Label-Guided Generative Adversarial Network for Realistic Image Synthesis

Junchen Zhu, Lianli Gao, Jingkuan Song, Yuan-Fang Li, Feng Zheng, Xuelong Li, IEEE Fellow and Heng Tao Shen, IEEE Fellow

Abstract—Generating photo-realistic images from labels (e.g., semantic labels or sketch labels) is much more challenging than the general image-to-image translation task, mainly due to the large differences between extremely sparse labels and detail rich images. We propose a general framework Lab2Pix to tackle this issue from two aspects: 1) how to extract useful information from the input; and 2) how to efficiently bridge the gap between the labels and images. Specifically, we propose a Double-Guided Normalization (DG-Norm) to use the input label for semantically guiding activations in normalization layers, and use global features with large receptive fields for differentiating the activations within the same semantic region. To efficiently generate the images, we further propose Label Guided Spatial Co-Attention (LSCA) to encourage the learning of incremental visual information using limited model parameters while storing the well-synthesized part in lower-level features. Accordingly, Hierarchical Perceptual Discriminators with Foreground Enhancement Masks are proposed to toughly work against the generator thus encouraging realistic image generation and a sharp enhancement loss is further introduced for high-quality sharp image generation. We instantiate our Lab2Pix for the task of label-to-image in both unpaired (Lab2Pix-V1) and paired settings (Lab2Pix-V2). Extensive experiments conducted on various datasets demonstrate that our method significantly outperforms state-of-the-art methods quantitatively and qualitatively in both settings.

Index Terms—Generative Adversarial Networks (GANs), Label-to-Image Synthesis, Photo-realistic Image Generation

1 Introduction

Generating photo-realistic images from labels (e.g., semantic labels or sketch labels), which we refer to as label-to-image, or Lab2Pix hereinafter, can be considered as a subtask of image-to-image translation, which is valuable to many applications including datasets synthesis and image processing. Recently, great progress [1], [2], [3], [4], [5], [6] on image synthesis has been made especially with the advance of Generative Adversarial Networks (GANs) [7]. Label-to-image synthesis is one of the most challenging problems among all types of image synthesis tasks, due to the complexity of scenes that contain multiple objects of different categories. Thus, in order to synthesize high-quality images, it is necessary to focus on both global shapes as well as detailed textures for each object.

Both paired and unpaired data have been used to train Lab2Pix models. Recent paired-data methods [1], [3], [5], [8] and unpaired-data methods [4], [9], [10], [11], [12] have achieved remarkable abilities of generating realistic images from simple scenes. However, for more challenging multi-object or complex-objects scenarios, existing architectures still exhibit unsatisfactory performance. For instance, some state-of-the-art works [3], [4] are unable to synthesize details well on objects with complex textures, while some others [12], [13] require significant computational resources to portray details of a single object.

As pioneering general image translation frameworks, the paired-data method Pix2Pix [8] and unpaired-data method CycleGAN [10] are the first to translate labels to real images. Some works [5], [6], [9], [14] leverage the advantage of multi-stage learning to stabilize the training process and improve the quality of synthesized samples. However, such multi-stack architectures result in a tremendous increase in the number of parameters and training time. Different from the multi-stage design, some works [4], [11] propose to add extra modules (e.g., dilated convolution) to improve performance. These methods show noticeable improvements on the background but inappreciable effects on foreground objects. Besides, inspired by the idea of disentanglement [15], some works [12], [13] propose to encode a whole image as two one-dimension latent code parts (i.e. content and style). Since the code length limits the expression of detailed textures for multi objects, the quality of the synthesized details is poor in complex multi-object scenes.

Label-to-image synthesis is a challenging task due to two major reasons. (1) Compared with other image translation tasks, Lab2Pix suffers from the tremendous differences between the input labels and output images, which has not been specifically considered by existing methods. The input labels only contain pixel-level category attributes while the output images are semantically rich. (2) The sparse information in input labels makes it hard to extract useful features for guiding generation and constraining the synthesized images. For example, in addition to local pixel features, global contents for all instances should be considered for generation. Previous works usually consider this task as a normal image-to-image

• J. Zhu, L. Gao, J. Song and H. Shen are with the Future Media Center and School of Computer Science and Engineering, The University of Electronic Science and Technology of China, Chengdu, China, 611731. Y. Li is with Faculty of Information Technology, Monash University. X. Li is with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P.R. China. X. Li is also with the Key Laboratory of Intelligent Interaction and Applications (Northwestern Polytechnical University), Ministry of Industry and Information Technology, Xi'an 710072, P. R. China. E-mail: lianli.gao@uestc.edu.cn

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3186752

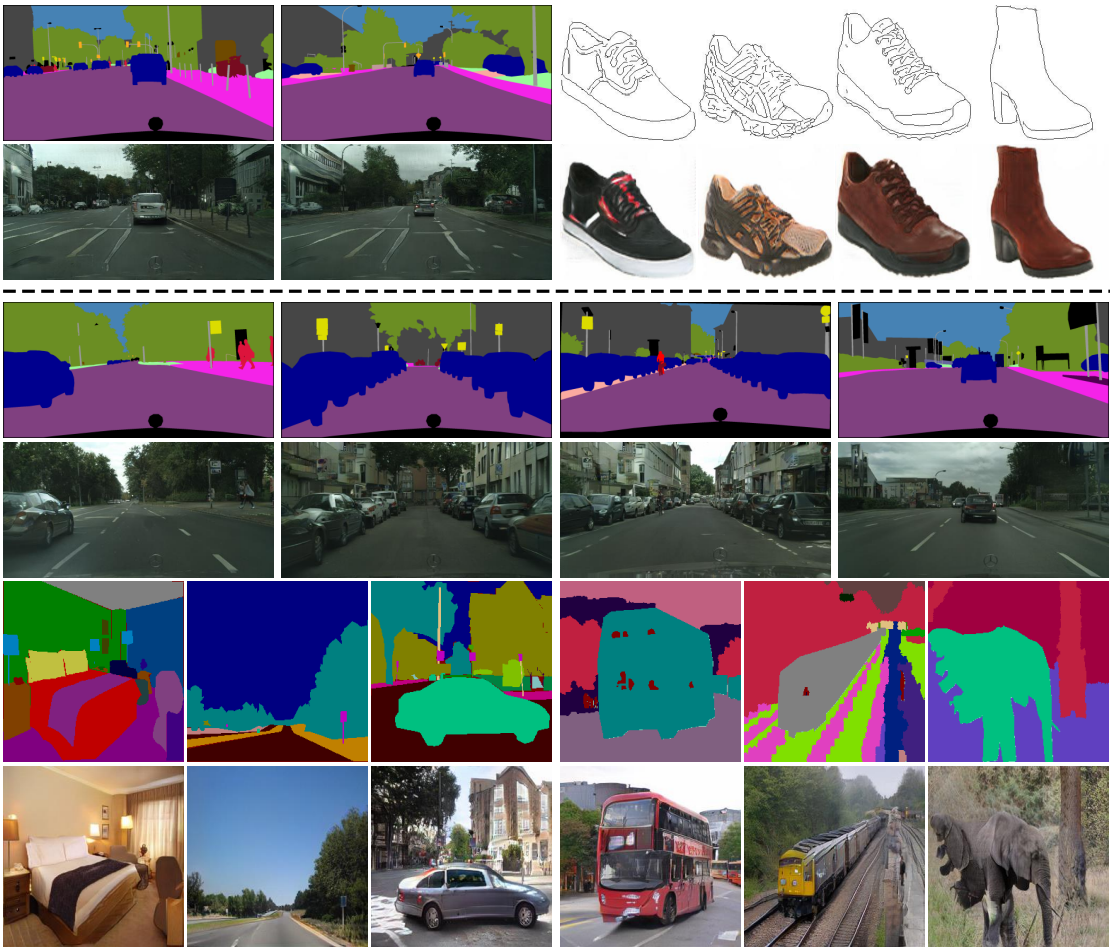


Fig. 1: Some synthesized examples of our Lab2Pix-V1 (above the dash) and Lab2Pix-V2 (below the dash). Our Lab2Pixs take label maps as the inputs and predicts the corresponding realistic images with unpaired-data and paired-data learning. The task is extremely hard since the generated samples are supposed to match the input label maps and keep realistic in complex scenes at the same time. The generated samples from our model are colorful and photo-realistic and contain detailed textures.

translation, and barely consider the special attributes in the input label map, the output raw image, and the huge gap between them. Thus, they often achieve incomplete generation and obtain results with blank holes and few details.

In this paper, we focus on two aspects in the label-to-image task. First, to comprehensively extract features from input label maps with sparse information, we propose Double-Guided Normalization (DG-norm) and Label Guided Spatial Co-attention (LSCA) for the image generator. Specifically, DG-norm provides extra global information (e.g., the entire shape of an object and its neighboring instances) in guiding image generation compared with existing methods that only consider pixel-level attributes. LSCA prevents the network from losing well-synthesized parts under limited model parameters with the guidance of label maps. Second, to constrain output images thus encouraging photo-realistic image generation from label maps, we design a novel and powerful Hierarchical Perceptual Discriminator (HPD) and a general Sharp Enhancement Loss. In detail, Hierarchical Perceptual Discriminators are designed in different structures for different scale images, to discriminate objects with both low-level visual concepts and high-level semantic information. Compared with existing ones, HPD provides hierarchical discrimination to fully consider different objects in complex scenes described by output images with the help of auxiliary perceptual features. Furthermore, we take advantage of data itself to make blurry samples, and add them as negative samples into the adversarial training to boost clear image generation which we term as the sharp enhancement loss. Additionally, we propose a novel Foreground Enhancement Mask in adversarial loss calculation to focus more on the challenging foreground generation with the label guidance. Different from existing methods, we fully consider the sparse information contained in the input label and the abundant detail described in the output image. We propose several modules and loss functions to boost the complete feature extraction and expression. Meanwhile, we introduce simple fusion modules to maintain the large-span translation with the limited model.

Based on the above components, we propose a unified GAN framework Lab2Pix, illustrated in Fig.2, with two versions (Lab2Pix-V1 [16] in Fig.7 and Lab2Pix-V2 in Fig.8) for the challenging Lab2Pix task in the unpaired-data and the paired-data settings, respectively. Both generators produce

multi-scale images in one forward pass and each image is distinguished by one independent discriminator. To stabilize the unpaired-data training, we further propose Image Consistency Loss and Cycle Segmentation Loss. To confirm our model's ability of generating high-resolution images, we build Lab2Pix-V2-H to synthesize double-scale samples of basic Lab2Pix-V1 and Lab2Pix-V2.

The major contributions of this paper can be summarized as follows:

1) To extract useful information from sparse labels, we propose a Double-Guided Normalization (DG-Norm), where the input label is utilized for semantically guiding generation. The global feature with large receptive fields is added to differentiate the activations within the same semantic region.
2) To efficiently generate the images using limited model parameters, we propose a Label Guided Spatial Co-Attention (LSCA) to encourage the incremental learning of visual information while storing the well synthesized part in lower-level features.
3) To encourage realistic and clear image generation with abundant details, we equip our model with a set of novel hierarchical perceptual discriminators and constraints including sharpness enhancement loss, image consistency loss and cycle segmentation loss.
4) We instantiate our Lab2Pix framework for the task of label-to-image with both unpaired and paired settings. Extensive experiments on six benchmark datasets demonstrate that both our models achieve state-of-the-art results both quantitatively and qualitatively.

Our source code and models are available at https://github.com/RoseRollZhu/Lab2Pix.

2 Related Work

2.1 Conditional GANs

Generative Adversarial Networks (GANs) [7] are proposed to synthesize various data. In general, they can be divided into conditional and unconditional types. Conditional GANs provide approaches for users to control synthesized data with some additional information. For instance, categories [17], [18], sketches [19], [20], descriptions [21], [22], bounding boxes [23], [24] and special attribute codes [25], [26] have all been used as the input guidance.

Most of the earlier studies on conditional GANs are based on paired-data learning. Later architectures were prposed [27], [28], [29], [30] to support unpaired-data learning. Compared with the unconditional setting, conditional GANs rely more on paired data for training. The accurate mapping functions between condition values and generated results are difficult to learn if corresponding samples are not given. Because of this, with the same or similar structures, models with paired learning usually perform much better (e.g., [11]). In general, the performances of paired-data learning are much better than unpaired-data ones. In this work, we employ the conditional GAN model trained with both paired and unpaired data.

2.2 Image Synthesis from Label

Image-to-image translation, usually tackled by GANs [7], is to synthesize images in the target domain from the source domain (e.g., image style translation [31], [32], object translation [33], [34], image super-resolution [35], [36]). Label-to-mage synthesis, a subtask of image-to-image translation, limits the source domain to label maps (e.g., semantic label maps or sketch label maps) and the target domain to real-world images. Totally, this task can be divided into paired-data and unpaired-data training settings.

In the paired-data training setting, the model are fed with label maps and corresponding images for training. The pioneering work Pix2Pix [8] directly applies U-Net [37] to generate the images. CRN [1] suggests to synthesize the images from low resolution to high resolution progressively, which may stabilize the training process and improve image quality. Pix2PixHD [5] seeks to address super high resolution image generation by splitting the task to multi separate stages. The state-of-the-art method SPADE [3] proposes the spatial adaptive normalization for labels to guide generation without erasing useful information. Inspired by SPADE [3], many works have been proposed. CC-FPSE [38] design a conditional convolution and semantics-embedding strategy for label maps to better guide the generation and discrimination. TSIT [39] adopts a versatile two-stream framework with multi-scale feature normalizations to integrate the content and style of generated images. Compared with SPADE only focusing on local information, we introduce a double-guided normalization to fully consider the local and global features of label maps for complex objects.

For the more challenging unpaired-data training setting, unpaired label maps and images are used for training. CycleGAN [10] firstly proposes a cyclic architecture to support this task. Most works follow the basic cyclic structure to train with unpaired data. SCAN [9] uses a two-stack architecture to synthesize high-resolution images progressively, where the first stack processes data in half scale, and the parameters of each stack are updated iteratively. Inspired by the segmentation network, SPAP [4] designs a coarse-to-fine fusion structure with dilated convolutions. They use convolutions with different dilation sizes to capture multi-scale information of the image. MUNIT [12] encodes image information as style and content parts and exchanges these two parts to synthesize images in different styles.

