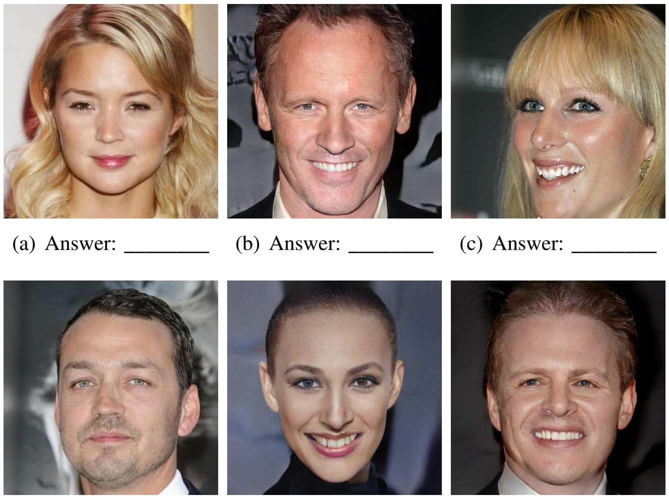
Layout-to-Image Translation with Double Pooling Generative Adversarial Networks

Hao Tang and Nicu Sebe

***Abstract*—In this paper, we address the task of layout-to-image translation, which aims to translate an input semantic layout to a realistic image. One open challenge widely observed in existing methods is the lack of effective semantic constraints during the image translation process, leading to models that cannot preserve the semantic information and ignore the semantic dependencies within the same object. To address this issue, we propose a novel Double Pooing GAN (DPGAN) for generating photo-realistic and semantically-consistent results from the input layout. We also propose a novel Double Pooling Module (DPM), which consists of the Square-shape Pooling Module (SPM) and the Rectangle-shape Pooling Module (RPM). Specifically, SPM aims to capture short-range semantic dependencies of the input layout with different spatial scales, while RPM aims to capture long-range semantic dependencies from both horizontal and vertical directions. We then effectively fuse both outputs of SPM and RPM to further enlarge the receptive field of our generator. Extensive experiments on five popular datasets show that the proposed DPGAN achieves better results than state-of-the-art methods. Finally, both SPM and SPM are general and can be seamlessly integrated into any GAN-based architectures to strengthen the feature representation. The code is available at https://github.com/Ha0Tang/DPGAN.**

***Index Terms*—GANs, Pooling, Layout-to-Image Translation**



(a) Answer: ________ (b) Answer: ________ (c) Answer: ________

(d) Answer: ________ (e) Answer: ________ (f) Answer: ________

Fig. 1: 'Real vs. Fake' game: Can you guess which image is real and which has been generated by the proposed DPGAN?

I. Introduction

In Figure 1 we show a 'Real vs. Fake' game, in which a mix of 'real' images are collected from the real world and 'fake' images are generated by our GAN model. The goal is to guess which image is real and which one has been generated by the proposed GAN model. Now you can check your answers below[1]. This should be a very challenging and difficult task, considering the recent progress in Generative Adversarial Networks (GANs) [1].

In this paper, we aim to address the challenging layout-to-image translation task, which has a wide range of real-world applications such as content generation and image editing [2], [3], [4]. This task has been widely investigated in recent years [4], [5], [6], [7], [8], [9]. For example, Park et al. [5] proposed the GauGAN model with a novel spatially-adaptive normalization to generate realistic images from semantic layouts. Tang et al. [9] proposed the LGGAN framework with a novel local generator for generating realistic small objects and detailed local texture. Despite the interesting exploration of these methods, we can still observe blurriness and artifacts in their generated results because the existing methods lack an effective semantic dependency modeling to maintain the semantic information of the input layout, causing intra-object semantic inconsistencies such as the fence, buses, and pole generated by GauGAN in Figure 2.

To solve this limitation, we propose a novel Double Pooling GAN (DPGAN) and a novel Double Pooling Module (DPM). The proposed DPM consists of two sub-modules, i.e., Square-shape Pooling Module (SPM) and Rectangle-shape Pooling Module (RPM). In particular, SPM aims to capture short-range and local semantic dependencies, leading pixels within the same object to be correlated. Simultaneously, RPM aims to capture long-range and global semantic dependencies from both horizontal and vertical directions. Finally, we propose seven image-level and feature-level fusion strategies to effectively combine the outputs of both SPM and RPM for generating high-quality and semantically-consistent images.

Overall, the contributions of our paper are:

- We propose a novel Double Pooing GAN (DPGAN) for the challenging task of layout-to-image translation, which can effectively capture semantic dependencies among different locations of the input layout for generating photo-realistic and semantically-consistent images.
- We design a novel Double Pooling Module (DPM), which consists of the Square-shape Pooling Module (SPM) and

Hao Tang is with the Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8092, Switzerland. E-mail: hao.tang@vision.ee.ethz.ch

Nicu Sebe is with the Department of Information Engineering and Computer Science (DISI), University of Trento, Trento 38123, Italy. E-mail: sebe@disi.unitn.it.

Corresponding author: Hao Tang.

[1] Answers: (a) real; (b) fake; (c) real; (d) real; (e) fake; (f) fake.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3109531, IEEE Transactions on Image Processing

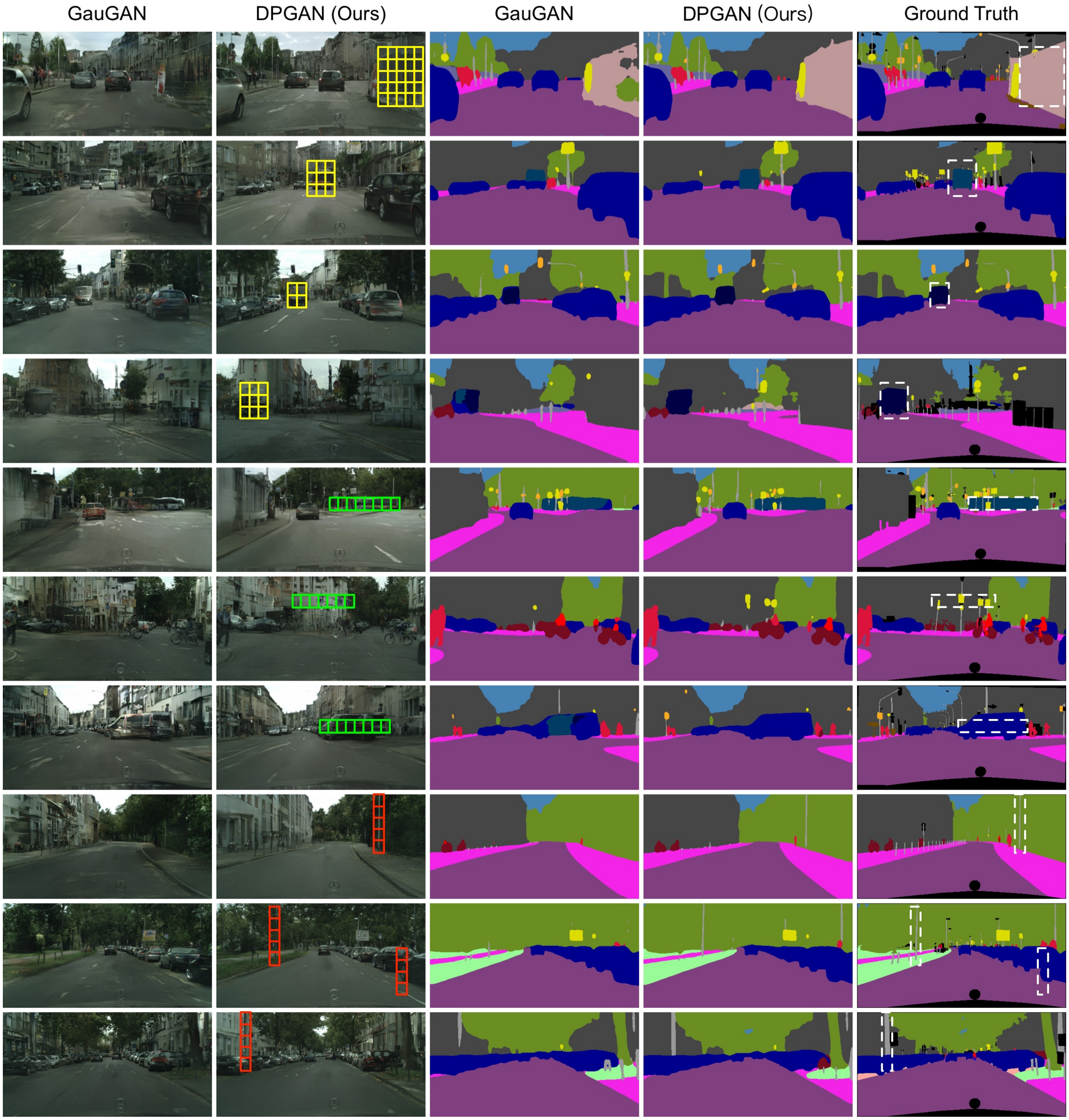


Fig. 2: Visualization of our generated semantic maps compared with those from GauGAN [5] on Cityscapes. The proposed SPM (yellow grids) captures short-range and local dependencies, while the proposed RPM (i.e., HRPM and VRPM) captures long-range and global semantic correlations from both horizontal direction (green grids) and vertical direction (red grids), respectively. Equipped with both modules, the proposed DPGAN can enlarge the receptive field, thus improves the intra-object semantic consistency. Most improved regions are highlighted in the ground truths with white dash boxes.

