Local Attention Pyramid for Scene Image Generation

Sang-Heon Shim, Sangeek Hyun, DaeHyun Bae, Jae-Pil Heo*
Sungkyunkwan University

Abstract

In this paper, we first investigate the class-wise visual quality imbalance problem of scene images generated by GANs. The tendency is empirically found that the class-wise visual qualities are highly correlated with the dominance of object classes in the training data in terms of their scales and appearance frequencies. Specifically, the synthesized qualities of small and less frequent object classes tend to be low. To address this, we propose a novel attention module, Local Attention Pyramid (LAP) module tailored for scene image synthesis, that encourages GANs to generate diverse object classes in a high quality by explicit spread of high attention scores to local regions, since objects in scene images are scattered over the entire images. Moreover, our LAP assigns attention scores in a multiple scale to reflect the scale diversity of various objects. The experimental evaluations on three different datasets show consistent improvements in Frechet Inception Distance (FID) and Frechet Segmentation Distance (FSD) over the state-of-the-art baselines. Furthermore, we apply our LAP module to various GANs methods to demonstrate a wide applicability of our LAP module.

1. Introduction

Generative Adversarial Networks (GANs) [7] has lead the remarkable progress in image generation tasks. Its recent advances have reached to generate images nearly indistinguishable from real-world images even on the large-scale benchmarks [15,16]. Nonetheless, generating diverse objects in a scene image has not been received sufficient attention in the field. The datasets which most approaches have been focusing on usually have a single object centered at the image such as human faces. However, the real-world scene images have various objects in diverse regions, as all we know. In this aspect, we raise a question “can the GANs produce scene images including diverse objects in a high quality?”.

In the GANs framework, the generator learns to produce high-quality samples to deceive the discriminator. In the scene image generation, the generator tends to first synthesize big and frequently appeared object classes since it can be a shortcut to reduce a discrepancy between real and fake distributions. This leads the generator to become just an expert at drawing the dominant objects in the scene. It can cause a significant class-wise visual quality imbalance problem, especially low qualities for the object classes with small scales or low appearance frequencies. Let us denote those object classes as non-dominant objects throughout this paper.

To support the above raised argument, we first conduct preliminary study in the scene image generation with the state-of-the-art GANs ($i.e.$ StyleGAN [13]). We measure the class-wise quality scores based on the segmentation results (Sec. 3 and Fig. 1). The pilot study shows that GANs tend to produce non-dominant objects in lower quality. This empirical evidence motivates us to develop a way to ensure balanced quality over diverse object classes.

To mitigate the aforementioned problem, we propose a simple yet effective attention module, Local Attention Pyramid (LAP) tailored for the scene generation, that spreads attentions over the entire image regions and drives high local attentions in various scales. Specifically, our LAP module receives feature maps as its input and first determine coarse locations of each object class by employing depthwise convolution layers. Since depthwise convolution has no intervention between channels, we can infer the feature maps that maintain feature representations of each channel which is highly related to parts of the object [1]. We then divide feature maps into several feature patches and perform an instance normalization for each patch, before feeding them into the sigmoid function. By doing this, LAP amplifies the locally high activation scores in each patch. Thus, it spreads high attentions to diverse regions that can encourage the features of various objects scattered in image. To handle the diversity of object scales, our LAP module infers multiple attention maps with various patch sizes based on the feature map pyramid.

Our main contributions are summarized as follows:

- We highlight that GANs suffer from the class-wise image quality imbalance problem in the scene synthesis task. In its empirical investigation, we found that GANs more concentrate on generating big and frequently ap-

[*] Corresponding author

peared objects and thus provide inferior visual qualities for the non-dominant objects.

- We introduce a novel Local Attention Pyramid (LAP) module to address the class-wise quality imbalance problem that scatters the attentions over different regions with locally high scores by reflecting the characteristics of the scene images.
- Since our LAP is a generic module, it can be applied to various GANs architectures, loss functions, and training strategies. We apply the LAP to various state-of-the-art GANs methods, and evaluate on large-scale scene image benchmarks with various quality measures. Experimental results show that the LAP significantly and consistently improves image quality with very few additional learnable parameters.

2. Related Work

Generative Adversarial Networks. There have been many great works to improve the visual quality of synthesized samples by developing the network architecture of GANs [3, 12–14, 21–23]. As the most pioneering work, Radford et al. [22] proposed DCGAN that employs strided and transposed convolutions. Recently, StyleGAN [13, 14] proposed to inject latent codes to each convolution layer in order to disentangle the factors of variation. MSG-GAN [12] proposed to connect the intermediate layers between the generator and the discriminator to allow the discriminator to look at multiple scales of the intermediate outputs of the generator. In this work, we propose a module based on a novel local attention pyramid mechanism applicable for existing GANs architectures to improve the visual quality of various objects located in diverse regions. We validate the merits of our method by applying our LAP module to the several aforementioned GANs architectures.

Attention mechanism. In recent years, simple and effective attention mechanisms [9, 10, 24, 26] have been proposed in the computer vision literature. Typically, they receive feature maps of a convolution block as input and refine them by attention mechanism. For instance, SENet [9] proposed to employ a channel attention mechanism to enhance inter-channel relationships. On the other hand, Woo et al. [24] introduced CBAM which utilizes both spatial and channel attentions in a sequential way. In the domain of GANs, a few attention techniques have proposed mainly based on the self-attention mechanism [6, 26]. Zhang et al. [26] introduced a self-attention layer for modeling long-range dependency between spatially distant features. Daras et al. [6] reduced the computational complexity of the self-attention process by computing attention maps in a multi-step manner. In this work, we explicitly distribute the activation values to infer the high attention scores for each region. It helps to enhance the feature representation of various objects since the objects are spread out in the scene images.