Different from prior works, which usually design the architecture for the general image-to-image translation, our Lab2Pix models consider the specific properties that make label-to-image more challenging, i.e., the significant differences between the input labels and output images.

3 Proposed Method

Given a label map M (i.e., a semantic map or a sketch map), we aim to synthesize a high-resolution photo-realistic image with an end-to-end label-to-image network. Moreover, our design is supposed to support both the unpaired-data and paired-data training, which means the real image Y provided in each training pair shares the same or different layout with the provided M. Considering the challenging label-to-image task, we can conclude that the input labels only contain pixel-level category attributes and no semantic information, while the output images are supposed to be aligned with the input labels, photo-realistic and semantically rich. To generate high-quality images under limited model parameters,

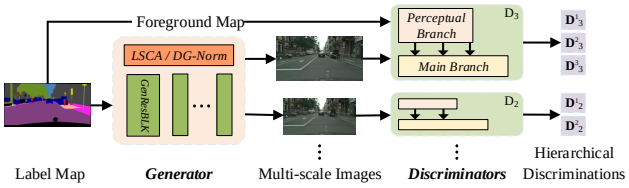


Fig. 2: Illustration of our proposed unified Lab2Pix (including Lab2Pix-V1 and Lab2Pix-V2) framework. The generator takes a label map to synthesize multi-scale images, and independent discriminators give hierarchical discriminative results for each image based on the foreground map extracted from the label map.

the network is supposed to dig information from input label maps as much as possible and bridge the gap between the sparse labels and detailed rich images. Thus, we propose a few modules in generators to process the label maps, a set of strong discriminators and extra loss functions to encourage realistic image generation, which compose a unified Lab2Pix framework as illustrated in Fig. 2.

3.1 Extract Information from Sparse Label

We propose three modules to extract information from sparse labels for comprehensive label processing. The three modules address the challenging label processing in different ways and can coexist in one model.

3.1.1 Adaptive Label Encoder.

To generate images, we are supposed to encode labels at the front stage of the network. Given a semantic label, each pixel contains a semantic category information. In contrast, a sketch label contains few information and its informative pixels are extremely sparse. Therefore, we design two label encoders: the semantic label encoder and the sketch label encoder to adaptively handle semantic and sketch information. Their detailed structure is shown in Fig. 3. Both label encoders take a label map M and a randomly generated standard Gaussian noise z as input. In this task, we add a fully connected layer to z and then reshape it to a 3-dimensional noise feature f_z. Then, we combine the noise feature f_z with a label feature. Specifically, for the sketch label encoder, we design a sketch encoder with multiple convolutions to extract the overall feature $\mathrm{f}_{l_{sk}}$ of the sketch labels. The strides are set as 2 for expanding the receptive field of each pixel. Finally, we concatenate f_z with $\mathrm{f}_{l_{sk}}$ as the output of the sketch label encoder. For the semantic label encoder, we use two stacked generative residual blocks (GenResBlk) to encode the input label map and fuse it with the noise information. Specifically, we input f_z and M to the first GenResBlk and obtain a coarse feature, as illustrated in Fig. 3. Then, we inject the coarse feature and M to the second GenResBlk for further encoding and take the result as the output of the semantic label encoder. Here, no extra encoder is required, since every pixel in M provides rich guiding information.

3.1.2 Generative Residual Blocks (GenResBlk).

GenResBlk is the conditional residual block that synthesizes image features at a specific scale along with the whole

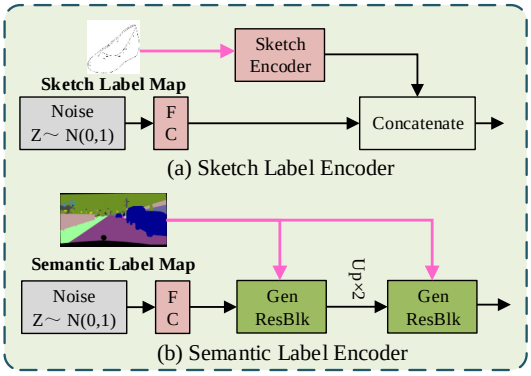


Fig. 3: The structure of our Adaptive Label Encoder. It separately encodes the sketch and semantic label maps according to their characteristics.

generation. Inspired by the previous work [3], we adopt the normalization layer for the label to guide the image synthesis described as a function G_{rb}. The entire process of GenResBlk can be described as follows:

$$\mathrm{d}_{t+1} = \mathrm{G}^t_{rb}(\mathrm{d}_t, \mathrm{M}), \tag{1}$$

where d_t is the input of the t-th GenResBlk and d_{t+1} is upscaled from d_t by a factor of 2. We find that SPADE [3] only processes the label maps with one-layer embeddings (only two layers of 3×3 convolutions) as the input. Thus, the guidance information only contains pixel-level category attributes. This design works fine for objects with similar textures in different patches (e.g., sky, road, grass), but gives unsatisfactory performances in instances with complex textures (e.g., vehicle, animals). Each patch in these instances contains different sub-objects, which means the generator needs to locate each body part from the global shape for high-quality synthesis.

The category information is not enough for high-quality image synthesis. If the global information (e.g., global shapes, global positions) is ignored in the guidance, the generated objects in the scenes may not be reasonable even though some patches are well synthesized. Thus, we design a global encoder to provide an extra feature for generation. The encoder consists of several convolutions with stride 2. We only want to obtain the global feature, so the global encoder only gives one final result. The process can be expressed as follows:

$$\mathrm{H} = \mathrm{E}_{global}(\mathrm{M}). \tag{2}$$

Then, unlike baseline GenResBlk G_{rb}, with the help of the global encoder, we add global information as an extra condition to guide generation in GenResBlks. We design the novel Double-Guided Normalization (DG-Norm), which is shown in Fig. 4. Let E^i denote the activation before normalization. In DG-Norm, E^i will go through a batch normalization BN, which can be expressed as follows:

$$BN(\mathrm{E}^i) = \gamma_{BN}\left(\frac{\mathrm{E}^i - \mu(\mathrm{E}^i)}{\sigma(\mathrm{E}^i)}\right) + \beta_{BN}. \tag{3}$$

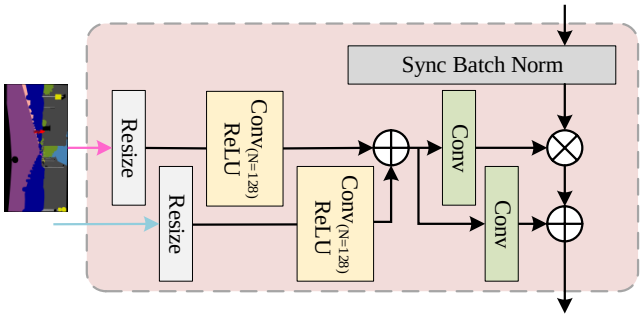


Fig. 4: The structure of our DG-Norm. The input feature is normalized by batch normalization first. Then we use both the label map and global feature to predict the new distribution parameters which effect the normalized feature.

where γ_{BN} and β_{BN} are learned parameters in the batch normalization. $\mu(\mathrm{E}^i)$ and $\sigma(\mathrm{E}^i)$ can be calculated as follows:

$$\begin{aligned}\mu(\mathrm{E}^i)_c &= \frac{1}{NH^iW^i}\sum_{n,h,w}\mathrm{E}^i_{nchw},\\ \sigma(\mathrm{E}^i)_c &= \sqrt{\frac{1}{NH^iW^i}\sum_{n,h,w}(\mathrm{E}^i_{nchw})^2-\mu(\mathrm{E}^i)_c},\end{aligned} \tag{4}$$

where H^i and W^i are the height and width of E^i. Then, we can de-normalize the feature with new parameter, the function can be expressed as:

$$DG(\mathrm{E}^i) = \gamma_{DG}\cdot(BN(\mathrm{E}^i)+1)+\beta_{DG}, \tag{5}$$

where γ_{DG} and β_{DG} can be obtained as in Fig. 4. We resize the label map and global feature to the same size as the input E^i. We fuse the guidance information by element-wise addition. The total process of proposed new GenResBlk G_{rb*} containing DG-Norm can be described as follows:

$$\mathrm{d}^{'}_{t+1} = \mathrm{G}^t_{rb*}(\mathrm{d}^{'}_t, \mathrm{M}, \mathrm{H}), \tag{6}$$

where $\mathrm{d}^{'}_t$ is the input of the t-th GenResBlk, and $\mathrm{d}^{'}_{t+1}$ is upscaled from $\mathrm{d}^{'}_t$ by a factor of 2.

3.1.3 Label Guided Spatial Co-attention (LSCA).

As the resolution goes higher at the back stage of generation, the classical GenResBlk, which is equipped with convolution and normalization layers to process data in low dimension, may not be able to maintain all object features. Objects features include lower-level visual features like textures and colors, middle-level visual features like object part attributes, and high-level visual features like object semantic information. Note that, the lower-level coarse and high-level fine structures of objects with simple textures (e.g., grass) are similar, which helps these textures to be synthesized well enough at the early generation stage. As a result of limited network parameters, these simple visual features may be cleaned away quickly during the later generation stage when the generator focuses on complex object textures, or the model has to give up synthesizing incremental fine details for these objects.

To address this issue, we propose the LSCA to relieve the information loss by producing a co-attention map to refine image features. It fuses features at different scales and

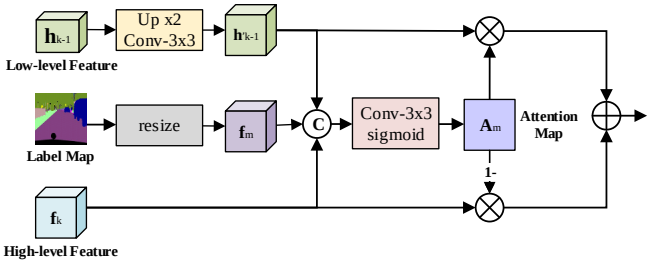


Fig. 5: The structure of an Label Guided Spatial Co-Attention (LSCA) block. Our LSCA fuses features in different layers by an attention map with label guidance. C denotes the operation of concatenating.

dimensions with the label guidance. The structure of our proposed LSCA is demonstrated in Fig. 5. It has three inputs: previous low-level visual features h_{k-1}, current high-level visual features f_k, which is the output of current GenResBlk, and the label map M with semantic information. First, to avoid the Checkerboard-Artifacts issue [40], h_{k-1} is upscaled by a factor of 2 and operated by a 3×3 convolution to produce a merge-able low-level feature h'_{k-1}. Besides, we resize M to the same size as h'_{k-1} and f_k. The resized M is defined as f_m. Second, all the above three features are concatenated to obtain an attention map A_m by passing it in a convolution layer activated by a sigmoid function. Finally, h'_{k-1} and f_k are filtrated by A_m to yield the fused feature f_k, described as:

$$\mathrm{f}_k = \mathrm{h}'_{k-1}\cdot\mathrm{A}_m + \mathrm{f}_k\cdot(1-\mathrm{A}_m), \tag{7}$$

where $\cdot$ represents element-wise product with broadcasting and $+$ indicates an element-wise sum operation.

3.2 Bring the Gap between Label and Image

We propose a set of novel discriminators and loss functions to constrain the synthesized images thus encourage high-quality images with rich details generation from label maps.

3.2.1 Hierarchical Perceptual Discriminator.

For a GAN network, competition in this minimax two-player game drives both models to improve their performance until the counterfeit samples are indistinguishable from the genuine samples [7]. In this paper we propose novel hierarchical perceptual discriminators D, whose discriminative ability is strong and competitive in contrast to our generator's generative power thus encourage high-quality image generation.

The translation process from label to image consists of the synthesis of a variety of visual concepts for multiple objects, such as textures, various compositional parts, and categorized attributes. Thus, fully checking the object details, parts and category information is beneficial for improving the ability of a discriminator. Specifically, we design three independent discriminators (i.e., D_1, D_2 and D_3) to consider multi-scale information. All the discriminators are designed in PatchGAN [8] style which means that no fully-connected layer is used to capture global information. All the real samples are resized to the size of the generated samples. Specifically, we equip our discriminators with three novel model designs (i.e., hierarchical discrimination, mini-inception block and perceptual branch) and one novel function design (i.e., foreground enhancement mask), which make them significantly

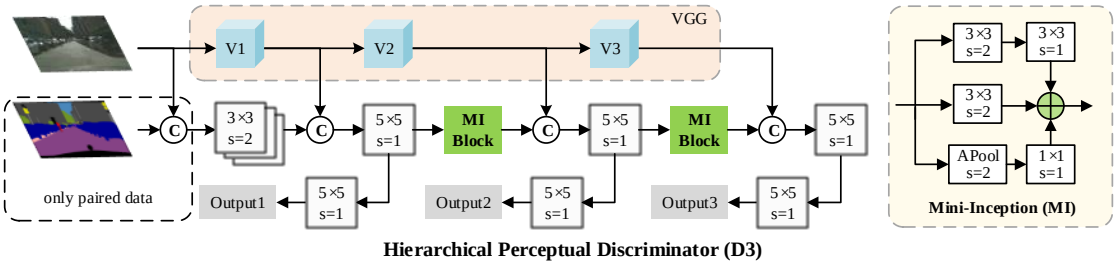


Fig. 6: The proposed hierarchical perceptual discriminator structure, which includes two branches. The perceptual branch take the images as the input. The concatenation of images and the corresponding label maps are fed to the main branch in Lab2Pix-V2, while in Lab2Pix-V1, only images are fed. v_1, v_2 and v_3 indicate Conv_{1_1}-pool_3, Conv_{4_1}-pool_4 and Conv_{5_1}-pool_5 of pretrained VGG16. s represents stride in convolution and pooling. C denotes the operation of concatenating.

different from existing ones. The details will be described in the following subsections.

