Few-shot Semantic Image Synthesis Using StyleGAN Prior

Yuki Endo
University of Tsukuba
endo@cs.tsukuba.ac.jp

Yoshihiro Kanamori
Univeristy of Tsukuba
kanamori@cs.tsukuba.ac.jp

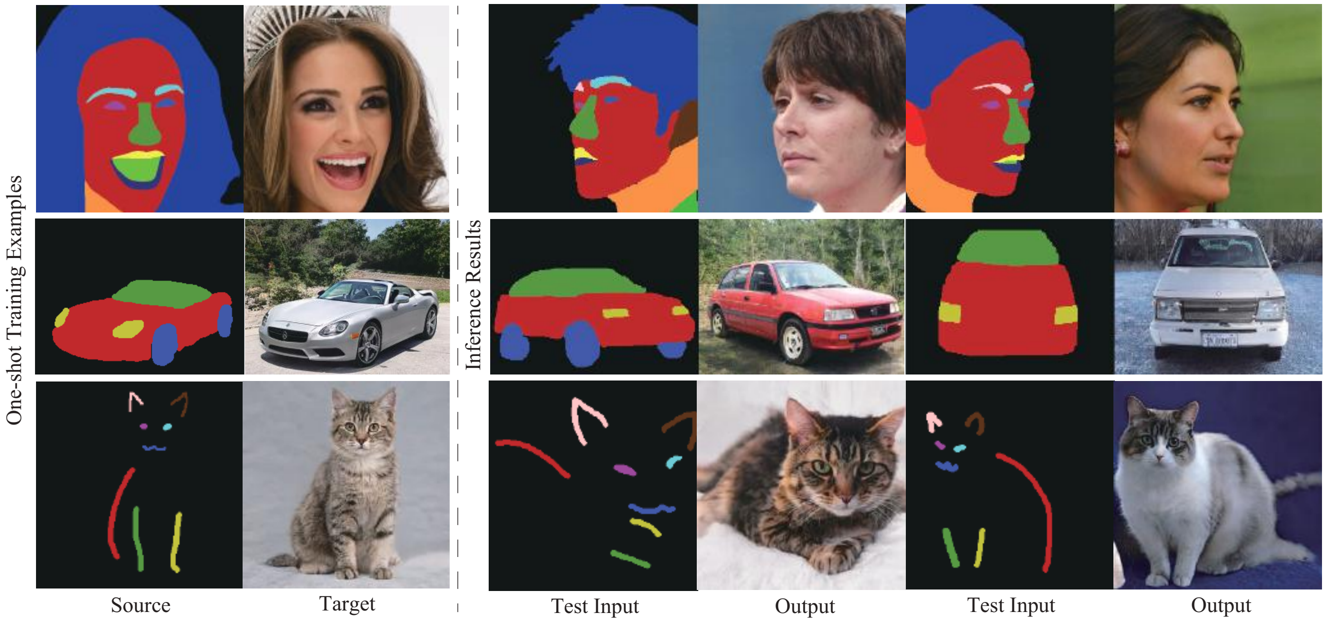

Figure 1: Our method can synthesize photorealistic images from dense or sparse semantic annotations using a single training pair and a pre-trained StyleGAN.

Abstract

This paper tackles a challenging problem of generating photorealistic images from semantic layouts in few-shot scenarios where annotated training pairs are hardly available but pixel-wise annotation is quite costly. We present a training strategy that performs pseudo labeling of semantic masks using the StyleGAN prior. Our key idea is to construct a simple mapping between the StyleGAN feature and each semantic class from a few examples of semantic masks. With such mappings, we can generate an unlimited number of pseudo semantic masks from random noise to train an encoder for controlling a pre-trained StyleGAN generator. Although the pseudo semantic masks might be too coarse for previous approaches that require pixel-aligned masks, our framework can synthesize high-quality images from not only dense semantic masks but also sparse inputs such as landmarks and scribbles. Qualitative and quantitative results with various datasets demonstrate improvement over previous approaches with respect to layout fidelity and visual quality in as few as one- or five-shot settings.

1. Introduction

Semantic image synthesis is a powerful technique for generating images with intuitive control using spatial semantic layouts. A drawback is that most existing techniques require substantial training data in source and target domains for high-quality outputs. Even worse, annotations of pixel-wise labels (e.g., semantic masks) are quite costly.

In this paper, we present the first method for few-shot semantic image synthesis, assuming that we can utilize many unlabeled data with only a few labeled data of the target domain. Imagine that you have a large dataset of car or cat photos, but only a single annotated pair is available (Figure 1, the 2nd and 3rd rows). In this scenario, we utilize the state-of-the-art generative adversarial network (GAN), *StyleGAN* [16, 17], pre-trained using the unlabeled dataset. Namely, we achieve high-quality image synthesis by exploring StyleGAN's latent space via *GAN inversion*. What is challenging here is that, although common GAN inversion techniques [1, 2] assume that test inputs belong to the same domain as GAN's training data (*e.g*., facial photographs), our test and training data are in different domains, *i.e*., semantic layouts and photographs. How to invert the input in a different domain into GAN's latent space is an open question, as introduced in the latest survey [37].

To bridge the domain gaps for the first time, we construct a mapping between the semantics predefined in the few-shot examples and StyleGAN's latent space. Inspired by the fact that pixels with the same semantics tend to have similar StyleGAN features [8], we generate pseudo semantic masks from random noise in StyleGAN's latent space via simple nearest-neighbor matching. This way, we can draw an unlimited number of training pairs by only feeding random noise to the pre-trained StyleGAN generator. After integrating an encoder on top of the fixed StyleGAN generator, we then train the encoder for controlling the generator using the pseudo-labeled data in a supervised fashion. Although our pseudo semantic masks might be too noisy or coarse for the previous pixel-aligned approach [23], our method works well with such masks thanks to the tolerance to misalignment. Our approach integrates semantic layout control into pre-trained StyleGAN models publicly available on the Web [24], via pseudo labeling even from a single annotated pair with not only a dense mask but also sparse scribbles or landmarks.

In summary, our major contributions are three-fold:

- We explore a novel problem of few-shot semantic image synthesis, where the users can synthesize high-quality, various images in the target domains even from very few and rough semantic layouts provided during training.
- We propose a simple yet effective method for training a StyleGAN encoder for semantic image synthesis in few-shot scenarios, via pseudo sampling and labeleing based on the StyleGAN prior, without hyper parameter tuning for complicated loss functions.
- We demonstrate that our method significantly outperforms the existing methods w.r.t. layout fidelity and visual quality via extensive experiments on various datasets.

2. Related Work

Image-to-Image translation There are various image-to-image (I2I) translation methods suitable for semantic image synthesis; the goals are, *e.g*., to improve image quality [6, 22, 23, 32, 31], generate multi-modal outputs [23, 18, 44, 10], and simplify input annotations using bounding boxes [41, 30, 19]. However, all of these methods require large amounts of training data of both source and target domains and thus are unsuitable for our few-shot scenarios.

Table 1: Feeding types and required amounts of data for existing semantic image synthesis (SMIS), "*few-shot*" image-to-image translation (I2I), and ours.

Method	Feeding	Training		Test
		source	target	target
SMIS [14, 23, 22]	paired	large	large	none
"Few-shot" I2I [21, 36]	unpaired	large	large	small
Benaim and Wolf [4]	unpaired	small	large	none
Few-shot SMIS (ours)	paired	small	large	none

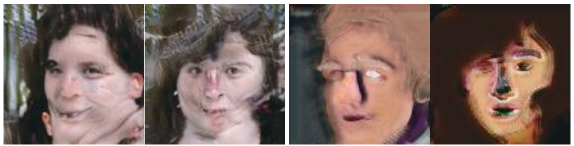

Figure 2: Results of one-shot semantic image synthesis with general few-shot I2I translation (SEMIT [36] and Benaim and Wolf [4]) using the same training data as ours. The input semantic masks are the same as those used in Figure 7.

FUNIT [21] and SEMIT [36] are recently proposed methods for "*few-shot*" I2I translation among different classes of photographs (*e.g*., dog, bird, and flower). However, their meaning of "few-shot" is quite different from ours; they mean that only a few target class data are available in test time, but assume sufficient data of both source and target classes in training time (with a difference in whether the image class labels are fully available [21] or not [36]). Contrarily, we assume only a few source data, *i.e*., ground-truth (GT) semantic masks, in training time. These "few-shot" I2I translation methods do not work at all in our settings, as shown in Figure 2.

Benaim and Wolf [4] presented a one-shot unsupervised I2I translation framework for the same situation as ours. However, their "*unpaired*" approach suffers from handling semantic masks, which have less distinctive features than photographs (see Figure 2). Moreover, their trained model has low generalizability, specialized for the single source image provided during training. In other words, their method needs to train a model for each test input, while our method does not. Table 1 summerizes the differences of problem settings between each method.

Latent space manipulation Recent GAN inversion (e.g., Image2StyleGAN [1, 2]) can control GAN outputs by inverting given images into GAN's latent space. There have been also many attempts to manipulate inverted codes in disentangled latent spaces [7, 15, 27, 28, 12]. However, inverting semantic masks into a latent space defined by photographs is not straightforward because how to measure the discrepancy between the two different domains (*i.e.*, semantic masks and photographs) is an open question [37]. Note that we cannot use pre-trained segmentation networks in our few-shot scenrarios. Our method is the first attempt of GAN inversion for semantic masks into StyleGAN's latent space defined by photographs.

Few-shot semantic image synthesis To the best of our knowledge, there is no other few-shot method dedicated to semantic image synthesis. An alternative approach might be to use few-shot semantic segmentation [9, 35, 20, 33, 34, 39] to annotate unlabeled images to train image-to-image translation models. In recent few-shot semantic segmentation methods based on a meta-learning approach, however, training episodes require large numbers of labeled images of various classes other than target classes to obtain common knowledge. Therefore, this approach is not applicable to our problem setting.

3. Few-shot Semantic Image Synthesis

3.1. Problem setting

Our goal is to accomplish semantic image synthesis via semi-supervised learning with N_u unlabeled images and N_l labeled pairs both in the same target domain, where $N_u \gg N_l$. In particular, we assume few-shot scenarios, setting $N_l = 1$ or 5 in our results. A labeled pair consists of a one-hot semantic mask $\mathbf{x} \in \{0,1\}^{C\times W\times H}$ (where C, W, and H are the number of classes, width, and height) and its GT RGB image $\mathbf{y} \in \mathbb{R}^{3\times W\times H}$. A semantic mask can be a dense map pixel-aligned to $\mathbf{y}$ or a sparse map (e.g., scribbles or landmarks). In a sparse map, each scribble or landmark has a unique class label, whereas unoccupied pixels have an "*unknown*" class label. Hereafter we denote the labeled dataset as $\mathcal{D}_l = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N_l}$ and the unlabeled dataset as $\mathcal{D}_u = \{\mathbf{y}_i\}_{i=1}^{N_u}$.