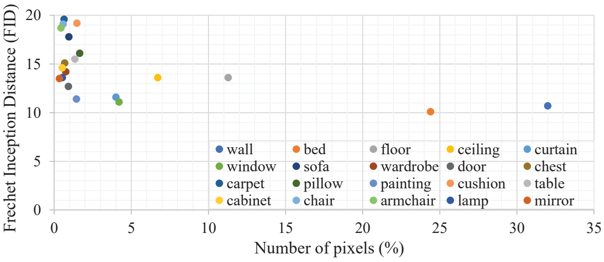


Figure 1. Class-wise FID scores with their pixel percentages $(\frac{\text{the number of pixels of a class}}{\text{total number of pixels}})$ in synthesized scene images. The class-wise visual qualities tend to be negatively correlated with the pixel percentages. It is an empirical evidence that GANs generates less frequent or small-scaled objects in lower qualities, while concentrating more on a few but dominant object classes. Note that, the StyleGAN [13] is utilized for this experiment. For each class, we compute class-wise FID score from 5K of real and fake patches.

3. Problem Definition

In this paper, we restrict our discussion on the scene image generation with GANs. Although there are various objects in the real scene images, GANs tend to concentrate on generating few but dominant objects in high quality but neglect other diverse objects. Since the dominant objects are large-sized and frequently appeared in training dataset, the learning of GANs framework mainly focuses on them, as discussed earlier. Thus, it may be natural to have the low visual qualities especially in the case of non-dominant objects.

To support the raised argument, we perform an experiment to investigate the tendency of class-wise generation quality with respect to the total number of pixels corresponding to each class in a dataset by utilizing the state-of-the-art GANs, StyleGAN. In the experiment, we meausre the class-wise FID scores. Specifically, given a set of real and fake samples, we first compute the segmentation maps using a pre-trained segmentation model [25]. The class-wise cropped image patches are then collected based on the segmentation results. Finally, the class-wise FID values between real and fake patches are computed. One desirable trend might be the consistent FID values across different object classes. However, the experimental results show the tendency that the FID values become worse from dominant to non-dominant objects as reported in Fig. 1. For instance, the FID of the *chair* class are about 9 points higher than the *bed* class.

The aforementioned problem guides us a motivation that GANs can produce diverse objects with higher visual quality, if the GANs try to learn the object concepts in diverse regions during training phase. To this end, we propose a

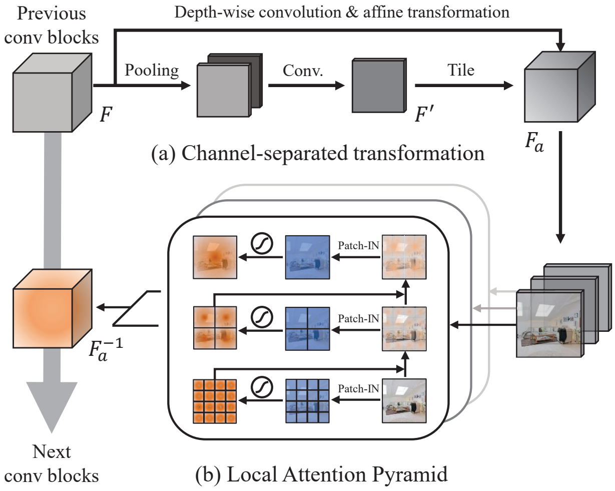

Previous
conv blocks
Depth-wise convolution & affine transformation
Pooling
Conv.
Tile
F
F'
F_a
(a) Channel-separated transformation
Patch-IN
Patch-IN
Patch-IN
F_a^{-1}
Next
conv blocks
(b) Local Attention Pyramid

Figure 2. Overall framework of our proposed attention module. (a) Channel-separated transformation infers coarse location of class-wise spatial attention. We employ depthwise convolution layers to prevent intervention between channels, since it can cause the loss of feature representations of each channel that is highly related to parts of object. (b) Local Attention Pyramid divides the feature maps into multi-scale patches, and applies patch instance normalization followed by the sigmoid into the patches. Then, the computed local attention map are combined in a recursive manner. LAP encourages the model to generate diverse objects with higher quality by amplifying locally high attention scores in every multi-scale patches.

local attention mechanism for scene GANs. Our proposed attention mechanism assumes that a scene image has various objects on its diverse regions. Thus, we aim to explicitly guarantee high attention scores for each spatial region. We first explain our attention mechanism on Sec. 4 and validate the effectiveness against various baseline GANs architectures as well as off-the-shelf attention modules on Sec. 5.

4. Our Approach

Our attention module receives a 3D feature map and transforms it for the next convolution block. We have two main components, 1) channel-separated transformation, and 2) local attention pyramid, as illustrated in Fig. 2. Intuitively, the first component (Sec. 4.1) derives rough locations of each object class by computing channel-wise spatial attention scores without any normalization, while the second one (Sec. 4.2) performs the score normalization with various patch sizes to deal with object scale diversity.

4.1. Channel-separated Transformation

In a high level, we first determine coarse locations of each object class or its parts by transforming the input features. Since each channel of GANs is highly related to parts of an object class as discussed in [1], we compute a spatial attention score map for each channel unlike typical attention mechanisms utilizing a global attention map.