the Rectangle-shape Pooling Module (RPM). SPM aims at capturing short-range and local semantic dependencies, while RPM aims at modeling long-range and global semantic dependencies from both horizontal and vertical directions. Both SPM and RPM are general and can be readily applied to existing GAN-based frameworks without modifying the architecture of the network.

- We conduct extensive experiments on five popular datasets with different image resolutions, i.e., ADE20K [10], DeepFashion [11], Cityscapes [12], CelebAMask-HQ [13], and Facades [14]. Both qualitative and quantitative results demonstrate that the proposed DPGAN is able to produce better results than state-of-the-art approaches.

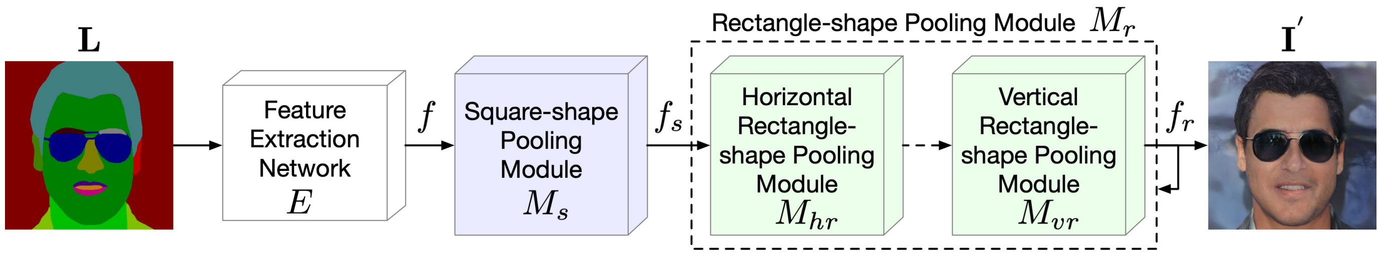

L
Feature Extraction Network E
f
Square-shape Pooling Module M_s
f_s
Rectangle-shape Pooling Module M_r
Horizontal Rectangle-shape Pooling Module M_{hr}
Vertical Rectangle-shape Pooling Module M_{vr}
f_r
I′

Fig. 3: Overview of the generator G of our proposed DPGAN, which consists of a feature extraction network E, a Square-shape Pooling Module M_s, and a Rectangle-shape Pooling Module M_r. All components are trained in an end-to-end fashion so that M_s and M_r can benefit from each other by capturing both long-range and short-range semantic dependencies.

II. Related Work

Generative Adversarial Networks (GANs) [1] are widely used techniques to generate high-quality images [15], [16], [17], [18], [19], [19], videos [20], and 3D objects [21]. A GAN framework contains a generator and a discriminator, where the generator tries to generate realistic images to fool the discriminator while the discriminator aims to accurately tell whether an image is real or fake. Furthermore, Mirza and Osindero propose Conditional GANs (CGANs) [22] based on GANs by incorporating extra guidance information to generate user-specific images, e.g., category labels [23], [24], [25], text descriptions [26], [27], human pose/gesture [28], [29], [30], [31], [32], attention maps [33], [34], [35], [36].

Layout-to-Image Translation aims to turn semantic layouts into realistic images [5], [6], [8], [9], [37], [38], [39], [40], [41]. For example, Park et al. [5] proposed GauGAN with a novel spatially-adaptive normalization to generate realistic images. Although GauGAN [5] has achieved promising results, we still observe unsatisfactory aspects mainly in the generated scene details and intra-object completions (see Figure 2), which we believe are mainly due to the lack of short-range and long-range semantic constrains in the input layout. The proposed SPM and RPM explicitly address this problem.

Pooling operations are commonly used in semantic segmentation tasks [42], [43], [44], [45] to improve the receptive field. For example, Zhao et al. [42] proposed a pyramid pooling module to capture the global context information in the scene parsing task. Hou et al. [44] proposed a new strip pooling, which considers a long but narrow kernel, for the scene parsing task. However, to the best of our knowledge, our idea of using pooling modules to capture both short-range and long-range semantic dependencies has not been investigated by any existing layout-to-image translation or even GAN-based image generation approaches.

III. Double Pooling GANs

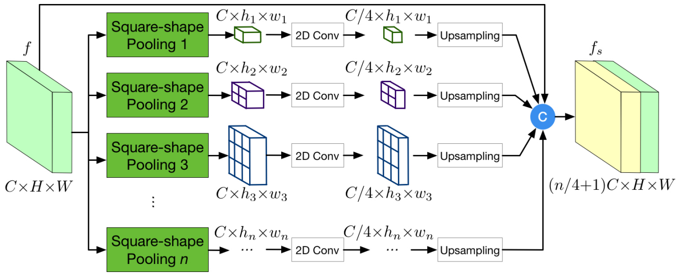

f
$C{\times}H{\times}W$
Square-shape Pooling 1
Square-shape Pooling 2
Square-shape Pooling 3
Square-shape Pooling n
$C{\times}h_1{\times}w_1$
$C{\times}h_2{\times}w_2$
$C{\times}h_3{\times}w_3$
$C{\times}h_n{\times}w_n$
2D Conv
$C/4{\times}h_1{\times}w_1$
$C/4{\times}h_2{\times}w_2$
$C/4{\times}h_3{\times}w_3$
$C/4{\times}h_n{\times}w_n$
Upsampling
C
f_s
$(n/4{+}1)C{\times}H{\times}W$

Fig. 4: The proposed Square-shape Pooling Module (SPM) which aims to capture short-range and local semantic dependencies. Our SPM is a n-level pooling module with different square-kernel size, i.e., (h_1, w_1), (h_2, w_2), $\cdots$, (h_n, w_n), where $\{h_i{=}w_i\}_{i=1}^n$. The symbol Ⓒ denotes channel-wise concatenation.

Overview. We start by presenting the details of the proposed Dual Pooling GANs (DPGAN), which consists of a generator G and discriminator D. An illustration of the proposed generator G is shown in Figure 3, which mainly consists of three components, i.e., a feature extraction network E extracting deep features from the input layout $\mathbf{L}$, a Square-shape Pooling Module (SPM) modeling short-range and local semantic dependencies, and a Rectangle-shape Pooling Module (RPM) capturing long-range and global semantic dependencies from both horizontal and vertical directions. SPM and RPM together form our proposed Double Pooling Module (DPM). Moreover, we propose seven image-level and feature-level fusion methods to combine both the outputs of SPM and RPM.

Feature Extraction Network. As shown in Figure 3, the network E receives the semantic layout $\mathbf{L}$ as input and outputs the deep feature f, which can be formulated as,

$$f = E(\mathbf{L}). \tag{1}$$

Then, f is fed into the proposed SPM and RPM for learning short-range and long-rang semantic dependencies, respectively.

Square-Shape Pooling Module. Existing layout-to-image translation methods such as [5], [6], [9], [40] directly use deep features generated by convolutional operations, leading to limited effective fields-of-views and thus generating different textures in the pixels with the same label. To model short-range and local semantic dependencies over the deep feature f, we propose a Square-shape Pooling Module (SPM). Note that the idea of the proposed SPM is inspired by the pyramid pooling module proposed in [42] and we extend the original module used in image segmentation to a completely different image generation task.