3.2. Overview

The core of our method is to find appropriate mappings between semantics defined by a few labeled pairs $\mathcal{D}_l$ and StyleGAN's latent space defined by an unlabeled dataset $\mathcal{D}_u$. Specifically, we first extract a feature vector representing each semantic class, which we refer to as a *representative vector*, and then find matchings with StyleGAN's feature map via (k-)nearest-neighbor search. Such matchings enable *pseudo labeling*, *i.e.*, to obtain pseudo semantic masks from random noise in StyleGAN's latent space, which are then used to train an encoder for controlling the pre-trained StyleGAN generator. A similar approach is the prototyping used in recent few-shot semantic segmentation [9, 35, 39]. Our advantage is that our method suffices with unsupervised training of StyleGAN models, whereas the prototyping requires supervised training of feature extractors (e.g., VGG [29]).

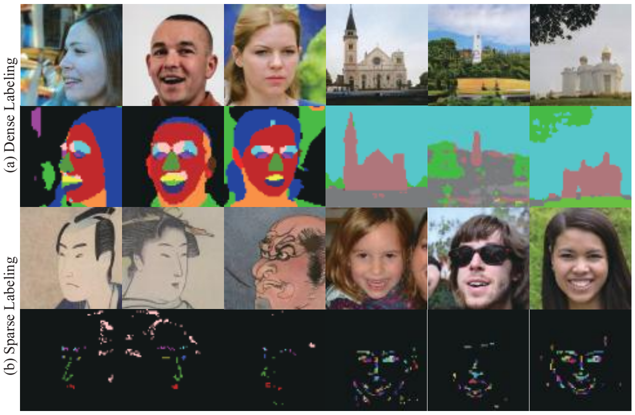

Figure 3: Our densely (a) and sparsely (b) pseudo-labeled examples.

Our pseudo semantic masks are often noisy, distorted (see Figure 3), and thus inadequate for conventional approaches of semantic image synthesis or image-to-image translation, which require pixel-wise correspondence. However, even from such low-quality pseudo semantic masks, we can synthesize high-quality images with spatial layout control by utilizing the pre-trained StyleGAN generator. This is because the StyleGAN generator only requires latent codes that encode spatially global information.

As an encoder for generating such latent codes, we adopt the *Pixel2Style2Pixel* (pSp) encoder [26]. The inference process is the same as that of pSp; from a semantic mask, the encoder generates latent codes that are then fed to the fixed StyleGAN generator to control the spatial layout. We can optionally change or fix latent codes that control local details of the output images. Please refer to Figure 3 in the pSp paper [26] for more details.

Hereafter we explain the pseudo labeling process and the training procedure with the pseudo semantic masks.

3.3. Pseudo labeling

We elaborate on how to calculate the representative vectors and pseudo labeling, for which we propose different approaches to dense and sparse semantic masks.

3.3.1 Dense pseudo labeling

Figure 4 illustrates the pseudo labeling process for dense semantic masks. We first extract StyleGAN's feature maps corresponding to the semantic masks in $\mathcal{D}_l$. If pairs of se-

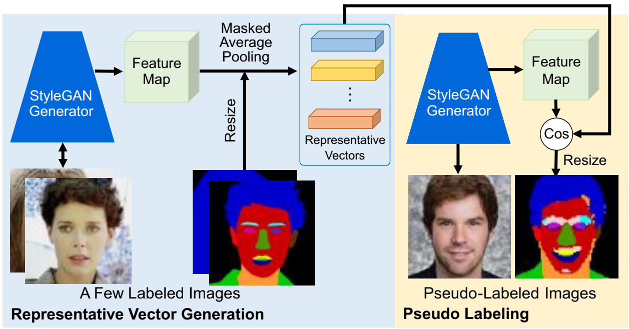

Figure 4: Dense pseudo labeling. Left: We compute a representative vector of each class via masked average pooling over the feature maps of few-shot examples. Right: We assign pseudo labels to sampled images via nearest-neighbor matching based on cosine similarity between the representative vectors and the feature maps of the sampled images.

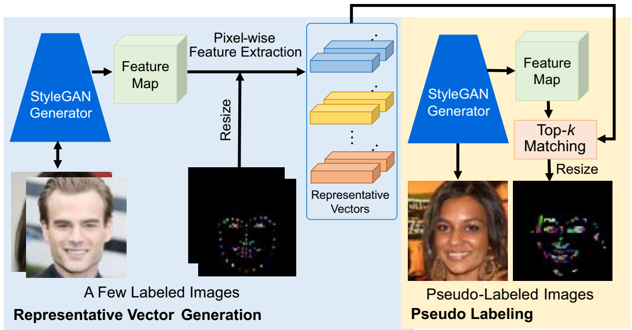

Figure 5: Sparse pseudo labeling. Left: Unlike the dense version, we extract representative vectors for all labeled pixels. Right: For each representative vector, we take top-k correspondences and assign its class label to the corresponding pixels whose similarites are above a threshold t.

mantic masks $\mathbf{x}$ and GT RGB images $\mathbf{y}$ are available in $\mathcal{D}_l$, we first invert $\mathbf{y}$ into the StyleGAN's latent space via optimization and then extract the feature map via forward propagation. Otherwise, we feed one or a few noise vectors to the pre-trained StyleGAN generator, extract the feature maps and synthesized images, and manually annotate the synthesized images to create semantic masks. Next, we extract a representative vector $\mathbf{v}_c$ for each semantic class c from the pairs of extracted feature maps and semantic masks, following the approach by Wang *et al.* [35] for prototyping. Specifically, we apply the masked average pooling to the feature map $\mathbf{F}_i \in \mathbb{R}^{Z\times W'\times H'}$ (where Z, W', and H' are the number of channels, width, and height) using a resized semantic mask $\mathbf{x}'_i \in \mathbb{R}^{C\times W'\times H'}$, and then average over each pair i in $\mathcal{D}_l$:

$$\mathbf{v}_c = \frac{1}{N_l}\sum_{i=1}^{N_l} \frac{\sum_{x,y}\mathbf{F}_i^{x,y}\,\mathbb{1}\left[\mathbf{x}_i^{(c,x,y)}=1\right]}{\sum_{x,y}\mathbb{1}\left[\mathbf{x}_i^{(c,x,y)}=1\right]}, \tag{1}$$

where (x, y) denote pixel positions, and $\mathbb{1}\left[\cdot\right]$ is the indicator function that returns 1 if the argument is true and 0 otherwise.

After obtaining representative vectors, we generate pseudo semantic masks for training our encoder. Every time we feed random noise to the pre-trained StyleGAN generator, we extract a feature map $\mathbf{F}'$ and then calculate a semantic mask via nearest-neighbor matching between the representative vectors and the pixel-wise vectors in $\mathbf{F}'$. In all of our results, feature maps $\mathbf{F}'$ are at resolution of 64×64 and extracted from the layer closest to the output layer of the StyleGAN generator. Class label $c^{(x,y)}$ for pixel (x, y) is calculated as follows:

$$c^{(x,y)} = \mathrm{argmax}_{c\in C}\cos(\mathbf{v}_c, \mathbf{F}'^{(x,y)}). \tag{2}$$

As a distance metric, we adopt the cosine similarity $\cos(\cdot,\cdot)$, inspired by the finding [8] that StyleGAN's feature vectors having the same semantics form clusters on a unit sphere. Finally, we enlarge the semantic masks to the size of the synthesized images. Figure 3(a) shows the examples of pseudo labels for dense semantic masks.

3.3.2 Sparse pseudo labeling

Figure 5 illustrates the pseudo labeling process for sparse semantic masks. As explained in Subsection 3.1, sparse semantic masks have a class label for each annotation (e.g., a scribble and landmark) and an "unknown" label. Here we adopt a pseudo-labeling approach different from the dense version due to the following reason. We want to retain the spatial sparsity in pseudo semantic masks so that the pseudo semantic masks resemble genuine ones as much as possible. However, if we calculate nearest-neighbors for a representative vector of each annotation as done in the dense version, the resultant pseudo masks might form dense clusters of semantic labels. Alternatively, as a simple heuristics, we consider each pixel in each annotation has its representative vector and calculate a one-to-one matching between each annotated pixel and each pixel-wise vector. In this case, however, many annotated pixels might match an identical pixel-wise vector (*i.e.*, many-to-one mapping), which results in fewer samples in pseudo semantic masks. Therefore, we calculate top-k (i.e., k-nearest-neighbors) instead of one-nearest-neighbor to increase matchings. In the case of many-to-one mappings, we assign the class label of an annotation that has the largest cosine similarity. To avoid outliers, we discard the matchings if their cosine similarities are lower than a threshold t and assign the "unknown" label. Figure 3(b) shows the examples of pseudo labels for sparse semantic masks.

We set $k = 3$ and $t = 0.5$ in all of our results in this

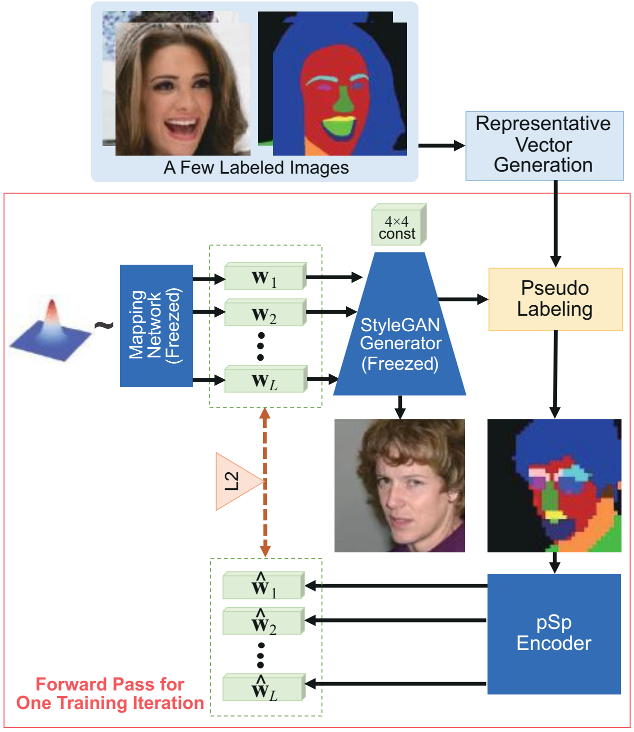

Figure 6: Training iteration of the encoder. We first generate images from noise vectors via the mapping network and the StyleGAN generator. We then compute pseudo semantic masks using the *representative vectors* (Figures 4 and 5). We optimize the encoder parameters based on L2 loss between latent codes.

paper. The supplementary material contains pseudo-labeled results with different parameters.

3.4. Training procedure

Figure 6 illustrates the learning process of our encoder. First, we explain the forward pass in the training phase. We feed a random noise $\mathbf{z}$ sampled from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to the encoder and obtain latent codes $\{\mathbf{w}_i\}_{i=1}^{L}$ (where L is the number of layers to input/output latent codes) via the pre-trained StyleGAN's mapping network f. We feed the latent codes to the pre-trained StyleGAN generator to synthesize an image while extracting the intermediate layer's feature map. From this feature map and representative vectors, we create a pseudo semantic mask, which is then fed to our encoder to extract latent codes $\{\hat{\mathbf{w}}_i\}_{i=1}^{L}$.