In typical attention mechanisms, the locations to focus are computed by convolution operations. However, the general convolution operation is not appropriate for our purpose to compute class-wise spatial attention scores since it involves multiple channels during the computation. Such intervention among channels can cause a loss of channel-wise information highly related to parts of the object [1]. In downstream tasks, condensation operations help to highlight discriminative features, however, they disturb to spread focuses to the diverse objects in generation tasks. As a result, the convolution layer, where the filters are fully-connected across the channel axis, is not suitable for our LAP module.

To meet our purpose, we utilize the depthwise convolution layer, since it does not allow the intervention among channels. This way, we can compute locations to focus for each channel independently. Specifically, we design our channel-separated transformation component upon the CBAM [24] and add a pathway of depthwise convolution layers on it. Note that, we do not perform any normalization or activation unlike other attention modules but pass the raw values to the LAP, since the LAP encourages to generate objects in diverse scales based on normalization within various sized patches.

Let us describe the practical implementation of our channel-separated transformation. Given an input intermediate feature map $F \in \mathbb{R}^{C\times H\times W}$, we first squeeze the input feature map along channel dimension through poolings and convolution layers as follows:

$$F' = \mathrm{Conv}\big(\big[\mathrm{AvgPool}(F); \mathrm{MaxPool}(F)\big]\big), \qquad (1)$$

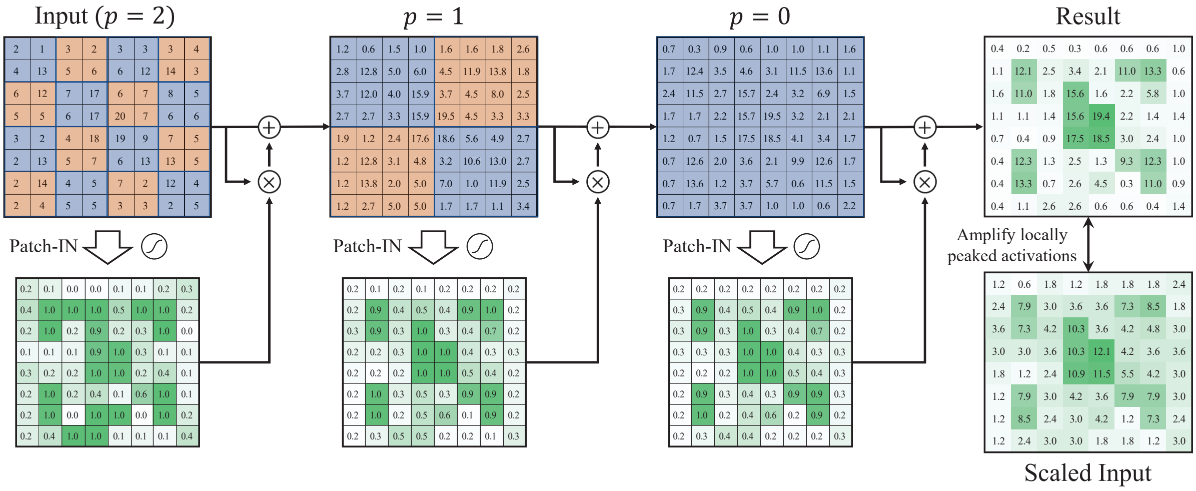
where $\mathrm{Conv}(\cdot)$, $\mathrm{AvgPool}(\cdot)$ and $\mathrm{MaxPool}(\cdot)$ indicate a convolution operation with the 3×3 kernel size, average pooling and max pooling, respectively. By doing this, the general feature representation for locations to focus is contained to the $F' \in \mathbb{R}^{1\times H\times W}$ [24]. As aforementioned, we perform the channel-separated transformation from the input feature map F, and then modulate the F' as follows:

$$F_a = \mathrm{Conv}_d^{\gamma}(F) \odot T(F', C) + \mathrm{Conv}_d^{\beta}(F), \qquad (2)$$

where Conv_d^{γ} and Conv_d^{β} denote the 3×3 depthwise convolution operations but having different learnable parameters, $\odot$ indicates an element-wise multiplication and $T(f, n)$ is the tile function that copies a feature map f with n times along channel dimension. We feed the computed feature maps $F_a \in \mathbb{R}^{C\times H\times W}$ to our local attention pyramid.

4.2. Local Attention Pyramid

We can directly utilize the transformed feature map computed in Sec. 4.1 with a proper activation function such as the sigmoid. However, we still have a concern that the channels corresponding to the dominant objects can suppress the



2	1	3	2	3	3	3	4
4	13	5	6	6	12	14	3
6	12	7	17	6	7	8	5
5	5	6	17	20	7	6	6
3	2	4	18	19	9	7	5
2	13	5	7	6	13	13	5
2	14	4	5	7	2	12	4
2	4	5	5	3	3	2	5

1.2	0.6	1.5	1.0	1.6	1.6	1.8	2.6
2.8	12.8	5.0	6.0	4.5	11.9	13.8	1.8
3.7	12.0	4.0	15.9	3.7	4.5	8.0	2.5
2.7	2.7	3.3	15.9	19.5	4.5	3.3	3.3
1.9	1.2	2.4	17.6	18.6	5.6	4.9	2.7
1.2	12.8	3.1	4.8	3.2	10.6	13.0	2.7
1.2	13.8	2.0	5.0	7.0	1.0	11.9	2.5
1.2	2.7	5.0	5.0	1.7	1.7	1.1	3.4

0.7	0.3	0.9	0.6	1.0	1.0	1.1	1.6
1.7	12.4	3.5	4.6	3.1	11.5	13.6	1.1
2.4	11.5	2.7	15.7	2.4	3.2	6.9	1.5
1.7	1.7	2.2	15.7	19.5	3.2	2.1	2.1
1.2	0.7	1.5	17.5	18.5	4.1	3.4	1.7
0.7	12.6	2.0	3.6	2.1	9.9	12.6	1.7
0.7	13.6	1.2	3.7	5.7	0.6	11.5	1.5
0.7	1.7	3.7	3.7	1.0	1.0	0.6	2.2