Hierarchical Discrimination. As shown in Fig. 7 and Fig. 8, for both models, we have multiple discriminators to discriminate multi-scaled images X_i. The structure of D_3 with full components is shown in Fig. 6. A $W \times H$ image contains more precise high-level visual information than its half-sized $(0.5W \times 0.5H)$ image. Thus, D_3 has three levels of outputs to recognize visual concepts: low-, middle- to high-level, while D_2 removes the third output and focuses on capturing low- and middle-level visual concepts. Similarly, D_1 is designed with the first output to capture low-level visual concepts.

If we only directly input the images to discriminators which is a typical and suitable condition for most image generation tasks (including the unpaired setting in our label-to-image), the three discriminators in our model are defined as follows:

$$\begin{aligned} \mathrm{D}_1(\mathrm{X}_1) &= \{\mathrm{D}_1^1(\mathrm{X}_1)\}, \\ \mathrm{D}_2(\mathrm{X}_2) &= \{\mathrm{D}_2^1(\mathrm{X}_2), \mathrm{D}_2^2(\mathrm{X}_2)\}, \\ \mathrm{D}_3(\mathrm{X}_3) &= \{\mathrm{D}_3^1(\mathrm{X}_3), \mathrm{D}_3^2(\mathrm{X}_3), \mathrm{D}_3^3(\mathrm{X}_3)\}, \end{aligned} \tag{8}$$

where D_i^l indicates the output of D_i in level l.

Mini-Inception Block. To increase the depth of the discriminator network while keeping the computational budget constant, we borrow the idea of Inception [41] and design a mini-inception block to further improve our discriminator. As Fig. 6 shows, it has three parallel branches. The different branches extract features in different levels and combine them as the output.

Perceptual Branch. In VGG-GAN [42], a pre-trained deep classification network is embedded inside the discriminator to improve the robustness and efficiency of perceptual losses. However, directly embedding a pre-trained deep classification network works for facial images with a single object but fails for natural images with several objects. To address this issue, we consider a pre-trained deep classification network as a supplementary perceptual branch to our discriminator. Our framework is different from VGG-GAN in two important aspects. Firstly, our discriminators are designed mainly based on an independent encoder structure. Secondly, we add the perceptual branch to boost their performance as a supplementary module. If the perceptual part is removed, our discriminators can still distinguish samples with a weaker ability, while discriminators in VGG-GAN would be non-functional. As shown in Fig. 6, we divide the perceptual branch into three parts v_1, v_2 and v_3. The perceptual information extracted from each part is combined to the main branch by concatenation. In addition, we choose VGG16 [43] pre-trained on the ImageNet [44], and all parameters are fixed during training.

Apart from basic discriminator structure design, we propose novel Foreground Enhancement Mask in the adversarial loss calculation. Compared with substances in the background, foreground objects have more complex textures, which make them difficult to synthesize. Enabling the discriminator to focus more on the foreground may boost the ability of foreground object generation. Motivated by this, we multiply the prediction result with a weight map W_i^l to increase the weight of foreground parts when calculating the GAN loss function. Different from the current self-attention mechanism in image generation [45], our function uses an accurate weight map to lead the attention on foreground parts with little computational cost. Specifically, a label map M can be manually divided into two parts: foreground pixels (e.g., vehicle, bicycle and sign) with a small number A and background pixels (e.g., sky, building and road) with a large number B. To obtain W_i^l, we create a foreground enhance mask M_{en} which only contains two values. All background pixels have value P and all foreground pixels have value $T \times P$, where T is a hyper-parameter to control the rate of enhancement. For unpaired-data learning, since real samples do not have the corresponding label map, it is important to keep the mean value of the whole mask M_{en} to 1. Thus, P can be calculated as:

$$P = (A + B)/(T \times A + B) \tag{9}$$

Our PatchGAN-style discriminators' outputs share the same spatial layout of the input images and its label map M. Thus, we can adjust each pixel's weight of the discrimination result D_i^l by simply multiplying a M_{en}-related enhancement map W_i^l, which shares the same scale with D_i^l. To obtain W_i^l, we perform average pooling on M_{en} with different kernel sizes. Based on adversarial loss calculation functions proposed in previous works [7], [46], we use L_{adv}^R and L_{adv}^F to indicates adversarial loss functions for real and fake samples. Thus, the GAN loss function for the generator is:

$$\mathcal{L}_1^G(\mathrm{X}) = \sum_i \frac{1}{\sum_{l=1}^i \lambda_{il}} \sum_{l=1}^i \lambda_{il} (\mathbb{E}_{\mathcal{M}}^{il}[L_{adv}^R(\mathrm{D}_i^l(\mathrm{X}_i))]). \tag{10}$$

The GAN loss function for the discriminators can be divided into two parts: real prediction loss and fake prediction loss, which can be expressed as:

$$\mathcal{L}_1^D(\mathrm{X},\mathrm{Y}) = \frac{1}{2}(\mathcal{L}_1^D(\mathrm{Y})_R + \mathcal{L}_1^D(\mathrm{X})_F). \tag{11}$$

We obtain the fake prediction loss as:

$$\mathcal{L}_1^D(\mathrm{X})_F = \sum_i \frac{1}{\sum_{l=1}^i \lambda_{il}} \sum_{l=1}^i \lambda_{il} \mathbb{E}_{\mathcal{M}}^{il}[L_{adv}^F(\mathrm{D}_i^l(\mathrm{X}_i))]. \tag{12}$$

As for real prediction loss, the foreground enhancement mask is not available in unpaired-data learning. Thus, it can be calculated as:

$$\mathcal{L}_1^D(\mathrm{Y})_R = \sum_i \frac{1}{\sum_{l=1}^i \lambda_{il}} \sum_{l=1}^i \lambda_{il} \mathbb{E}[L_{adv}^R(\mathrm{D}_i^l(\mathrm{Y}_i))], \tag{13}$$

where

$$\mathbb{E}_{\mathcal{M}}^{il}[q] = \mathbb{E}[\mathrm{W}_i^l \cdot q], \tag{14}$$

and $\cdot$ is element-wise dot production and λ_{il} is the hyper-parameter. We set $\lambda_{i1} = 1$ and $\lambda_{i(l+1)} = \frac{1}{2}\lambda_{il}$. Note that the sketch maps lack semantic categorical information, thus we do not use the foreground enhancement mask, which is equivalent to setting W_i^l as a matrix of ones.

Finally, the loss function of our GAN is:

$$\mathcal{L}_1(\mathrm{X},\mathrm{Y}) = \mathcal{L}_1^G(\mathrm{X}) + \mathcal{L}_1^D(\mathrm{X},\mathrm{Y}). \tag{15}$$

3.2.2 Loss Functions

We propose a novel and general Sharpness Enhancement Loss for the photo-realistic image generation. One major difficulty in synthesizing high-resolution images is that the network may fail to penalize real but blurry images. To solve this problem, we downscale real images and upscale the downsampled ones both with a scale factor 2 to obtain real but blurry images, and treat them as fake samples. If discriminators can differentiate these samples, they will force the generator to synthesize sharp and realistic images in return. Specifically, we manually pre-process the training samples Y into three resolutions Y_i^f $(i = 1, 2, 3)$, where Y_1^f has the lowest resolution. When training our discriminators, we only consider the ground-truth image Y_i as the real sample. The sharpness enhancement loss is a supplement to the generative adversarial loss. On the one hand, we directly add the extra fake prediction loss. On the other hand, we need to increase the real prediction loss with the same weight. Since this function itself only adds the loss on the fake part to the total adversarial loss, we keep the real-fake balance by adding the real part loss value with the same weight. The loss function is defined as:

$$\mathcal{L}_2(\mathrm{Y},\mathrm{Y}^f) = \frac{1}{2}(\mathcal{L}_1^D(\mathrm{Y})_R + \mathcal{L}_1^D(\mathrm{Y}^f)_F). \tag{16}$$

4 Lab2Pix Model

To evaluate the proposed modules and optimization strategies, we instantiate our Lab2Pix framework with two models (i.e., Lab2Pix-V1 and Lab2Pix-V2) for the label-to-image task under unpaired-data and paired-data settings respectively. According to the big gap of the two different settings, we slightly adjust the models and propose extra novel loss functions for better performance.

4.1 Lab2Pix-V1

The Lab2Pix-V1 model is an end-to-end label-to-image network to synthesize a high-resolution photo-realistic image trained with unpaired data.

4.1.1 Model.

Our Lab2Pix-V1 model mainly consists of one generator and three independent discriminators. The generator G_{v1} is in essence a mapping function, which transfers a label map M of size $W \times H$ into an image X_i of size $W \times H$ finally. Inspired by the success of progressive generation scheme in other tasks [47], [48], our generator produces three different scale images in one forward process. The generation process can be defined as follows:

$$\mathrm{X}_i = \mathrm{G}_{v1}(\mathrm{z},\mathrm{M}), \quad i = 1, 2, 3 \tag{17}$$

where z is a 128-dim noise providing the style information of the image X_i, X_3 is the final $W \times H$ synthesized image, and X_1 and X_2 are synthesized images of lower resolutions. Specifically, the generator produces outputs of three scales in a coarse-to-fine manner to keep training stable when no paired data is provided. Note that the scale of X_{i+1} is as twice as X_i. We use the discriminators described in the previous sections directly, and we notate them as $\mathrm{D}_{v1\_i}(i = 1, 2, 3)$.

4.1.2 Auxiliary Loss Functions.

To help stabilize the unpaired-data training and help the model converge, we propose two novel auxiliary loss functions in Lab2Pix-V1.

Image Consistency Loss. In StackGAN++ [6], a color-consistency regularization, e.g., color mean value and covariance, is proposed to make sure the multi-scale generated samples are consistent. This constraint works for synthesizing a single object, but not for our case where images contain multiple objects with complex textures. In addition, as the resolution of synthesized image increases, the training process tends to be more unstable especially with unpaired-data training. Inspired by StackGAN++ [6], we postulate that if we keep the synthesized images at different scales with similar global structures and contents, the network will tend to be more stable. Consequently, we propose an image consistency loss to guarantee the similarity of the generated images in our unpaired-data model.

Specifically, the generator outputs X_i $(i = 1, 2, 3)$ at one time. We consider two adjacent outputs as a pair, and two pairs: $(\mathrm{X}_1, \mathrm{X}_2)$ and $(\mathrm{X}_2, \mathrm{X}_3)$ are acquired. We adopt a VGG16 [43] pre-trained on the ImageNet [44] to process each synthesized image to obtain five features respectively from 'Conv_{1_2}', 'Conv_{2_2}', 'Conv_{3_2}', 'Conv_{4_2}' and '$Conv_{5\_2}$'. Let $\Phi_l (l = 1, 2, 3, 4, 5)$ be the l-th output. The loss function can be described as:

$$\mathcal{L}_{v1\_3}(\mathrm{X}) = \sum_{i=1}^{2} \sum_l \|\Phi_l(\mathcal{P}_2(\mathrm{X}_{i+1})) - \Phi_l(\mathrm{X}_i)\|_2, \tag{18}$$

where X means the set of X_i and $\mathcal{P}_2$ indicates the pooling with stride 2.

Cycle Segmentation Loss. To support the unpaired-data training process where the input labels are not paired to the input images, we design a cycle segmentation loss. The training dataset consists of data from two domains: the label

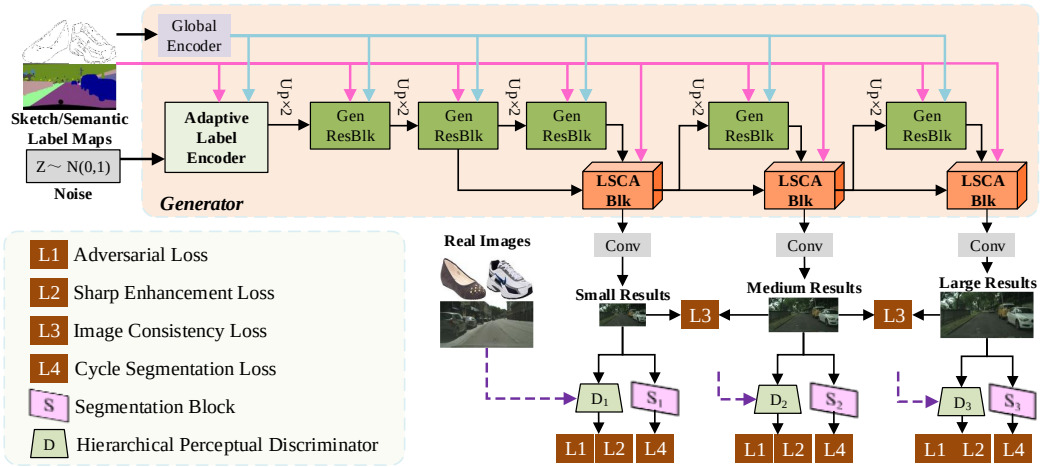


Fig. 7: The proposed unpaired-data Lab2Pix-V1 structure. It takes either a sketch label map or a semantic label map as input to produce photo-realistic images. The generator use an adaptive label encoder to separately encode the sketch and semantic label maps according to their characteristics, and gradually outputs higher-resolution (small, medium and large) images in one forward process. The structures of different images is guaranteed to be close by the image consistency loss, while the correspondence of the output image and input label is verified by the cycle segmentation loss.

map domain $\mathcal{M}_d$ and the image domain $\mathcal{Y}_d$. Our generator learns a mapping function $\mathrm{G}: \mathcal{M}_d \rightarrow \mathcal{Y}_d$, while we apply segmentation networks ICNet [49] to learn another mapping function $\mathrm{S}: \mathcal{Y}_d \rightarrow \mathcal{M}_d$. Since our generator progressively synthesizes images of three different scales, we apply three independent segmentation networks S_1, S_2 and S_3 to obtain their semantic maps or sketch maps. Consequently, our cycle segmentation loss is defined as:

$$\mathcal{L}_{v1\_4}(\mathrm{X},\mathrm{M}) = -\sum_i \frac{1}{HW} \sum_{h=1,w=1}^{H,W} \log \frac{e^{\mathrm{S}_i^{\overline{n},h,w}(\mathrm{X}_i)}}{\sum_{n=1}^{N} e^{\mathrm{S}_i^{n,h,w}(\mathrm{X}_i)}}, \quad (19)$$

where H, W is the height and width of the image. N is the class number of the whole dataset. $\mathrm{S}_i^{n,h,w}$ represents the output in position (h,w) of predicted class n. $\overline{n}$ is the correct class of pixel in position (h,w). For the sketch-to-image task, there are only two classes ($N=2$): sketch pixels and blank pixels.

