a patch-based self-supervision scheme that enables training. Experiments on the COCO-stuff dataset show significant improvements over the existing methods. Furthermore, our approach provides interpretability and can be extended to other content manipulation tasks.

## References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2

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

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018. 2, 5

[4] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 40–48, 2018. 2

[5] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1520, 2017. 2

[6] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. In *Advances in Neural Information Processing Systems*, pages 352–361, 2018. 2, 3

[7] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001. 2

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 7

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[11] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018. 2

[12] 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. ACM, 2001. 2

[13] 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 *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 6

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4

[15] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 783–791, 2017. 2

[16] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 2

[17] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. 2

[18] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. 2

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1, 2, 3

[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 3

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

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

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

[25] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–51, 2018. 2

[26] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017. 2

[27] Jianxin Lin, Yingce Xia, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Conditional image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5524–5532, 2018. 1, 2, 3, 6, 10, 11, 12

[28] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010. 5

[29] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 2

[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 1

[31] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4990–4998, 2017. 2, 7

[32] Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. *arXiv preprint arXiv:1805.11145*, 2018. 2

[33] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017. 2

[34] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[35] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 2

[36] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 4, 5, 6, 10, 11

[37] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018. 1

[38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 2

[39] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018. 2

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[41] Miao Wang, Guo-Ye Yang, Ruilong Li, Run-Ze Liang, Song-Hai Zhang, Peter Hall, Shi-Min Hu, et al. Example-guided style consistent image synthesis from semantic labeling. *arXiv preprint arXiv:1906.01314*, 2019. 1, 2, 3, 6, 10, 11, 12

[42] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 2, 3

[43] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 1, 2, 3

[44] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 2

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[46] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. *arXiv preprint arXiv:1903.09760*, 2019. 2, 5, 7, 12

[47] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 1

[48] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 2

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6

[50] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2

## Appendix A. The Synthesis Module

As shown in Fig. 11, our image synthesis module (the dash block on the right) takes the image features map $F_{x,1}^{(i)}$ and segmentation features map $F_{c,1}^{(i)}$ as inputs to output a new image $\hat{x}_{2\to1}$. Specifically, at each scale, a SPADE residue block [36] with upsampling layer takes the concatenation of $F_{x,1}^{(i)}$ and $F_{c,1}^{(i)}$ as input to generate an upsampled feature map or image.

## Appendix B. MSCA Pretraining

As shown in Fig. 12, an auxiliary feature decoder (the dash block on the right) is used to pretrain the feature extractors and the MSCA modules. Specifically, at each scale, the concatenation of $F_{x,1}^{(i)}$ and $F_{c,1}^{(i)}$ at each scale is fed into a $1 \times 1$ convolutional layer to reconstruct the ground-truth VGG feature of $x_1$ at the corresponding scale. We weighted sum the L1 losses between predictions and ground-truth at each scales, then apply backpropagation to update weights of the whole model. We pretrain the model for 20 epochs. Because of the light-weight design of the feature decoder, the pretraining step only takes around 12 hours, and around 4% of the total training time.

## Appendix C. Results of [27, 41]

We provide additional results of conditional image-to-image translation (Conditional I2I) [27] and style-guided synthesis [41] in Fig. 13, column 9 and 10. To train the