The framework of SPM is elaborated in Figure 4. Specifically, we first separately feed f into n square-shape pooling layers to produce n new feature maps with different spatial scales. Consider the n-th pooing layer in Figure 4, whose

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3109531, IEEE Transactions on Image Processing

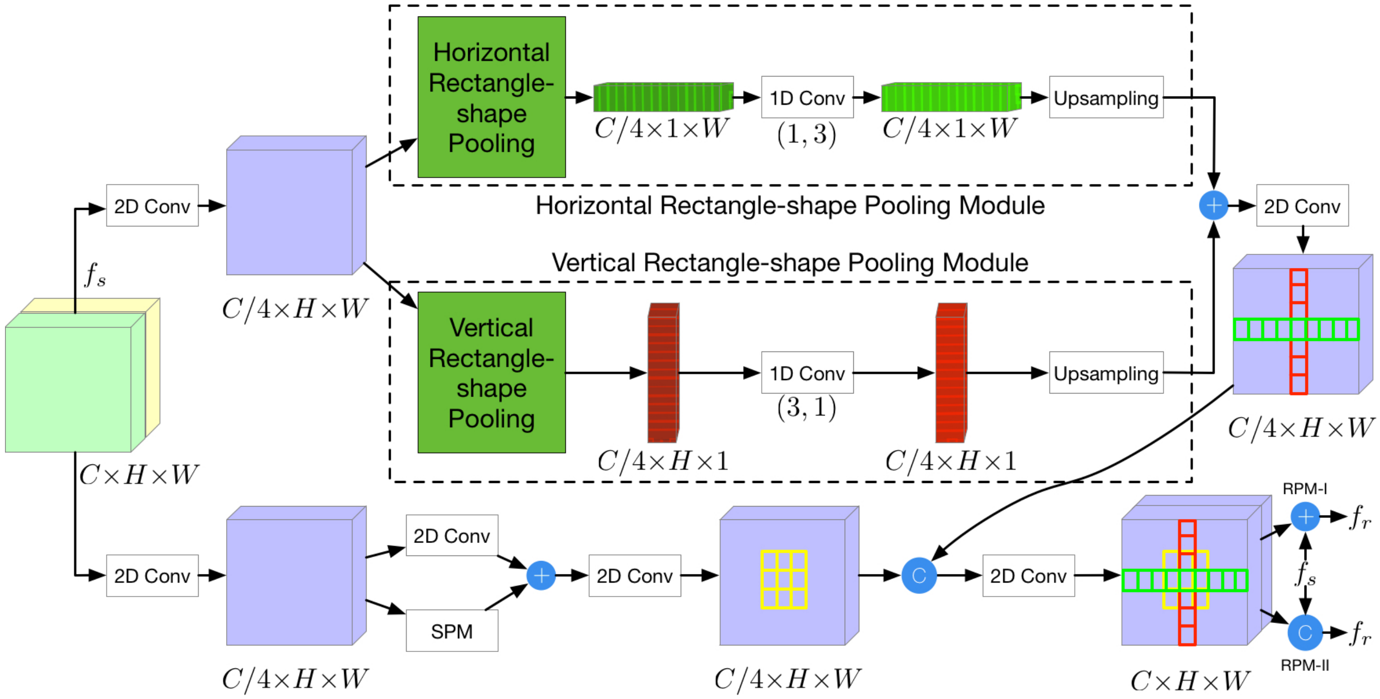


Fig. 5: The proposed Rectangle-shape Pooling Module (RPM) which consists of a Horizontal Rectangle-shape Pooling Module (HRPM) and a Vertical Rectangle-shape Pooling Module (VRPM), aiming to capture long-range and global semantic dependencies from horizontal and vertical direction, respectively. The yellow, green, and red grids represent short-dependency, horizontal long-dependency, and vertical long-dependency, respectively. The symbols $\oplus$, and ⓒ denote element-wise addition, and channel-wise concatenation, respectively.

input is the deep feature $f\in\mathbb{R}^{C\times H\times W}$. The output is the pooled feature map $f^n\in\mathbb{R}^{C\times h_n\times w_n}$, where $(h_n,\ w_n)$ is the targeted output size of the n-th pooling layer. We then feed the pooled feature f^n through a convolutional layer for reducing the number of channels, leading to a new feature map $\hat{f}^n\in\mathbb{R}^{C/4\times h_n\times w_n}$. After that, we perform an upsampling operation on $\hat{f}^n$ to obtain the feature map $\tilde{f}^n$, which has the same spatial size with f. Mathematically,

$$\tilde{f}^n=\mathrm{up}_n(\mathrm{conv}(\mathrm{pl}_n(f))), \tag{2}$$

where $\mathrm{up}(\cdot)$, $\mathrm{conv}(\cdot)$ and $\mathrm{pl}(\cdot)$ denote upsampling, convolutional layer, and square-pooing layer, respectively.

Next, we concatenate all n learned feature maps and the input feature f to produce the final feature $f_s\in\mathbb{R}^{(n/4+1)C\times H\times W}$. The computation process can be expressed as follow,

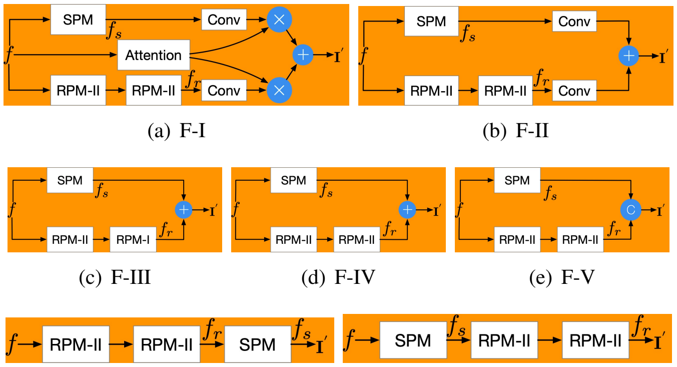
$$f_s=\mathrm{concat}(\tilde{f}^1,\tilde{f}^2,\tilde{f}^3,\cdots,\tilde{f}^n,f), \tag{3}$$

where $\mathrm{concat}(\cdot)$ denotes channel-wise concatenation. By doing so, the feature f_s has short-range and local semantic dependencies with different spatial scales. Therefore, the pixels with the same semantic label can achieve mutual gains, thus we are improving intra-object semantic consistency (see the first to fourth rows in Figure 2).

Rectangle-Shape Pooling Module. The proposed SPM captures only short-range semantic dependencies, as shown in Figure 2. To capture long-range and global semantic dependencies, we can increase the kernel size of the square pooling. However, this inevitably incorporates lots of irrelevant regions when processing rectangle-shaped and narrow objects such as the bus and pole shown in Figure 2.

To alleviate this limitation, we propose a novel Rectangle-shape Pooling Module (RPM), which aims to capture long-range and global semantic dependencies from both horizontal and vertical directions. The idea of the proposed RPM is inspired by the strip pooling module proposed in [44] and the framework of RPM is illustrated in Figure 5. It consists of a Horizontal Rectangle-shape Pooling Module (HRPM) and a Vertical Rectangle-shape Pooling Module (VRPM). HRPM captures long-range dependencies from horizontal and narrow objects (e.g., the fifth to seventh rows of Figure 2), while VRPM captures long-range correlations from vertical and narrow objects (e.g., the eighth to tenth rows of Figure 2).

As shown in Figure 5, given the feature $f_s\in C\times H\times W$ produced by SPM, we first feed it into a convolution layer to reduce the number of the channels and obtain a new feature $f_s^1\in C/4\times H\times W$. Note that we use C to represent the number of channels of f_s for simplicity, which is different from the one used in Figure 4. Then f_s^1 is separately fed into HRPM and VRPM to capture both horizontal and vertical long-range dependencies. Specifically, in HRPM, f_s^1 is first fed into a horizontal rectangle-shape pooling layer to obtain a new feature $f_h\in C/4\times1\times W$. After that, we put f_h through a 1D convolutional layer to obtain the feature $\hat{f}_h$. Next, an upsampling operation is performed on $\hat{f}_h$ to expand the spatial size and then output the feature $\tilde{f}_h$. Similarly, in VRPM, f_s^1 is first fed into a vertical rectangle-shape pooling layer



(a) F-I (b) F-II

(c) F-III (d) F-IV (e) F-V

(f) F-VI (g) F-VII

Fig. 6: The proposed seven image-level (a-b) and feature-level (c-g) fusion strategies. The symbols $\oplus$, $\otimes$, and ⓒ denote element-wise addition, element-wise multiplication, and channel-wise concatenation, respectively.