4.1.3 Optimization.

With previously defined loss functions in Formulas (15), (16), (18) and (19), we obtain the overall loss function to optimize our network, which is expressed as follows:

$$\begin{aligned}\mathcal{L}_{v1}(\mathrm{X},\mathrm{Y},\mathrm{Y}^f,\mathrm{M}) = \quad & \mathcal{L}_{v1\_1}(\mathrm{X},\mathrm{Y}) + \lambda_2 \mathcal{L}_{v1\_2}(\mathrm{Y},\mathrm{Y}^f) \\ & + \lambda_3 \mathcal{L}_{v1\_3}(\mathrm{X}) + \lambda_4 \mathcal{L}_{v1\_4}(\mathrm{X},\mathrm{M}),\end{aligned} \quad (20)$$

where λ_2, λ_3 and λ_4 are weights for each auxiliary loss.

In addition, the whole network is required to learn parameters of G_{v1}, D_{v1_i} $(i=1,2,3)$ and S_i $(i=1,2,3)$. Therefore, we consider G_{v1} and S_i $(i=1,2,3)$ as net_1, and $\mathrm{D}_{v1\_i}(i=1,2,3)$ as net_2. When optimizing the parameters of net_1, the parameters of net_2 are fixed and vice versa. We train the network iteratively until convergence.

4.2 Lab2Pix-V2

The Lab2Pix-V2 model is trained in a paired-data manner, with paired data, where the given label map M and ground truth Y indicate the same semantic content.

4.2.1 Model.

Like Lab2Pix-V1, Lab2Pix-V2 consists of one generator and several independent discriminators, and its generator G_{v2} maps a label map M of size $W \times H$ to an image X_i of size $W \times H$ finally. The mapping function G_{v2} can be described as follows:

$$\mathrm{X} = \mathrm{G}_{v2}(\mathrm{z},\mathrm{M}). \quad (21)$$

Note that, the generator only produces one final image X of size $H \times W$ itself since the paired data guarantees the relative stability of training. It has been proved efficient [5] to discriminate images in multiple scales for high-resolution image generation. Thus, we downsample X with different kernel sizes to obtain smaller images X_1 and X_2, and we rename X to X_3 to maintain consistency in notations. The framework of Lab2Pix-V2 is illustrated in Fig. 8. Note that, we do not add the LSCA into this model on account of our limited memory when experimenting with the same training parameters with our competitors. The framework is illustrated in Fig. 8.

We use three independent discriminators (i.e., D_{v2_1}, D_{v2_2} and D_{v2_3}), which share similar structures with those in Lab2Pix-V1, to consider multi-scale information. However, to save memory, we address the label and image matching issue in Lab2Pix-V2 discriminators instead of additional loss functions. Specifically, we concatenate the images with label maps into the discriminators of Lab2Pix-V2, which is illustrated in Fig. 6. For each of the three discriminators D_{v2_i} $(i=1,2,3)$, the input of the discriminator can be expressed as:

$$\mathrm{X}_i^* = (\mathrm{X}_i,\mathrm{M}), \quad \mathrm{Y}_i^* = (\mathrm{Y}_i,\mathrm{M}). \quad (22)$$

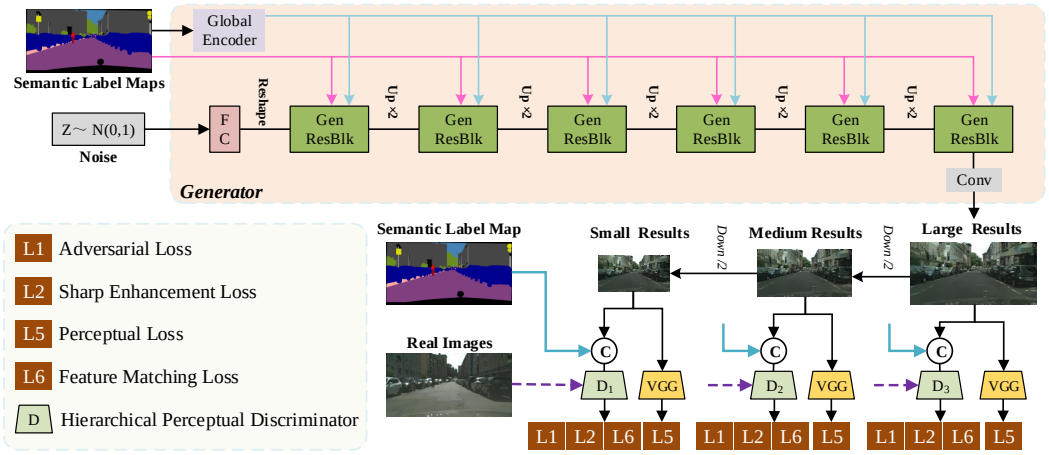


Fig. 8: The proposed paired-data Lab2Pix-V2 structure. The generator outputs one high-resolution image in one forward pass and we downsample it to obtain images in lower resolutions. The generated samples and real samples are concatenated with the label maps respectively before inputted to D_i. Note that, we give up LSCA owing to our limited hardware settings.

The three discriminators in our model are defined as:

$$\begin{aligned}&\mathrm{D}_{v2\_1}(\mathrm{X}_1^*) = \{\mathrm{D}^1_{v2\_1}(\mathrm{X}_1^*)\},\\ &\mathrm{D}_{v2\_2}(\mathrm{X}_2^*) = \{\mathrm{D}^1_{v2\_2}(\mathrm{X}_2^*), \mathrm{D}^2_{v2\_2}(\mathrm{X}_2^*)\},\\ &\mathrm{D}_{v2\_3}(\mathrm{X}_3^*) = \{\mathrm{D}^1_{v2\_3}(\mathrm{X}_3^*), \mathrm{D}^2_{v2\_3}(\mathrm{X}_3^*), \mathrm{D}^3_{v2\_3}(\mathrm{X}_3^*)\},\end{aligned} \tag{23}$$

where $\mathrm{D}^l_{v2\_i}$ indicates the output of D_{v2_i} on level l. Thus, the GAN loss function for the generator is:

$$\mathcal{L}^G_{v2\_1}(\mathrm{X}^*) = \sum_i \frac{1}{\sum_{l=1}^i \lambda_{il}} \sum_{l=1}^i \lambda_{il} (\mathbb{E}^{il}_{\mathcal{M}}[L^R_{adv}(\mathrm{D}^l_{v2\_i}(\mathrm{X}_i^*))]). \tag{24}$$

The GAN loss function for the discriminators can also be divided into two parts: real prediction loss and fake prediction loss. The discriminator loss can be expressed as:

$$\mathcal{L}^D_{v2\_1}(\mathrm{X}^*, \mathrm{Y}^*) = \frac{1}{2}(\mathcal{L}^D_{v2\_1}(\mathrm{Y}^*)_R + \mathcal{L}^D_{v2\_1}(\mathrm{X}^*)_F). \tag{25}$$

The fake prediction loss can be calculated as:

$$\mathcal{L}^D_{v2\_1}(\mathrm{X}^*)_F = \sum_i \frac{1}{\sum_{l=1}^i \lambda_{il}} \sum_{l=1}^i \lambda_{il} \mathbb{E}^{il}_{\mathcal{M}}[L^F_{adv}(\mathrm{D}^l_{v2\_i}(\mathrm{X}_i^*))]. \tag{26}$$

We calculate real prediction loss as:

$$\mathcal{L}^D_{v2\_1}(\mathrm{Y}^*)_R = \sum_i \frac{1}{\sum_{l=1}^i \lambda_{il}} \sum_{l=1}^i \lambda_{il} \mathbb{E}^{il}_{\mathcal{M}}[L^R_{adv}(\mathrm{D}^l_{v2\_i}(\mathrm{Y}_i^*))]. \tag{27}$$

We set all hyper parameters in the same way as Lab2Pix-V1.

4.2.2 Auxiliary Loss Functions.

To guarantee the quality of the synthesized images, we follow the previous work [3], [5] to use the perceptual loss $\mathcal{L}_{v2\_5}$ and the discriminator feature matching loss $\mathcal{L}_{v2\_6}$ in this paired-data architecture. The perceptual loss is defined as:

$$\mathcal{L}_{v2\_5}(\mathrm{X}, \mathrm{Y}) = \sum_i \sum_l \lambda_{Ml} \|\Phi_l(\mathrm{X}_i) - \Phi_l(\mathrm{Y}_i)\|_1, \tag{28}$$

where Φ_l $(l = 1, 2, 3, 4, 5)$ is the l-th output of the pretrained VGG19 network, and λ_{Ml} $(l = 1, 2, 3, 4, 5)$ is the weight for each part. We set $\lambda_{M1} = 1/32$, $\lambda_{M2} = 1/16$, $\lambda_{M3} = 1/8$, $\lambda_{M4} = 1/4$ and $\lambda_{M5} = 1$.

The discriminator feature matching loss can be expressed as:

$$\mathcal{L}_{v2\_6}(\mathrm{X}, \mathrm{Y}) = \sum_i \frac{1}{k} \sum_k \|\mathrm{D}^{(k)}_{v2\_i}(\mathrm{X}_i) - \mathrm{D}^{(k)}_{v2\_i}(\mathrm{Y}_i)\|_1, \tag{29}$$

where $\mathrm{D}^{(k)}_{v2\_i}$ indicates the k-th layer output of D_{v2_i}. The two loss functions optimize the distribution of generated samples to be close to that of real samples in different ways.

To match the paired-data setting, we slightly adjust the original Sharpness Enhancement Loss function described in the previous section. Specifically, we resize the real image samples to obtain the blurry ones and use them with unchanged labels to calculate the additional adversarial loss. The loss function is defined as:

$$\mathcal{L}_{v2\_2}(\mathrm{Y}^*, \mathrm{Y}^{f*}) = \frac{1}{2}(\mathcal{L}^D_{v2\_1}(\mathrm{Y}^*)_R + \mathcal{L}^D_{v2\_1}(\mathrm{Y}^{f*})_F), \tag{30}$$

where Y^{f*} is obtained in the same way as Y^*.

4.2.3 Optimization.

With the previous defined loss functions in Formulas (24), (25) and (30), we form the total loss to optimize our network, which can be described as:

$$\begin{aligned}\mathcal{L}_{v2}(\mathrm{X}, \mathrm{Y}, \mathrm{Y}^f, \mathrm{M}) =& \mathcal{L}_{v2\_1}(\mathrm{X}^*, \mathrm{Y}^*) + \lambda_2 \mathcal{L}_{v2\_2}(\mathrm{Y}^*, \mathrm{Y}^{f*})\\ &+ \lambda_5 \mathcal{L}_{v2\_5}(\mathrm{X}, \mathrm{Y}) + \lambda_6 \mathcal{L}_{v2\_6}(\mathrm{X}, \mathrm{Y})\end{aligned} \tag{31}$$

where λ_2, λ_5 and λ_6 are weights for each loss. In particular,

$$\mathcal{L}_{v2\_1}(\mathrm{X}^*, \mathrm{Y}^*) = \mathcal{L}^G_{v2\_1}(\mathrm{X}^*) + \mathcal{L}^D_{v2\_1}(\mathrm{X}^*, \mathrm{Y}^*) \tag{32}$$

Lab2Pix-V2 only consists of G_{v2} and D_{v2_i} $(i = 1, 2, 3)$. Therefore, we directly train the two adversarial components G_{v2} and $\mathrm{D}_{v2\_i} (i = 1, 2, 3)$ iteratively until convergence.

5 Experiments

5.1 Datasets

We evaluate our proposed methods on six publicly available datasets. These include four label-to-image datasets:

Fig. 9: Ablation study results on the Cityscapes dataset with unpaired-data training. Lab2Pix-V1 with all components obtains the best results.

Fig. 10: Ablation study results on the Cityscapes dataset with paired-data training. Lab2Pix-V2 with all components obtains the best results.

Cityscapes [50], COCO-Stuff [51], ADE20K [52], and Facades [53], and two sketch-to-image datasets: Edges2shoes and Edges2handbags (provided by Pix2Pix [8]). Cityscapes contains 2048×1024 resolution images recorded in street scenes from 50 different cities. Each street scene image is annotated with labels from 35 categorizes. It has $2,975$ samples for training and 500 for validation. All images are resized to 512×256 or 1024×512 through nearest neighbor interpolation. COCO-Stuff consists of various scene images with various resolutions. The objects in each image belong to 182 different categories. The dataset consists of $118,000$ training samples and $5,000$ validation samples. ADE20K is similar to COCO-Stuff. It defines 150 semantic categories and contains $20,210$ training samples and $2,000$ validation samples. Both images from COCO-Stuff and ADE20K are resized to 256×256 resolution. Facades contains 12 semantic labels and all images are 256×256. It has 400 training samples and 100 validation samples. The other two datasets are sketch datasets with image resolution 256×256. Edges2shoes has approximately $50,000$ samples with $49,825$ for training and 200 for validation, while Edges2handbags is the larger one with $138,567$ for training and 200 for validation.

5.2 Implementation details

Lab2Pix-V1 training. For the semantic label to image task, we set batch size $N = 1$, hyper-parameter $T = 2$ and epochs as 100. During the whole training process, we linearly increase λ_2 from zero to one. The learning rates are set as 0.0002 for the first 50 epochs, and then decay to 0 in the remaining 50 epochs. For the sketch label to image task, we train the network for 10 epochs with $N = 4$. λ_2 and the learning rate are set as 1 and 0.0002 respectively. In addition, for all experiments, we set $\lambda_3 = 5e^{-6}$ and $\lambda_4 = 10$. The Adam optimizer [54] is adopted with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and we choose vanilla adversarial loss calculation [7] for a better performance. All experiments are conducted on an NVIDIA Titan Xp GPU.

