Image Synthesis from Layout with Locality-Aware Mask Adaption

Zejian Li[1] , Jingyu Wu[1] , Immanuel Koh[2] , Yongchuan Tang[1] , Lingyun Sun[1,3]
[1]Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Zhejiang University
[2]Singapore University of Technology and Design
[3]Collaborative Innovation Center of AI by MOE and Zhejiang Provincial Government
{zejianlee, wujingyu}@zju.edu.cn, immanuel_koh@sutd.edu.sg, {yctang, sunly}@zju.edu.cn

Abstract

This paper is concerned with synthesizing images conditioned on a layout (a set of bounding boxes with object categories). Existing works construct a layout-mask-image pipeline. Object masks are generated separately and mapped to bounding boxes to form a whole semantic segmentation mask (layout-to-mask), with which a new image is generated (mask-to-image). However, overlapped boxes in layouts result in overlapped object masks, which reduces the mask clarity and causes confusion in image generation. We hypothesize the importance of generating clean and semantically clear semantic masks. The hypothesis is supported by the finding that the performance of state-of-the-art LostGAN decreases when input masks are tainted. Motivated by this hypothesis, we propose Locality-Aware Mask Adaption (LAMA) module to adapt overlapped or nearby object masks in the generation. Experimental results show our proposed model with LAMA outperforms existing approaches regarding visual fidelity and alignment with input layouts. On COCO-stuff in 256×256, our method improves the state-of-the-art FID score from 41.65 to 31.12 and the SceneFID from 22.00 to 18.64.

1. Introduction

This paper is concerned with image generation from layouts, a specific task of conditional image synthesis. A layout is a set of bounding boxes with object categories, representing the positions, sizes and classes of objects in an image. The layout-to-image generation task is to convert the bounding boxes to a photorealistic image without segmentation annotation [37]. This task remains a challenging problem but provides a promising approach to understanding visual relations in images via analysis-by-synthesis. It also has a wide range of applications such as human-computer collaborative creation, where a potentially desired picture is generated according to the layout given by a human.

Existing works construct an effective layout-mask-image pipeline [32, 38]. As a layout only provides a coarse configuration of the desired image, a semantic segmentation mask is generated first and then translated to the final image (Fig. 1). A semantic segmentation mask specifies the category distribution of each pixel [7]. In the degraded and simple case, each pixel belongs to only a single category. How-

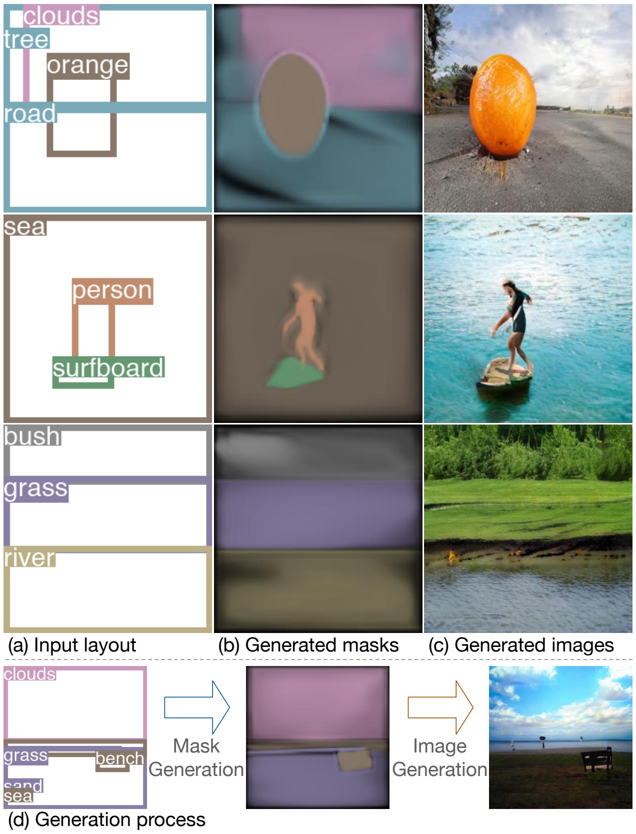

Figure 1. **An illustration of image generation from layouts** (bounding boxes with categories). Based on the input layout as shown in column (a), the generator first synthesizes a semantic mask in (b) and then translates the mask to an image in (c). (d) summarizes the generation process. Masks and images are generated by our model with the Locality-Aware Mask Adaption.

ever, a generated mask may not be "clean" and semantically clear but has overlapping masks and mixed categories of pixels. This is because the overall semantic mask is formed by aggregating object masks generated separately [32, 38]. As a result, when the bounding boxes overlap, which is a common situation, the object masks also overlap and the category of pixels can be confusing. Such an ambiguous semantic mask causes difficulty for image generation, as it is not clear for the generator to synthesize which object in the overlapped area. Therefore, we hypothesize that a semantically clear mask with little overlaps is important for high-quality image generation.

Hypothesis justification. To test our hypothesis, we train the mask-to-image component of LostGAN [32] with ground-truth masks (Sec. 3). When also tested on ground-truth masks, this mask-to-image model has a significantly better performance than the original layout-to-image one on visual fidelity and layout alignment. When tested with increasingly tainted masks, the model's performance decays gradually (Fig. 2). This shows the impact of mask clarity on image quality and supports our hypothesis.