to obtain a new feature $f_v \in C/4 \times H \times 1$. After that, we use a 1D convolutional layer to obtain the feature $\hat{f}_v$. Next, an upsampling operation is performed on $\hat{f}_v$ to expand the spatial size and then output the feature $\tilde{f}_v$. We finally sum both $\tilde{f}_h$ and $\tilde{f}_v$ incorporating both horizontal and vertical long-range dependencies,

$$\tilde{f}_s = \tilde{f}_h + \tilde{f}_v. \tag{4}$$

Taking into account the advantages of short-range dependency modeling, we also consider incorporating SPM in our PRM to make the feature representations more discriminative. As shown in Figure 5, we first feed f_s into another convolution layer to reduce the number of the channels and obtain a new feature $f_s^2 \in C/4 \times H \times W$. We then capture short-range semantic dependencies by using Equation (5).

$$f_s^{'} = \mathrm{conv}(\mathrm{conv}(f_s^2) + \mathrm{SPM}(f_s^2)), \tag{5}$$

where $\mathrm{SPM}(\cdot)$ is the model proposed in Figure 4. After that, we combine both short-range and long-range semantic dependencies by using,

$$f_r^{'} = \mathrm{conv}(\mathrm{concat}(f_s^{'}, \tilde{f}_s)). \tag{6}$$

In this way, $f_r^{'}$ is more discriminative than f_s by aggregating different types of contextual information via various pooling operations, leading to better results. Finally, we also propose two methods to add the input feature f_s, constituting a residual connection [46]. The first one (RPM-I) is performing an element-wise addition:

$$f_r = f_r^{'} + f_s. \tag{7}$$

The second one (RPM-II) is performing a channel-wise concatenation:

$$f_r = \mathrm{concat}(f_r^{'}, f_s). \tag{8}$$

Fusion of SPM and RPM. To take full advantage of both short-range and long-range semantic information, we further aggregate the outputs from these two pooling modules. Specifically, we propose seven aggregation methods as shown in Figure 6:

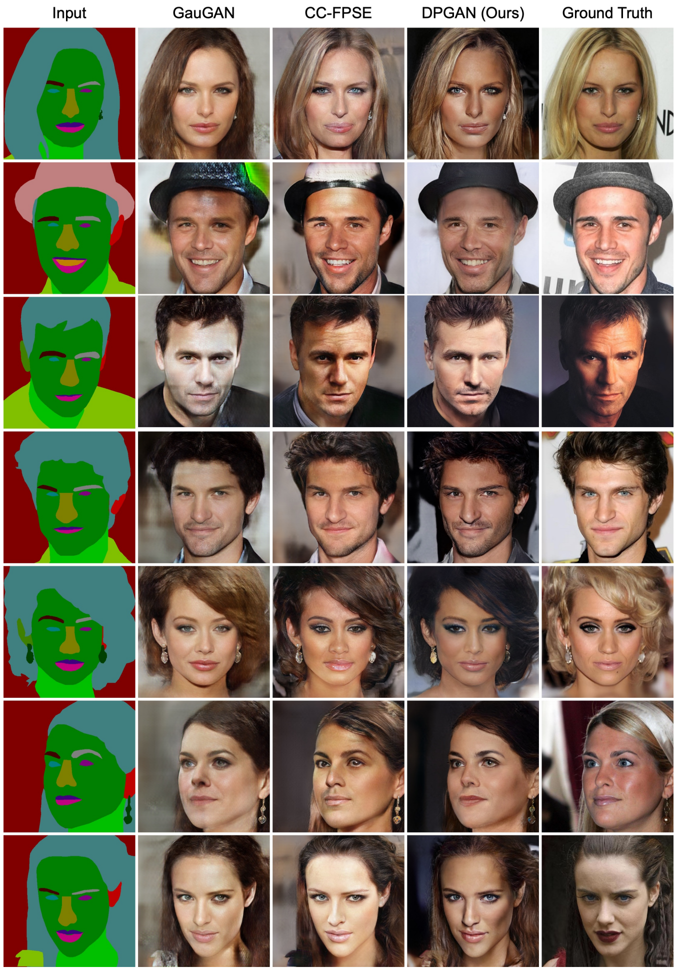


Fig. 7: Qualitative comparison on CelebAMask-HQ. From left to right: Input, GauGAN [5], CC-FPSE [6], DPGAN (Ours), and Ground Truth. We see than DPGAN generates more convincing details than GauGAN and CC-FPSE, e.g., the hair, the hat, and the face skin in the first, second, and third row, respectively.

- (1) F-I is an image-level fusion strategy based on the attention fusion method proposed in LGGAN [9]. The formulation of F-I can be expressed as: $\mathbf{I}^{'} = \mathrm{conv}(f_s) \times A_1 + \mathrm{conv}(f_r) \times A_2$, where A_1 and A_2 are attention masks produced by an attention decoder.
- (2) F-II is also an image-level fusion method represented as: $\mathbf{I}^{'} = \frac{\mathrm{conv}(f_s) + \mathrm{conv}(f_r)}{2}$.
- (3) The other five methods are feature-level based fusion: F-III, F-IV and F-V are parallel structures, while F-VI and F-VII are cascading structures. Mathematically, F-III can be written as $\mathbf{I}^{'} = \mathrm{conv}(f_s + f_r)$.
- (4) The difference between F-IV and F-III is that F-IV uses two 'RPM-II' to produce the feature f_r, while F-III adopts one 'RPM-II' and one 'RPM-I' to generate f_r.
- (5) The difference between F-V and F-IV is that F-V uses a channel-wise concatenation operation to combine both f_s and f_r, thus it can be expressed as: $\mathbf{I}^{'} = \mathrm{conv}(\mathrm{concat}(f_s, f_r))$.
- (6) Both F-VI and VII are cascading structures, and the

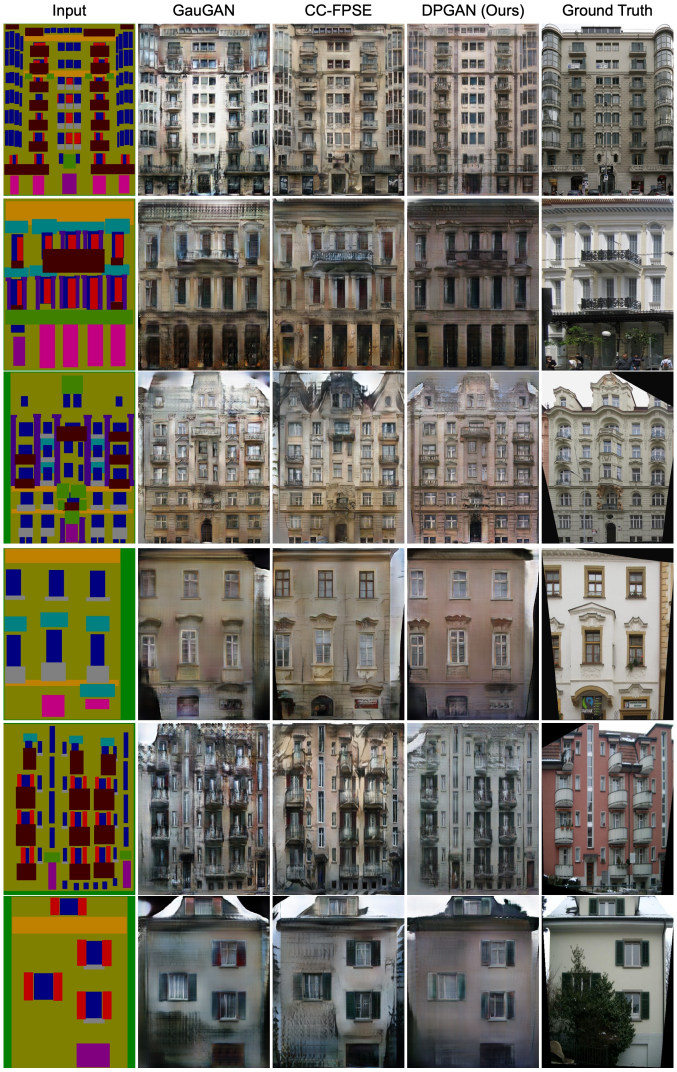


Fig. 8: Qualitative comparison on Facades. From left to right: Input, GauGAN [5], CC-FPSE [6], DPGAN (Ours), and Ground Truth. We see that DPGAN generates more clear architecture structures with fewer artifacts than GauGAN and CC-FPSE.

difference between them is the order of RPM and SPM. We can represent F-VI as: $\mathbf{I}^{'}{=}\mathrm{conv}(f_s)$.