Lab2Pix-V2 training. We design 4 GenResBlks apart from those in the semantic label encoder and one downsample operation on X_3 to obain X_2 for 512×256 and 256×256. In the higher-resolution (1024×512) synthesis model, 5 additional GenResBlks and two downsample operations on X_3 are set to obtain X_1 and X_2. For the Cityscapes dataset, we set batch size $N = 20$ for 512×256 image synthesis, and $N = 8$ for 1024×512 image synthesis. We train the model for 200

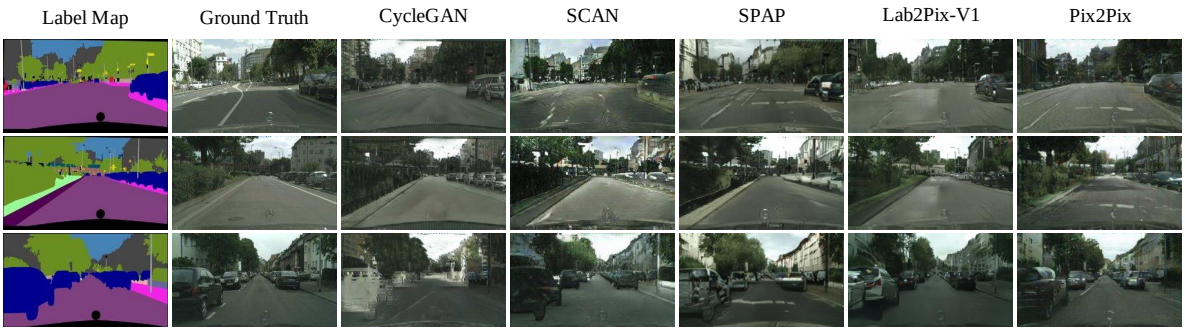


Fig. 11: Comparison on the Cityscapes dataset with unpaired-data training. Images of CycleGAN, SCAN and SPAP are from paper [4]. Our Lab2Pix-V1 generates more realistic images than other unpaired-data methods including CycleGAN, SCAN and SPAP. Note that Pix2Pix is one of the paired-data methods.

Fig. 12: Comparison on the Edges2shoes and Edges2-handbags datasets with unpaired-data training.

epochs. For the COCO-Stuff and ADE20K datasets, we set batch size $N = 40$ for 256×256 image generation. The model is optimized for 100 epochs. The learning rates for all networks are set as 0.0002 at first and linearly decay to 0 in the last half training epochs. We set hyper-parameters $T = 5$, $\lambda_2 = 1$, $\lambda_5 = 10$ and $\lambda_6 = 10$. The Adam optimizer [54] is adopted in all experiments with $\beta_1 = 0$ and $\beta_2 = 0.9$, and we choose hinge version adversarial loss calculation [46] for a stable training. All experiments are conducted on 4 NVIDIA Tesla V100 GPUs.

5.3 Evaluation Metrics

Following previous works [3], [4], [5], [9], [10], we adopt segmentation networks to obtain three standard segmentation metric scores to evaluate our method on the semantic label to image task. They are per-pixel accuracy (PPA), per-class accuracy (PCA) and mean class IoU (C-IoU) which are usually named as FCN scores. Specifically, we feed the generated samples to the segmentation networks and compare the predictions with the original labels. We adopt FCN [55] for unpaired-data learning on cityscapes dataset, DRN-D-105 [56] for paired-data learning on cityscapes dataset, DeepLabV2 [57] for COCO-Stuff dataset and UperNet101 [58] for ADE20K dataset. To ensure efficiency, all segmentation networks have been pre-trained on the corresponding datasets. Besides, FID [59] is also used with Inception V3 model [60] in our evaluation, which can measure the distance between the generated and real samples in terms of data distributions. It is widely adopted in various image synthesis tasks, including single-object synthesis [2], [6] and multi-object synthesis [3], [23]. Thus, we apply FID to all but Facades dataset. For the Facades dataset, we utilize PSNR and SSIM [61] that are also used in previous works [4], [9], [10]. The PSNR value represents the disparity between the fake and real samples on the pixel level, while SSIM measures the content variation in terms of image luminance, contrast and structure. Compared with PSNR, SSIM works better in evaluating the quality of the synthesized image. Note that our models are not sensitive to random seeds. Thus, we only report each score with single stable value for each experiment following competitors.

TABLE 1: Ablation study of the proposed unpaired-data Lab2Pix-V1 on the Cityscapes dataset. FCN scores are obtained with FCN [55].

baseline	setup	PPA	PCA	C-IoU	FID
A_u	SPADE w/ $\mathcal{L}_4$	0.48	0.09	0.08	133.2
B_u	A_u w/ PG	0.33	0.11	0.08	117.7
C_u	B_u w/ LSCA	0.48	0.15	0.12	83.5
D_u	C_u w/ $\mathcal{L}_3$	0.54	0.18	0.14	83.4
E_u	D_u w/ HPD	0.64	0.22	0.18	76.0
F_u	E_u w/ FE	0.69	0.23	0.18	67.9
G_u	F_u w/ $\mathcal{L}_2$	0.77	0.22	0.18	65.0
H_u	Lab2Pix-V1	0.76	0.25	0.20	67.4

5.4 Ablation Study

We conduct ablation studies on both the paired-data and unpaired-data settings to fully demonstrate the advantage of our proposed components and loss functions. We first introduce the Baseline Models that we use.

For Lab2Pix-V1, we set SPADE with proposed cycle segmentation loss ($\mathcal{L}_4$) as the basic baseline A_u and progres-

Fig. 13: Comparison on the Cityscapes dataset with paired-data training.

TABLE 2: Ablation study of the proposed unpaired-data Lab2Pix-V1 and paired-data Lab2Pix-V2 on Cityscapes dataset. FCN scores are obtained with DRN-D-105 [56].

baseline	setup	PPA	PCA	C-IoU	FID
A_s	SPADE	0.930	0.681	0.592	57.4
B_s	A_s w/ HPD	0.934	0.711	0.621	49.2
C_s	B_s w/ FE	0.935	0.727	0.636	48.0
D_s	C_s w/ $\mathcal{L}_2$	0.934	0.731	0.639	46.0
E_s	Lab2Pix-V2	0.936	0.738	0.646	45.5

sively add each component to obtain Lab2Pix-V1. Totally, seven baselines (i.e., A_u, B_u, C_u, D_u, E_u, F_u and G_u) are constructed, and Lab2Pix-V1 is denoted as H_u. Based on A_u, we make B_u by adopting the progressive generation scheme. We add LSCA to B_u and obtain C_u. Then, $\mathcal{L}_3$ is further added to compose baseline D_u. We equip D_u with our hierarchical perceptual discriminators (HPD) to make baseline E_u. With foreground enhancement masks (FE) added, baseline F_u is constructed based on E_u. Baseline G_u adds the sharpness enhancement loss ($\mathcal{L}_2$) to F_u. Finally, we add DG-Norm to G_u and obtain the full model H_u. We show quantitative results in Tab. 1, and demonstrate the qualitative results in Fig. 9.

For Lab2Pix-V2, we take SPADE as the basic baseline A_s and gradually add each component to the model to obtain Lab2Pix-V2. We add our hierarchical perceptual discriminators (HPD) to A_s to obtain B_s. Then, foreground enhancement masks (FE) are further added to train the baseline C_s. Baseline D_s adds the sharpness enhancement loss ($\mathcal{L}_2$) to C_s. Finally, with the global encoder (GE) and DG-Norm, the full model Lab2Pix-V2 is denoted by E_s. The quantitative results are shown in Tab. 2, and the qualitative results are demonstrated in Fig. 10.

We can make the following observations from the ablation analysis results from Tab. 1 and Tab. 2. Comparing A_u with B_u which obtain comparable metrics, naively adding progressive generation scheme to the Lab2Pix model is incapable of improving performance. We observe that pure progressive generation training of Lab2Pix will be unstable. For all metrics, baseline C_u is lower than the baseline D_u, which confirms the importance of $\mathcal{L}_3$. Also, as we can see in Fig. 9, the generated samples without $\mathcal{L}_3$ are significantly worse than those with $\mathcal{L}_3$. These pictures are rendered in an unreasonable way, which proves $\mathcal{L}_3$ can stabilize the synthesis process when the progressive generation scheme is adopted. Comparing D_u with E_u and A_s with B_s, baseline models with HPD (i.e., E_u and B_s) significantly decrease the PPA scores. This means that HPD tends to improve the quality of generation especially for those relative large objects. Baseline G_u outperforms F_u and D_s outperforms C_s in almost every metric, which indicates $\mathcal{L}_2$ improves the entire structure of images as it is designed to. Take row one in Fig. 9 as an example, the vehicles synthesized by G_u are clear and easy to recognize. If we remove $\mathcal{L}_2$ or HPD, these parts become indistinct. Row three in Fig. 10 confirms this again in the paired-data setting. Compared with baseline F_u, which is equipped with the foreground enhancement mask FE, baseline E_u increases its FID, which indicates that FE works effectively in unpaired-data learning. As for paired-data learning, baseline B_s, which is not equipped with FE, obtains lower scores in all metrics compared with C_s, which is equipped with FE. Especially, the 16% gap in PCA and 15% gap in C-IoU are the biggest among all adjacent baselines. This proves that FE does improve the generation of foreground objects. Take row five of Fig. 9 and row three of Fig. 10 as examples, the removal of FE leads to the dim boundary between foreground and background objects. Recall that the baselines G_u and D_s are not equipped with DG-Norm while H_u and E_s are. From comparison of D_s with E_s, and G_u with H_u, we can conclude that our proposed DG-Norm increases model performance, especially in terms of PCA and C-IoU which reflect the quality of relative small and complex foreground objects.

Fig. 14: Comparison on the Cityscapes dataset with paired-data training. In each sample set, the left six images from top left to bottom right are the label map (a), the ground truth (b), the result of SPADE (c) , CC-FPSE (d), TSIT (e) and Lab2Pix-V2 (f) in 512×256 respectively. The right large image is the result of Lab2Pix-V2-H (g) in 1024×512.

TABLE 3: Quantitative results of different methods on Cityscapes and Facades datasets. For all metrics, higher is better. Note that Pix2Pix is a paired-data method.

Method	PPA	PCA	C-IoU	PSNR	SSIM
Dataset	Cityscapes			Facades	
CycleGAN	0.52	0.17	0.11	11.72	0.20
SCAN	0.64	0.20	0.16	10.67	0.17
SPAP	0.73	0.22	0.17	12.20	0.21
Lab2Pix-V1	0.76	0.25	0.20	11.85	0.28
Pix2Pix	0.71	0.25	0.18	-	-

TABLE 4: Quantitative results of different methods on the Edges2shoes and Edges2handbags datasets. Memory usage is measured with batch size 4. Time usage ratio indicates the entire training time normalized by that of Lab2Pix-V1. Lab2Pix-V1* is trained for ten epochs while all the other models are trained for five epochs.

Method	FID		Memory(GB)	Time
Datasets	shoes	handbags	Training Info	
CycleGAN	137.9	98.0	11	1.4
MUNIT	105.5	87.4	12	2.8
Lab2Pix-V1	100.1	81.1	8	1.0
Lab2Pix-V1*	76.7	78.7	8	2.0

In summary, by observing all metrics in Tab. 1 and Tab. 2 and qualitative comparison results in Fig. 9 and Fig. 10, the results clearly demonstrate the effectiveness of our proposed Lab2Pix-V1 and Lab2Pix-V2 models in generating natural photo-realistic images in both paired-data and unpaired-data settings. In detail, LSCA and image consistency loss ($\mathcal{L}_3$) are the most essential for Lab2Pix-V1, while HPD and foreground enhancement mask make the most important contributions to Lab2Pix-V2. Note that, if we remove the cycle segmentation loss ($\mathcal{L}_4$) which is used to bring the layout of input label and output image closer, training will not converge. Because the generated images are not required to be similar to the input labels when lacking the necessary constraint of this loss function.

5.5 Comparison with State-of-the-art Models

For the semantic label to image task, we compare our Lab2Pix models with six state-of-the-art methods on the Cityscapes, COCO-Stuff, ADE20K and Facades datasets. The baseline models include three unpaired-data methods CycleGAN [10], SCAN [9] and SPAP [4], and five paired-data methods Pix2Pix [8], Pix2PixHD [5], SPADE [3], CC-FPSE [38] and TSIT [39]. As for the sketch label to image task, CycleGAN [10] and MUNIT [12] are chosen as the baseline models.

