SEAN: Image Synthesis with Semantic Region-Adaptive Normalization

Peihao Zhu[1] Rameen Abdal[1] Yipeng Qin[2] Peter Wonka[1]

[1]KAUST [2]Cardiff University

A. Additional Implementation Details

Generator. Our generator consists of several SEAN ResBlks. Each of them is followed by a nearest neighbor upsampling layer. Note that we only inject the style codes $\mathbf{ST}$ into the first 6 SEAN ResBlks. The other inputs are injected to all SEAN ResBlks. The architecture of our generator is shown in Figure 1.

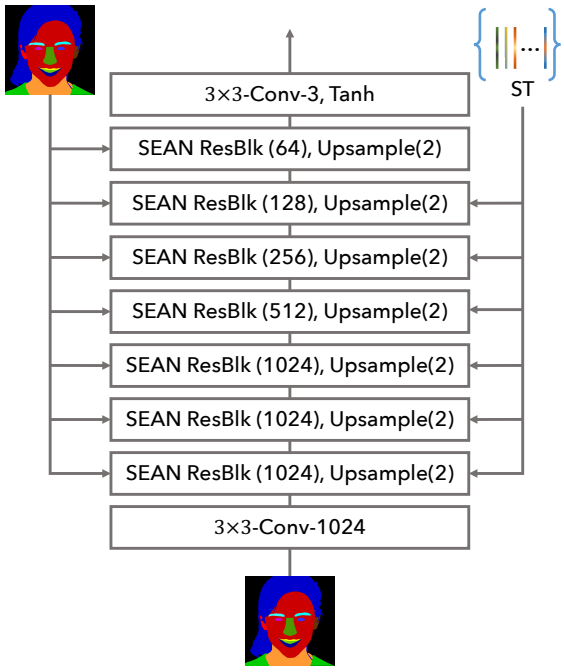

Figure 1: SEAN Generator. The style codes $\mathbf{ST}$ and segmentation mask are passed to the generator through the proposed SEAN ResBlks. The number of feature map channels is shown in the parenthesis after each SEAN ResBlk. To better illustrate the architecture, we omit the learnable noise inputs and per-style Conv layers in this figure. These details are shown in Fig.3 of the main paper (see A_{ij} and B_{ij}).

Discriminator. Following SPADE [9] and Pix2PixHD [12], we employed two multi-scale discriminators with instance normalization (IN) [11] and Leaky ReLU (LReLU). Similar to SPADE, we apply spectral normalization [8] to all the convolutional layers of the discriminator. The architecture of our discriminator is shown in Figure 2.

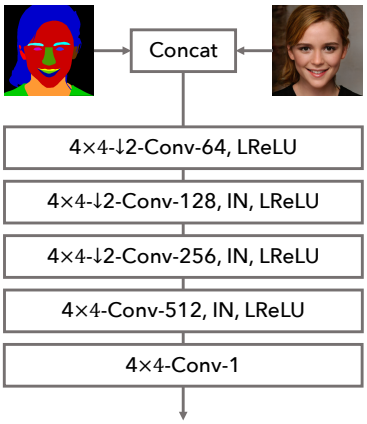

Figure 2: Following SPADE and Pix2PixHD, our discriminator takes the concatenation of a segmentation mask and a style image as inputs. The loss is calculated in the same way as PatchGAN [3].

Style Encoder. Our style encoder consists of a "bottleneck" convolutional neural network and a region-wise average pooling layer (Figure 4). The inputs are the style image and the corresponding segmentation mask, while the outputs are the style codes $\mathbf{ST}$.

Loss function. The design of our loss function is inspired by those of SPADE and Pix2PixHD which contains three components:

(1) *Adversarial loss.* Let E be the style encoder, G be the SEAN generator, D_1 and D_2 be two discriminators at different scales [12], $\mathbf{R}$ be a given style image, $\mathbf{M}$ be the corresponding segmentation mask of $\mathbf{R}$, we formulate the conditional adversarial learning part of our loss function as:

$$\min_{E,G} \max_{D_1,D_2} \sum_{k=1,2} \mathcal{L}_{\text{GAN}}\left(E, G, D_k\right) \tag{1}$$