In the backward pass, we optimize the encoder using the following loss function:

$$\mathcal{L} = \mathbb{E}_{\mathbf{w} \sim f(\mathbf{z})} \|\hat{\mathbf{w}} - \mathbf{w}\|_2^2. \tag{3}$$

This loss function indicates that our training is quite simple because backpropagation does not go through the pre-trained StyleGAN generator. Algorithm 1 summarizes the whole process of training. In the supplementary material, we also show the intermediate pseudo semantic masks and reconstructed images obtained during the training iterations.

Algorithm 1 Few-shot learning of StyleGAN encoder

```
Input: A labeled set $\mathcal{D}_l$ and unlabeled set $\mathcal{D}_u$
  Train StyleGAN using $\mathcal{D}_u$
  Compute representative vectors using $\mathcal{D}_l$
  for each training iteration do
    Sample latent codes according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$
    Feed the latent codes to the generator
    Pseudo labeling using representative vectors
    Feed the pseudo semantic masks to the encoder
    Compute the loss $\mathcal{L}$ as in Eq. (3)
    Compute the gradient and optimize the encoder
  end for
```

4. Experiments

We conducted experiments to evaluate our method. The supplementary material contains implementation details.

4.1. Datasets

We used public StyleGAN2 [17] models pre-trained with FFHQ (human faces) [16, 17], LSUN (car, cat, and church) [40, 17], ukiyo-e [25], and anime face images [11]. To evaluate our method quantitatively, we used the preprocessed CelebAMask-HQ datasets [43], which contains face images and corresponding semantic masks (namely, 2,000 for test and 28,000 for training). In addition, we extracted face landmarks as sparse annotations using OpenPose [5]. The numbers of "*ground-truth*" face landmarks are reduced to 1,993 for test and 27,927 for training because OpenPose sometimes failed. We used also LSUN church [40] (300 in a validation set and 1,000 in a training set) for the quantitative evaluation. Because this dataset does not contain semantic masks, we prepared them using the scene parsing model [42] consisting of the ResNet101 encoder [13] and the UPerNet101 decoder [38]. For our experiments of N_l-shot learning, we selected N_l paired images from the training sets, while the full-shot version uses all of them.

4.2. Qualitative results

Figure 7 compares the results generated from semantic masks of the CelebAMask-HQ dataset in a one-shot setting. Figure 8 also shows the results generated from sparse landmarks in a five-shot setting. The pixel-aligned approach, SPADE [23] and pix2pixHD++ [23], generates images faithfully to the given layouts, but the visual quality is very low. Meanwhile, pSp [26], which uses pre-trained

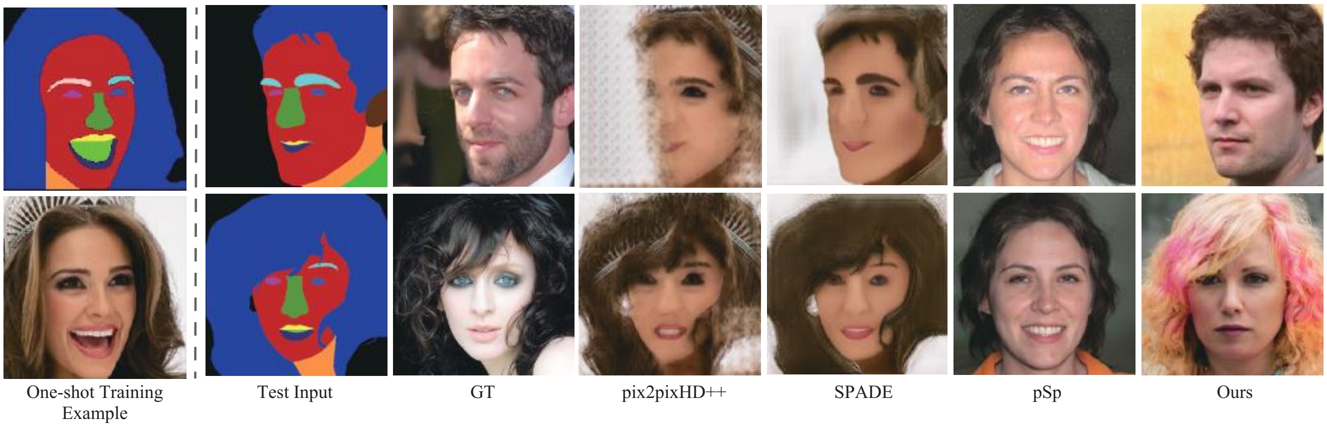

Figure 7: Comparison of face images generated from dense semantic masks in a one-shot setting.

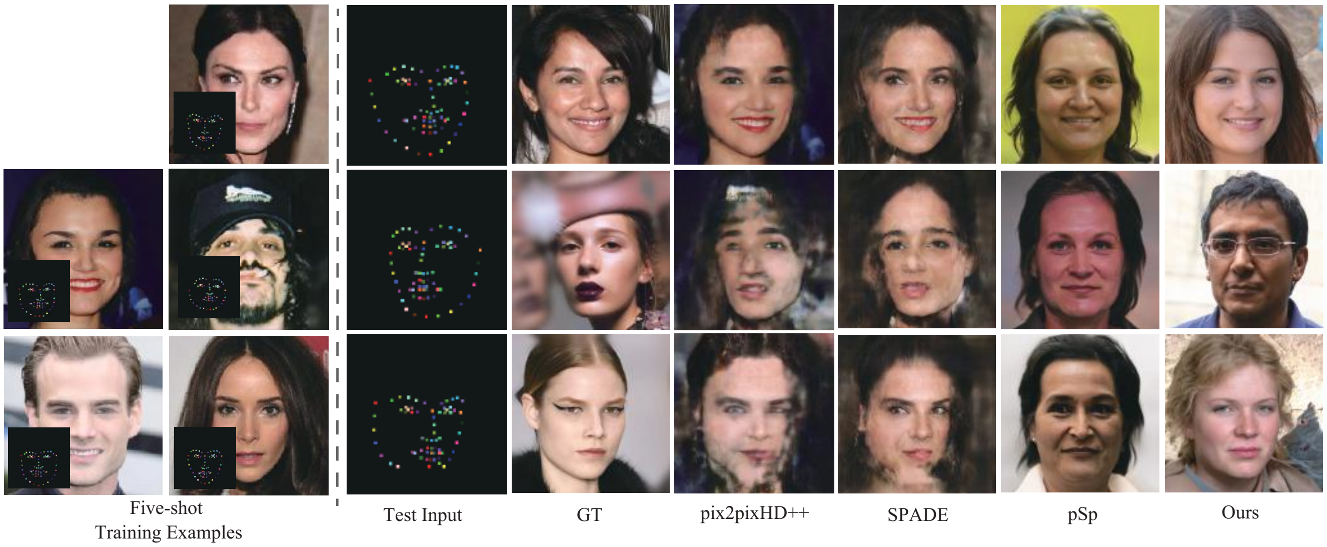

Figure 8: Comparison of face images generated from sparse landmarks in a five-shot setting.

StyleGANs, is able to generate realistic images. Although pSp can optionally generate multi-modal outputs, it ignores the input layouts due to over-fitting to the too few training examples. In contrast, our method produces photorealistic images corresponding to the given layouts. We can see the same tendency in comparison with the LSUN church dataset in Figure 9.

The benefit of our few-shot learning approach is not to need many labeled data. We therefore validate the applicability of our method to various domains where annotations are hardly available in public. Figures 1 and 10 show car, cat, and ukiyo-e images generated from semantic masks and scribbles. Again, pSp does not reflect the input layouts on the results, whereas our method controls output semantics accordingly (e.g., the cats' postures and the ukiyo-e hairstyles). Interestingly, our method works well with cross lines as inputs, which specify the orientations of anime faces (Figure 11).

Finally, we conducted a comparison with the pixel-aligned approach using our pseudo labeling technique. Figure 12 shows the results of SPADE, pix2pixHD++, and ours, which were trained up to 100,000 iterations with the appropriate loss functions. Because our pseudo semantic masks are often misaligned, the pixel-aligned approach failed to learn photorealistic image synthesis, whereas ours succeeded. Please refer to the supplementary material for more qualitative results.

4.3. Quantitative results

We quantitatively evaluated competitive methods and ours with respect to layout fidelity and visual quality. For each dataset, we first generate images from test data (i.e., semantic masks/landmarks in CelebA-HQ and semantic masks in LSUN church) using each method and then ex-

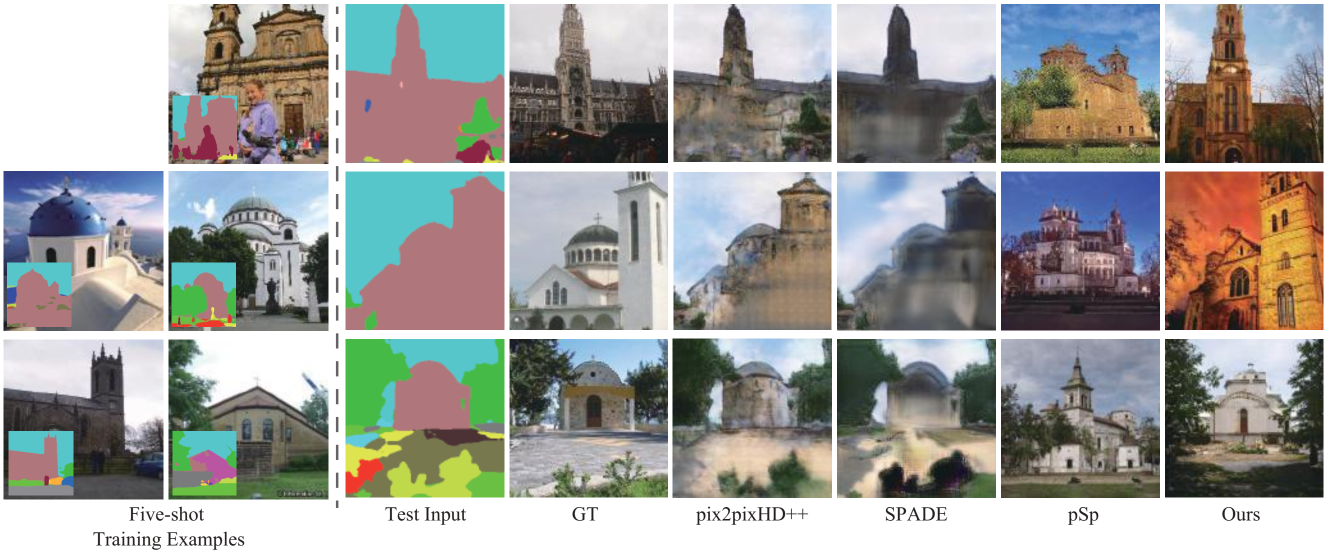

Figure 9: Comparison of church images generated from dense semantic masks in a five-shot setting.

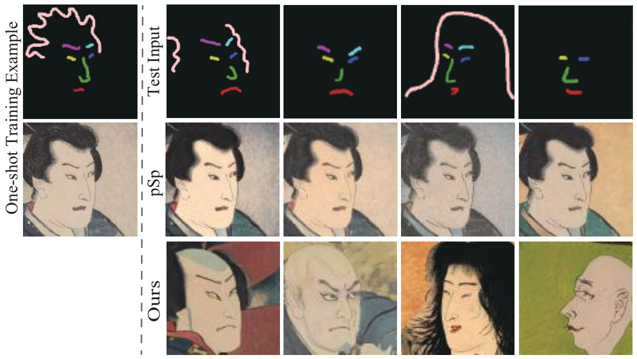

Figure 10: Comparison of ukiyo-e images generated from sparse scribbles in a five-shot setting.