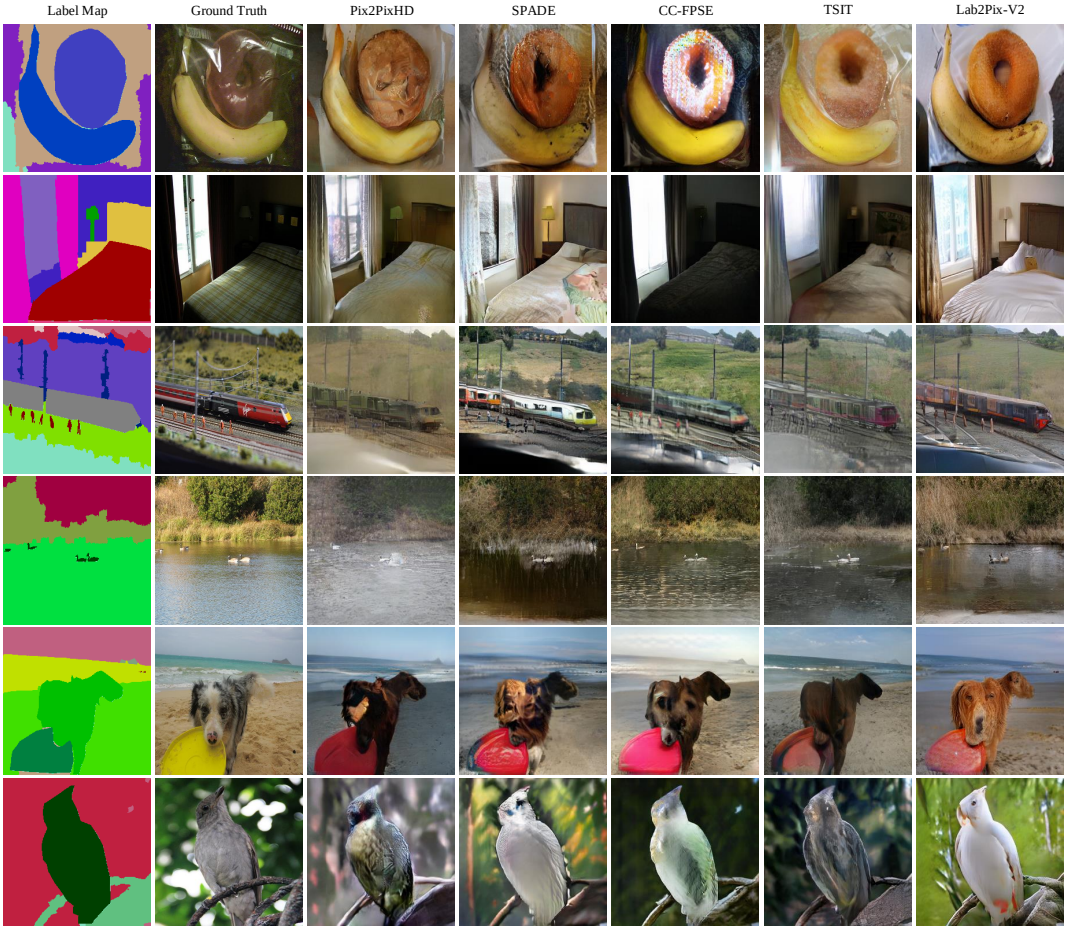


Fig. 15: Comparison on the COCO-Stuff dataset with paired-data training.

5.5.1 Lab2Pix-V1 Comparison

We choose three unpaired-data competitors: CycleGAN [10], SCAN [9], SPAP [4] and one paired-data method Pix2Pix [8] for Lab2Pix-V1 on two label-to-image datasets: Cityscapes [50] and Facades [53]. CycleGAN [10] and MUNIT [12] are chosen to compare with Lab2Pix-V1 on two sketch-to-image datasets: Edges2shoes and Edges2Handbags. The quantitative results are presented in Tab. 3 and Tab. 4, and the qualitative results are shown in Fig. 11 and Fig. 12.

For the label-to-image task, the results are shown in Tab. 3. We can observe the significant improvements of Lab2Pix-V1 over the baselines. It consistently outperforms all unpaired-data methods on all the metrics except for the PSNR score. However, it is widely agreed that SSIM is a better metric than PSNR in terms of evaluating image quality. More specifically, compared with SPAP, the current best unpaired-data model, our Lab2Pix-V1 outperforms it by 0.03, 0.03, 0.03 and 0.07 for PPA, PCA, C-IoU and SSIM, respectively. Compared with the paired-data method Pix2Pix, we surpass it by 0.05 on PPA and 0.02 on C-IoU, and achieve the same PCA score. These results verify the advantage of our proposed method. In addition, Fig. 11 shows a qualitative comparison with these methods on the Cityscapes dataset. Compared with SPAP, for instance, Lab2Pix-V1 generates more photo-realistic images, which render sharper boundaries for adjacent objects (e.g., vehicles) and more natural textures and details for objects.

For the sketch-to-image task, a comparison is conducted on the Edges2shoes and Edges2handbags datasets. The quantitative results are shown in Tab. 4. For a fair comparison, we also report training information about memory and time cost. Both CycleGAN and MUNIT are trained on the same datasets with officially recommended training settings except that we keep the same batch size (i.e., 4) with Lab2Pix-V1. For CycleGAN, MUNIT and Lab2Pix-V1, training epochs are all set as five, while Lab2Pix-V1* increases the training epochs to ten. From Tab. 4, we make the following observations by comparing with the current best model MUNIT. Firstly, our Lab2Pix-V1 model requires less computational resources (8GiB vs. 12GiB). Secondly, our Lab2Pix-V1 model achieves slightly better FID scores on both datasets, but significantly decreases the training time. Specifically, the training time of Lab2Pix-V1 is only approximately 36% of that of MUNIT. Thirdly, trained for twice the number of epochs than Lab2Pix-V1, our Lab2Pix-V1* model reaches the best FID scores on both datasets, but still uses less memory and training time than MUNIT. All quantitative results demonstrate that our method achieves the best trade-off between quality and efficiency (both time and memory cost). Furthermore, qualitative synthesized examples are shown in Fig. 12, which demonstrate that our proposed methods are able to generate photo-realistic images with convincing natural textures and

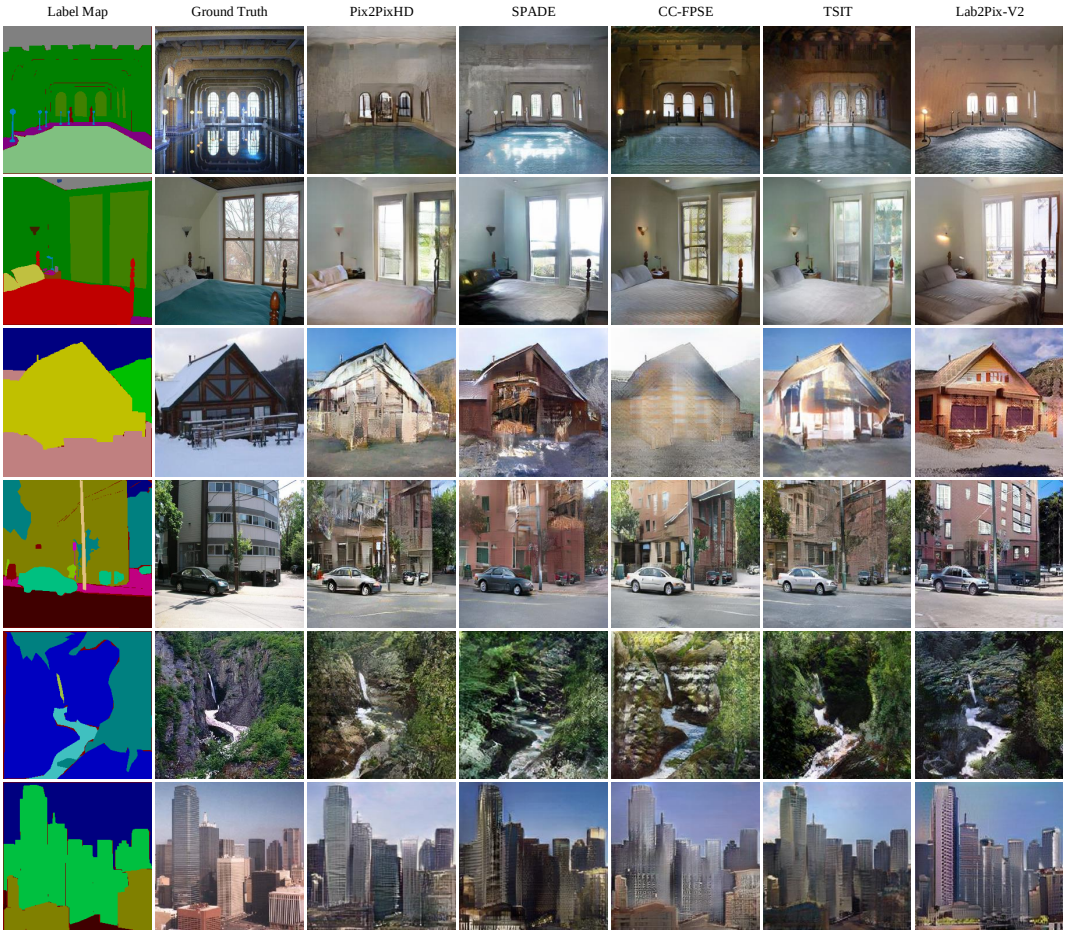


Fig. 16: Comparison on the Ade20K dataset with paired-data training.

details about shoes and bags, showing better performance than the baselines.

5.5.2 Lab2Pix-V2 Comparison

We compare our Lab2Pix-V2 model with Pix2PixHD [5], SPADE [3], CC-FPSE [38] and TSIT [39] on three datasets: Cityscapes [50], COCO-Stuff [51] and ADE20K [52]. The quantitative results are presented in Tab. 5. The qualitative results for the three datasets are respectively shown in Fig. 13, Fig. 15 and Fig. 16. More randomly selected samples are attached to the supplementary material. As can be observed in Tab. 5, Lab2Pix-V2 obtains the best performance in FID on all three datasets. Besides, we can observe that our Lab2Pix-V2, CC-FPSE and TSIT all obtain larger values in almost all FCN scores compared to the ground truth on all datasets. However, real samples set the standard of realistic and label-matched results, which makes an apparent conflict with the traditional evaluation strategy that "higher scores indicate better performance". We think higher FCN scores only indicate these methods generate samples easier for segmentation networks to make pixel-level classification, instead of they synthesize images in better image quality or matching input labels better. To address this issue, we add a user study to evaluate our method.

On the Cityscapes dataset, our method obtains comparable FCN scores with CC-FPSE and TSIT, and significantly outperforms other methods. Specifically, it surpasses baseline SPADE by 2.4% on PCA and 2.4% on C-IoU. Besides, our Lab2Pix-V2 model achieves significantly better FID values than all other methods. Around 15 point reduction in FID is made by Lab2Pix-V2 compared with the baseline model SPADE. The qualitative comparison shown in Fig. 13 also confirms our improvements. For instance, in rows four and eight, only Lab2Pix-V2 synthesizes clear and realistic textures of both vehicles and buildings in these images. In Fig. 14, we give samples generated by Lab2Pix-V2-H. We can observe that with our proposed components, the images in higher resolution give more detailed textures, especially in those areas with small foreground objects.

As for COCO-Stuff, our Lab2Pix-V2 model achieves comparable performances on FCN scores compared with SPADE and CC-FPSE, and obtains 4.5%, 6.2% and 5.3% higher FCN scores compared with TSIT. Meanwhile, 5.4 reduction in FID compared with the baseline SPADE shows our Lab2Pix-V2 is able to synthesize more realistic images. In Fig. 15, Lab2Pix-V2 can generate both simple and complex scenes well. For example, in the first and last two rows, our Lab2Pix-V2 model is able to generate reasonable and clear object textures in the simple synthesis of one or two main objects. In terms of complex multi-object scenes, the generated samples of Lab2Pix-V2 show sharp boundaries between each two neighboring instances, and clear and vivid objects.

In experiments on ADE20K, our Lab2Pix-V2 obtains

TABLE 5: Quantitative results of different methods on Cityscapes, COCO-Stuff and ADE20K datasets. We demonstrate the metrics obtained by real images for reference. Each method is trained with three different random seeds, and each test sample for each trained model is inferred three times with different random noises.

Method	PPA(%)	PCA(%)	C-IoU(%)	FID
	Cityscapes			
Ground Truth	93.0	72.7	62.1	-
Pix2PixHD	92.2±0.2	64.4±1.5	55.5±1.6	70.3±3.8
SPADE	93.4±0.1	70.4±0.6	61.2±0.6	62.3±0.9
CC-FPSE	93.5±0.1	72.5±0.9	63.0±0.8	54.8±0.7
TSIT	93.6±0.1	71.5±0.1	62.4±0.3	85.9±0.9
Lab2Pix-V2	93.6±0.0	72.8±0.5	63.6±0.6	47.7±1.2
	COCO-Stuff			
Ground Truth	59.6	39.3	28.3	-
Pix2PixHD	56.4±0.1	32.0±0.2	22.8±0.1	25.7±0.3
SPADE	61.6±0.5	38.1±0.4	28.0±0.4	22.2±0.6
CC-FPSE	63.6±0.3	41.6±0.1	30.9±0.1	17.8±0.2
TSIT	58.4±0.1	34.7±0.1	25.2±0.2	22.1±0.9
Lab2Pix-V2	62.9±0.3	40.9±0.6	30.5±0.5	16.8±0.5
	ADE20K			
Ground Truth	77.3	44.8	33.6	-
Pix2PixHD	78.2±0.9	35.8±0.4	32.5±0.4	38.2±0.1
SPADE	81.2±0.1	48.0±0.6	41.0±0.3	34.3±0.3
CC-FPSE	82.3±0.3	50.3±0.8	42.9±0.7	33.2±0.9
TSIT	79.8±0.2	43.4±0.3	36.9±0.2	35.3±0.6
Lab2Pix-V2	80.8±0.4	47.7±0.1	40.7±0.2	28.2±0.5

comparable FCN scores but significantly better FID (i.e., over 5 point reduction) compared with all other methods. As we can observe in Fig. 16, in the first two rows, Lab2Pix-V2 can synthesize more detailed textures (e.g., ripples of water, folds on bed) in indoor scenes. For outdoor scenes that are presented in the last four rows, Lab2Pix-V2 paints recognizable appearances on each object while keeping their shapes strictly following the input label maps.

In the user study, we invite 10 participators which are normal people without professional skills in image processing and recognition for each dataset, and we randomly choose 100 samples for each participator. Each sample contains synthesized images of all five methods from the same input label map. We also provide the corresponding input label and ground truth image for reference. The generated images in each sample are randomly shuffled for a fair comparison. Participators are asked to rank the five images in each sample according to the image quality and the matching degree with the input label with no time limitation. Approaches will obtain from 5 to 1 point as they are chosen from the best to the worst. We demonstrate the mean scores obtained by all methods on each dataset and the Top1 and Top2 rates in Tab.6. It can be observed that images synthesized by Lab2Pix-V2 are preferred on all datasets, which verifies the superiority of our method.