Our method. Motivated by the above hypothesis, we propose Locality-Aware Mask Adaption (LAMA) module in the context of layout-to-image generation (Sec. 4). After a raw semantic mask is formed by combining object masks, LAMA aims to adapt the raw mask to a cleaner one by considering objects' local relation (Fig. 3). It scales the mask values of each object in each pixel individually with a learned matching mechanism. Empirically, when two object masks originally overlap, LAMA enables the mask of the background category to shrink precisely in pixels where the foreground mask is located (Fig. 4). Finally, the generated mask and the category are injected into the image generation pipeline via normalization layers.

Experimental results. Empirically the proposed model outperforms the state-of-the-art methods in terms of visual quality of images and layout alignment (Sec. 6.1). On COCO-stuff in 256×256, our method improves the state-of-the-art FID [10] score from 41.65 to 31.12 and the SceneFID [35] from 22.00 to 18.64. Besides, a quantitative comparison show LAMA refines the raw masks to cleaner ones with smaller entropies (Sec. 6.2). In addition, the contribution of LAMA module is exemplified in a control experiment. The performance of our model decays without LAMA, while that of LostGAN [32] is improved when combined with LAMA (Sec 6.3).

Our main contribution lies in three aspects:

1. **Conceptual contribution.** A hypothesis on the importance of generating clean semantic masks is presented and preliminarily verified by experimental results.

2. **Technical contribution.** Locality-Aware Mask Adaption (LAMA) is proposed for generating a clean and sharp semantic mask to facilitate image generation. Our contribution of LAMA is orthogonal to existing works, and LAMA can be easily integrated with other methods.

3. **Metric contribution.** An new and challenging evaluation metric termed YOLO scores is proposed to measure the layout alignment of objects.

We provide a complete implementation with PyTorch [27] including source code and evaluation metrics[1].

Table 1. A brief summary of how existing methods aggregate overlapped object masks/features. $\diamond$ means masks are generated with BiConvLSTM [31] modeling mask relations during generation.

Methods	Trained with GT masks	Aggregating overlapped object masks/features
Layout2Im [37, 38]	No	ConvLSTM
LostGAN [32, 34]	No	Normalize
OC-GAN [35]	No	Normalize and concatenate with layout boundaries
Hong *et al.* [13]	Yes	Sum $\diamond$
Obj-GAN [20]	Yes	Maxpooling $\diamond$
OP-GAN [11, 12]	Yes	Sum in global pathway and replacement in object path
SG2IM [15]	Yes	Sum
Ashual and Wolf [1]	Yes	Normalize
Ours	No	Adapt and normalize

2. Related Works

Image Generation from Layouts. Existing layout-to-image methods divide the task into generating semantic masks from layouts and image synthesis from masks. Layout2Im [37] proposes the layout-to-image task, and it generates features for bounding boxes to form masks and styles of objects. Its extension [38] further generates a mask directly for each box. Similarly, LostGAN-V1 [32] generates masks for boxes individually and forms a whole semantic mask, and the semantic mask is injected into the mask-to-image generator via ISLA-Norm layers. LostGAN-V2 [34] further integrate masks learned from feature maps at different generation stages. From another perspective, OC-GAN [35] improves layout fidelity by maximizing the similarity between the image embeddings and the scene graph inferred from the layout. Additionally, DCL [33] maps the generated mask and the inferred mask from the generated image to maximize structural consensus. These methods use only layouts and images. BachGAN [21] and Attribute-Guided Layout2Im [22] further use masks and attribute annotations. Layout-to-image generation also serves as a useful subprocess in text-to-image translation [11, 12, 13, 20] and image generation from scene graphs [1, 15].

Aggregating overlapped bounding boxes and masks. Bounding boxes inevitably overlap in real or generated layouts, so in the layout-to-image generation generated ob-

[1] https://github.com/ZejianLi/LAMA

ject masks also overlap and give confusing pixelwise features. The overlap problem has been noticed by existing works [35, 38]. Existing method use operations like sum, normalization, maxpooling, replacement or ConvLSTM [31] to aggregate overlapped object masks or features (Tab. 1). The sum and normalization operation do not alleviate overlap. When the real masks are not available, maxpooling and replacement allow the model to show wrong objects without passing gradients to correct ones. Zhao *et al*. [38] shows aggregating masks with ConvLSTM [31] has a better performance than with sum operation, as ConvLSTM models object relations and thus considers overlaps. Sylvain *et al*. [35] finds overlapped boxes of the same category results in unclear object boundaries and proposes to concatenate semantic masks with layout boundaries. Different from existing works, our method uses pixelwise adaption and normalization. Our adaption module learns to scale mask values according to neighborhood mask configurations and gives more clear semantic masks.

3. An Experiment on Impacts of Mask Clarity

A control experiment is conducted to test our hypothesis by investigating the impacts of mask clarity on the performance. We train the mask-to-image component of LostGAN [32] with ground-truth masks and test it on ground-truth or tainted masks. The component has a GauGAN-like architecture [26]. It stacks ResNet blocks [9] with ISLA-Norm layers [32]. The ISLA-Norm transforms the generated mask to modulation parameters applied on batch-normalized [14] features to propagate mask information.