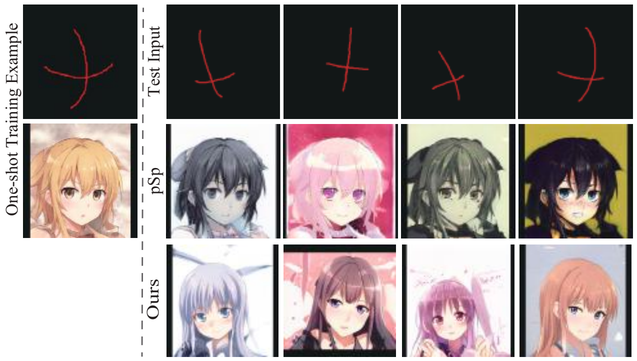

Figure 11: Comparison of anime face images generated from cross lines in a five-shot setting.

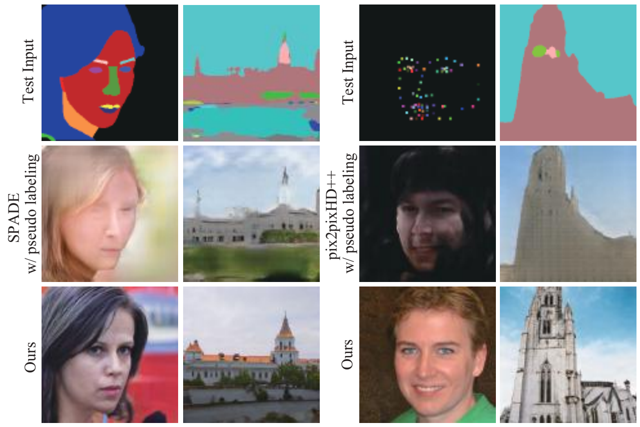

Figure 12: Comparison with the pixel-aligned approach (SPADE [23] and pix2pixHD++ [23]) trained with our pseudo-labeled data.

tract the corresponding semantic masks/landmarks for evaluation, as done in Subsection 4.1. As evaluation metrics for parsing, we used Intersection over Union (IoU) and accuracy. As for IoU, we used mean IoU (mIoU) for CelebA-HQ. For LSUN church, we used frequency weighted IoU (fwIoU) because our "ground-truth" (GT) semantic masks synthesized by [42] often contain small noisy-labeled regions, which strongly affect mIoU. As a landmark metric, we computed RMSE of Euclidean distances between landmarks of generated and GT images. If landmarks cannot be detected in generated images, we counted them as *N/A*. We used Fréchet Inception Distance (FID) as a metric for visual quality.

Table 2 shows the quantitative comparison in few-shot settings, except for the bottom row, where all labeled images in the training datasets were used. In the five-shot

Table 2: Quantitative comparison on each dataset. N_l is the number of labeled training data, * means training the model using our pseudo-labeled images sampled from the pre-trained StyleGANs, and *Full* means using all labeled data in each traininig set. *N/A* indicates the number of images in which landmarks cannot be detected.

		CelebAMask-HQ			CelebALandmark-HQ			LSUN ChurchMask		
Method	N_l	mIoU↑	accu↑	FID↓	RMSE↓	N/A↓	FID↓	fwIoU↑	accu↑	FID↓
pix2pixHD++ [23]	5	62.2	92.7	82.7	43.5	31	109.7	71.8	78.3	133.2
pix2pixHD++*	5	38.7	87.6	98.0	28.9	40	125.6	68.0	76.7	100.6
SPADE [23]	5	64.8	92.4	79.3	23.8	35	121.4	63.6	70.6	193.2
SPADE*	5	40.1	88.8	88.5	33.7	11	138.8	70.3	78.9	101.7
pSp [26]	1	24.8	61.7	93.4	37.0	0	87.9	29.2	42.0	96.5
pSp [26]	5	28.4	67.6	96.9	37.1	0	87.5	34.4	47.9	89.5
Ours	1	38.3	81.7	59.1	31.5	0	82.1	41.8	54.0	65.1
Ours	5	41.5	82.5	53.9	31.4	0	77.4	48.6	62.1	56.9
pSp [26]	Full	49.2	87.9	67.4	26.6	0	106.6	50.4	63.8	76.2

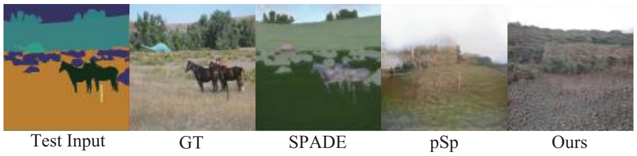

Figure 13: Failure case with semantic classes that do not appear in few-shot training examples ("*animal*" in this case).

setting, the pixel-aligned approach (i.e., pix2pixHD++ [23] and SPADE [23]) records consistently high IoU, accuracy, and FID scores. These scores indicate that the output images are aligned to the semantic masks relatively better but the image quality is lower, as we can see from the qualitative results. The larger numbers of undetected faces (denoted as "N/A") also indicate low visual quality. We confirmed that our pseudo labeling technique does not yield consistent improvements for the pixel-aligned approaches (indicated with "*"). In contrast, ours yields lower FID scores than the pixel-aligned approach and pSp [26] (even in the full-shot setting) consistently and is overall improved by increasing N_l from 1 to 5. Ours also outperforms pSp in the few-shot settings w.r.t. all the metrics except for N/A. The qualitative full-shot results are also included in the supplementary material.

5. Discussion

Here we summarize the pros and cons of the related methods and ours. The pixel-aligned approach [23] preserves spatial layouts specified by the semantic masks but fails to learn from our noisy pseudo labels due to the sensitivity to misaligned semantic masks. Contrarily, ours on top of pSp [26] is tolerant of misalignment and thus works well with our pseudo labels. However, it is still challenging to reproduce detailed input layouts and to handle layouts that StyleGAN cannot generate. A future direction is to overcome these limitations by, e.g., directly manipulating hidden units corresponding to semantics of input layouts [3]. Another limitation is that we cannot handle semantic classes unseen in the few-shot examples. Figure 13 shows such an example with a more challenging dataset, ADE20K [42]. Please refer to the supplementary material for more results.

It is also worth mentioning that our method outperformed the full-shot version of pSp in FID. This is presumably because our pseudo sampling could better explore StyleGAN's latent space defined by a large unlabeled dataset (e.g., 70K images in FFHQ and 48M images in LSUN church) than pSp, which uses limited labeled datasets for training the encoder.

6. Conclusion

In this paper, we have proposed a simple yet effective method for few-shot semantic image synthesis for the first time. To compensate for the lack of pixel-wise annotation data, we generate pseudo semantic masks via (k-)nearest-neighbor mapping between the feature vector of the pre-trained StyleGAN generator and each semantic class in the few-shot labeled data. In each training iteration, we can generate a pseudo label from random noise to train an encoder [26] for controlling the pre-trained StyleGAN generator using a simple L2 loss. The experiments with various datasets demonstrated that our method can synthesize higher-quality images with spatial control than competitive methods and works well even with sparse semantic masks such as scribbles and landmarks.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV2019*, pages 4431–4440. IEEE, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR2020*, pages 8293–8302. IEEE, 2020. 2

[3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN dissection: Visualizing and understanding generative adversarial networks. In *7th International Conference on Learning Representations*, 2019. 8

[4] Sagie Benaim and Lior Wolf. One-shot unsupervised cross domain translation. In *NeurIPS 2018*, pages 2108–2118, 2018. 2

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 5

[6] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice*, pages 1520–1529, 2017. 2

[7] Chia-Hsing Chiu, Yuki Koyama, Yu-Chi Lai, Takeo Igarashi, and Yonghao Yue. Human-in-the-loop differential subspace search in high-dimensional latent space. *ACM Trans. Graph.*, 39(4):85, 2020. 3

[8] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of gans. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5770–5779, 2020. 2, 4

[9] Nanqing Dong and Eric P. Xing. Few-shot semantic segmentation with prototype learning. In *British Machine Vision Conference 2018, BMVC 2018*, page 79, 2018. 3

[10] Yuki Endo and Yoshihiro Kanamori. Diversifying semantic image synthesis and editing via class- and layer-wise vaes. *Comput. Graph. Forum*, 39(7):519–530, 2020. 2

[11] Aaron Gokaslan. Making Anime Faces With StyleGAN, 2020. https://www.gwern.net/Faces. 5

[12] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable GAN controls. *CoRR*, abs/2004.02546, 2020. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 770–778, 2016. 5

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, 2017. 2

[15] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2020. 3

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2, 5

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8107–8116. IEEE, 2020. 2, 5, 11

[18] Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional IMLE. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 4219–4228, 2019. 2

[19] Yandong Li, Yu Cheng, Zhe Gan, Licheng Yu, Liqiang Wang, and Jingjing Liu. Bachgan: High-resolution image synthesis from salient object layout. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2020. 2

[20] Lizhao Liu, Junyi Cao, Minqian Liu, Yong Guo, Qi Chen, and Mingkui Tan. Dynamic extension nets for few-shot semantic segmentation. In *MM '20: The 28th ACM International Conference on Multimedia*, pages 1441–1449, 2020. 3

[21] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 10550–10559, 2019. 2

[22] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and Hongsheng Li. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 568–578, 2019. 2

[23] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2, 5, 7, 8, 20

[24] Justin Pinkney. Awesome Pretrained StyleGAN2, 2020. https://github.com/justinpinkney/awesome-pretrained-stylegan2. 2

[25] Justin Pinkney. Ukiyo-e Yourself with StyleGAN 2, 2020. https://www.justinpinkney.com/ukiyoe-yourself. 5

[26] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *CoRR*, abs/2008.00951, 2020. 3, 5, 8, 11, 15, 20

[27] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9240–9249, 2020. 3

[28] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *CoRR*, abs/2007.06600, 2020. 3

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015. 3

[30] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 10530–10539, 2019. 2

[31] Hao Tang, Song Bai, and Nicu Sebe. Dual attention gans for semantic image synthesis. In *MM '20: The 28th ACM International Conference on Multimedi0*, pages 1994–2002, 2020. 2

[32] Hao Tang, Dan Xu, Yan Yan, Philip H. S. Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7867–7876, 2020. 2

[33] Pinzhuo Tian, Zhangkai Wu, Lei Qi, Lei Wang, Yinghuan Shi, and Yang Gao. Differentiable meta-learning model for few-shot semantic segmentation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 12087–12094, 2020. 3

[34] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *Computer Vision - ECCV 2020 - 16th European Conference*. 3

[35] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 9196–9205, 2019. 3, 4

[36] Yaxing Wang, Salman Khan, Abel Gonzalez-Garcia, Joost van de Weijer, and Fahad Shahbaz Khan. Semi-supervised learning for few-shot image-to-image translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4452–4461, 2020. 2

[37] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN Inversion: A Survey, 2021. 2, 3

[38] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Computer Vision - ECCV 2018 - 15th European Conference*, pages 432–448, 2018. 5

[39] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *Computer Vision - ECCV 2020 - 16th European Conference*. 3

[40] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. 5

[41] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8584–8593, 2019. 2

[42] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 5122–5130, 2017. 5, 7, 8, 20

[43] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. SEAN: image synthesis with semantic region-adaptive normalization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pages 5103–5112, 2020. 5

[44] Zhen Zhu, Zhiliang Xu, Ansheng You, and Xiang Bai. Semantically multi-modal image synthesis. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5466–5475, 2020. 2

A. Implementation Details

We implemented our method with PyTorch and ran our code on PCs equipped with GeForce GTX 1080 Ti. We used StyleGAN2 [17] as a generator and pSp [26] as an encoder. We trained the encoder using the Ranger optimizer [26] with a learning rate of 0.0001. The batch size (i.e., the number of pseudo-labeled images per iteration) was set to 2. We performed 100,000 iterations and took a day at most. Regarding our multi-modal results, please refer to Section D in this supplementary material.