- (7) F-VII is also shown in Figure 3, and it can be expressed as: $\mathbf{I}^{'}{=}\mathrm{conv}(f_r)$.

Optimization Objective. We follow GauGAN [5] and employ three different losses as our optimization objective, i.e., $\mathcal{L}{=}\lambda_{gan}\mathcal{L}_{gan}{+}\lambda_f\mathcal{L}_f{+}\lambda_p\mathcal{L}_p$, where $\mathcal{L}_{gan}$, $\mathcal{L}_f$ and $\mathcal{L}_p$ denote adversarial loss, discriminator feature matching loss, and perceptual loss, respectively. We set $\lambda_{gan}{=}1$, $\lambda_f{=}10$, and $\lambda_p{=}10$ in our experiments.

Training Details. We use the multi-scale discriminator [5] as our discriminator D. We use the Adam solver [47] and set $\beta_1{=}0$, $\beta_2{=}0.999$. Moreover, we set $n{=}4$ in the proposed SPM, and set $h_1{=}w_1{=}1$, $h_2{=}w_2{=}2, h_3{=}w_3{=}3$, and $h_4{=}w_4{=}6$, respectively. The kernel size of convolutional layers in the proposed SPM is set to $1{\times}1$. We set $n{=}2$ for SPM used in RPM, and set $h_1{=}w_1{=}12$ and $h_2{=}w_2{=}20$. The kernel size of 1D convolutional layers in RPM is $1{\times}3$ and $3{\times}1$, respectively. The proposed DPGAN is implemented by using PyTorch [48]. We conduct the experiments on NVIDIA DGX1 with 8 32GB V100 GPUs.

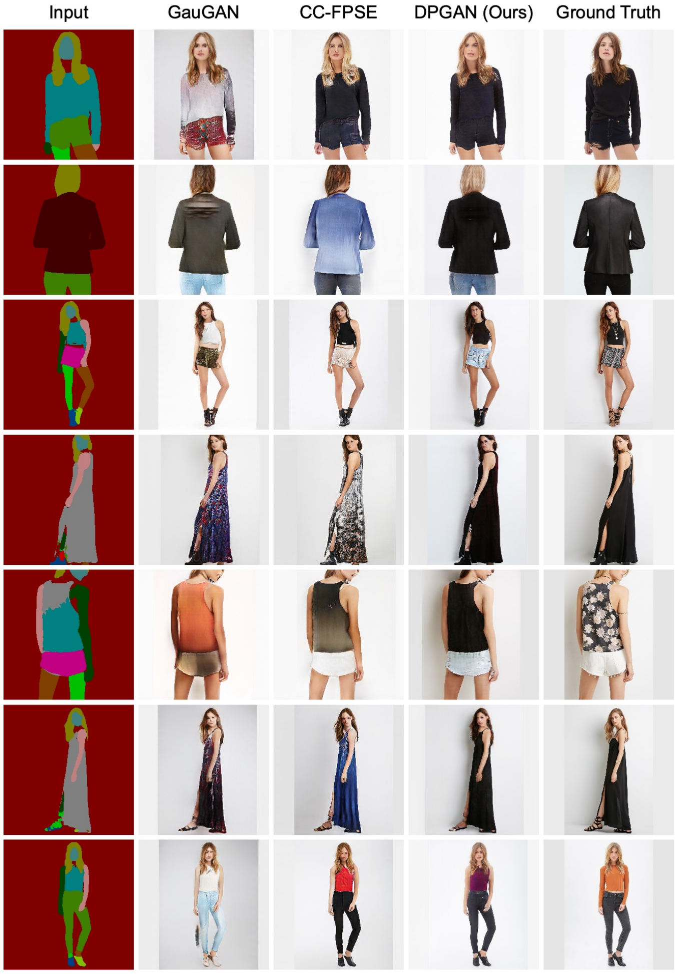


Fig. 9: Qualitative comparison on DeepFashion. From left to right: Input, GauGAN [5], CC-FPSE [6], DPGAN (Ours), and Ground Truth. We see that DPGAN generates more photo-realistic clothes than GauGAN and CC-FPSE.

IV. Experiments

Datasets. We follow GauGAN [5] and firstly conduct experiments on Cityscapes [12] and ADE20K [10] datasets. Cityscapes contains street scene images, and ADE20K contains both indoor and outdoor scenes. To further evaluate the robustness of our method, we conduct experiments on three more datasets with diverse scenarios, i.e., DeepFashion [11], CelebAMask-HQ [13], and Facades [14]. DeepFashion contains human body images, CelebAMask-HQ contains human facial images, and Facades contains facade images with diverse architectural styles. Experiments are conducted using different image resolutions to validate that our DPGAN can also generate high-resolution images, i.e., ADE20K (256×256), DeepFashion (256×256), Cityscapes (512×256), Facades (512×512), and CelebAMask-HQ (512×512).

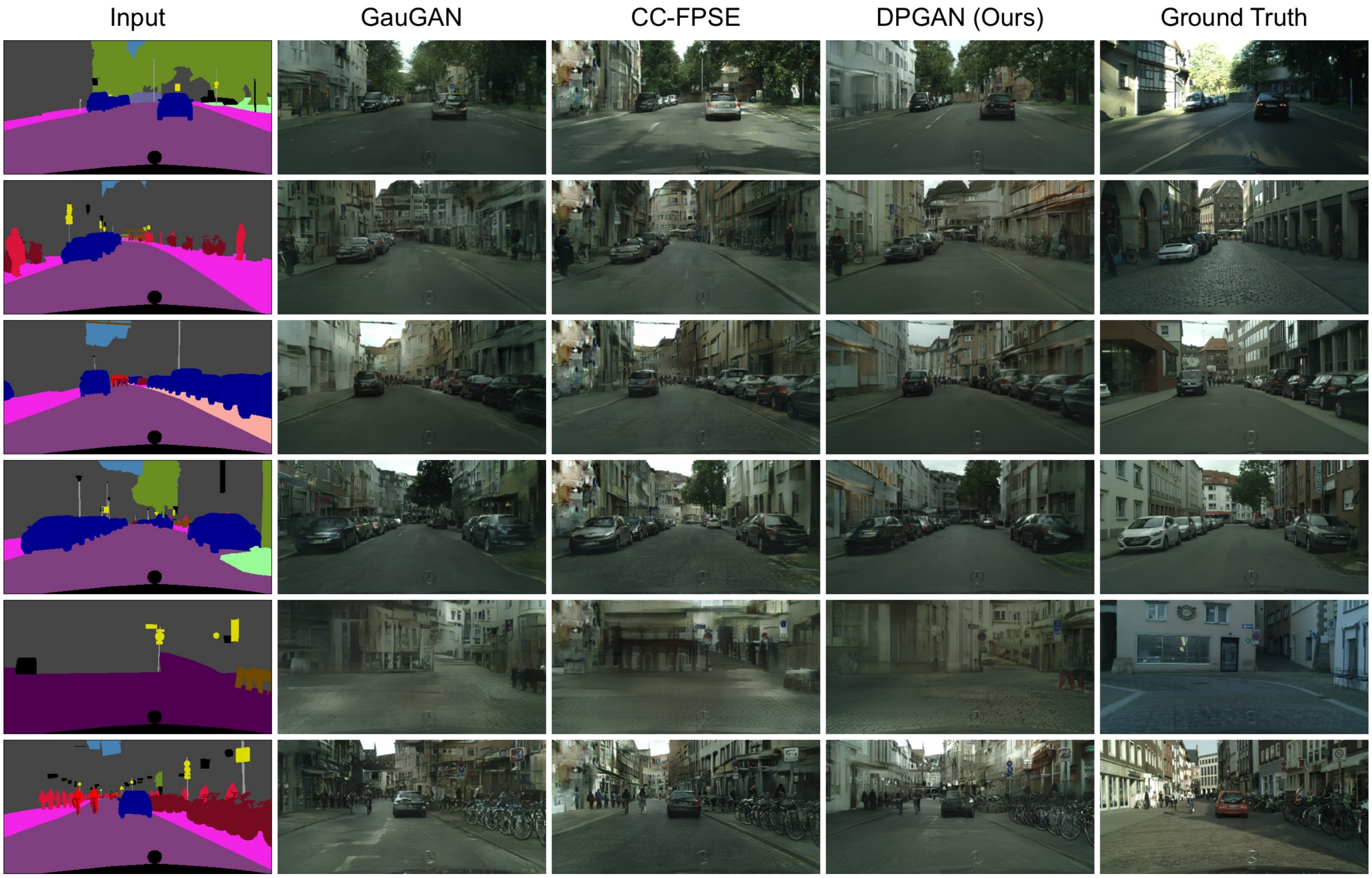


Fig. 10: Qualitative comparison on Cityscapes. From left to right: Input, GauGAN [5], CC-FPSE [6], DPGAN (Ours), and Ground Truth. We see that DPGAN produces more realistic images with fewer artifacts than both leading methods.