5.6 Failure Examples

In Fig. 17, we give some failure examples generated by our Lab2Pix-V2 model. We can observe that those objects in multivariate shapes (e.g., various vehicles, trains and pedestrians) are not be synthesized well. These instances are usually relatively small and rare foreground objects that are able to move or turn themselves. For example, face generation is an independent generation task addressed by many works [47], [62]. We will direct our future work towards these issues by investigating solutions for more fine-grained synthesis.

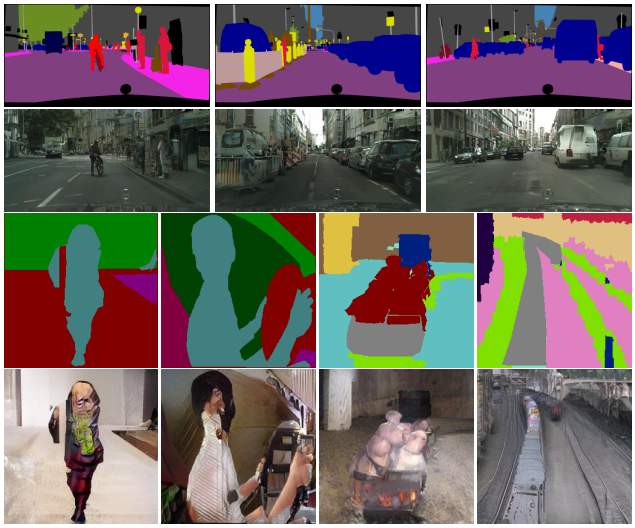


Fig. 17: Some failure examples in our Lab2Pix-V2 experiments on the Cityscapes, COCO-Stuff and ADE20K datasets.

6 Conclusion

In this paper, we propose a novel end-to-end, GAN-based framework, Lab2Pix, for the label to image synthesis task. Recognizing the large gap between label maps and real images and challenges in both unpaired and paired data settings, we propose effective models for both settings, namely Lab2Pix-V1 and Lab2Pix-V2. Specifically, for the generator, we design the Label Guided Spatial Co-Attention (LSCA) to gradually refine synthesized images under limited parameters, and Double-Guided Normalization (DG-Norm) to combine local and global characterizations for generation. To encourage realistic images generation, the discriminators are designed in a hierarchical architecture to discriminate image contents in different sizes and complex textures with foreground enhancement masks leading to focus on challenging foreground objects. In addition, the sharpness enhancement loss is proposed to better constrain the network to generating high-resolution realistic images. Our Lab2Pix framework obtains the state-of-the-art performances on six public datasets both qualitatively and quantitatively, outperforming current best models in both unpaired and paired data settings. We will study fine-grained texture generation as a future work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62020106008, No. 62122018, No. 61772116, No. 61872064, No. 61871470).

References

[1] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in ICCV, 2017, pp. 1511–1520.
[2] L. Gao, D. Chen, J. Song, X. Xu, D. Zhang, and H. T. Shen, "Perceptual pyramid adversarial networks for text-to-image synthesis," in AAAI, vol. 33, no. 01, Jul. 2019, pp. 8312–8319.

TABLE 6: Results of the user study on Cityscapes, COCO-Stuff and ADE20K datasets. Higher score indicates the results of this method are considered to have higher quality by human. We also provide the Top1 and Top2 (chosen as the best and the best two samples) rates of each method in the user study.

Method	Mean Score	Top1(%)	Top2(%)	Mean Score	Top1(%)	Top2(%)	Mean Score	Top1(%)	Top2(%)
	Cityscapes			COCO-Stuff			ADE20K		
Pix2PixHD	2.50±1.27	8.5	21.5	3.00±1.36	9.4	21.9	2.63±1.77	12.3	27.9
SPADE	2.81±1.42	13.6	28.6	3.10±1.47	15.2	26.5	2.84±1.76	15.4	32.8
CC-FPSE	3.41±1.59	28.2	50.4	3.16±1.57	25.6	49.3	3.12±1.96	22.8	43.9
TSIT	2.61±1.50	13.0	25.2	3.10±1.66	18.8	35.0	3.02±1.81	18.5	39.6
Lab2Pix-V2	3.67±1.66	36.7	62.0	3.21±1.93	30.9	47.1	3.39±1.98	31.1	51.6

[3] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in CVPR, 2019.

[4] W. Sun and T. Wu, "Learning spatial pyramid attentive pooling in image synthesis and image-to-image translation," arXiv preprint arXiv:1901.06322, 2019.

[5] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in CVPR, 2018, pp. 8798–8807.

[6] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 8, pp. 1947–1962, 2018.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in NeurIPS, 2014, pp. 2672–2680.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in CVPR, 2017.

[9] M. Li, H. Huang, L. Ma, W. Liu, T. Zhang, and Y. Jiang, "Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks," in ECCV, 2018, pp. 184–199.

[10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in ICCV, 2017, pp. 2223–2232.

[11] D. E. van der Ouderaa, Worrall, "Reversible gans for memory-efficient image-to-image translation," in CVPR, 2019.

[12] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in ECCV, 2018, pp. 172–189.

[13] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in NeurIPS, 2017, pp. 700–708.

[14] E. L. Denton, S. Chintala, R. Fergus et al., "Deep generative image models using a laplacian pyramid of adversarial networks," in NeurIPS, 2015, pp. 1486–1494.

[15] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in CVPR, 2016, pp. 2414–2423.

[16] L. Gao, J. Zhu, J. Song, F. Zheng, and H. T. Shen, "Lab2pix: Label-adaptive generative adversarial network for unsupervised image synthesis," in ACM MM, 2020.

[17] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.

[18] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in ICLR, 2019.

[19] W. Chen and J. Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," in CVPR, June 2018.

[20] Y. Lu, S. Wu, Y.-W. Tai, and C.-K. Tang, "Image generation from sketch constraint using contextual gan," in ECCV, 2018, pp. 205–220.

[21] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Manigan: Text-guided image manipulation," in CVPR, 2020.

[22] S. Nam, Y. Kim, and S. J. Kim, "Text-adaptive generative adversarial networks: manipulating images with natural language," in NeurIPS, 2018, pp. 42–51.

[23] W. Sun and T. Wu, "Image synthesis from reconfigurable layout and style," in ICCV, 2019, pp. 10 531–10 540.

[24] Y. Li, Y. Cheng, Z. Gan, L. Yu, L. Wang, and J. Liu, "Bachgan: High-resolution image synthesis from salient object layout," in CVPR, 2020.

[25] J. Song, J. Zhang, L. Gao, X. Liu, and H. T. Shen, "Dual conditional gans for face aging and rejuvenation," in IJCAI, 2018, pp. 899–905.

[26] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in CVPR, 2018, pp. 8789–8797.

[27] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," in ICLR, 2017.

[28] H. Shu, Y. Wang, X. Jia, K. Han, H. Chen, C. Xu, Q. Tian, and C. Xu, "Co-evolutionary compression for unpaired image translation," in ICCV, 2019, pp. 3235–3244.

[29] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in ICCV, 2017, pp. 2849–2857.

[30] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in ICML, 2017, pp. 1857–1865.

[31] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in ICCV, 2017, pp. 1510–1519.

[32] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in ECCV, 2016, pp. 694–711.

[33] S. Ma, J. Fu, C. W. Chen, and T. Mei, "DA-GAN: instance-level image translation by deep attention generative adversarial networks," in CVPR, 2018, pp. 5657–5666.

[34] S. Mo, M. Cho, and J. Shin, "Instagan: Instance-aware image-to-image translation," in ICLR, 2019.

[35] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in CVPR, 2017, pp. 105–114.

[36] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: enhanced super-resolution generative adversarial networks," in ECCV, 2018, pp. 63–79.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in MICCAI. Springer, 2015, pp. 234–241.

[38] X. Liu, G. Yin, J. Shao, X. Wang, and H. Li, "Learning to predict layout-to-image conditional convolutions for semantic image synthesis," in NeurIPS, 2019.

[39] L. Jiang, C. Zhang, M. Huang, C. Liu, J. Shi, and C. C. Loy, "Tsit: A simple and versatile framework for image-to-image translation," in ECCV, 2020.

[40] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," Distill, vol. 1, no. 10, p. e3, 2016.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in CVPR, 2015, pp. 1–9.

[42] D. Sungatullina, E. Zakharov, D. Ulyanov, and V. Lempitsky, "Image manipulation with perceptual discriminators," in ECCV, 2018, pp. 579–595.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in ICLR, May 2015.

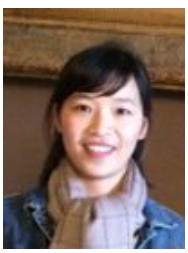
[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in CVPR. Ieee, 2009, pp. 248–255.

[45] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in ICML, 2019, pp. 7354–7363.

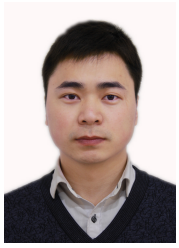
[46] J. H. Lim and J. C. Ye, "Geometric gan," arXiv preprint arXiv:1705.02894, 2017.

[47] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in CVPR, 2019, pp. 4401–4410.
[48] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in ICLR, 2018.
[49] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," arXiv preprint arXiv:1704.08545, 2017.
[50] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in CVPR, 2016.
[51] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in CVPR, 2018.
[52] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in CVPR, 2017.
[53] R. Tyleček and R. Šára, "Spatial pattern templates for recognition of objects with regular structure," in German Conference on Pattern Recognition (GCPR). Springer, 2013, pp. 364–374.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[55] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in CVPR, 2015, pp. 3431–3440.
[56] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in CVPR, 2017, pp. 472–480.
[57] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," IEEE TPAMI, vol. 40, no. 4, pp. 834–848, 2017.
[58] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in ECCV, 2018, pp. 418–434.
[59] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in NeurIPS, 2017, pp. 6626–6637.
[60] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in CVPR, 2016, pp. 2818–2826.
[61] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli et al., "Image quality assessment: from error visibility to structural similarity," IEEE TIP, vol. 13, no. 4, pp. 600–612, 2004.
[62] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in CVPR, 2020, pp. 8107–8116.

Junchen Zhu is working toward the Ph.D. degree in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include image, video, and 3D model synthesis and adversarial learning in computer vision and deep learning.



Lianli Gao (Member, IEEE) received the Ph.D. degree in information technology from The University of Queensland (UQ), Brisbane, QLD, Australia, in 2015. She is currently a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. She is focusing on integrating natural language for visual content understanding. Dr. Gao was the winner of the IEEE Trans. on Multimedia 2020 Prize Paper Award, the Best Student Paper Award in the Australian Database Conference, Australia, in 2017, the IEEE TCMC Rising Star Award in 2020, and the ALIBABA Academic Young Fellow.



Jingkuan Song (Senior Member, IEEE) is currently a Professor with the University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include large-scale multimedia retrieval, image/video segmentation and image/video understanding using hashing, graph learning, and deep learning techniques. Dr. Song has been an AC/SPC/PC Member of IEEE Conference on Computer Vision and Pattern Recognition for the term 2018–2021, and so on. He was the winner of the Best Paper Award in International Conference on Pattern Recognition, Mexico, in 2016, the Best Student Paper Award in Australian Database Conference, Australia, in 2017, and the Best Paper Honorable Mention Award, Japan, in 2017.

Yuanfang Li is a Senior Lecturer at Faculty of Information Technology, Monash University, Australia. He received his Ph.D. in computer science from National University of Singapore in 2006. His research interests include knowledge graphs, knowledge representation and reasoning, ontology languages, and software engineering.

Feng Zheng (Member, IEEE) received the Ph.D. degree from The University of Sheffield, U.K. He is currently an Assistant Professor with the Southern University of Science and Technology (SUSTech), China. His research interests include machine learning (ML), computer vision (CV), and human–computer interaction (HCI).

Xuelong Li (M'02-SM'07-F'12) is a full professor with School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P.R. China.

Heng Tao Shen (Fellow, IEEE) is the Dean of School of Computer Science and Engineering, the Executive Dean of AI Research Institute at University of Electronic Science and Technology of China (UESTC). He obtained his BSc with 1st class Honours and PhD from Department of Computer Science, National University of Singapore in 2000 and 2004 respectively. His research interests mainly include Multimedia Search, Computer Vision, Artificial Intelligence, and Big Data Management. He is/was an Associate Editor of ACM Transactions of Data Science, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, IEEE Transactions on Knowledge and Data Engineering, and Pattern Recognition. He is a Member of Academia Europaea, Fellow of ACM, IEEE and OSA.