B. Sparse Pseudo Labeling with Different Parameters

Figure 14 shows the sparsely pseudo-labeled results (right) for the StyleGAN sample (lower left) using different parameters k and t with a one-shot training pair (upper left). As explained in Subsubsection 3.3.2 in our paper, k is used for top-k matching between per-pixel feature vectors and representative vectors, whereas t is a threshold of cosine similarity. For all of our other results, we set $k = 3$ to reduce the number of misfetches of matched pixels and $t = 0.5$ to reduce outliers.

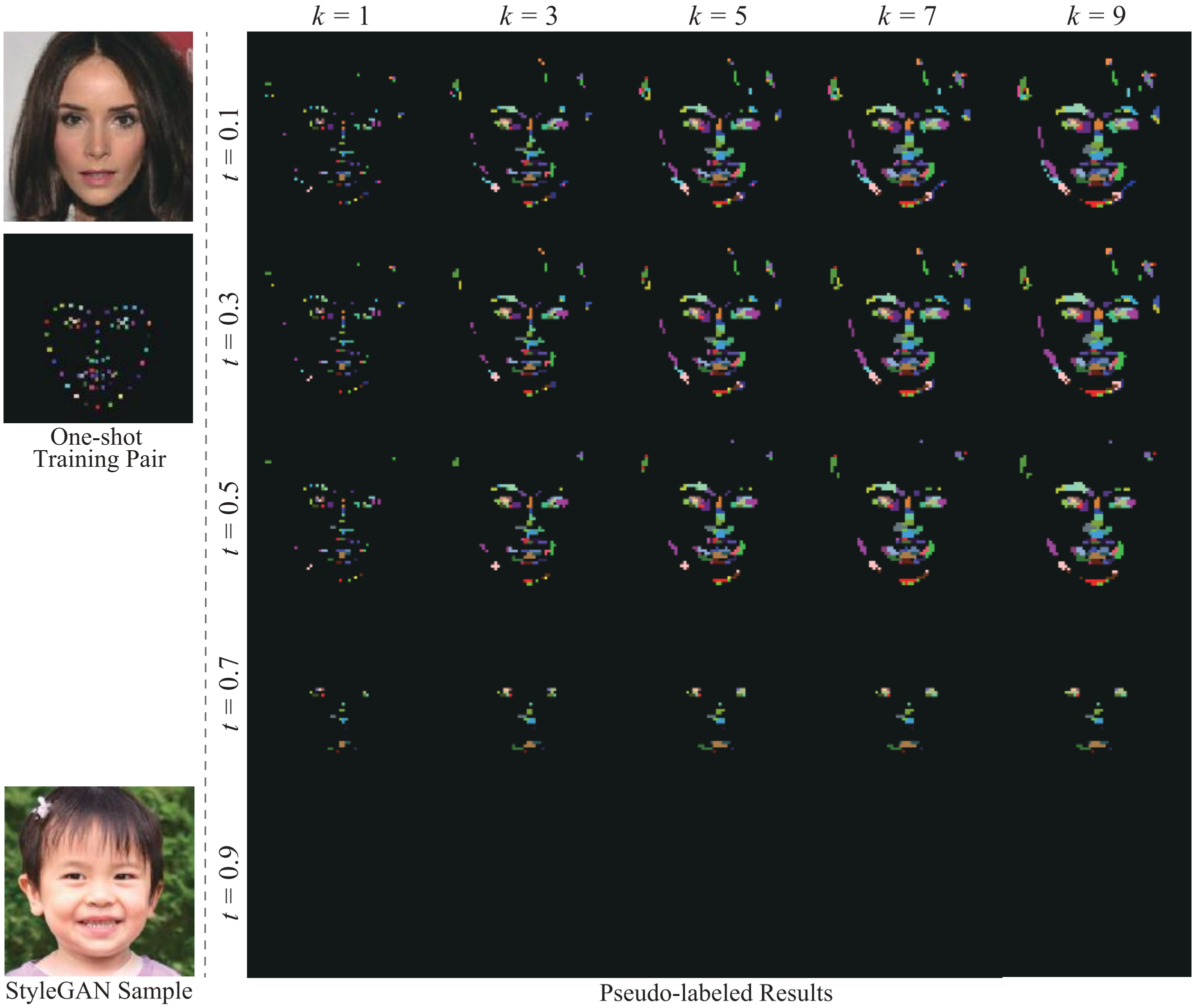

Figure 14: Sparsely pseudo-labeled results with different parameters k and l.

C. Images Reconstructed from Pseudo Semantic Masks During Training Procedure

Figures 15, 16, 17, 18, 19, and 20 show the intermediate outputs in one-shot settings during training iterations, which is explained in Subsection 3.4 of our paper. For each set of results, we fed random noise vectors to the pre-trained StyleGAN generator to obtain synthetic images (top row) and feature vectors, from which we calculated pseudo semantic masks (middle row). We then used the pseudo masks to train the pSp encoder to generate latent codes for reconstructing images (bottom row). It can be seen that the layouts of the bottom-row images reconstructed from the middle-row pseudo semantic masks gradually become close to those of the top-row StyleGAN samples as the training iterations increase.

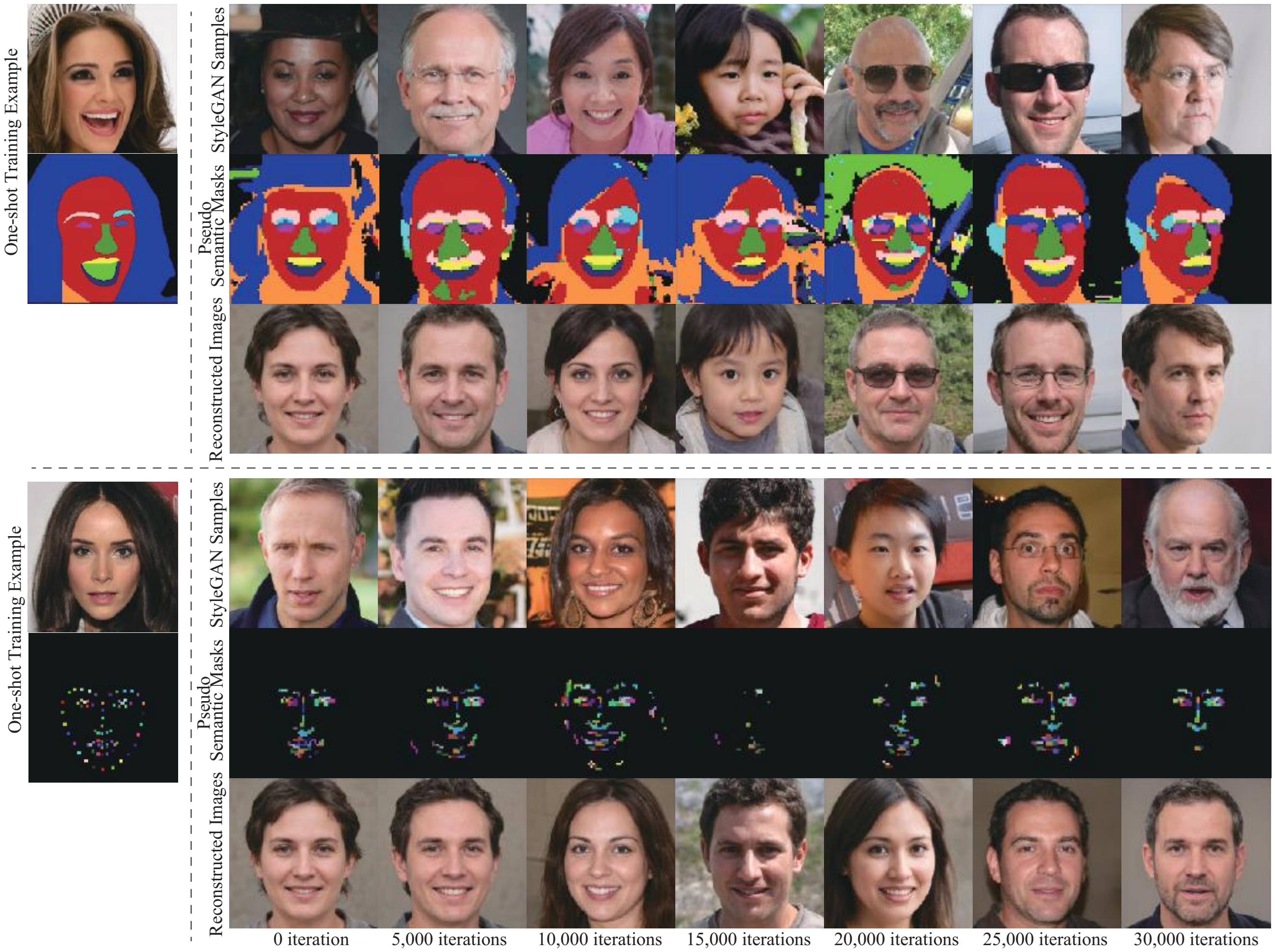

Figure 15: Intermediate training outputs with the StyleGAN pre-trained with the CelebA-HQ dataset.

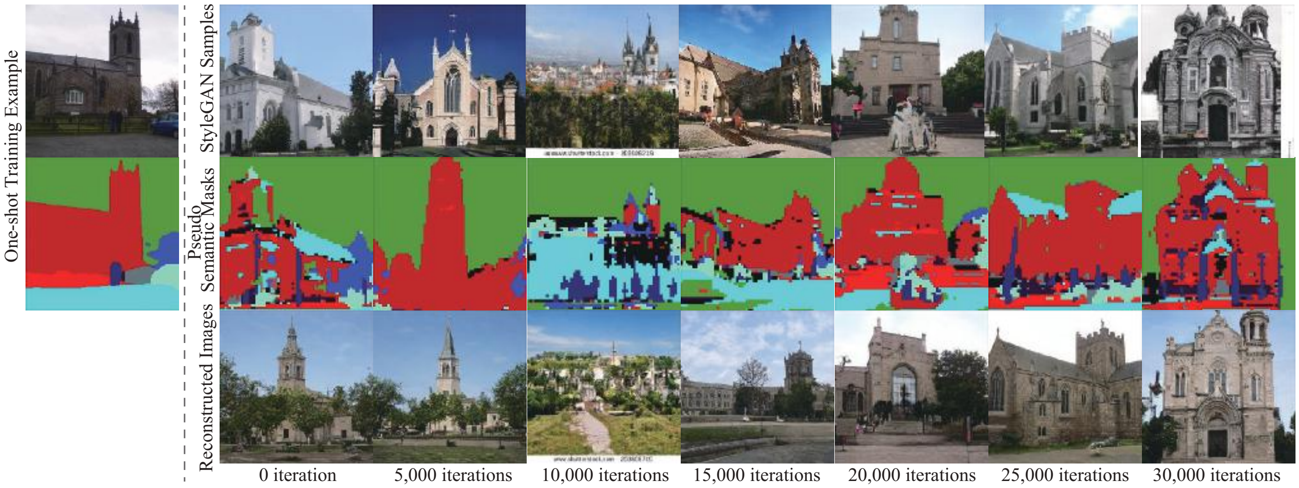

Figure 16: Intermediate training outputs with the StyleGAN pre-trained with the LSUN church dataset.