0.4	0.2	0.5	0.3	0.6	0.6	0.6	1.0
1.1	12.1	2.5	3.4	2.1	11.0	13.3	0.6
1.6	11.0	1.8	15.6	1.6	2.2	5.8	1.0
1.1	1.1	1.4	15.6	19.4	2.2	1.4	1.4
0.7	0.4	0.9	17.5	18.5	3.0	2.4	1.0
0.4	12.3	1.3	2.5	1.3	9.3	12.3	1.0
0.4	13.3	0.7	2.6	4.5	0.3	11.0	0.9
0.4	1.1	2.6	2.6	0.6	0.6	0.4	1.4

0.2	0.1	0.0	0.0	0.1	0.1	0.2	0.3
0.4	1.0	1.0	1.0	0.5	1.0	1.0	0.2
0.2	1.0	0.2	0.9	0.2	0.3	1.0	0.0
0.1	0.1	0.1	0.9	1.0	0.3	0.1	0.1
0.3	0.2	0.2	1.0	1.0	0.2	0.4	0.1
0.2	1.0	0.2	0.4	0.1	0.6	1.0	0.1
0.2	1.0	0.0	1.0	1.0	0.0	1.0	0.2
0.2	0.4	1.0	1.0	0.1	0.1	0.1	0.4

0.2	0.1	0.2	0.1	0.2	0.2	0.2	0.2
0.2	0.9	0.4	0.5	0.4	0.9	1.0	0.2
0.3	0.9	0.3	1.0	0.3	0.4	0.7	0.2
0.2	0.2	0.3	1.0	1.0	0.4	0.3	0.3
0.2	0.2	0.2	1.0	1.0	0.5	0.4	0.2
0.2	1.0	0.3	0.5	0.3	0.9	0.9	0.2
0.2	1.0	0.2	0.5	0.6	0.1	0.9	0.2
0.2	0.3	0.5	0.5	0.2	0.2	0.1	0.3

0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
0.3	0.9	0.4	0.5	0.4	0.9	1.0	0.2
0.3	0.9	0.3	1.0	0.3	0.4	0.7	0.2
0.3	0.3	0.3	1.0	1.0	0.4	0.3	0.3
0.2	0.2	0.2	1.0	1.0	0.5	0.4	0.3
0.2	0.9	0.3	0.4	0.3	0.9	0.9	0.3
0.2	1.0	0.2	0.4	0.6	0.2	0.9	0.2
0.2	0.3	0.4	0.4	0.2	0.2	0.2	0.3

1.2	0.6	1.8	1.2	1.8	1.8	1.8	2.4
2.4	7.9	3.0	3.6	3.6	7.3	8.5	1.8
3.6	7.3	4.2	10.3	3.6	4.2	4.8	3.0
3.0	3.0	3.6	10.3	12.1	4.2	3.6	3.6
1.8	1.2	2.4	10.9	11.5	5.5	4.2	3.0
1.2	7.9	3.0	4.2	3.6	7.9	7.9	3.0
1.2	8.5	2.4	3.0	4.2	1.2	7.3	2.4
1.2	2.4	3.0	3.0	1.8	1.8	1.2	3.0

Figure 3. A toy example describing our local attention pyramid module. The example illustrates inferring local attention pyramid on 8×8 feature map with pyramid level $p = 2$. The most right image shows the output of LAP module. To clearly compare "Result" and "Input", we scale the "Input" to have the same amount of total values with "Result". As shown, the LAP amplifies locally peaked activations.

others in the end, since they are highly likely to cover spatially broad regions in the feature maps. Since it is harmful for generation of diverse objects in high quality and channels of the feature map have different distributions, a proper channel-wise normalization should be performed. Moreover, the importance of locally high scores (i.e. having greater scores than its neighbors) should be distinguished depending on the object class related to the channels. For instance, a locally high scored cell related to a small-scaled object needs to receive more attention than one for a large object.

The aforementioned issue motivates us to propose our Local Attention Pyramid (LAP). The LAP is developed to amplify the locally high activation scores within different sized patches. A base operation of the LAP is to divide a channel of the feature map into a grid and perform a normalization within each grid cell independently. After the normalization, the whole channel proceeds to the sigmoid function. Such an independent normalization within a patch results the emphasis of local peaks and spread of high attentions to diverse regions. Furthermore, we perform the patch-wise normalization with various sized patches based on a spatial pyramid, since the size of patch for the normalization is closely related to the object scale. As a result, the LAP assigns attention scores with consideration on the scale diversity of various objects.

Attention map at pyramid level p. For the sake of simplicity, we explain with the assumption that the shape of both F_a and feature patches are square. Given an input F_a, the patch size s^p at pyramid level p is defined as follows:

$$s^p = \tfrac{l}{2^p}, \tag{3}$$

where l is the length of longer side of F_a. We then normalize each $s^p \times s^p$ sized feature patch based on the following instance normalization function:

$$\text{Patch-IN}(f, k_h, k_w, s_h, s_w), \tag{4}$$

where Patch-IN is performed for the input f with a $k_h \times k_w$ sized sliding window and a stride size $s_{(\cdot)}$ along the spatial axis. The Patch-IN function calibrates the highest activated value of a feature patch to nearby 1. Thus, the sigmoid results of the Patch-IN output always include high attention scores within the patch. Our LAP module computes the attention map $M(F_a, s^p)$ as follows:

$$M(F_a, s^p) = \sigma\big(\text{Patch-IN}(F_a, s^p, s^p, s^p, s^p)\big), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function. In practice, we set the $k_{(\cdot)} = s_{(\cdot)} = s^p$ so that the Patch-IN operates on non-overlapping feature patches.