TABLE I: User study. The numbers indicate the percentage of users who favor the results of our proposed DPGAN over the competing methods.

AMT ↑	Cityscapes	ADE20K	DeepFashion	Facades	CelebAMask-HQ
Ours vs. GauGAN [5]	65.78	68.72	66.85	67.54	69.91
Ours vs. CC-FPSE [6]	62.21	64.36	63.16	64.54	67.18

Evaluation Metrics. We follow GauGAN [5] and adopt mean Intersection-over-Union (mIoU), pixel accuracy (Acc), and Fréchet Inception Distance (FID) [49] as the evaluation metrics on Cityscapes and ADE20K. For DeepFashion, CelebAMask-HQ, and Facades datasets, we use FID and Learned Perceptual Image Patch Similarity (LPIPS) [50] to evaluate the quality of the generated images.

A. Comparisons with State-of-the-Art

We adopt GauGAN [5] as our backbone and insert the proposed Double Pooling Module (DPM) before the last convolution layer to form our final model, i.e., DPGAN.

Qualitative Comparisons. We first compare the proposed DPGAN with GauGAN [5] and CC-FPSE [6] on DeepFashion, CelebAMask-HQ, and Facades datasets. Note that we used the source code provided by the authors to generate the results of GauGAN and CC-FPSE on these three datasets for fair comparisons. Visualization results are shown in Figures 7, 8, and 9. We can see that the proposed DPGAN generates more photo-realistic and semantically-consistent results than both GauGAN and CC-FPSE.

Moreover, we compare DPGAN with two leading methods on both Cityscapes and ADE20K datasets, i.e., GauGAN [5] and CC-FPSE [6]. Comparison results are shown in Figures 10 and 11. We can see that our DPGAN produces more clear and visually plausible results than both leading methods, further validating the effectiveness of our proposed DPGAN.

User Study. We follow the same evaluation protocol of GauGAN and also perform a user study. The results compared with GauGAN and CC-FPSE are shown in Table I. We see that users strongly favor the results generated by our proposed DPGAN on all datasets, further validating that the generated images by our DPGAN are more photo-realistic.

Quantitative Comparisons. Although the user study is more suitable for evaluating the quality of the generated image in this task, we also follow GauGAN and use mIoU, Acc, FID, and LPIPS for quantitative evaluation. The results compared

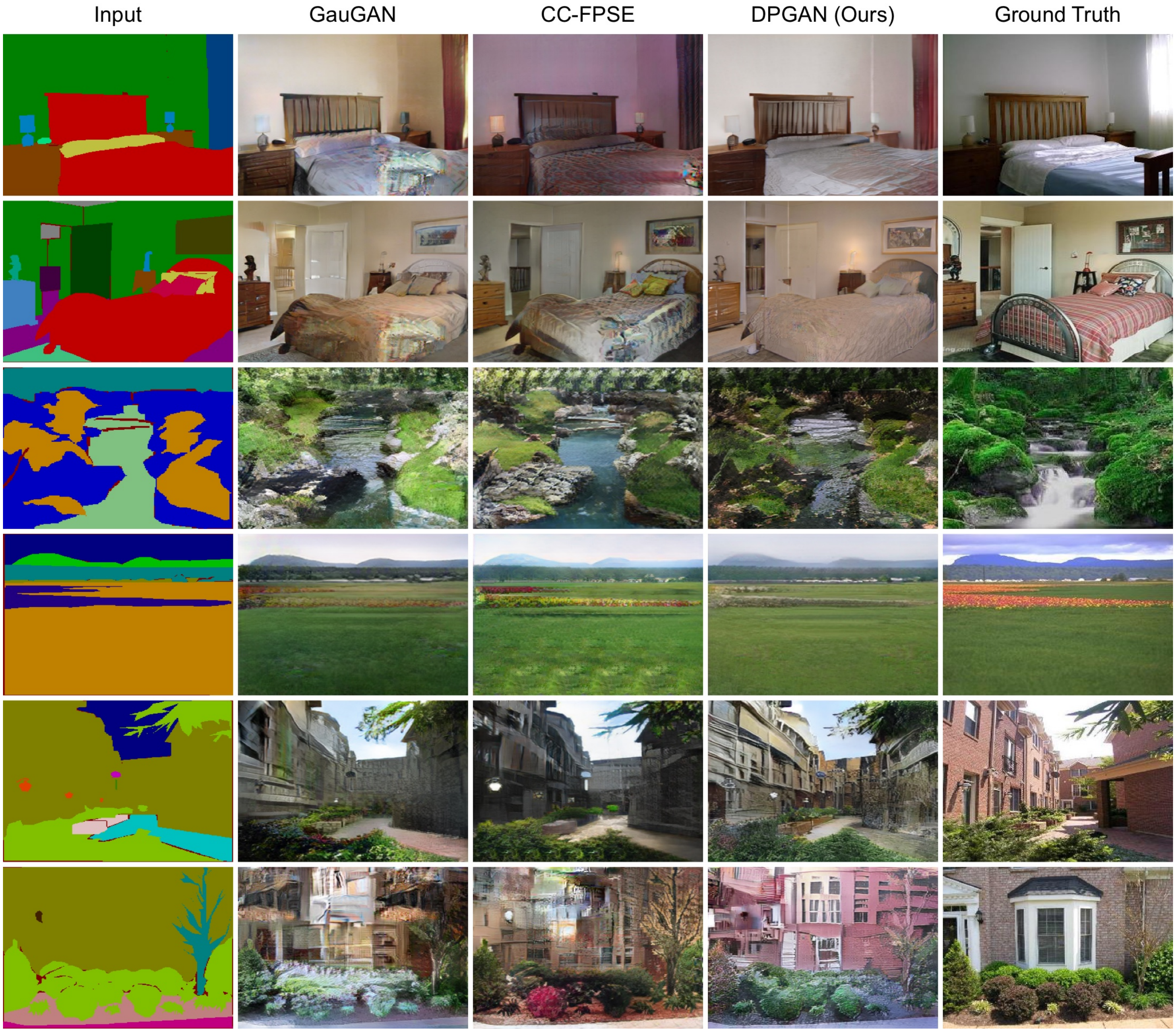


Fig. 11: Qualitative comparison on ADE20K. From left to right: Input, GauGAN [5], CC-FPSE [6], DPGAN (Ours), and Ground Truth. We see that DPGAN produces realistic images while respecting the spatial semantic layout at the same time.

TABLE II: Quantitative comparison of different methods on DeepFashion, Facades, and CelebAMask-HQ.

Method	DeepFashion		Facades		CelebAMask-HQ	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
GauGAN	22.8	0.2476	116.8	**0.5437**	42.2	0.4870
+ DPM (Ours)	**20.8**	**0.2455**	**115.1**	0.5503	**25.1**	**0.4823**

with several leading methods are shown in Tables II and III. Firstly, we observe that the proposed DPGAN achieves the best results compared with GauGAN on DeepFashion, CelebAMask-HQ, and Facades datasets, as shown in Table II. Moreover, we can see that DPGAN achieves competitive results compared with other leading methods on both Cityscapes and ADE20K datasets in Table III.

Visualization of Generated Semantic Maps. We follow GauGAN and adopt the pretrained DRN-D-105 [54] on the generated Cityscapes images to produce semantic maps. The results compared with those produced by GauGAN are shown in Figure 2. We clearly see that the proposed SPM and RPM can capture short-range and long-range semantic dependencies, leading to more semantically-consistent and realistic results than GauGAN.

TABLE III: Quantitative comparison of different methods on Cityscapes and ADE20K. The results of other methods are reported from their papers.