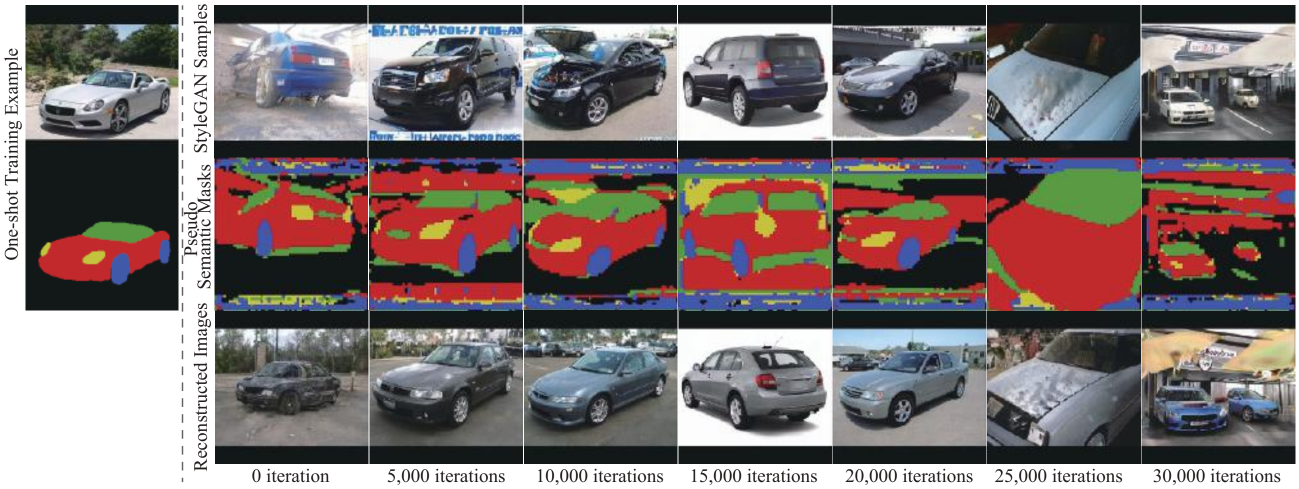

Figure 17: Intermediate training outputs with the StyleGAN pre-trained with the LSUN car dataset.

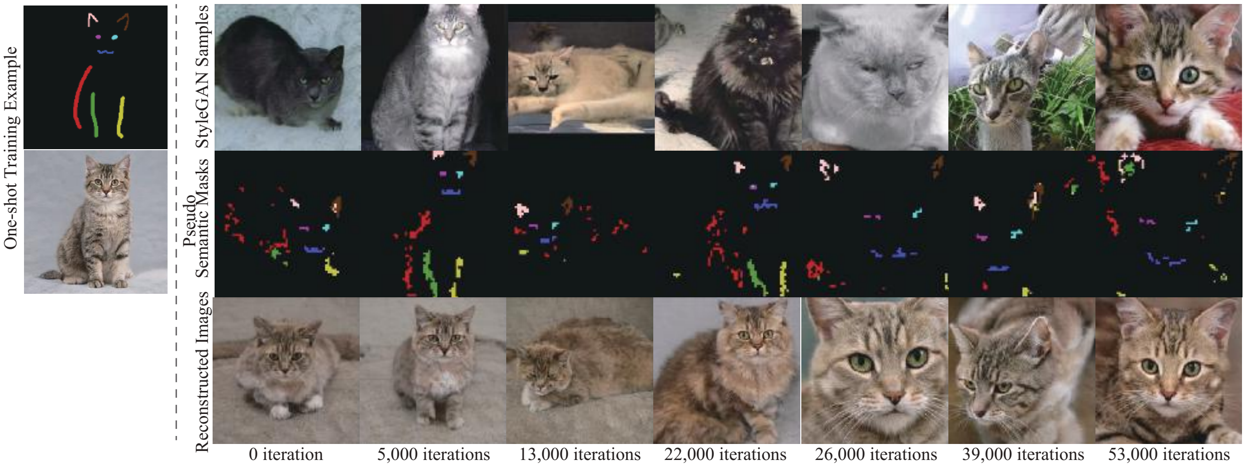

Figure 18: Intermediate training outputs with the StyleGAN pre-trained with the LSUN cat dataset.

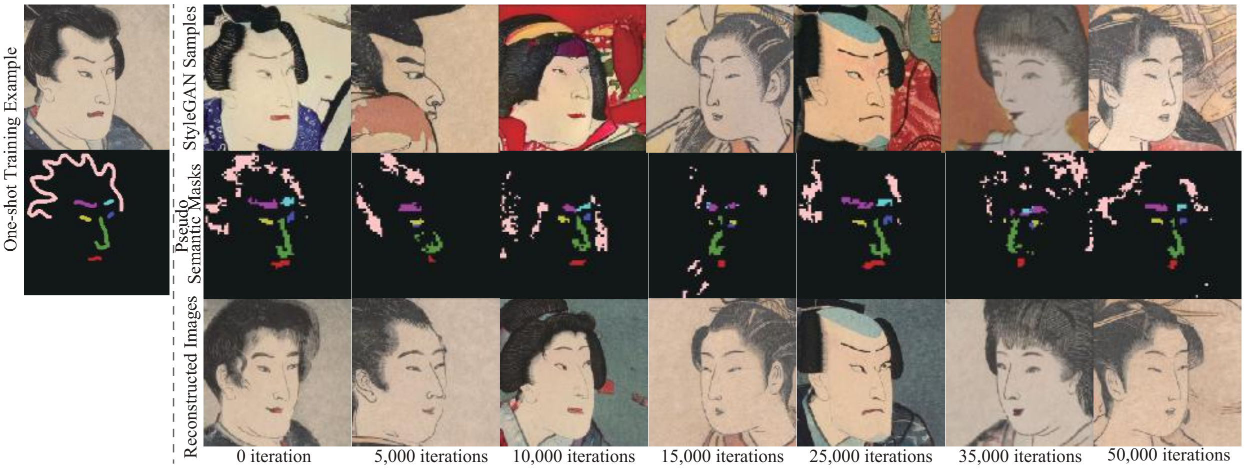

Figure 19: Intermediate training outputs with the StyleGAN pre-trained with the ukiyo-e dataset.

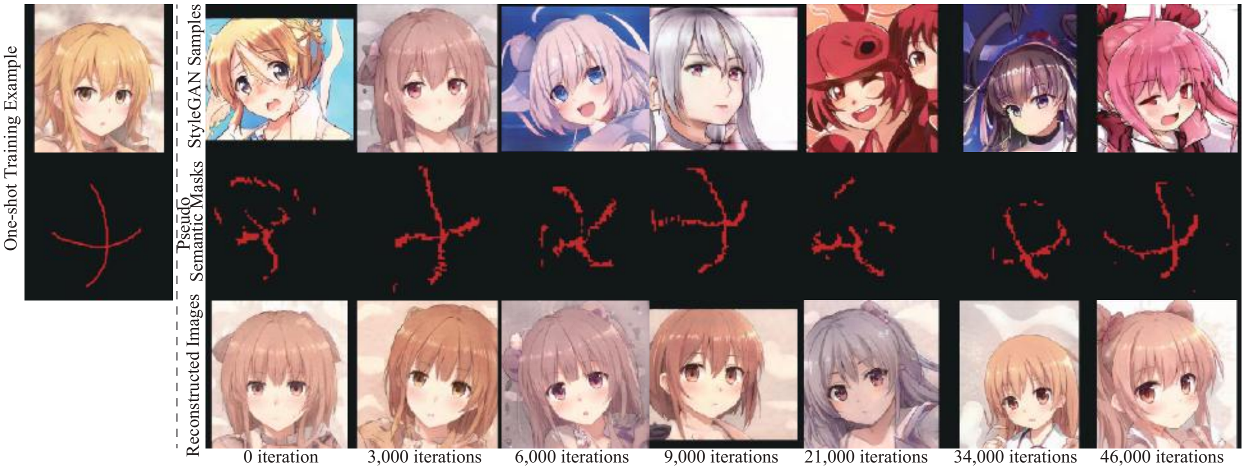

Figure 20: Intermediate training outputs with the StyleGAN pre-trained with the anime face dataset.

D. Multi-modal Results

Figure 21 demonstrates that our method can generate multi-modal results. To obtain multi-modal outputs in test time, we follow the same approach as pSp [26]; we fed latent codes encoded from an input layout to the first l layers of the generator and random noise vectors to the other layers. While we used $l = 8$ for other results in our paper and this supplementary matrial, here we used different values of l to create various outputs. Specifically, we set $l = 8, 5, 7, 5, 5, 5$, and 5 from the top rows in Figure 21. As explained in the pSp paper [26], smaller l affects coarser-scale styles whereas larger l changes finer-scale ones.