Formally, the model is trained to generate an image based on a given semantic mask $\boldsymbol{M}$. A semantic mask with $\boldsymbol{m}$ objects $\boldsymbol{M} \in [\mathbf{0},\mathbf{1}]^{\boldsymbol{m}\times\boldsymbol{W}\times\boldsymbol{H}}$ describes the object each pixel belongs to, where $\boldsymbol{W}$ and $\boldsymbol{H}$ are the width and height. We denote the ground-truth mask as $\bar{\boldsymbol{M}}$ and a tainted mask as $(1-\tau)\times\bar{\boldsymbol{M}}+\frac{\tau}{m}\times\mathbf{1}(\bar{\boldsymbol{M}})$ where $\boldsymbol{\tau}\in[0,1]$ and $\mathbf{1}(\bar{\boldsymbol{M}})$ is the all-one tensor with the same size as $\bar{\boldsymbol{M}}$. A larger τ results in an unclear semantic mask. The experiment is performed on COCO-stuff [4] in 128 × 128, with FID [10] and SceneFID [35] to estimate generative quality. Both metrics are introduced in Sec 5.

The result is shown in Fig. 2. When tested with ground-truth masks ($\boldsymbol{\tau}=\mathbf{0}$), the model has a significantly better performance than the original LostGAN. As $\boldsymbol{\tau}$ increases and masks become unclear and ambiguous, the performance decays. This shows the positive impact of mask clarity on generative quality and supports our hypothesis. More results are in Tab. 5. A concurrent work [5] performs a similar experiment, and the result is consistent with ours. We also present the performance of our model. With LAMA module and other modifications, our performance is better than LostGAN with ground-truth masks in SceneFID.

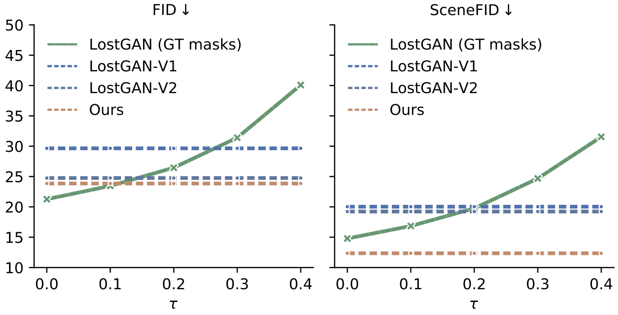

Figure 2. **Results in the experiment on mask clarity (Sec. 3).** Models are evaluated with FID [10] and SceneFID [35]. LostGAN (GT masks) is the LostGAN [32] model trained with ground-truth semantic masks but tested with tainted masks. A larger τ results in more ambiguous masks. Dotted lines label performances of the original LostGANs [32, 34] and ours for reference.

4. Method

In this section, we describe our layout-to-image method. We first summarize the overall pipeline in Sec. 4.1 and then mainly introduce our proposed Locality-Aware Mask Adaption module in Sec. 4.2.

4.1. Overview

Layout-to-mask. Given a layout $\boldsymbol{L}$ of $\boldsymbol{m}$ objects, the layout-to-mask stage is to generate a semantic mask $\boldsymbol{M}$. A layout $\boldsymbol{L}$ is formalized as $\{(\boldsymbol{b}_1,\boldsymbol{c}_1),\ldots,(\boldsymbol{b}_m,\boldsymbol{c}_m)\}$. A bounding box $\boldsymbol{b}_i$ is a subset of the image lattice describing the position and size $(\boldsymbol{w}_i,\boldsymbol{h}_i)$ of the $\boldsymbol{i}^{th}$ object, while $\boldsymbol{c}_i$ is the object category, for $\boldsymbol{i}\in\{\mathbf{1},\ldots,\boldsymbol{m}\}$.

To form a raw semantic mask, object masks are generated separately and mapped to bounding boxes. For each object, an object feature $\boldsymbol{o}_i\in\mathbb{R}^{\boldsymbol{d}}$ contains the category and the object size. It is formed as $\boldsymbol{o}_i=[\boldsymbol{y}_i,\boldsymbol{z}_i,\boldsymbol{w}_i,\boldsymbol{h}_i]$. Here $\boldsymbol{y}_i\in\mathbb{R}^{\boldsymbol{d}_y}$ is the embedding of category $\boldsymbol{c}_i$, $\boldsymbol{z}_i\in\mathbb{R}^{\boldsymbol{d}_z}$ is a stochastic variation, and $\boldsymbol{d}=\boldsymbol{d}_y+\boldsymbol{d}_z+\mathbf{2}$. Given a set of object features $\boldsymbol{O}=[\boldsymbol{o}_1,\ldots,\boldsymbol{o}_m]$, the generator $\boldsymbol{F}_{om}$ transform them to $\boldsymbol{m}$ object masks of size 32 × 32. These masks are resized and mapped to the corresponding bounding boxes to form the raw mask $\tilde{\boldsymbol{M}}$ [32]. Finally, $\tilde{\boldsymbol{M}}$ is adapted to the final mask $\boldsymbol{M}$ with our proposed locality-aware mask adaption module (Fig. 3).

Mask-to-image. This stage is to generate an image given the generated semantic mask $\boldsymbol{M}$. We adopt the mask-to-image component of LostGAN-V1 [32] and have three modifications. Firstly, inspired by Batch-Instance Normalization [25], we replace Batch Normalization with Batch-Group Normalization (BGN) in the ISLA-norm layers [32]. BGN further utilizes information across channels when the batch size is small. Secondly, noise injection [16] is applied after convolution layers to introduce more stochasticity. Thirdly, we use ReZero [2] to stabilize the training. Our