Method	Cityscapes			ADE20K		
	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓
CRN [2]	52.4	77.1	104.7	22.4	68.8	73.3
SIMS [51]	47.2	75.5	49.7	-	-	-
Pix2pixHD [4]	58.3	81.4	95.0	20.3	69.2	81.8
GAN Compression [52]	61.2	-	-	-	-	-
BachGAN [53]	-	70.4	73.3	-	66.8	49.8
PIS [7]	64.8	82.4	96.4	-	-	-
SelectionGAN [34]	63.8	82.4	65.2	40.1	81.2	33.1
DAGAN [40]	66.1	82.6	60.3	40.5	81.6	31.9
LGGAN [9]	68.4	83.0	57.7	41.6	81.8	31.6
GauGAN [5]	62.3	81.9	71.8	38.5	79.9	33.9
+ DPM (Ours)	65.2 (**+2.9**)	82.6 (**+0.7**)	53.0 (**-18.8**)	39.2 (**+0.7**)	80.4 (**+0.5**)	31.7 (**-2.2**)
CC-FPSE [6]	65.5	82.3	54.3	43.7	82.9	31.7
+ DPM (Ours)	66.9 (**+1.4**)	82.8 (**+0.5**)	51.9 (**-2.4**)	44.8 (**+1.1**)	83.2 (**+0.3**)	30.3 (**-1.4**)
TSIT [8]	65.9	82.7	59.2	38.6	80.8	31.6
+ DPM (Ours)	66.7 (**+0.8**)	83.1 (**+0.4**)	56.1 (**-3.1**)	39.9 (**+1.3**)	81.2 (**+0.4**)	30.5 (**-1.1**)

B. Ablation Study

Baselines of DPGAN. We conduct an extensive ablation study on Cityscapes to evaluate each component of the proposed DPGAN. DPGAN has 13 baselines (i.e., B1-B13) as shown in Table IV.

- (1) B1 is our baseline and uses a GauGAN structure [5].
- (2) B2 uses the proposed Square-shape Pooling Module (SPM) to capture short-range semantic dependencies.
- (3) B3 employs the proposed Rectangle-shape Pooling Module (RPM) to capture long-range semantic dependencies from both horizontal and vertical directions. Note that B3 uses Equation (7) to generate the feature f_r.
- (4) The difference between B4 and B3 is that B4 uses Equation (8) to generate f_r.
- (5) B5 is based on B3 and uses the proposed RPM twice.
- (6) B6 is based on B4 and uses the proposed RPM twice.
- (7)-(13) B7 to B13 are seven fusion methods proposed in Figure 6, which aim to effectively combine both short-range and long-range dependencies for further enlarging the receptive field of our model.

TABLE IV: Ablation study of our DPGAN on Cityscapes.

No.	Setting	mIoU ↑	Acc ↑	FID ↓
B1	GauGAN [5]	62.3	81.9	71.8
B2	B1 + SPM	64.9	82.5	55.9
B3	B1 + RPM-I	63.9	82.3	55.4
B4	B1 + RPM-II	63.8	82.4	54.9
B5	B1 + 2 RPM-I	64.5	82.4	54.2
B6	B1 + 2 RPM-II	64.7	82.5	53.2
B7	B6 + SPM + F-I	64.2	82.5	53.2
B8	B6 + SPM + F-II	**65.2**	82.5	54.5
B9	B6 + SPM + F-III	64.8	**82.6**	53.3
B10	B6 + SPM + F-IV	65.0	**82.6**	53.6
B11	B6 + SPM + F-V	63.1	82.4	53.3
B12	B6 + SPM + F-VI	64.8	82.4	53.4
B13	B6 + SPM + F-VII	**65.2**	**82.6**	**53.0**

Ablation Analysis. The results of the ablation study are shown in Table IV. We can see that B2 achieves better results than B1 on all metrics, which confirms the importance of modeling short-range semantic dependencies. Both B3 and B4 outperform B1, confirming the effectiveness of modeling long-range semantic dependencies. B6 outperforms B4, and B5 outperforms B3, showing that adding more layers of RPM will further enlarge the receptive field. Moreover, we observe that B13 outperforms B6 on all metrics, showing that both short-range and long-range semantic dependencies are essential for generating high-quality results. Lastly, we observe that B13 achieves better results than B7-B12, demonstrating the effectiveness of the fusion strategy F-VII.

We also note that B7 and B11 achieve slightly deteriorated results compared with B6 on the mIoU metric, which means both F-I and F-V are not very suitable fusion strategies for the proposed SPM and RPM. Other fusion strategies such as F-II, F-IV, and F-VII achieve slightly different results in all the evaluation metrics. This is because the proposed SPM and RPM are powerful to capture both short-term and long-term dependencies and a simple fusion strategy such as F-II, F-IV, or F-VII can achieve good generation performance. This also further illustrates the effectiveness of both SPM and RPM.

Generalization of DPM. The proposed Double Pooling Module (DPM) is general and can be seamlessly integrated into any existing GAN-based architecture to improve the image translation performance. Therefore, to validate the generalization ability of the proposed DPM, we further conduct more experiments on both Cityscapes and ADE20K datasets. Specifically, we adopt CC-FPSE as our E and then combine CC-FPSE and our DPM to form the final model. We observe that the CC-FPSE model with our DPM (i.e., CC-FPSE +

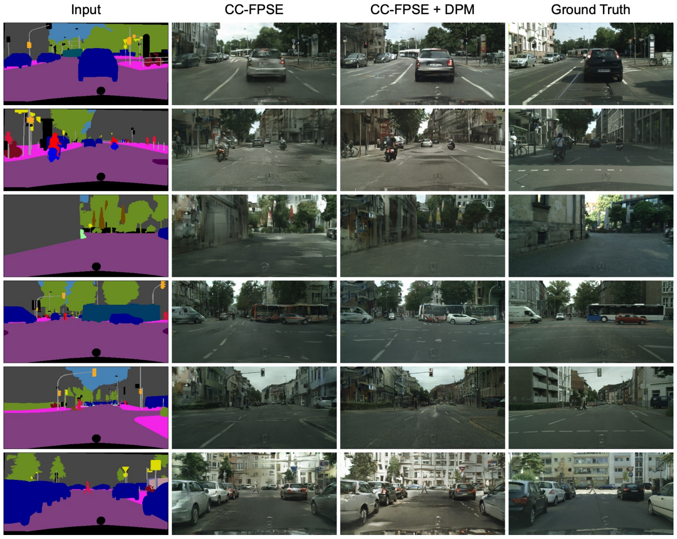


Fig. 12: The generalization ability of the proposed DPM on Cityscapes. We see that the CC-FPSE model with DPM (CC-FPSE + DPM) generates more realistic images with fewer artifacts than the CC-FPSE model without using DPM.

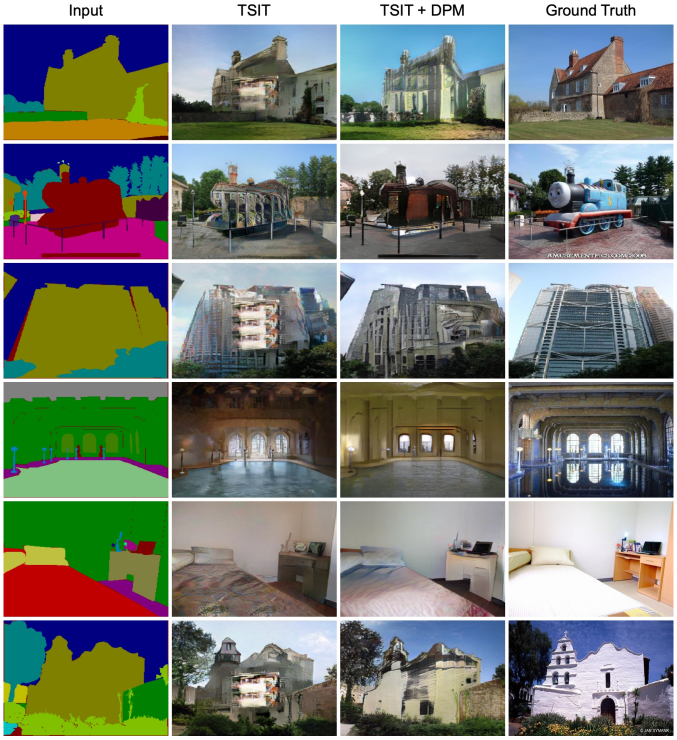


Fig. 13: The generalization ability of the proposed DPM on ADE20K. We see that the TSIT model with DPM (TSIT + DPM) generates more realistic images with fewer artifacts than the TSIT model without using DPM.

DPM) further improves all the three metrics, as shown in Table III. In the visualization results shown in Figure 12, we clearly observe that the CC-FPSE model with our DPM generates more realistic images with fewer artifacts than the CC-FPSE model without using our DPM on Cityscapes.