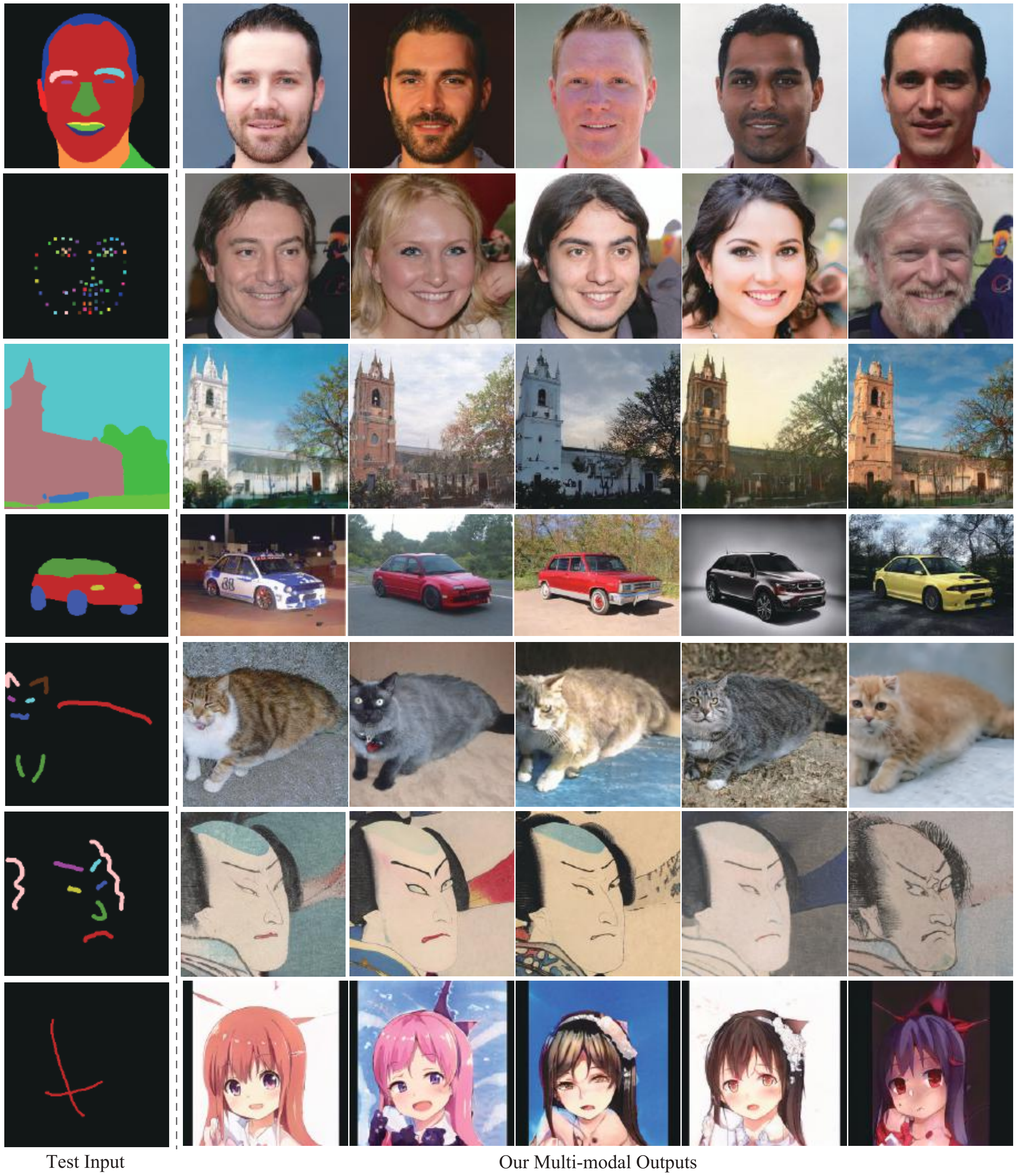

Figure 21: Multi-modal results of our method in few-shot settings.

E. Additional Qualitative Results

Figures 22, 23, 24, and 25 show the additional results. The corresponding few-shot training examples are the same as those shown in the paper.

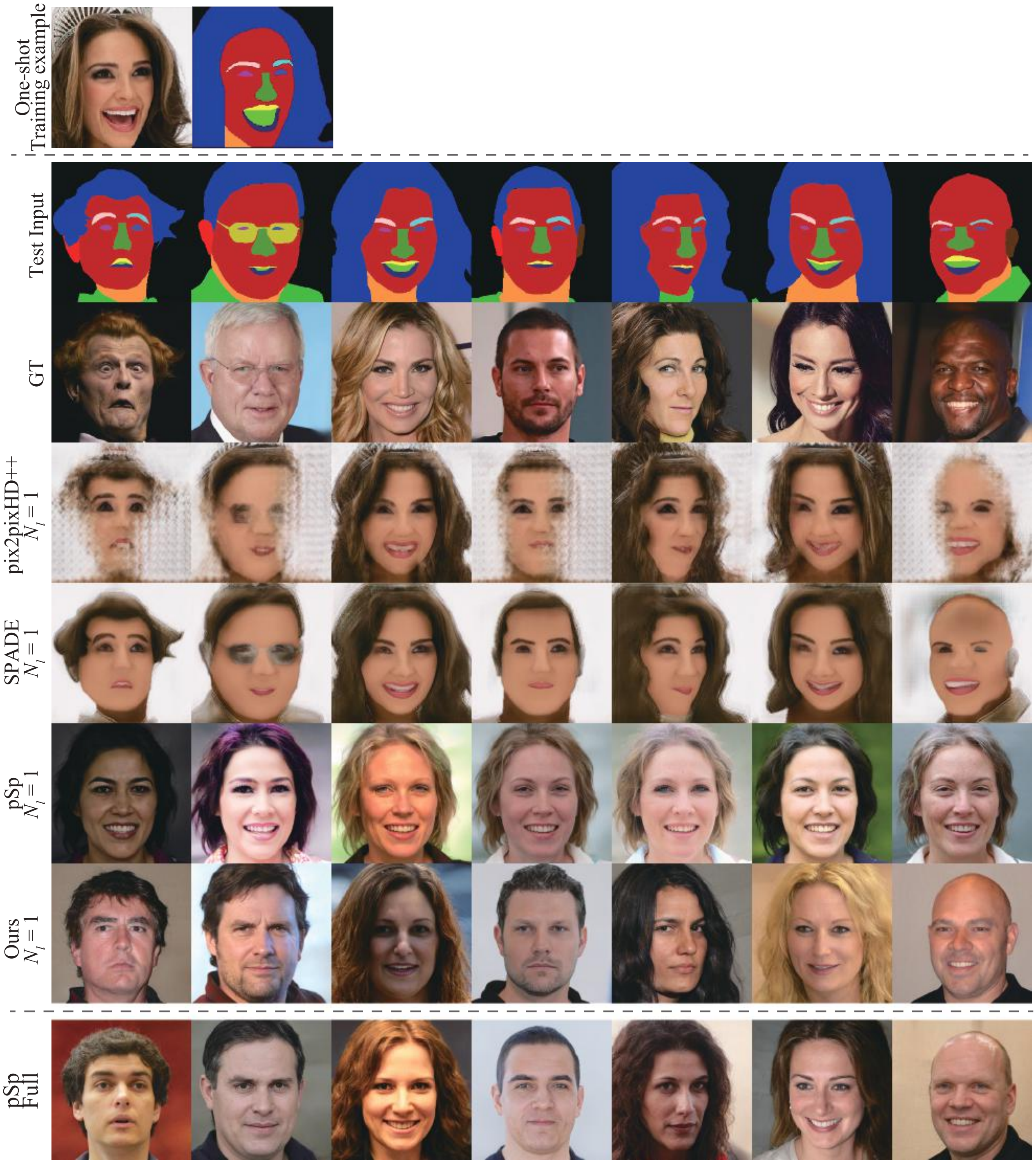

Figure 22: Additional visual comparison on the CelebAMask-HQ dataset.

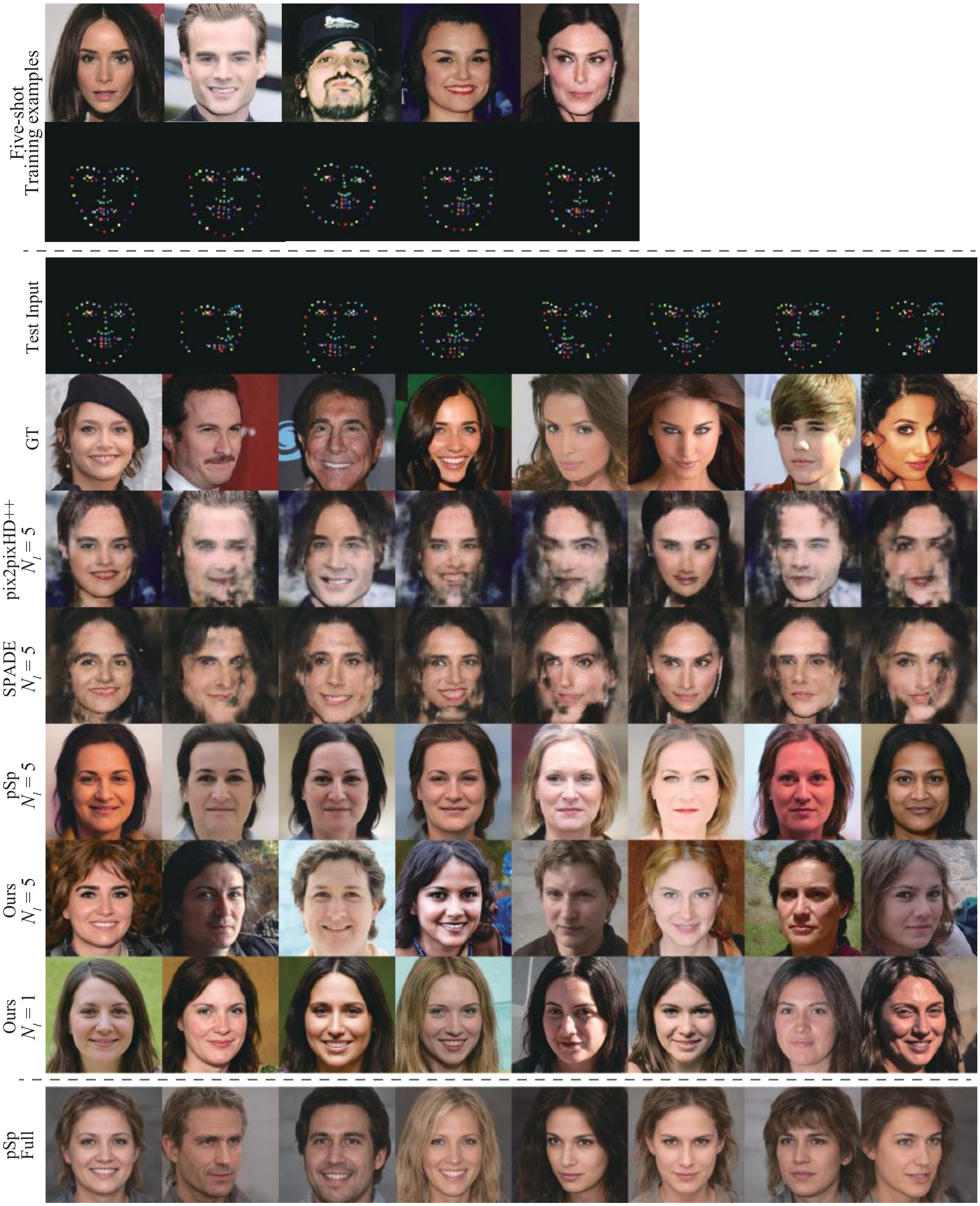

Figure 23: Additional visual comparison on the CelebALandmark-HQ dataset.