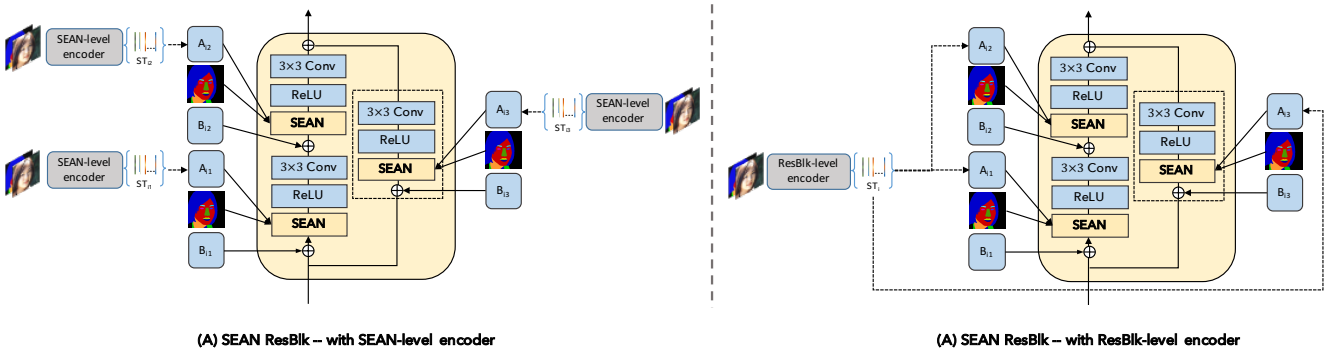

Figure 3: Detailed usage of the SEAN-level encoder and the ResBlk-level encoder within a SEAN ResBlk.

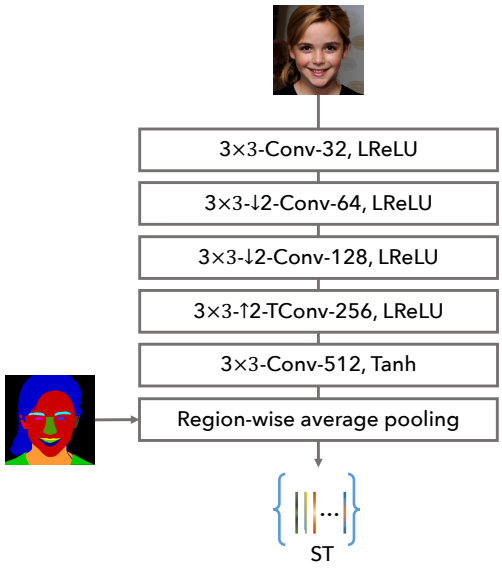

Figure 4: Our style encoder takes the style image and the segmentation mask as inputs to generate the style codes $\mathbf{ST}$.

Specifically, $\mathcal{L}_{\text{GAN}}$ is built with the Hinge loss that:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = &\mathbb{E}\left[\max(0, 1 - D_k(\mathbf{R}, \mathbf{M}))\right] \\ &+ \mathbb{E}\left[\max(0, 1 + D_k(G(\mathbf{ST}, \mathbf{M}), \mathbf{M}))\right]\end{aligned} \tag{2}$$

where $\mathbf{ST}$ is the style codes of $\mathbf{R}$ extracted by E under the guidance of $\mathbf{M}$:

$$\mathbf{ST} = E(\mathbf{R}, \mathbf{M}) \tag{3}$$

(2) *Feature matching loss* [12]. Let T be the total number of layers in discriminator D_k, $D_k^{(i)}$ and N_i be the output feature maps and the number of elements of the i-th layer of D_k respectively, we denote the feature matching loss term $\mathcal{L}_{FM}$ as:

$$\mathcal{L}_{FM} = \mathbb{E}\sum_{i=1}^{T}\frac{1}{N_i}\left[\left\|D_k^{(i)}(\mathbf{R}, \mathbf{M}) - D_k^{(i)}\left(G(\mathbf{ST}, \mathbf{M}), \mathbf{M}\right)\right\|_1\right] \tag{4}$$

(3) *Perceptual loss* [4]. Let N be the total number of layers used to calculate the perceptual loss, $F^{(i)}$ be the output feature maps of the i-th layer of the VGG network [10], M_i be the number of elements of $F^{(i)}$, we denote the perceptual loss $\mathcal{L}_{percept}$ as:

$$\mathcal{L}_{percept} = \mathbb{E}\sum_{i=1}^{N}\frac{1}{M_i}\left[\left\|F^{(i)}(\mathbf{R}) - F^{(i)}\left(G(\mathbf{ST}, \mathbf{M})\right)\right\|_1\right] \tag{5}$$

The final loss function used in our experiment is made up of the above-mentioned three loss terms as:

$$\min_{E,G}\left(\left(\max_{D_1,D_2}\sum_{k=1,2}\mathcal{L}_{\text{GAN}}\right) + \lambda_1\sum_{k=1,2}\mathcal{L}_{\text{FM}} + \lambda_2\mathcal{L}_{\text{percept}}\right) \tag{6}$$

Following SPADE and Pix2PixHD, we set $\lambda_1 = \lambda_2 = 10$ in our experiments.

Training details. We perform 50 epochs of training on all the datasets mentioned in the main paper. During training, all input images are resized to a resolution of 256×256, except for the CityScapes dataset [1] whose images are resized to 512×256. We use Glorot initialization [2] to initialize our network weights.

B. Additional Experimental Details

Table 3 (main paper). Supplementing row 5 and 6 in Table 3 of the main paper, Figure 3 shows how the two variants of style encoders (*i.e.* the SEAN-level encoder and the ResBlk-level encoder) are used in a SEAN ResBlk. Specifically, the SEAN-level encoders extract different style codes for each SEAN block while the same style codes extracted by the ResBlk-level encoder are used by all SEAN blocks within a SEAN ResBlk.