Multi-scale attention maps and their aggregation. We compute multi-scale attention maps in a recursive manner as illustrated in Fig. 3.

Let us denote the feature map at pyramid level p as F_a^p. We begin with the smallest patch size defined at the highest level (Eq. 3). At each level, our LAP module first computes an attention map based on Eq. 5. We then apply the attention to the current feature map and pass the result to the next level as the following recursive equation:

$$F_a^{p-1} = \alpha \cdot F_a^p + (1 - \alpha) \cdot (M(F_a^p, s^p) \odot F_a^p), \tag{6}$$

where α is a decay factor that controls the amount of influence from previous attention scores, and $\odot$ denote element-wise multiplication. Note that, we consistently set $\alpha = 0.5$

Dataset	Method	FID	FSD
COCO-stuff	StyleGAN	31.6	290.4
	StyleGAN (w/ ours)	**30.7**	**283.1**
	MSG-StyleGAN	76.7	655.8
	MSG-StyleGAN (w/ ours)	**65.3**	**563.1**
LSUN Bedroom	StyleGAN	4.2	38.4
	StyleGAN (w/ ours)	**3.4**	**34.5**
	MSG-StyleGAN	6.5	54.4
	MSG-StyleGAN (w/ ours)	**5.9**	**49.1**
	MSG-DCGAN	60.9	303.4
	MSG-DCGAN (w/ ours)	**24.9**	**238.6**
LSUN Combined	StyleGAN	4.6	37.1
	StyleGAN (w/ ours)	**3.9**	**33.2**
	MSG-StyleGAN	7.7	74.2
	MSG-StyleGAN (w/ ours)	**6.6**	**56.5**
	MSG-DCGAN	81.4	615.1
	MSG-DCGAN (w/ ours)	**46.6**	**426.7**

Table 1. Experimental results on large-scale scene datasets. We train baseline models with the official code and recommended hyperparameters. All models are trained for generating samples of 256×256 resolution.

for all the following experiments in this paper. We perform the aforementioned recursion while the patch size s^p covers the whole size of the F_a. The F_a^{-1} is the final output of our LAP module, and we add it to the intermediate feature maps F as illustrated in Fig. 2.

Multiple aspect ratios. In practical implementation, we adopt multiple aspect ratios of feature patches to reflect various object shapes. Specifically, our LAP module has three different kinds of patch shapes: square, wide, and long shapes. Therefore, the Eq. 3 is redefined as follows:

$$\begin{cases} s_h^p = \frac{H}{2^p}, & s_w^p = \frac{W}{2^p} \quad \text{square} \\ s_h^p = \frac{H}{2^p}, & s_w^p = W \quad \text{wide} \\ s_h^p = H, & s_w^p = \frac{W}{2^p} \quad \text{long} \end{cases} \tag{7}$$

where H and W denote the height and width of the F_a, respectively. And p denotes the pyramid level. According to Eq. 7, we also rewrite Eq. 5 as follows:

$$M(F_a, s_h^p, s_w^p) = \sigma\big(\text{Patch-IN}(F_a, s_h^p, s_w^p, s_h^p, s_w^p)\big), \tag{8}$$

where the stride size is the same with the kernel sizes so that the Patch-IN is performed on non-overlapping patches.

5. Experiments

In this section, we evaluate our proposed method by extensive experiments with the state-of-the-art unconditional GANs approaches including MSG-DCGAN, MSG-StyleGAN [12], and StyleGAN [13].

Method	# params
MSG-DCGAN	41.77M
MSG-DCGAN (w/ ours)	+ 0.1M (+ 0.2%)
StyleGAN	49.13M
StyleGAN (w/ ours)	+ 0.1M (+ 0.2%)
MSG-StyleGAN	49.13M
MSG-StyleGAN (w/ ours)	+ 0.1M (+ 0.2%)

Table 2. The number of learnable parameters of baselines and ours.

5.1. Datasets

We utilize the following benchmarks to evaluate the performance of our LAP and baselines:

COCO-stuff [4] is derived from COCO dataset [19]. It contains $118,000$ training images captured from indoor and outdoor scenes. It has 182 semantic classes.

LSUN bedroom is consist of 3M bedroom images. It has more than 20 semantic classes, which were investigated by employing the pre-trained segmentation model [25] in the published work [1, 2].

LSUN combined is a custom dataset that is a combination of LSUN kitchen, LSUN dining room, and LSUN living room. The dataset contains 4.1M images. We found more than 20 semantic classes by using a segmentation model [25].

5.2. Implementation Details

We apply our LAP module in every convolution blocks of the generator, excepts for 4^2 sized feature maps. In detail, we apply different pyramid level p according to the size of feature map. Specifically, starting from the pyramid level $p = 1$ at 8^2 sized feature maps, p is linearly increased for every 4 times up-scaled feature maps. Thus, $p = 2$ and $p = 3$ is starting at 32^2 and 128^2 sized feature maps, respectively.

For the training details, we follow the default training settings in the official implementation of StyleGAN [13] and MSG-GAN [12]. For instance, we allow training iterations until the discriminator sees 25M real images and we employ non-saturating loss [7] with R1 regularization [20] using $\gamma = 10$. Also, the data augmentation by horizontal flipping is not employed for all experiments.