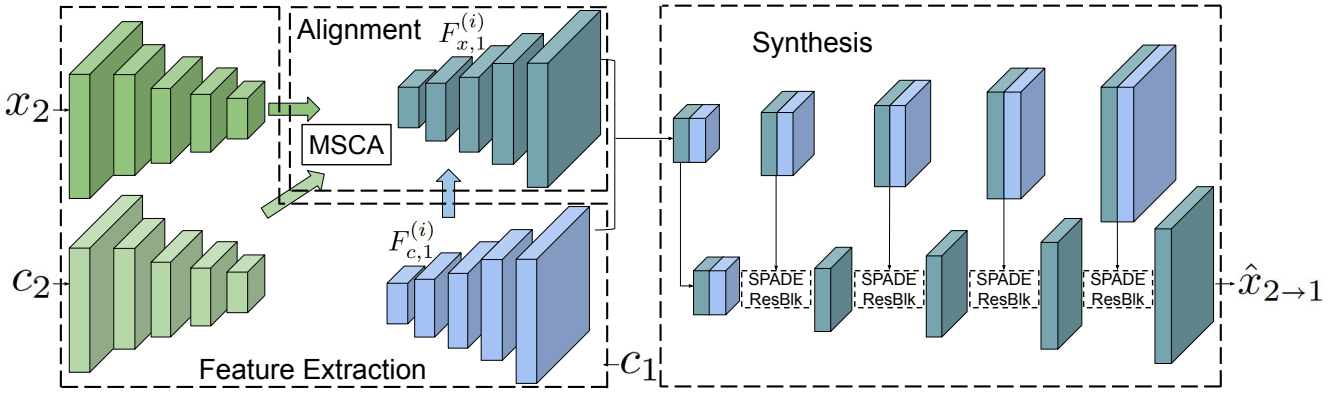


Figure 11. The details of the image synthesis module (the dash block on the right). The image synthesis module takes image features maps $F^{(i)}_{x,1}$ and segmentation features maps $F^{(i)}_{c,1}$ at all scale $i$ as inputs to output a new image $\hat{x}_{2\to1}$. Multiple SPADE residue blocks [36] with upsampling layers are used to upsample the spatial resolutions.

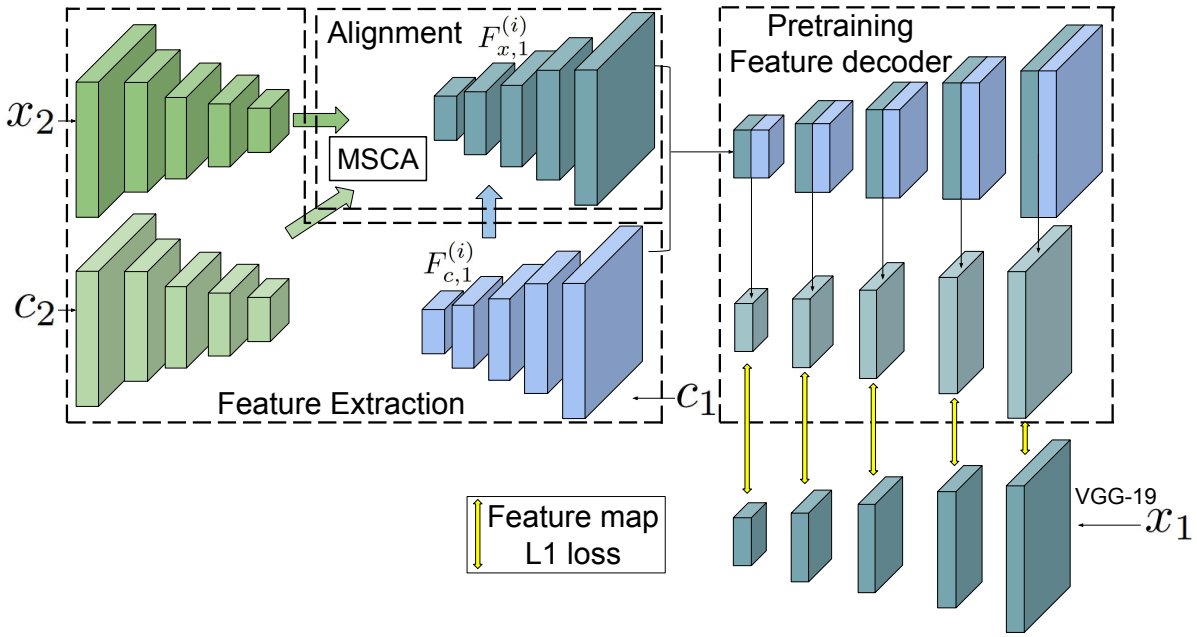


Figure 12. The details of the auxiliary feature decoder for feature extractor and MSCA pretraining (dash block on the right). At each scale $i$, the image features map $F^{(i)}_{x,1}$ and the segmentation features map $F^{(i)}_{c,1}$ are concatenated and feed to a $1 \times 1$ convolution layer to predict the VGG-19 features map of $x_1$.

model of [27], we resize images and semantic label maps to 64, the original resolution used in [27]. We test different learning rates and early stopping strategies to prevent the generator from model collapse. To implement [41], we train the model of [41] using our patch-based self-supervision. We test multiple learning rates and channel sizes of the generator. However, we could not achieves good results for [27] and [41]. We believe the disentanglement strategy of [27] is too challenging for the highly diversified COCO-stuff dataset. Meanwhile, input domain concatenation used in [41] may not be sufficient to capture and fuse the style information for the more challenging scene image dataset. In addition, spatially-adaptive normalization [36] might be required for [41] to better utilize the captured style coding.

## Appendix D. More Style Swapping Results

We show style swapping results on 12 diversified scenes in Fig. 14. As shown in the figure, our model can transfer styles to very different scene semantics and generate style consistent outputs given exemplar images.