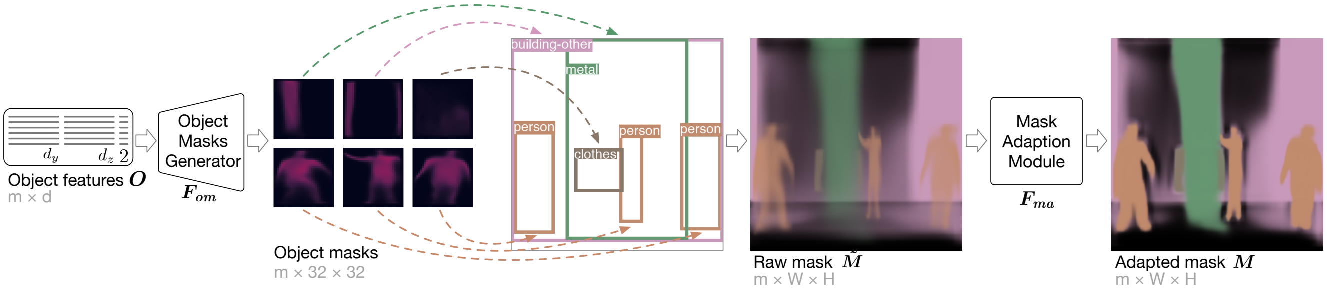

Figure 3. **An illustration of layout-to-mask generation.** The object masks are generated according to object features, which are the concatenation of category embeddings, the size of boxes and the stochastic variations. Then the object masks are resized and mapped to bounding boxes. The formed raw mask is adapted to alleviate overlappings and reduce semantic ambiguity.

ablation study shows both BGN and noise injection improve the model performance (Sec. 6.3).

4.2. Locality-Aware Mask Adaption

We propose Locality-Aware Mask Adaption (LAMA) module to adapt the raw mask to a more clean mask by diminishing mask overlapping. Mask overlapping means that objects share a large area in the semantic segmentation mask, and in the shared area objects have similar strength. In the raw mask $\tilde{\boldsymbol{M}}$ of $\boldsymbol{m}$ objects, each channel represent an object's mask, and a pixel may belong to more than one object simultaneously. See the raw mask in Fig. 3 for an example. Such overlap results in reduced clarity and causes difficulty in image generation (Sec. 3).

The mask overlapping problem derives from overlapped bounding boxes, and it is also caused by two issues. On the one hand, object masks are generated separately without considering their relations. On the other hand, object masks are aggregated by operations like sum or normalization which does not improve clarity. Our method tries to solve the problem from the aggregation part.

The proposed LAMA module adapts masks by scaling mask values of all objects in each pixel, which allows all pixels to choose the object they belong to. It determines the scaling based on the local mask configuration by matching embeddings of patches with the object embeddings. Specifically, given a pixel in the raw mask, the adaption module first aggregates the object information in the neighborhood as a query representation. Besides, the module also transforms the set of object features into a key representation. Then by matching the query and the key, LAMA forms the scaling factors of the pixel, which are applied to the raw mask values. By applying this scaling process to all pixels, LAMA adapts the whole raw mask (Fig. 4).

Formally, the adaption module $\boldsymbol{F_{ma}}$ takes the raw mask $\tilde{\boldsymbol{M}}$ and object features $\boldsymbol{O}$ as input and outputs the scaling factors $\boldsymbol{F_{ma}}(\tilde{\boldsymbol{M}}, \boldsymbol{O})$. Then the adapted mask is

$$\boldsymbol{M} = \boldsymbol{F_{ma}}(\tilde{\boldsymbol{M}}, \boldsymbol{O}) \odot \tilde{\boldsymbol{M}} \tag{1}$$

Here $\odot$ is the elementwise multiplication. The object features $\boldsymbol{O}$ are first transformed into object key and query $\boldsymbol{K}^{obj}, \boldsymbol{Q}^{obj} \in \mathbb{R}^{m \times d'}$ with a fully-connected (FC) layer, respectively. Here $\boldsymbol{d'}$ is $\lfloor \frac{d}{4} \rfloor$ by default. To aggregate the object information of pixels in $\tilde{\boldsymbol{M}}$, pixel query $\boldsymbol{Q}^{pix \in \mathbb{R}^{d' \times W \times H}}$ is defined as the sum of the object query weighted by the pixelwise raw mask strength.

$$\begin{aligned} \boldsymbol{Q}^{pix} &= \boldsymbol{Q}^{obj\top} \otimes \tilde{\boldsymbol{M}} \\ \text{where } \boldsymbol{Q}^{pix}_{kj} &= \sum_{i=1}^{m} \boldsymbol{Q}^{obj}_{ik} \tilde{\boldsymbol{M}}_{ij} \end{aligned} \tag{2}$$

The symbol $\otimes$ means dot product operation. Here $k \in \{1, \ldots, d'\}$ and j denotes a spatial position in the $W \times H$ lattice of $\boldsymbol{Q}^{pix}$ and $\tilde{\boldsymbol{M}}$. Namely, the pixel query in the j^{th} pixel $\boldsymbol{Q}^{pix}_{\cdot j}$ is the sum of object query $\boldsymbol{Q}^{obj}$ weighted by $\tilde{\boldsymbol{M}}_{\cdot j}$, the raw mask values in the j^{th} pixel. When one object mask dominates the j^{th} pixel of $\tilde{\boldsymbol{M}}$, the pixel query represents the only object. When several masks overlap on the j^{th} pixel, the pixel query represents the mixed objects.

Next, the pixel query is divided into patches and aggregated to form a local query. A local query $\boldsymbol{Q}^{loc} \in \mathbb{R}^{d' \times W \times H}$ represents the object configuration in the locality of each pixel. To summarize these configurations, two ResNet blocks with a convolutional kernel size of 3 are applied to the pixel query $\boldsymbol{Q}^{pix}$. Therefore, the local query represents the configurations of 5×5 patches centered at pixels, and thus LAMA is locality-aware. Such locality is important as we will show the performance decays without aggregating pixel query as local query (Sec. 6.3).