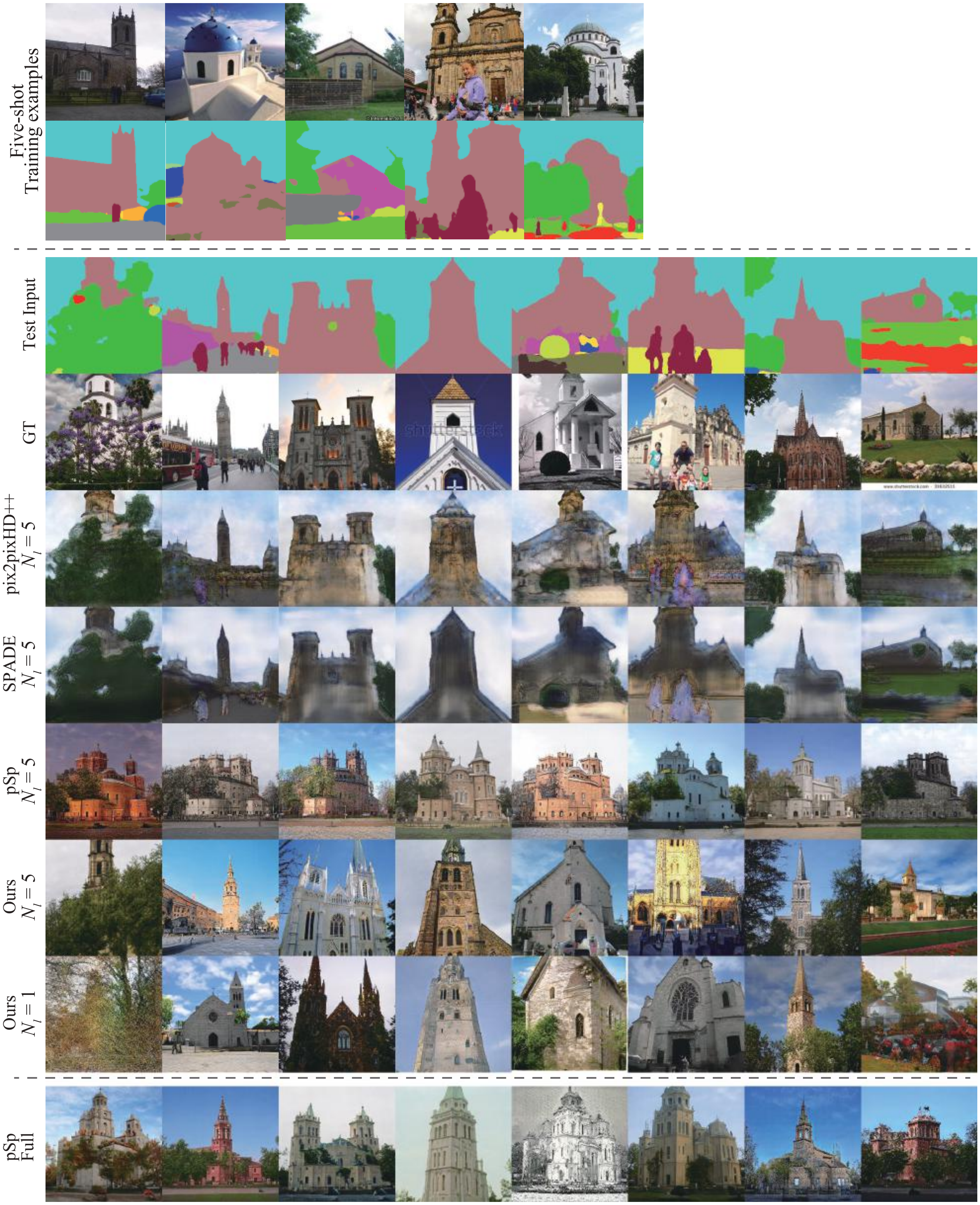

Figure 24: Additional visual comparison on the LSUN church dataset.

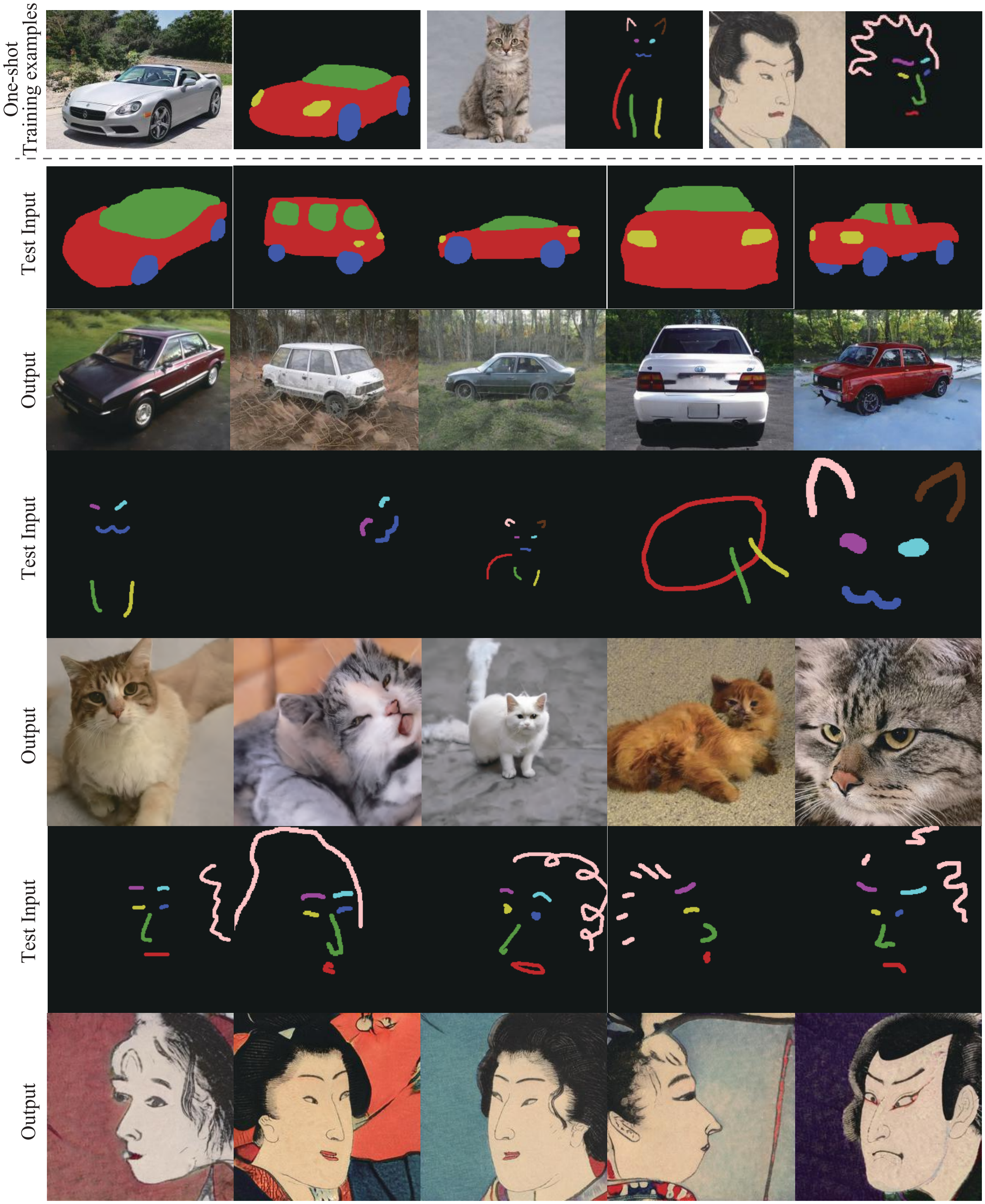

Figure 25: Our additional results obtained using various pre-trained StyleGANs in one-shot settings.

F. Limitation

Figure 26 and Table 3 show the results with a more challenging dataset, ADE20K [42], which consists of 20,210 training and 2,000 validation sets and contains indoor and outdoor scenes with 150 semantic classes. We used the training set without semantic masks to pre-train StyleGAN. Although our method can generate plausible images for some scenes, it struggles to handle complex scenes with diverse and unseen semantic classes.

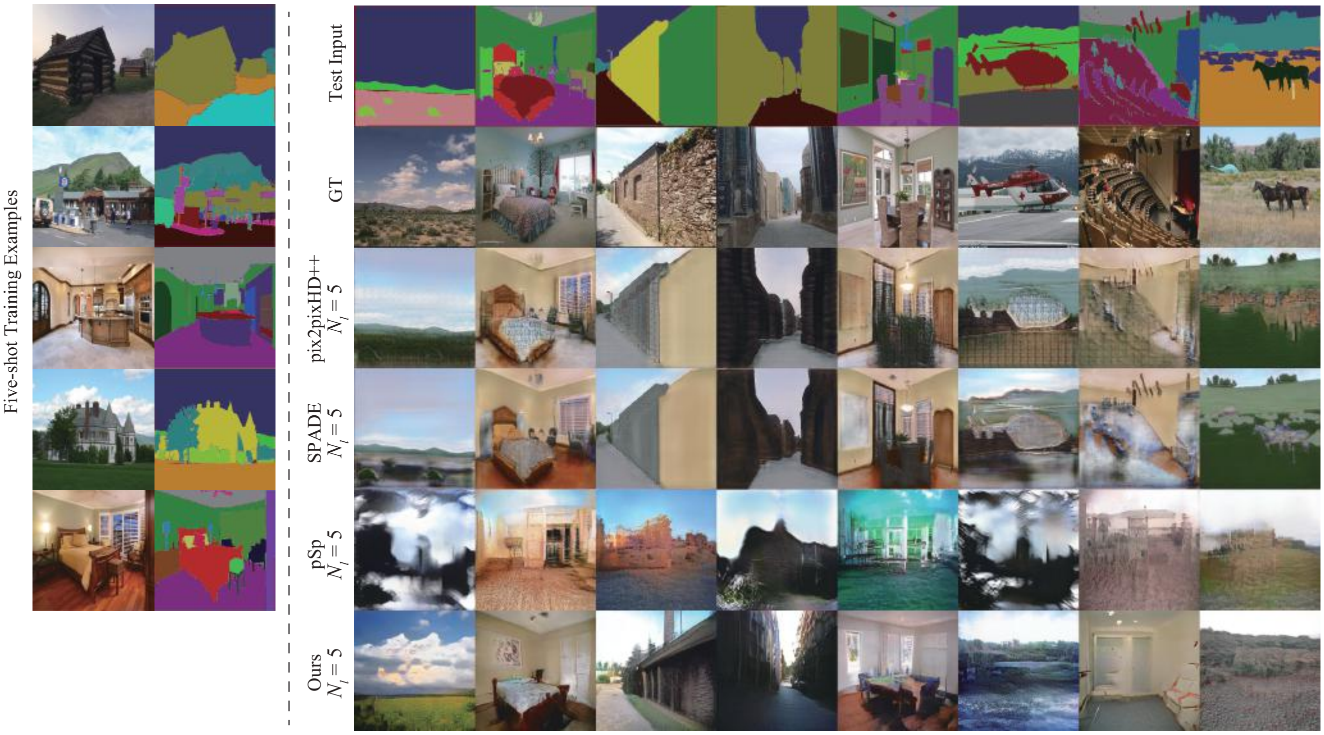

Figure 26: Qualitative comparison on the ADE20K dataset.

Table 3: Quantitative comparison on the ADE20K dataset.

		ADE20K		
Method	N_l	fwIoU↑	accu↑	FID↓
pix2pixHD++ [23]	5	39.2	56.0	110.0
pix2pixHD++*	5	18.7	31.5	142.8
SPADE [23]	5	42.3	58.8	98.1
SPADE*	5	23.1	38.6	129.8
pSp [26]	1	6.3	17.9	187.5
pSp [26]	5	8.8	18.5	177.0
Ours	1	10.8	19.8	155.4
Ours	5	15.8	28.3	95.1