Figure 6 (main paper). We used the Ground Truth (second column in Figure 6 of the main paper) as the style input for all methods. For Pix2pixHD, we generate the results by: (i) encoding the style image into a style vector; (ii) broadcasting the style vector and concatenating it to the mask input of the generator.

C. Justification of Encoder Choice

Figure 5 shows that the images generated by the unified encoder are of better visual quality than those generated by the SEAN-level encoder, especially for challenging inputs (e.g. extreme poses, unlabeled regions), which justifies our choice of unified encoder.

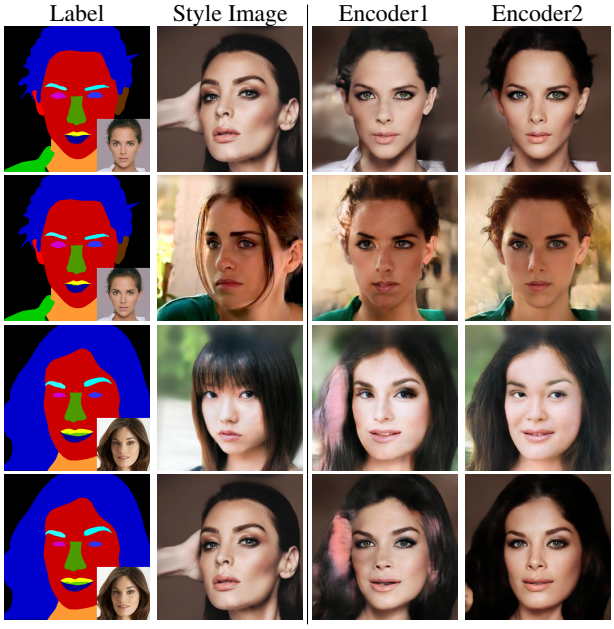

Figure 5: Encoder choice justification. Encoder1 is the SEAN-level encoder and Encoder2 is the unified encoder. SEAN-level encoder is more sensitive to the poses and unlabeled parts of the style image due to the overfitting. Using unified encoder can get more robust style transfer results.

D. Additional Analysis

ST-branch *vs*. Mask-branch. The contributions of ST-branch and mask-branch are determined by a linear combination (parameters α_β and α_γ). The resulting parameters are typically in the range of $0.35 - 0.7$ meaning that both branches are actively contributing to the result. See Fig 6 for one example. It is possible to completely drop the mask-branch, but the results will get worse. It was our initial intuition that the mask branch provides the rough structure and the ST-branch additional details. However, in the end, the interaction is quite complicated and cannot be understood by just varying the mixing parameter.

Extreme Cases. To further demonstrate SEAN's power in texture transfer, we show that highly complex textures from an artistic image can be transferred to a human face (Fig 7). In addition, our method is highly flexible that enables users to paint a semantic region at a spatially unreasonable location arbitrarily (Fig 8).

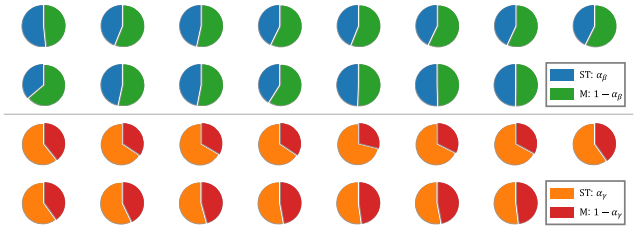

Figure 6: Contributions of ST-branch and Mask-branch for each SEAN normalization block. The pie charts and SEAN normalization blocks are in one-to-one correspondence.

Figure 7: Complex texture transfer.

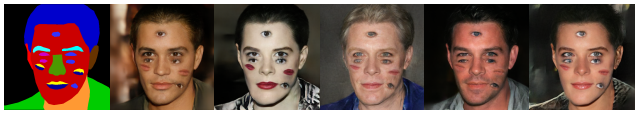

Figure 8: Spatially-flexible painting. Our method allows users to put eyes anywhere on a face.

E. User Study

We conducted a user preference study with Amazon Mechanical Turk (AMT) to illustrate our superior reconstruction results against existing methods (Table 1). Specifically, we created 600 questions for AMT workers to answer. In the end, our questions are answered by 575 AMT workers. For each question, we show the user a set of 5 images: a ground truth image, its corresponding segmentation mask, and 3 reconstruction images obtained by our method, Pix2PixHD [12] and SPADE [9]. The user is then asked to select the reconstructed image closest to the ground truth and with fewest artifacts. To relieve the impact of image orders and make a fair comparison, we picked 100 image sets randomly and created the 600 questions by enumerating all the 6 possible orders of the 3 reconstructed images in each of them.