The local query $\boldsymbol{Q}^{loc}$ is matched to object key $\boldsymbol{K}^{obj}$ with a dot product operation.

$$\begin{aligned} \boldsymbol{E} &= \boldsymbol{K}^{obj} \otimes \boldsymbol{Q}^{loc} \\ \text{where } \boldsymbol{E}_{ij} &= \sum_{k=1}^{d'} \boldsymbol{K}^{obj}_{ik} \boldsymbol{Q}^{loc}_{kj} \end{aligned} \tag{3}$$

Here the local query of the j^{th} position $\boldsymbol{Q}^{loc}_{\cdot j}$ is matched with the i^{th} object's key $\boldsymbol{K}^{obj}_{i \cdot}$. A matching value E_{ij} rep-

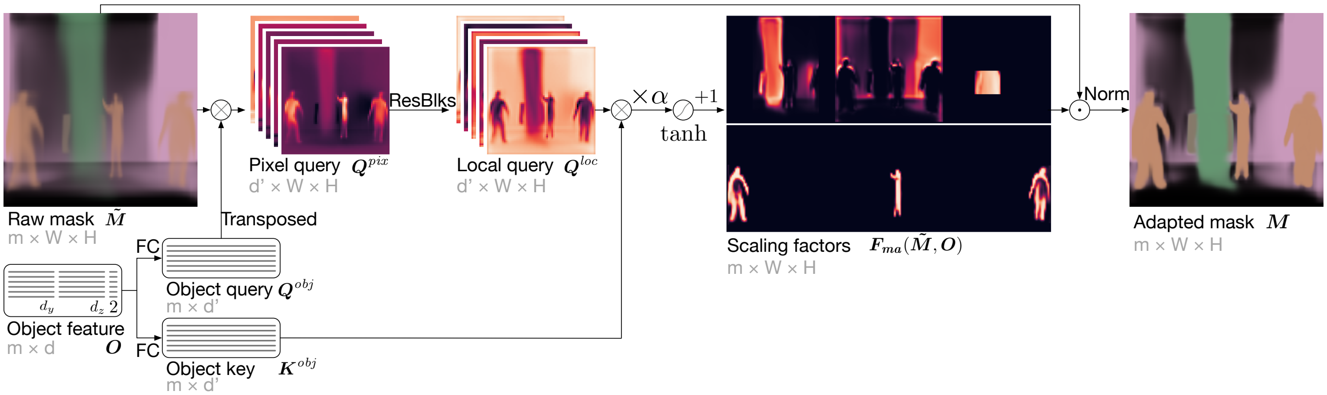

Figure 4. **LAMA's mask adaption procedure**. The example is identical to Fig. 3. Here ⊗ means the dot product operation and ⊙ element-wise multiplication. Object query is aggregated according to the raw mask as pixel query. With ResNet [8] blocks, pixel query in the locality is summarized as local query, which is matched with the object key to form the scaling factors. In the part of scaling factors, the second image of the first row shows the scaling factor of "building-other". Notice that the mask is shrunk precisely in the area where the "person" masks shown in the second row are located. This exemplifies LAMA's adaption ability.

resents the importance of the i^{th} object in the j^{th} position. Accordingly, the scaling factors is formed by:

$$\boldsymbol{F}_{ma}(\tilde{\boldsymbol{M}}, \boldsymbol{O}) = \tanh(\boldsymbol{\alpha}\boldsymbol{E}) + 1 \tag{4}$$

The scaling factors are in $(0, 2)$ and allow both shrinkage and enlargement of masks. A learnable parameter $\boldsymbol{\alpha}$ controls the strength of adaption, initialized as 0. The adapted mask is normalized to form the pixelwise object distribution as the final semantic mask. The main adaption process is summarized in Fig. 4. Components to learn in LAMA include two FC layers, two ResNet blocks and $\boldsymbol{\alpha}$.

Technically LAMA shifts pixelwise object distributions, but theoretically LAMA has an underlying assumption different from those of existing works. In existing works [32, 35], object masks are generated individually, which assumes masks are mutually independent. The layout-to-image task is ill-posed and thus the mask configuration is highly uncertain. Without ground-truth segmentations, the training of mask generation is weakly-supervised and relies on strong assumptions like independence. Layout2Im [37, 38] and Obj-GAN [20] use ConvLSTM [31] to generate or aggregate object masks, which assumes all objects are correlated. Our proposed LAMA has a different assumption that overlapped or adjacent boxes have masks correlated on visibility and appearance. This assumption is more general than independence but preserves locality. It allows the generative model to consider relations only among overlapped or nearby objects. With this local correlation assumption, LAMA boasts reconfigurability [34]. Reconfigurability means keeping most generated objects unchanged while moving, altering or adding a bounding box, which enables generative results to be more controllable.

LAMA module is expected to adapt masks but does not directly encourage a pixel to choose only one object. This is because if the wrong object is chosen in the overlapped area, the correct one will get little gradient and the convergence becomes difficult. Such incorrect appearances often happen in the early stage of training. Similarly, over-adaption is harmful at the beginning, so the model has an adaption strength $\boldsymbol{\alpha}$ initialized as 0 and begins the training without adaption. As the training proceeds, the model optimizes $\boldsymbol{\alpha}$ as needed to minimize the training loss. As $\boldsymbol{\alpha}$ can be absorbed in the computation of object key (Eq. 3), $\boldsymbol{\alpha}$ does not affect the final solution. We further discuss convergence issues when directly inferring new masks, reconfigurability, causal relations and other related topics in Sec. S2 of our supplementary material.