We consistently utilize 2 GPUs to train all the tested models, much fewer than 8 GPUs used in [14], due to the lack of research equipment. We realized that the performance of GANs can be different according to training configurations, such as numbers of GPUs and iterations, as discussed and reported in [12, 18]. For instance, the FID scores of StyleGAN trained on LSUN Churches are different in [12] and [14], due to the aforementioned issues. In this work, we compare all the tested models in a fair configuration and rather focus on clear demonstration of the merits of the LAP in terms of relative performance improvements by the LAP over the baselines.

Bedroom	wall	bed	floor	ceiling	window	curtain	pillow	cushion	painting	table	sofa	door	wardrobe	chest	carpet	chair	lamp	cabinet	armchair	mirror
StyleGAN	10.7	10.1	**13.6**	**13.6**	**11.1**	11.6	16.1	19.2	11.4	15.5	17.8	12.7	14.2	15.1	19.6	19.1	**13.6**	14.6	18.7	13.5
+ LAP	**10.0**	**9.6**	15.1	13.7	11.3	**11.4**	**15.9**	**19.1**	11.4	**14.7**	**17.1**	**11.8**	**12.8**	**14.1**	**18.3**	**17.8**	13.8	**13.7**	**18.6**	**12.8**

Combined	wall	floor	cabinet	ceiling	window	chair	table	sofa	surface	curtain	armchair	stove	kitchen table	painting	refrigerator	door	carpet	coffee table	cushion	shelf
StyleGAN	**11.0**	14.4	11.2	**13.1**	11.4	15.0	15.6	16.1	12.0	**13.0**	18.5	13.8	11.5	**12.2**	14.0	12.4	**19.1**	17.4	20.7	17.8
+ LAP	11.2	**13.5**	**10.8**	13.3	**10.9**	**14.9**	**14.7**	**14.5**	**11.7**	13.3	**16.2**	**12.8**	**10.8**	12.6	**12.0**	12.4	20.8	**16.4**	**18.0**	**16.5**

Table 3. Class-wise FID scores on LSUN bedroom (upper) and LSUN combined (lower). We compute class-wise FID scores from the class-wise cropped image patches based on a pre-trained segmentation model. Note that, "+ LAP" denotes StyleGAN (w/ ours).

5.3. Evaluation Metrics

FID. Frechet Inception Distance (FID) [8] has been widely used to measure the image generation quality. The FID score is computed by the sequential operations: first, inception features are extracted from a intermediate layer of inception model. Then, multidimensional Gaussian with mean m and covariance matrix C is obtained from the inception features, and the Frechet distance is computed as follows,

$$d^2\Big((m^r, C^r), (m^f, C^f)\Big) = \| m^r - m^f \|_2^2 + \text{Tr}\Big(C^r + C^f - 2(C^r C^f)^{1/2}\Big), \tag{9}$$

where Gaussians (m^r, C^r) and (m^f, C^f) are obtained from real and fake samples, respectively. Also, $\text{Tr}(\cdot)$ indicates the trace of a matrix. Note that, we measure FID scores from 50K of real and fake images throughout this paper.

Class-wise FID. To further quantify the per-class visual quality, we propose a class-wise FID. Instead of measuring from the whole-sized images, we first compute the segmentation maps using a pre-trained segmentation model. We then collect class-wise cropped image patches based on segmentation results. Finally, we measure the class-wise FID scores between real and fake patches by employing the Eq. 9. We denote it as class-wise FID. Note that, we consistently measure the class-wise FID scores from 5K of real and fake image patches throughout this paper.

FSD. Recently, Bau et al. [2] have proposed Frechet Segmentation Distance (FSD) for quantifying a degree of mode dropping in the scene datasets. Instead of using the inception features, it obtains Gaussian (m, C) from histograms of semantic segmentation labels. Specifically, the frequency of the class label c for a segmentation map x, $H(x, c)$, is computed based on the percentage of the pixels predicted to the class c as follows:

$$H(x, c) = \frac{1}{WH} \sum_{i=1}^{W \times H} B(x^i, c), \tag{10}$$

where

$$B(x^i, c) = \begin{cases} 1 & \text{when } s(x^i) = c \\ 0 & \text{when } s(x^i) \neq c, \end{cases} \tag{11}$$

and $s(x^i)$ denote a predicted label of a segmentation map x at pixel position i. The function $H(x, c)$ iterates over all semantic classes and a set of segmentation maps. After obtaining Gaussians (m^r, C^r) and (m^f, C^f) for real and fake samples, the Frechet distance is calculated from the Eq. 9. Note that, DeepLab-v2 [5] and UperNet101 [25] are employed as a pre-trained segmentation model for COCO-stuff and LSUN datasets, respectively. Following the Bau et al. [2], we measure FSD scores from the 10K real and fake images throughout this paper.

5.4. Quantitative Results

To show that our LAP is a generic module, we apply the LAP to three baselines that have different GANs architectures, loss functions, and training strategies. For instance, MSG-DCGAN trains a DCGAN based model with a relativistic-hinge loss function [11]. Also, they do not use progressive learning.

Traditional GANs architecture. We first apply our LAP module to MSG-DCGAN [12] and train on LSUN datasets. It is a hard task for the DCGAN based architecture to generate 256×256 fake samples. As reported in Table. 1, our LAP module achieves better visual quality than vanilla MSG-DCGAN. For instance, in LSUN bedroom, MSG-GAN (w/ ours) achieves the FID score of 24.9 that is significantly improved (36.0 FID points) over vanilla MSG-DCGAN. Note that, we tried to train DCGAN based architecture in COCO-stuff, however, it suffered from a serious mode collapse due to the higher scene complexity and lack of training images.