F. Additional Results

To demonstrate that the proposed per-region style control method builds the foundation of a highly flexible image-editing software, we designed an interactive UI for a demo. Our UI enables high quality image synthesis by transferring the per-region styles from various images to an arbi-

	Pix2PixHD [12]	SPADE [9]	Ours
Preference (%)	23.17	8.83	68.00

Table 1: User preference study (CelebAMask-HQ dataset). Our method outperforms Pix2PixHD [12] and SPADE [9] significantly.

trary segmentation mask. New styles can be created by interpolating existing styles. Please find the recorded videos of our demo in the supplementary material.

Figure 9 shows additional style transfer results on CelebAMask-HQ [6, 5, 7] dataset. Figure 10 and Figure 11 show additional style interpolation results on CelebAMask-HQ and ADE20K datasets.

Figure 12, 13, 14 and 15 show additional image reconstruction results of our method, Pix2PixHD and SPADE on the CelebAMask-HQ [6, 5, 7], ADE20K [13], CityScapes [1] and our Façades datasets respectively. It can be observed that our reconstructions are of much higher quality than those of Pix2PixHD and SPADE.

References

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4

[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. 2

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016. 1

[4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016. 2

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2017. 4

[6] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. *arXiv preprint arXiv:1907.11922*, 2019. 4

[7] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 4

[8] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks, 2018. 1

[9] 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, 3, 4, 7, 8, 9, 10

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 2

[11] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016. 1

[12] 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*, 2018. 1, 2, 3, 4, 7, 8, 9, 10