Figure 13. More visual comparisons to [27] and [41] (at columns 9 and 10, respectively). Example-guided scene synthesis is performed on **retrieved scenes** (top) and **arbitrary scenes** (buttom). Columns 1 to 3 (blue) depict the target label maps, exemplar label maps and the associated images, respectively. Columns 4 to 8 in (red) depict different methods and our model (Columns 8). Columns 13 and 14 (green) respectively depict original images from target label map and synthesized ground truth using [46] in the *retrieved* dataset. Our method clearly produces the most style-consistent and visually plausible images.

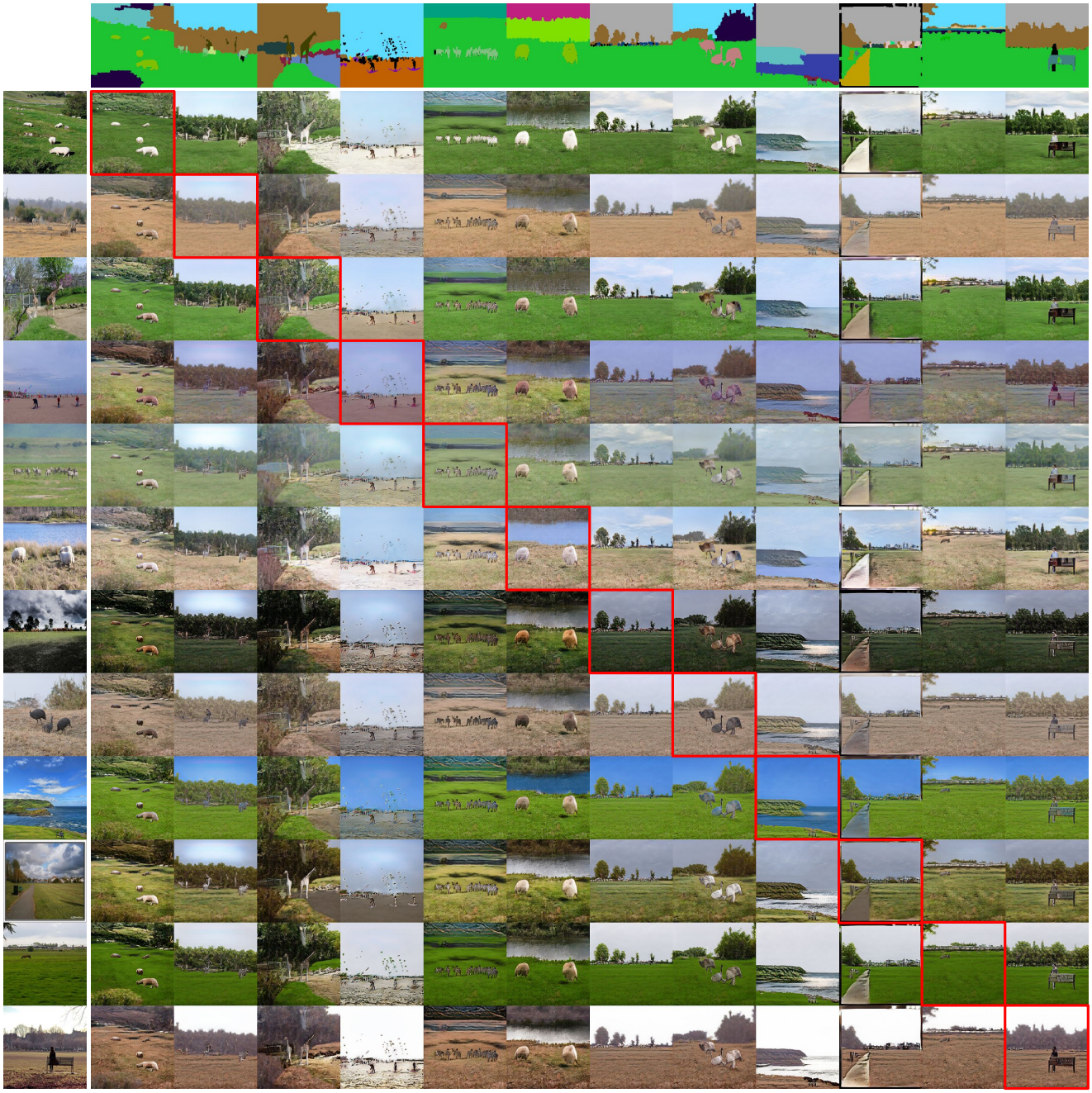

Figure 14. Style-structure swapping on 12 semantically unaligned and arbitrary scenes at resolution $256 \times 256$. Our model can generalize across very different scene semantics and synthesize images with reasonable and consistent styles. Note that the images along the diagonal (red boxes) are *self-reconstruction*, which are generally quite good. Please zoom in for details.