Moreover, we employ TSIT as our E and then combine TSIT and our DPM to form the final model. We observe that the TSIT model with our DPM (i.e., TSIT + DPM) further improves all the three metrics compared with the original TSIT, as shown in Table III. We also provide the visualization results in Figure 13, we clearly observe that the TSIT model with our DPM generates more realistic images with fewer artifacts than the TSIT model without using our DPM on ADE20K. Both experimental results validate that the proposed DPM can be integrated with other methods to further boost the translation performance.

V. Conclusions

We propose a novel Double Pooling GAN (DPGAN) for the challenging layout-to-image translation task. Specifically, we present a novel Double Pooling Module, which consists of the Square-shape Pooling Module (SPM) and the Rectangle-shape Pooling Module (RPM). SPM is used to capture short-range and local semantic dependencies. RPM is used to capture long-range and global semantic dependencies from both horizontal and vertical directions. The outputs of SPM and RPM are combined with the proposed fusion strategies to further effectively enlarge the receptive field of our model. Extensive experiments on five popular datasets demonstrate that DPGAN establishes new state-of-the-art results.

Acknowledgments

This work was supported by the EU H2020 AI4Media No. 951911 project, by the Italy-China collaboration project TALENT:2018YFE0118400, and by the PRIN project PREVUE.

References

[1] 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. 1, 3

[2] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *ICCV*, 2017. 1, 9

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017. 1

[4] 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. 1, 9

[5] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019. 1, 2, 3, 5, 6, 7, 8, 9

[6] X. Liu, G. Yin, J. Shao, X. Wang *et al.*, "Learning to predict layout-to-image conditional convolutions for semantic image synthesis," in *NeurIPS*, 2019. 1, 3, 5, 6, 7, 8, 9

[7] A. Dundar, K. Sapra, G. Liu, A. Tao, and B. Catanzaro, "Panoptic-based image synthesis," in *CVPR*, 2020. 1, 9

[8] 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. 1, 3, 9

[9] H. Tang, D. Xu, Y. Yan, P. H. Torr, and N. Sebe, "Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation," in *CVPR*, 2020. 1, 3, 5, 9

[10] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017. 2, 6

[11] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016. 2, 6

[12] 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. 2, 6

[13] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *CVPR*, 2020. 2, 6

[14] R. Tyleček and R. Šára, "Spatial pattern templates for recognition of objects with regular structure," in *GCPR*, 2013. 2, 6

[15] J. Zhang, J. Chen, H. Tang, W. Wang, Y. Yan, E. Sangineto, and N. Sebe, "Dual in-painting model for unsupervised gaze correction and animation in the wild," in *ACM MM*, 2020. 3

[16] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019. 3

[17] H. Tang, H. Liu, and N. Sebe, "Unified generative adversarial networks for controllable image-to-image translation," *IEEE TIP*, vol. 29, pp. 8916–8929, 2020. 3

[18] G. Liu, H. Tang, H. Latapie, and Y. Yan, "Exocentric to egocentric image generation via parallel generative adversarial network," in *ICASSP*, 2020. 3

[19] H. Tang and N. Sebe, "Total generate: Cycle in cycle generative adversarial networks for generating human faces, hands, bodies, and natural scenes," *IEEE TMM*, 2021. 3

[20] G. Liu, H. Tang, H. Latapie, J. Corso, and Y. Yan, "Cross-view exocentric to egocentric video synthesis," in *ACM MM*, 2021. 3

[21] H. Chen, H. Tang, H. Shi, W. Peng, N. Sebe, and G. Zhao, "Intrinsic-extrinsic preserved gans for unsupervised 3d pose transfer," in *ICCV*, 2021. 3

[22] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. 3

[23] 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. 3

[24] H. Tang, W. Wang, S. Wu, X. Chen, D. Xu, N. Sebe, and Y. Yan, "Expression conditional gan for facial expression-to-expression translation," in *ICIP*, 2019. 3

[25] H. Tang, X. Chen, W. Wang, D. Xu, J. J. Corso, N. Sebe, and Y. Yan, "Attribute-guided sketch generation," in *FG 2019*, 2019. 3

[26] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *ICCV*, 2017. 3

[27] M. Tao, H. Tang, S. Wu, N. Sebe, X.-Y. Jing, F. Wu, and B. Bao, "Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis," *arXiv preprint arXiv:2008.05865*, 2020. 3

[28] H. Tang, S. Bai, L. Zhang, P. H. Torr, and N. Sebe, "Xinggan for person image generation," in *ECCV*, 2020. 3

[29] H. Tang, D. Xu, G. Liu, W. Wang, N. Sebe, and Y. Yan, "Cycle in cycle generative adversarial networks for keypoint-guided image generation," in *ACM MM*, 2019. 3

[30] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *ICCV*, 2019. 3

[31] H. Tang, S. Bai, P. H. Torr, and N. Sebe, "Bipartite graph reasoning gans for person image generation," in *BMVC*, 2020. 3

[32] H. Tang, W. Wang, D. Xu, Y. Yan, and N. Sebe, "Gesturegan for hand gesture-to-gesture translation in the wild," in *ACM MM*, 2018. 3

[33] H. Tang, D. Xu, N. Sebe, and Y. Yan, "Attention-guided generative adversarial networks for unsupervised image-to-image translation," in *IJCNN*, 2019. 3

[34] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation," in *CVPR*, 2019. 3, 9

[35] B. Duan, W. Wang, H. Tang, H. Latapie, and Y. Yan, "Cascade attention guided residue learning gan for cross-modal translation," in *ICPR*, 2021. 3

[36] H. Tang, H. Liu, D. Xu, P. H. Torr, and N. Sebe, "Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks," *IEEE TNNLS*, 2021. 3

[37] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "Sean: Image synthesis with semantic region-adaptive normalization," in *CVPR*, 2020. 3

[38] E. Ntavelis, A. Romero, I. Kastanis, L. Van Gool, and R. Timofte, "Sesame: Semantic editing of scenes by adding, manipulating or erasing objects," in *ECCV*, 2020. 3

[39] Z. Zhu, Z. Xu, A. You, and X. Bai, "Semantically multi-modal image synthesis," in *CVPR*, 2020. 3

[40] H. Tang, S. Bai, and N. Sebe, "Dual attention gans for semantic image synthesis," in *ACM MM*, 2020. 3, 9

[41] H. Tang, X. Qi, D. Xu, P. H. Torr, and N. Sebe, "Edge guided gans with semantic preserving for semantic image synthesis," *arXiv preprint arXiv:2003.13898*, 2020. 3

[42] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017. 3

[43] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *CVPR*, 2019. 3

[44] Q. Hou, L. Zhang, M.-M. Cheng, and J. Feng, "Strip pooling: Rethinking spatial pooling for scene parsing," in *CVPR*, 2020. 3, 4

[45] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *ICCV*, 2019. 3

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 5

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015. 6

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019. 6

[49] 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. 7

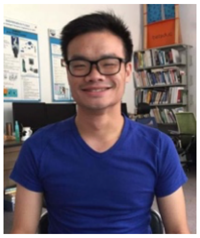
[50] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. 7

[51] X. Qi, Q. Chen, J. Jia, and V. Koltun, "Semi-parametric image synthesis," in *CVPR*, 2018. 9

[52] M. Li, J. Lin, Y. Ding, Z. Liu, J.-Y. Zhu, and S. Han, "Gan compression: Efficient architectures for interactive conditional gans," in *CVPR*, 2020. 9

[53] 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. 9

[54] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *CVPR*, 2017. 8



Hao Tang is currently a Postdoctoral with Computer Vision Lab, ETH Zurich, Switzerland. He received the master's degree from the School of Electronics and Computer Engineering, Peking University, China and the Ph.D. degree from Multimedia and Human Understanding Group, University of Trento, Italy. He was a visiting scholar in the Department of Engineering Science at the University of Oxford. His research interests are deep learning, machine learning, and their applications to computer vision.

Nicu Sebe is Professor in the University of Trento, Italy, where he is leading the research in the areas of multimedia analysis and human behavior understanding. He was the General Co-Chair of the IEEE FG 2008 and ACM Multimedia 2013. He was a program chair of ACM Multimedia 2011 and 2007, ECCV 2016, ICCV 2017 and ICPR 2020. He is a general chair of ACM Multimedia 2022 and a program chair of ECCV 2024. He is a fellow of IAPR.