[13] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4

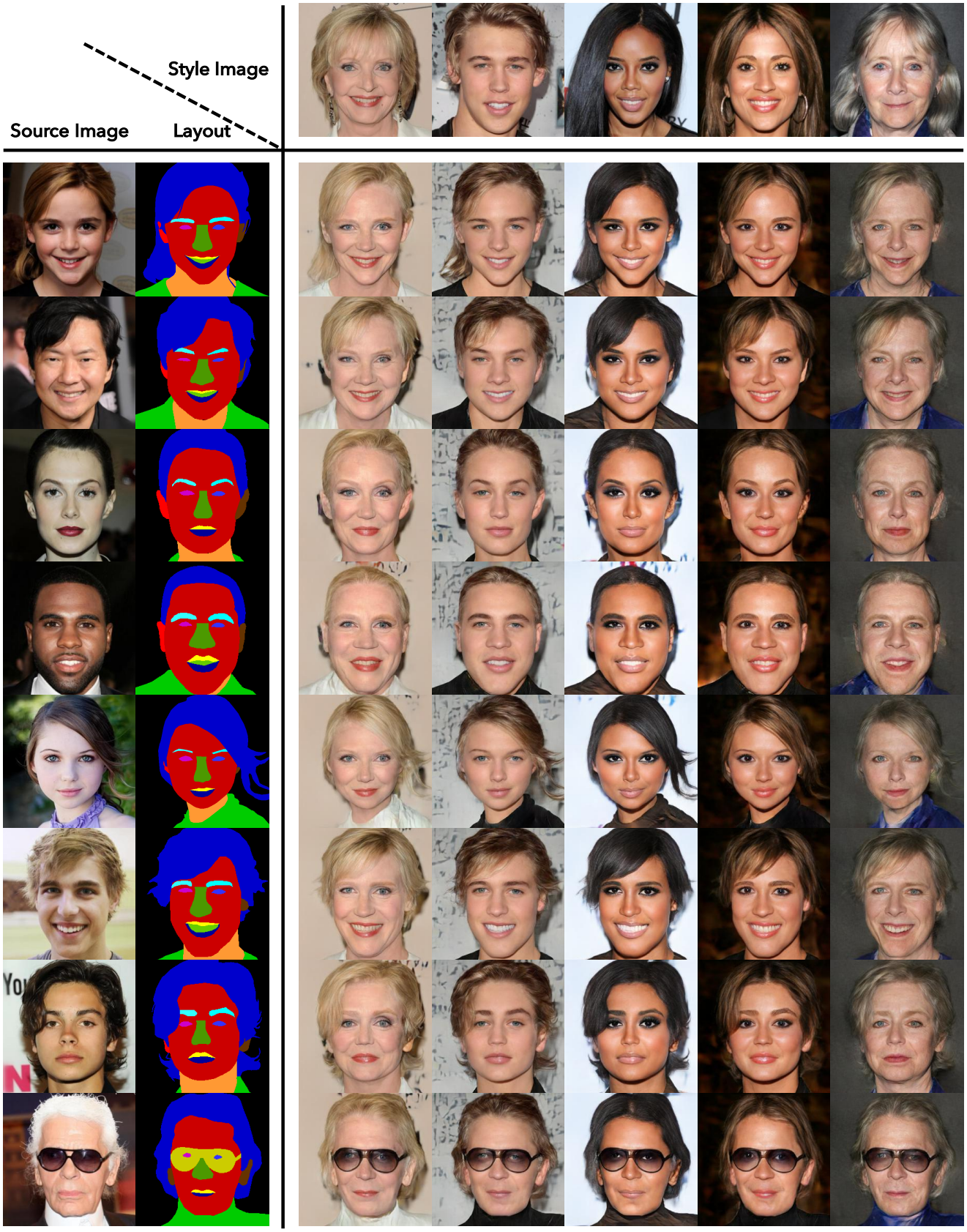

Figure 9: Style transfer on CelebAMask-HQ dataset

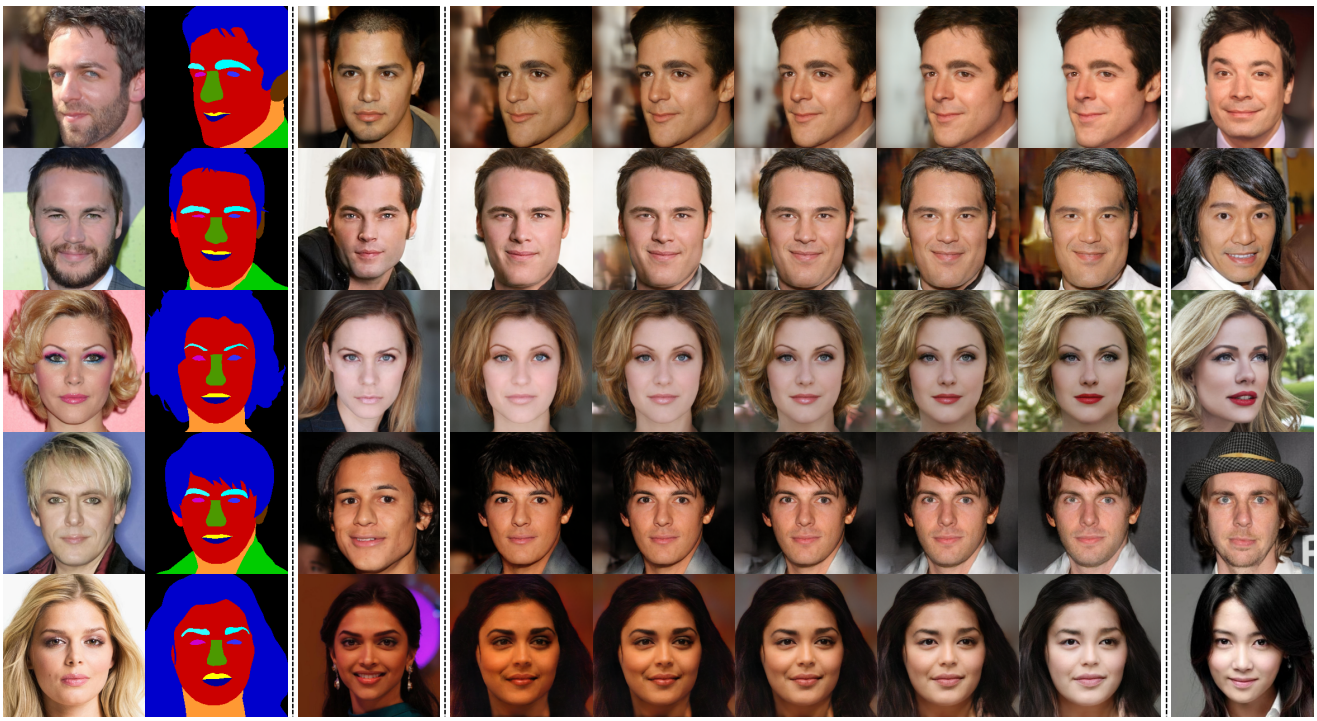

Figure 10: Style interpolation on CelebAMask-HQ dataset