Appendix
Randomly Chosen Samples for Comparison

In this part, we provide extra randomly sampled images to demonstrate the better performance of our Lab2Pix-V2.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3186752

Fig. 18: Comparison on Cityscapes dataset. Samples are randomly chosen.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3186752

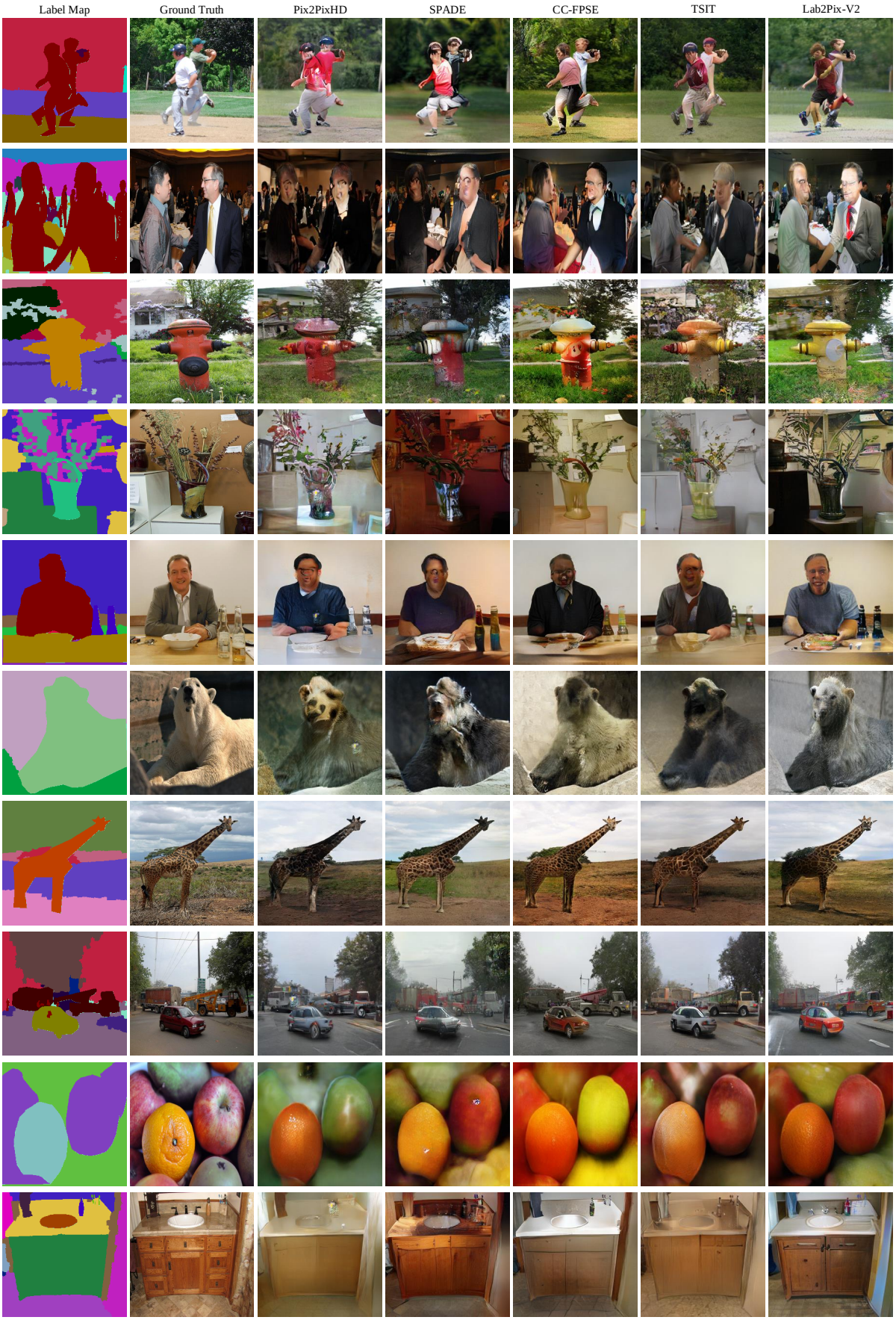


Fig. 19: Comparison on COCO-Stuff dataset. Samples are randomly chosen.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3186752

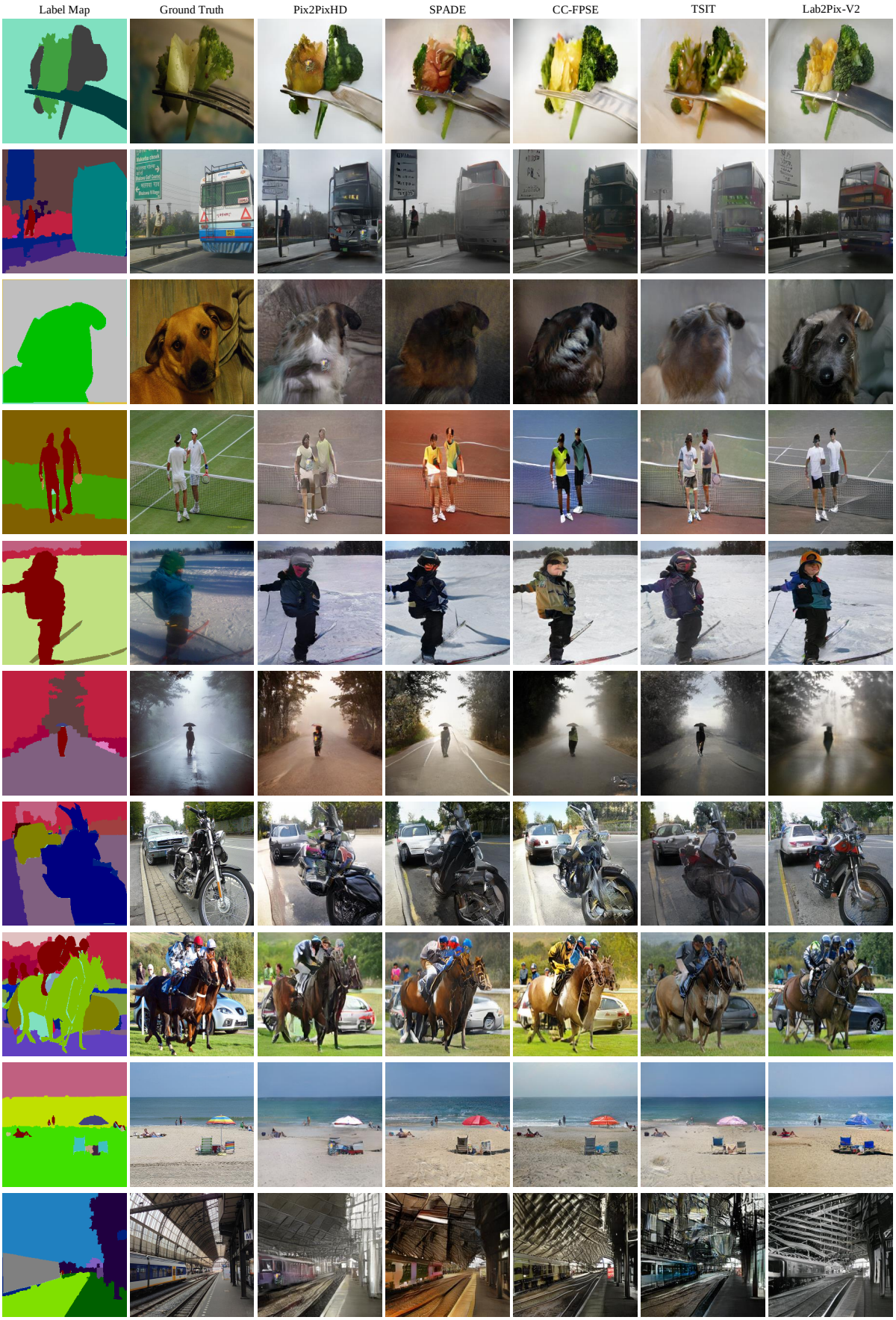


Fig. 20: Comparison on COCO-Stuff dataset. Samples are randomly chosen.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3186752

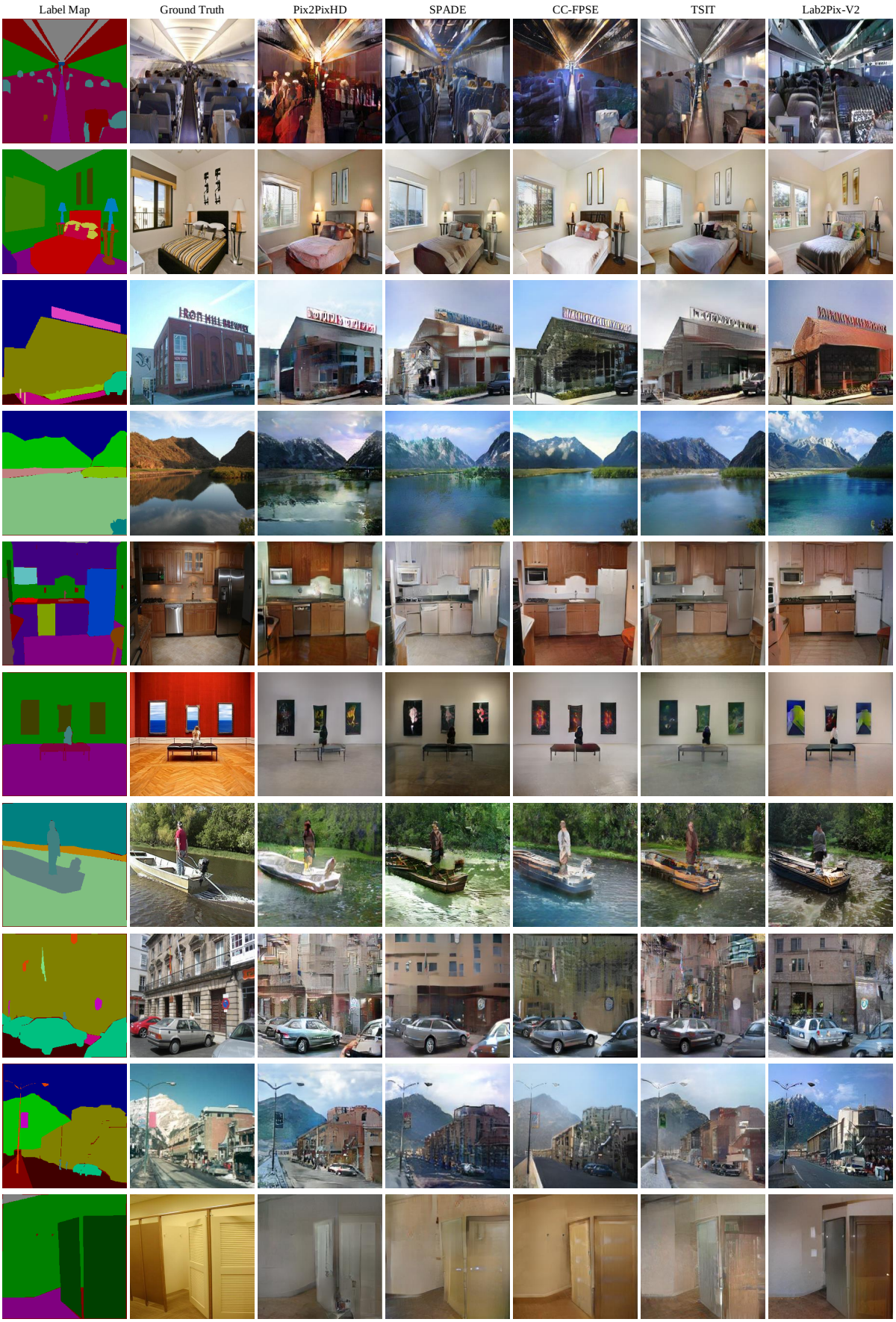


Fig. 21: Comparison on ADE20K dataset. Samples are randomly chosen.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3186752

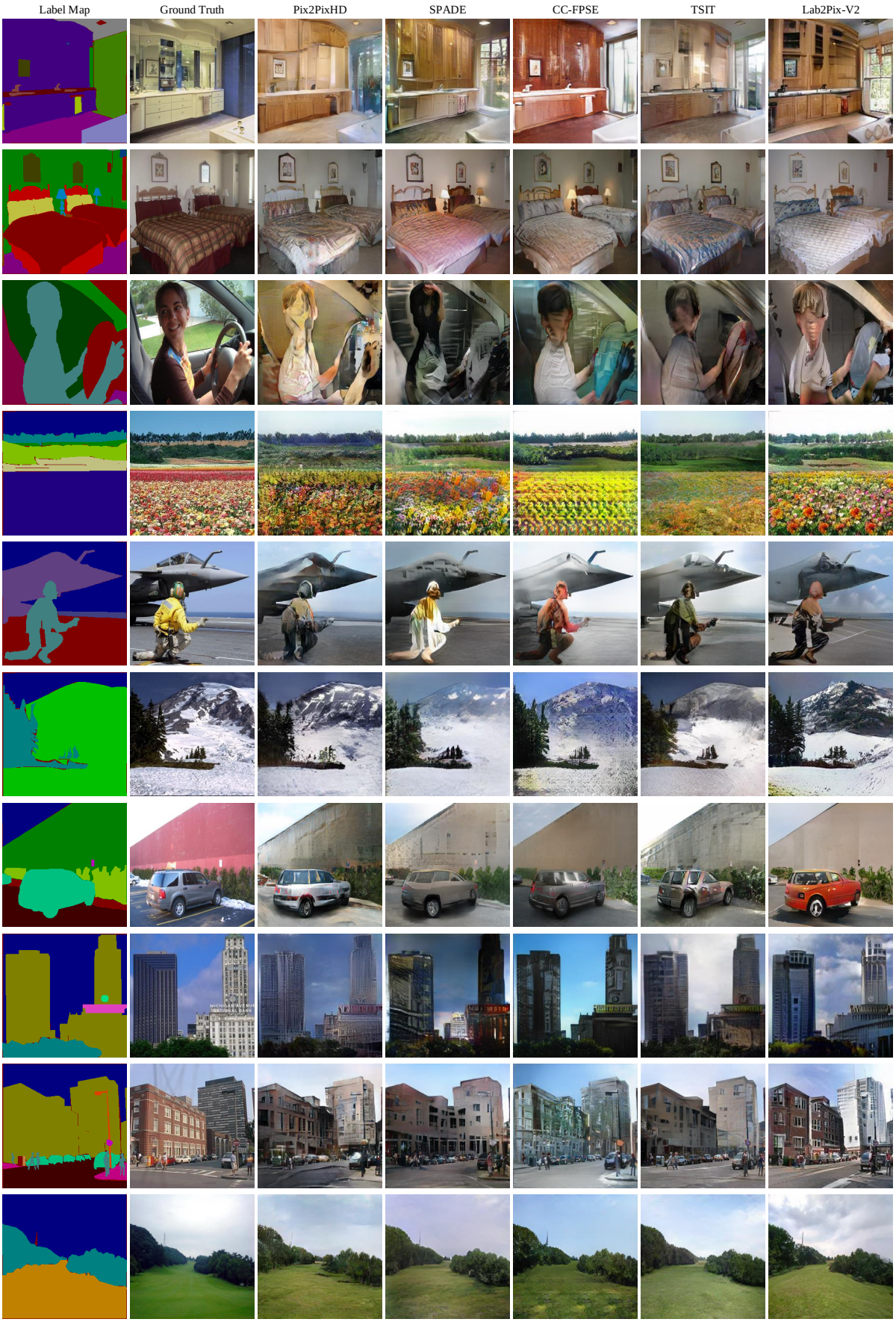


Fig. 22: Comparison on ADE20K dataset. Samples are randomly chosen.