In summary, we design LAMA with a trade-off between alleviating overlaps and facilitating convergence. Empirically we find our model spontaneously improves mask clarity with LAMA (Sec. 6.2).

4.3. Training

Loss. Following LostGAN [34], we use the adversarial training strategy [6, 23] to train our layout-to-image generation model with a discriminator. The discriminator consists of ResNet [8] blocks with Spectral Normalization [23]. To classify objects in bounding boxes, it uses ROI Align [7] to extract feature maps and identifies the objects with a projection discriminator [24]. The whole training loss consists of an adversarial hinge loss and a classification loss [24].

We do not adopt reconstruction loss in our training, because it may mislead the model to ignore inputs of stochastic variation. The reconstruction binds layouts to associated ground-truth images. To optimize the loss, the model may consider only the input boxes and thus generate a fixed mask to fit the training image. As stochastic variation is discouraged, the capacity of the model is limited. This analysis is

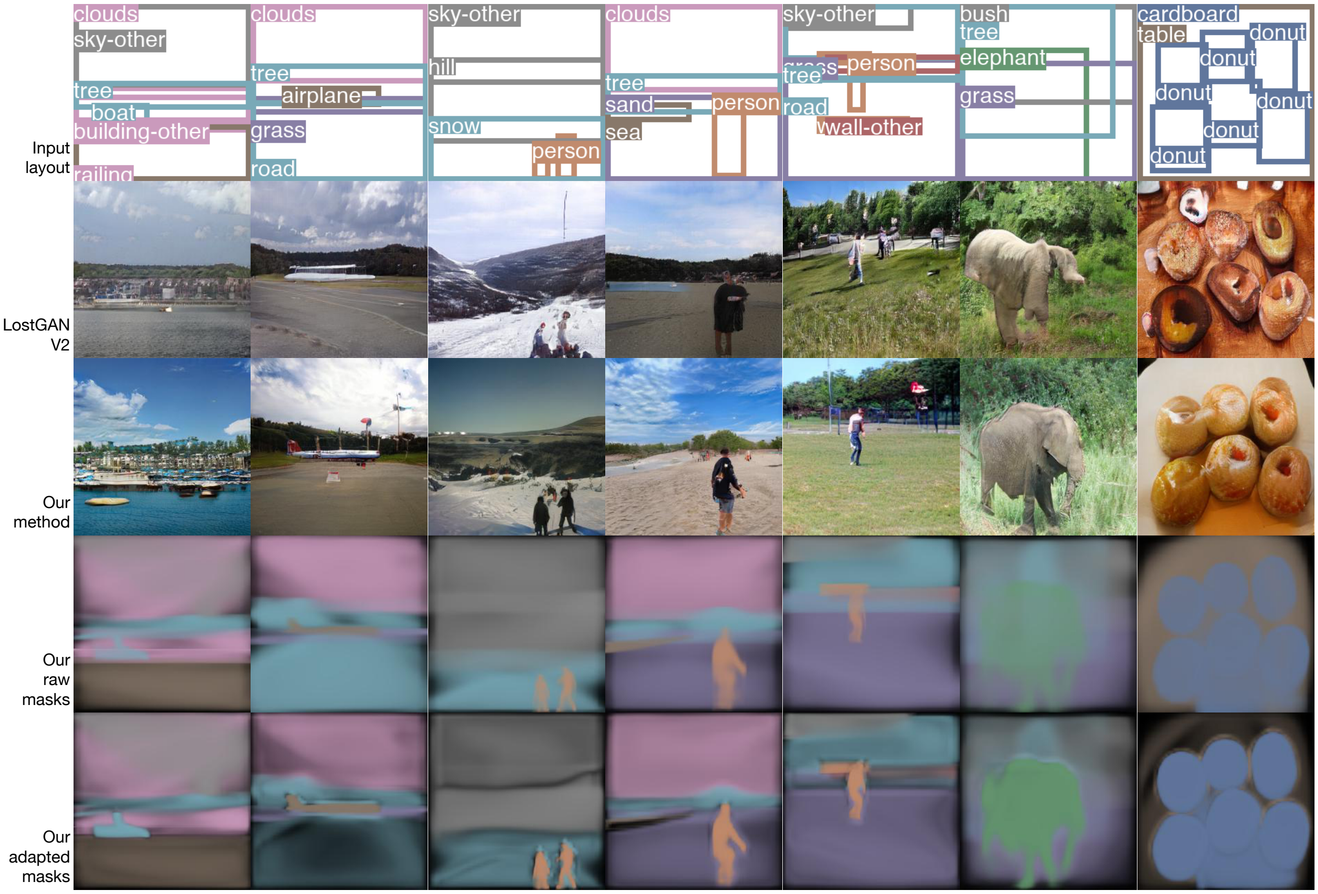

Figure 5. Examples of generated 256 × 256 images on COCO-Stuff by our methods and LostGAN-V2 [34]. Samples in the same column share the layout in the first row. Our generated masks before and after adaption are also presented.

supported by our ablation study (Sec. 6.3), where the performance of our model decays with the reconstruction loss.