Figure 11: Style interpolation on ADE20K dataset

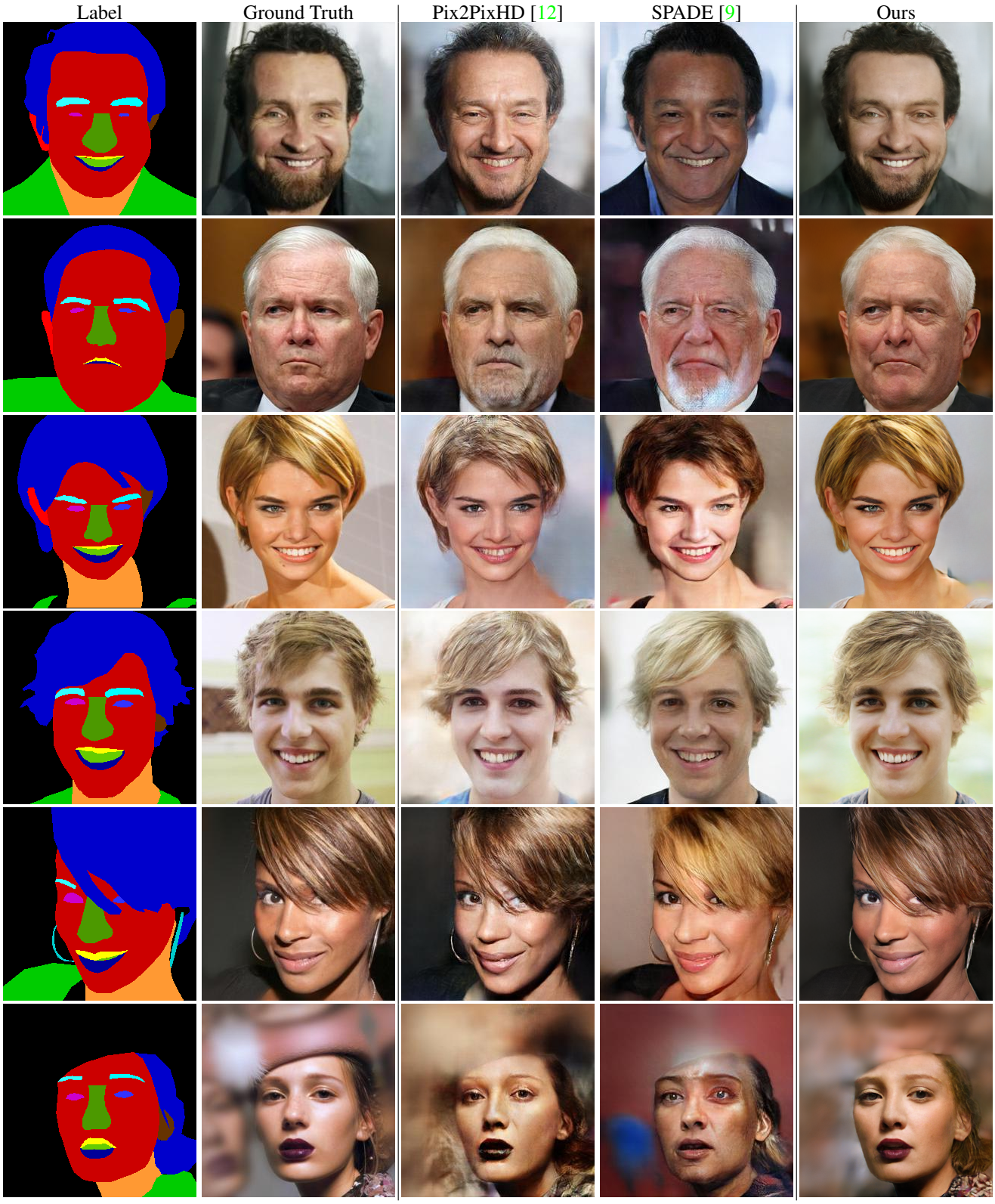

Figure 12: Visual comparison of semantic image synthesis results on the CelebAMask-HQ dataset. We compare Pix2PixHD, SPADE, and our method.