Recent GANs architecture. We further apply our LAP module to both StyleGAN and MSG-StyleGAN. We conduct extensive experiments on COCO-stuff and LSUN datasets and the results are reported in Table. 1. The results show that our LAP module consistently improves the FID and FSD scores of baseline methods. Specifically, in LSUN bedroom, StyleGAN (w/ ours) achieves an FID score of 3.4, improving 0.8 points, over vanilla StyleGAN, as well as increases the FSD score from 38.4 to 34.5. Similar trends can be found in other datasets and the experimental results of MSG-StyleGAN. Those results clearly validate that our LAP module boosts the learning capability of the recent scene image generator. Note that, we additionally provide

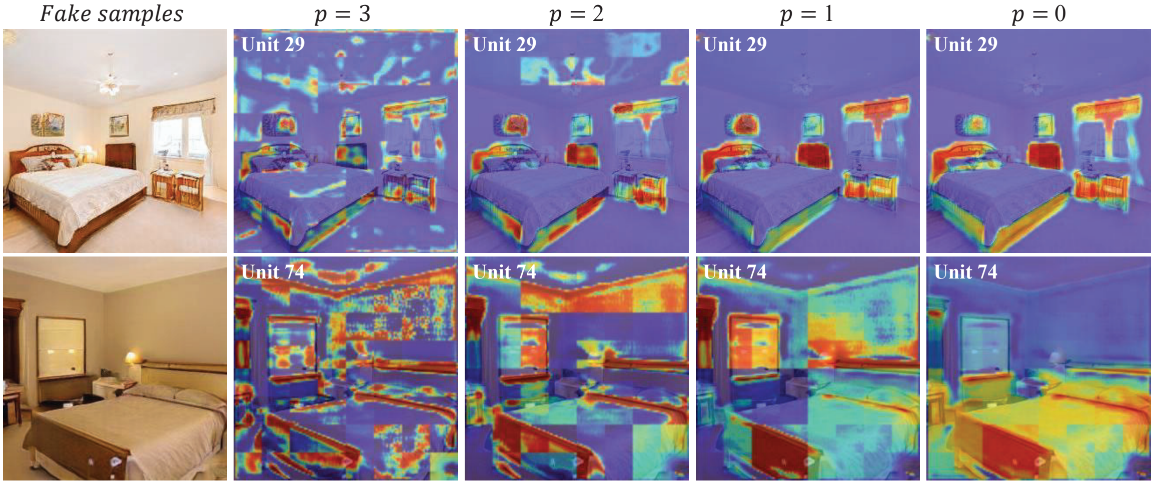


Figure 4. Visualization of local attention pyramid results. The images at the first column are fake samples generated by StyleGAN (w/ ours). Other columns show the visualized attention maps corresponding to each pyramid level.

the results of LSUN church, and the detailed quality analysis based on precision and recall measures according to [17] in Appendix.

The number of learnable parameters are reported in Table. 2. Our LAP module requires only 0.2% additional parameters from baseline methods. It tells that the improvement of GANs performance is not caused by increasing model capacity. Note that, the baseline models have the same number of convolution blocks so that the increment of model parameters are the same when LAP is applied.

Class-wise visual quality. Table. 3 reports the class-wise FID scores against generator of StyleGAN. As reported, our LAP module consistently improves the FID scores over diverse objects, so that it alleviates the class-wise visual quality imbalance problem in scene images generated by GANs. Note that, we did not measure the class-wise FID scores on COCO-stuff dataset mainly because the visual quality of generated samples are too low so that the pre-trained segmentation models do not segment the objects well. We also inform that relative per class improvements with respect to appearance frequency and size of objects are reported in Appendix.

5.5. Qualitative Results

Visualization of local attention pyramid is provided in Fig. 4. First, at $p = 3$, our local attention pyramid computes high attention scores for every smallest region. Thus, the high attention values are also multiplied to the empty regions such as $wall$. By recurrently inferring the attention maps from fine to coarse levels, our LAP module gradually produces the spatial attention maps focusing on diverse objects of the scene image.

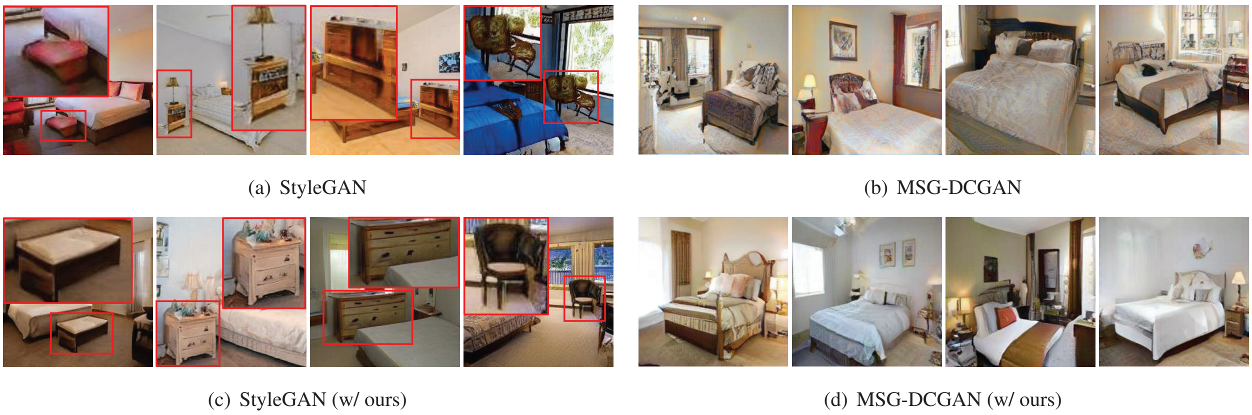
Generated samples are shown in Fig. 5. We utilize StyleGAN and MSG-DCGAN as baseline models for the qualitative comparison. For the samples generated by StyleGAN based models, the overall sample quality is similar. However, when looking in detail, the StyleGAN draws local objects with unrealistic shapes (see leftmost in upper rows), whereas StyleGAN with LAP produces more rigid object shapes (see leftmost in lower rows). For instance, the StyleGAN generator produces $chair$ sample without legs, which shows lower visual quality than the samples of StyleGAN with LAP. Comparison with models based on MSG-DCGAN shows more clear difference. Additional qualitative results are available in Appendix.