Training details. The model is trained for 200 epoches with Adam optimizer [17] with $\beta = (0, 0.999)$. The learning rate is 0.0003 for the discriminator and is 0.0001 for the generator, and both are decayed by 0.1 in the 120^{th} and 160^{th} epoch. The batch size is 32 for the resolution of 64×64 and 20 for 128×128 when trained on a single NVIDIA RTX 2080TI card. The size is 24 for 256×256 when trained on four 2080TI cards. Horizontal flip augmentation of training images and layouts is adopted [34, 35].

5. Experimental Setup

5.1. Datasets

Experiments are conducted on images and annotations of bounding boxes from COCO-Stuff 2017 [4] and Visual Genome dataset [18]. Segmengtation annotation is not used. The COCO-Stuff 2017 dataset contains bounding boxes of 80 thing and 91 stuff classes. Following SG2Im [15], objects covering less than 2% of the whole image are ignored, and only images with 3 to 8 objects are used. Thus, we have $74,777$ training and $3,097$ validatation images. For the Visual Genome dataset, we use the train-validation division and the selecting strategy of SG2Im [15]. Thus, we have $62,565$ training and $5,062$ validation samples, each with less than 10 objects from 178 categories.

5.2. Baselines

We mainly compare our method with leading layout-to-images methods, including Layout2Im [37] as well as its extension [38], LostGAN-V1 [32], LostGAN-V2 [34] and OC-GAN [35]. Particularly, we use the official implementation of Layout2Im, LostGAN-V1 and LostGAN-V2 to generate new images for comparison. Although BachGAN [21] and Attribute-Guided Layout2Im [22] perform a similar task, they require extra annotations and are not included in our comparison.

5.3. Metrics

Models are evaluated mainly from two aspects: the overall visual quality and diversity of generated images, and the fidelity and alignment of generated objects.

Table 2. Quantitative results on COCO-Stuff [4] and Visual Genome (VG) [18]. ↑ means a higher value is better, and vice versa. Best performances are highlighted. The results of other methods are taken from the original papers unless labeled. ○ and ⋆ denote results taken from [35] and [34], respectively. † denotes results on samples from pre-trained models or trained models with the official implementation.

Size	Methods	FID ↓		DS ↑		CAS ↑		SceneFID ↓	
		COCO	VG	COCO	VG	COCO	VG	COCO	VG
64×64	Layout2Im [37]	38.14	31.25	0.15 ± 0.06	0.17 ± 0.09	* 27.32	* 23.25	† 23.26	† 33.67
	Layout2Im + OWA [38]	40.19	33.54	0.09 ± 0.05	0.09 ± 0.11	-	-	-	-
	LostGAN-V1 [32]	34.31	34.75	0.35 ± 0.09	0.34 ± 0.10	28.81	27.50	† 10.50	† 5.15
	OC-GAN [35]	29.57	20.27	-	-	-	-	-	-
	Ours	**19.76**	**18.11**	**0.37 ± 0.10**	**0.37 ± 0.09**	**33.23**	**30.70**	**9.17**	**4.15**
128×128	LostGAN-V1 [32]	29.65	29.36	0.40 ± 0.09	0.43 ± 0.09	28.70	25.89	$^{\circ}$ 20.03	$^{\circ}$ 13.17
	LostGAN-V2 [34]	24.76	29.00	0.45 ± 0.09	0.42 ± 0.09	31.98	29.35	† 19.23	† 15.02
	OC-GAN [35]	36.31	28.26	-	-	-	-	16.76	9.63
	Ours	**23.85**	**23.02**	**0.46 ± 0.09**	**0.47 ± 0.09**	**34.15**	**32.81**	**12.35**	**8.28**
256×256	LostGAN-V2 [34]	42.55	47.62	**0.55 ± 0.09**	0.53 ± 0.09	30.33	28.81	† 22.00	† 18.27
	OCGAN [35]	41.65	40.85	-	-	-	-	-	-
	Ours	**31.12**	**31.63**	0.48 ± 0.11	**0.54 ± 0.09**	**30.52**	**31.75**	**18.64**	**13.66**

Inception Score (IS) [30] and **Frèchet Inception Distance (FID)** [10] measure overall visual quality. Specifically, we compute FID between generated images and validation sets. **Diversity Score (DS)** estimates the diversity of generated images. It is the perceptual similarity of deep features extracted from two generated images with the same layout. We use LPIPS [36] with a pre-trained AlexNet [19].

Classification Accuracy Score (CAS) [28] and **SceneFID** [35] measure the visual quality of objects. SceneFID is the FID between the resized 224×224 images of cropped objects from the generated and validation set. CAS score measures generated objects' quality with an auxiliary classifier. A ResNet-101 [8] is trained on generated object crops and tested on validation crops, and the testing accuracy is reported as the metric. Notice that a similar metric used in Layout2Im [37] and OC-GAN [35] is based on a classifier trained on training samples but tested on generated ones, and it may not consider the diversity of generated samples.

YOLO Scores are proposed to evaluate the alignment and fidelity of generated objects. We propose this metric based on the insight that layout-to-image generation is a reverse task of object detection. While SceneFID and CAS estimate the generative quality of object crops, YOLO Scores measure how generated objects are recognizable when even the layout is unknown. YOLO (You Only Look Once) [3, 29] is a series of object detection models, which infer layouts from images. By comparing the ground-truth and the inferred layout of a generated image, we can measure object alignment and visual fidelity. In practice, we use a YOLOv4 [3] model pre-trained on MS COCO dataset without stuff and report AP (Average Precision), AP_{50} and AP_{75}. The generated images are upsampled to 512×512 when tested.