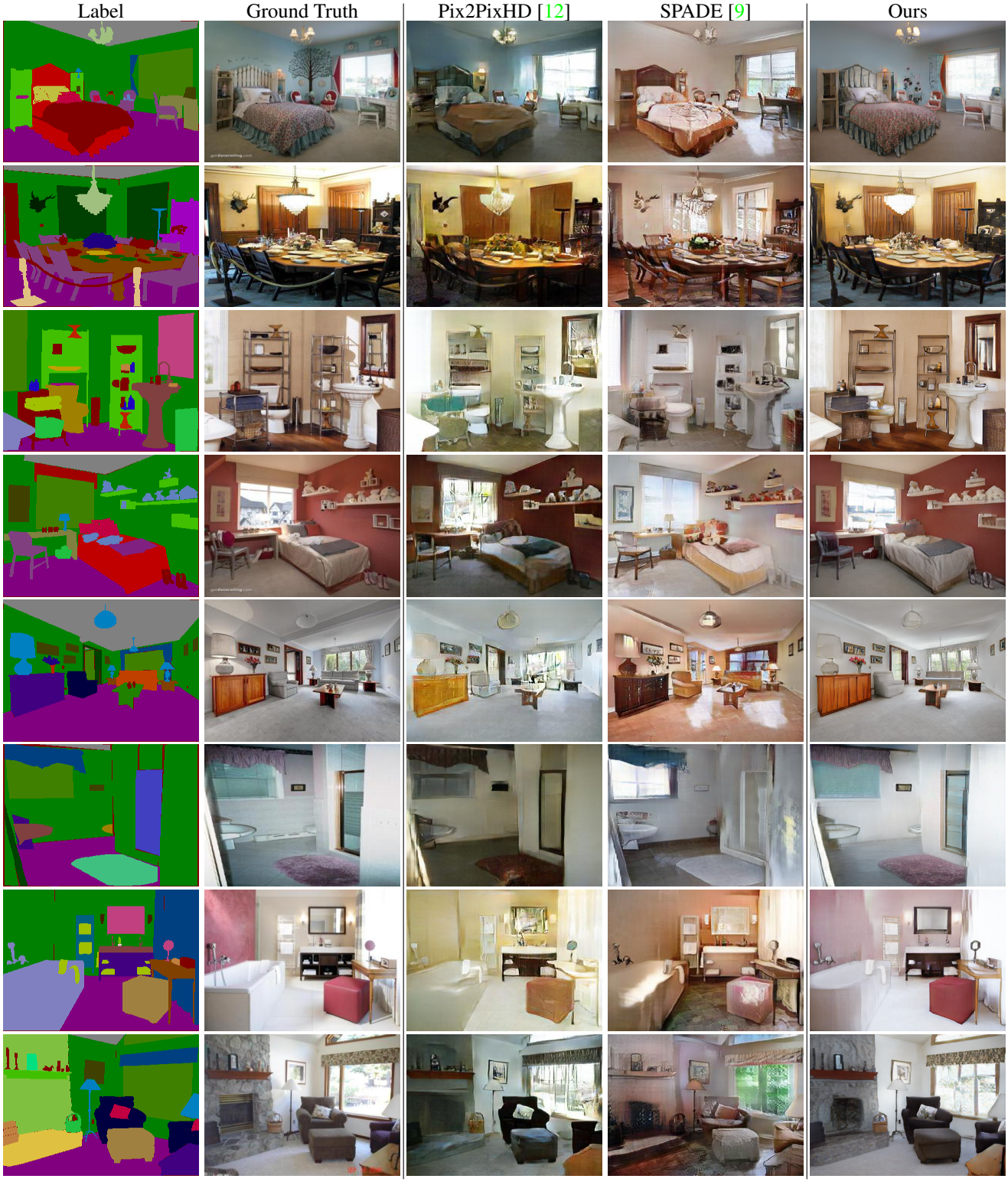

Figure 13: Visual comparison of semantic image synthesis results on the ADE20K dataset. We compare Pix2PixHD, SPADE, and our method.