Method		# params	FID	FSD
StyleGAN (w/ S.A.)	[26]	49.79M	5.9	46.9
StyleGAN (w/ CBAM)	[24]	49.36M	4.9	45.6
StyleGAN (w/ ours)		49.24M	**3.4**	**34.5**

Table 4. Comparison with prior attention mechanisms. All tested models are trained until the discriminator sees 25M images in the LSUN bedroom.

5.6. Comparison against attention mechanisms

We also compare LAP module against prior attention mechanisms. Specifically, we choose CBAM [24] and self-attention mechanism (S.A., [26]) for the comparison. We apply them to a baseline architecture of StyleGAN and train each model on LSUN bedroom. As reported in Table. 4, our LAP module achieves better visual quality scores than competing methods. Also, we have found that CBAM and S.A. rather degrade the image generation performance of StyleGAN, achieving FID of 4.9 and 5.9 respectively. We suspect that it is mainly because they infer a single attention



(a) StyleGAN

(b) MSG-DCGAN

(c) StyleGAN (w/ ours)

(d) MSG-DCGAN (w/ ours)

Figure 5. Qualitative comparison. The generated samples of StyleGAN and MSG-DCGAN contain objects with squeezed shapes and noisy textures, whereas the samples of our LAP show more rigid object shape and detailed texture.

map so that only few objects are highly likely to obtain high attention scores. Recall again that our LAP module achieves 3.4 FID points with requiring only small amount of additional learnable parameters.

5.7. Ablation Study

We further discuss about the effect of individual components of LAP module. To do this, we trained each ablated version of StyleGAN until the discriminator sees 12M real images in the LSUN bedroom. The scores are reported on Table. 5.

Channel-separated Transformation. We first investigate the effect of Sec. 4.1. Specifically, we add component of Sec. 4.1 and typical spatial attention mechanism to StyleGAN. This attention mechanism computes attention maps from whole-sized feature maps. As shown in second row of the table, employing both the channel-separated transformation and attention mechanism improves the FID from 6.5 to 5.3.

On the other hand, the aforementioned result may not clearly validate the merits of employing depthwise convolution layers. It is mainly because the performance gain can be obtained from the attention mechanism. To investigate it, we build a new ablated model where we remove the depthwise convolution layers from the model of the second row. Thus, it computes a spatial attention map with single channel but other operations are equal. As reported in the last row, the ablated model worsens the FID score from 6.5 to 8.3. It is a contrary result compared to the second row and the result confirms the necessity of channel-separated transformation.

Local Attention Pyramid. We then add the local attention pyramid to the second row's model. Thus, it is full implemented version of our proposed method. As reported in the third row, the full model further improves the visual quality performance, achieving an FID of 4.9. It is meaningful improvement since adding it requires only 11K additional parameters. Those ablation study confirms that each component of our LAP module is mutually beneficial.

Method	# params	FID	FSD
StyleGAN	49.13M	6.5	51.9
+ Sec. 4.1	+ 95K	5.3	52.8
+ Sec. 4.2	+ 11K	**4.9**	**47.0**
StyleGAN	49.13M	6.5	51.9
+ Eq. 1	+ 0.2K	8.3	53.4

Table 5. The ablation study of our method. We train all models until the discriminator sees 12M of real images in LSUN bedroom.

6. Conclusion

In this work, we raise a new problem that GANs tend to concentrate on generating big and frequently appeared objects rather than producing diverse objects with balanced visual quality. To alleviate it, we have proposed the Local Attention Pyramid (LAP) module to generate diverse objects with high-quality. Specifically, our LAP module explicitly guarantees to compute high attention scores to diverse regions. Since various objects are spread out in the scene image, the latent representation of diverse objects can be enhanced. In addition, our LAP module computes multi-scale attention maps for handling the diversity of objects scales. Extensive experiments have demonstrated that our LAP module consistently improves the visual quality in terms of FID and FSD metrics. Furthermore, we have also validated that LAP is a generic module by applying it to various prior GANs methods.

Acknowledgements. This work was supported in part by MCST/KOCCA (No. R2020070002), MSIT/IITP (No. 2020-0-00973, 2019-0-00421, 2020-0-01821, 2021-0-02003, and 2021-0-02068), and MSITKNPA/KIPoT (Police Lab 2.0, No. 210121M06).

References

[1] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[2] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4502–4511, 2019.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[6] Giannis Daras, Augustus Odena, Han Zhang, and Alexandros G Dimakis. Your local gan: Designing two dimensional local attention mechanisms for generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14531–14539, 2020.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

[10] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip Torr. Learn to pay attention. In *International Conference on Learning Representations*, 2018.

[11] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *International Conference on Learning Representations*, 2019.

[12] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[15] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, page 2012.

[17] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. COCO-GAN: generation by parts via conditional coordinating. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[20] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[21] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[22] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016.

[23] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2020.

[24] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[25] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

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