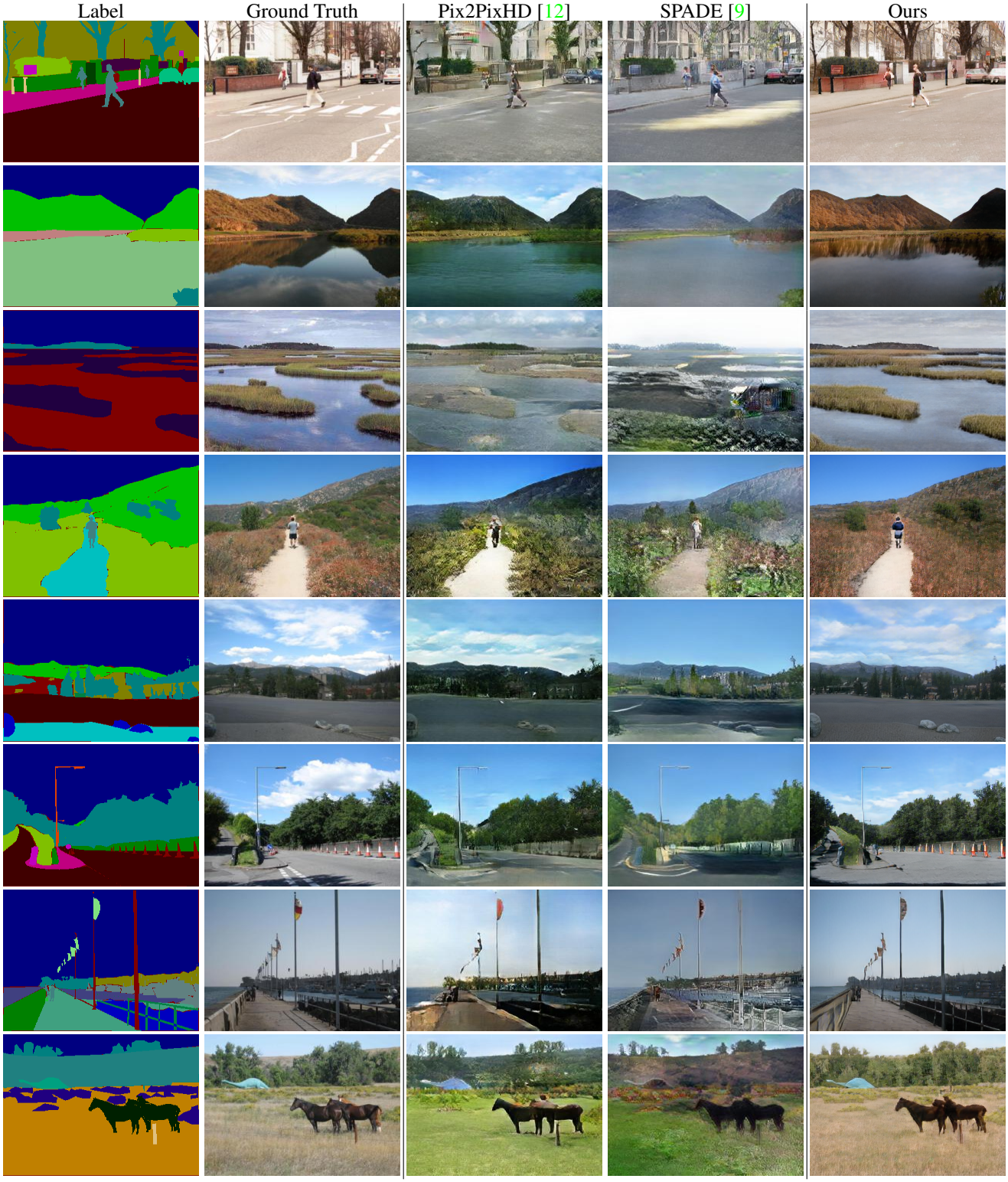

Figure 14: Visual comparison of semantic image synthesis results on the ADE20K dataset. We compare Pix2PixHD, SPADE, and our method.

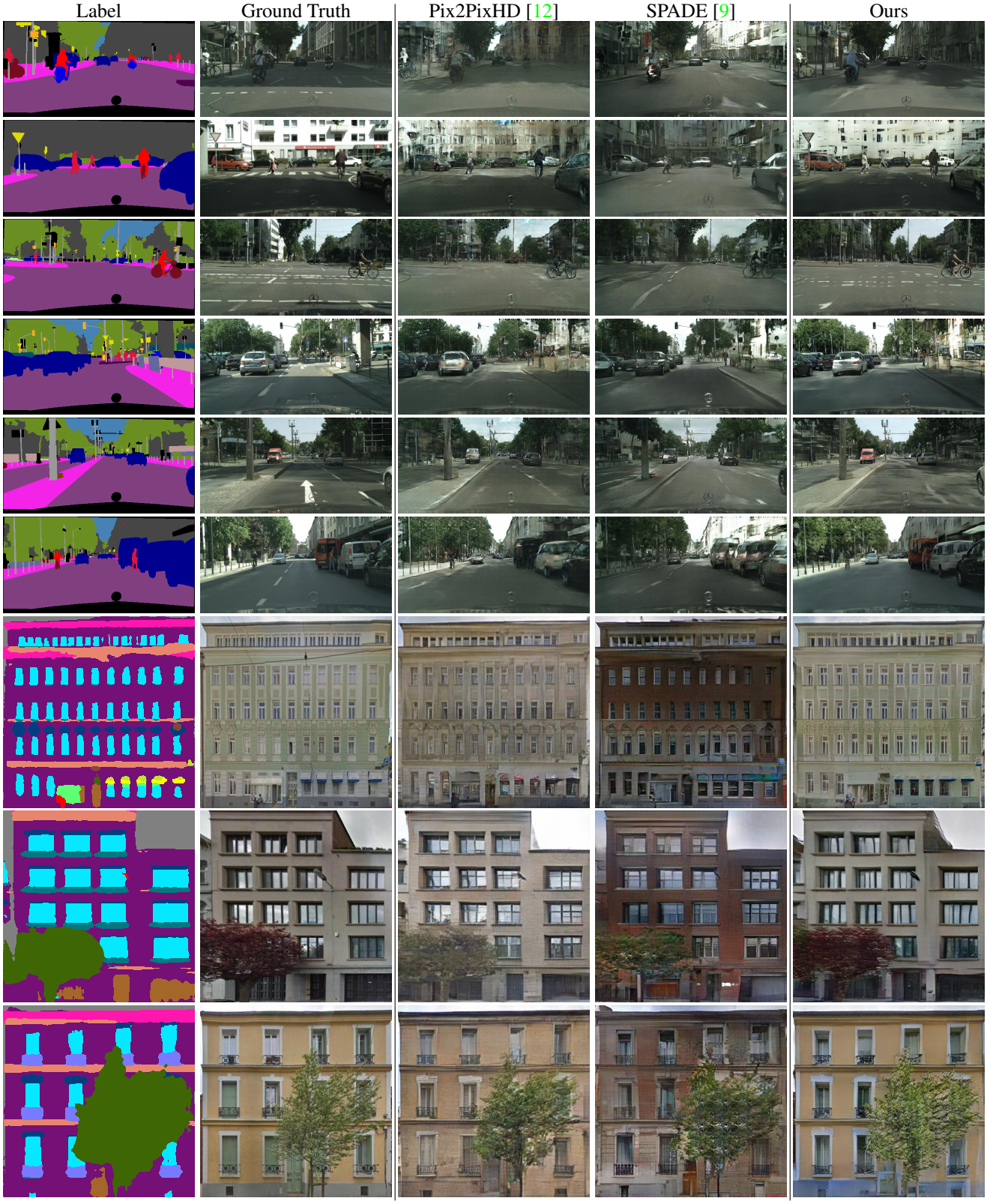

Figure 15: Visual comparison of semantic image synthesis results on the CityScapes and Façades dataset. We compare Pix2PixHD, SPADE, and our method.