Mask Entropy. We also evaluate the clarity of a generated mask with mask entropy, defined as the mean pixel-wise entropy. Given a mask $\boldsymbol{M}$ of $\boldsymbol{m}$ objects, we have

$$\mathcal{H}(\boldsymbol{M}) = -\frac{1}{W \times H} \sum_{j=1}^{W \times H} \sum_{i=1}^{m} \boldsymbol{M_{ij}} \log \boldsymbol{M_{ij}}. \tag{5}$$

The entropy value is in $[0, \log m]$. A larger value means a more unclear mask, and the value of an ideal mask is 0.

6. Results

6.1. Qualitative and Quantitative Results

Fig. 5 presents generated 256×256 images on COCO-Stuff as qualitative results. Our models generate visually appealing images as compared with state-of-the-art LostGAN-V2. Tab. 2 reports quantitative results of baseline methods and ours. Our model outperforms the existing methods in most cases. In terms of image quality, our model has lower FID values and higher diversity scores. On COCO-Stuff and Visual Genome in 256×256, our model improves state-of-the-art FID from 41.65 to 31.12 and from 40.85 to 31.63, respectively. Both are over 20% relative improvements. In terms of object quality, our model also has lower SceneFID values and higher CAS scores. On Visual Genome in 256×256, the challenging CAS accuracy is improved by 2.94%. On datasets in 256×256, SceneFID is improved from 22.00 to 18.64 and from 18.27 to 13.66, respectively. Both are over 15% relative improvements.

Tab. 3 reports the YOLO Scores of LostGANs [32, 34] and our proposed model. Ours outperforms the baselines consistently. Especially on images in 256×256, our model's AP is 4.1% higher than that of LostGAN-V2, which is a 45% improvement. Furthermore, the scores of on images in the validation set are also reported for comparison. The validation images are downsampled to 128×128 or 256×256 and then upsampled back to 512×512 for testing.

Table 5. Ablation studies on COCO-Stuff [18] of 128 × 128. Main performance differences between variants and the original model are labeled. See text in Sec. 6.3 for more details.

Methods	**FID** ↓	**DS** ↑	**CAS** ↑	**SceneFID** ↓	**AP** ↑	**AP$_{50}$** ↑	**AP$_{75}$** ↑
Ours	23.85	0.46 ± 0.09	34.15	12.35	7.9%	12.0%	8.9%
Ours w/o mask adaption	(+3.72) 27.57	0.46 ± 0.09	(−0.56) 33.59	(+4.63) 16.92	(−0.7%) 7.2%	10.6%	8.2%
Ours trained w/ GT masks	(−0.53) 23.32	0.32 ± 0.10	(+3.58) 37.73	(+0.69) 13.04	(+5.7%) 13.6%	19.3%	15.0%
Ours matching pixel query only	(+5.67) 29.52	0.45 ± 0.09	(−0.56) 33.59	(+6.04) 18.39	(−2.3%) 5.6%	8.9%	6.3%
Ours w/ BN only	(+0.64) 24.49	0.45 ± 0.10	(+0.44) 34.59	(+0.73) 13.08	(−1.1%) 6.8%	10.6%	7.1%
Ours w/ GN only	(+1.32) 25.17	0.45 ± 0.10	(−2.13) 32.02	(+0.93) 13.28	(−1.0%) 6.9%	10.5%	7.7%
Ours w/o noise injection	(+1.33) 25.18	0.41 ± 0.12	(−1.85) 32.30	(+1.77) 14.12	(−0.8%) 7.1%	10.9%	7.8%
Ours w/ reconstruction	(+7.12) 30.97	0.44 ± 0.08	(−0.40) 33.75	(+8.76) 21.11	(−2.7%) 5.2%	8.0%	5.7%
LostGAN-V1 [32]	29.65	0.40 ± 0.09	28.70	20.03	4.8%	8.4%	5.1%
LostGAN-V1 [32] w/ mask adaption	(−3.52) 26.13	0.43 ± 0.09	(+5.89) 34.59	(+2.71) 22.74	(+0.5%) 5.3%	8.9%	5.9%
LostGAN-V1 [32] w/ GT masks	(−8.38) 21.27	0.36 ± 0.11	(+10.35) 39.05	(−5.25) 14.78	(+6.5%) 11.3%	16.9%	12.5%

Table 3. Comparison of YOLO scores in experiments on the MS COCO dataset. A higher value is better.

Size	**Methods**	**AP** ↑	**AP$_{50}$** ↑	**AP$_{75}$** ↑
128×128	LostGAN-V1 [32]	4.8%	8.4%	5.1%
	LostGAN-V2 [34]	5.5%	9.2%	5.8%
	Ours	**7.9%**	**12.0%**	**8.9%**
	MS COCO val	33.1%	47.0%	36.9%
256×256	LostGAN-V2 [34]	9.1%	15.3%	9.8%
	Ours	**13.4%**	**19.7%**	**14.9%**
	MS COCO val	42.9%	60.2%	48.2%

Table 4. Mask entropies of our generated masks and α in Eq. 3.

Size	**Datasets**	**Raw Masks**	**Adapted Masks**	$\lvert\alpha\rvert$
64×64	COCO	0.31 ± 0.18	0.11 ± 0.07	0.73
	VG	0.18 ± 0.14	0.10 ± 0.09	0.66
128×128	COCO	0.30 ± 0.17	0.14 ± 0.11	0.55
	VG	0.45 ± 0.24	0.12 ± 0.09	0.66
256×256	COCO	0.33 ± 0.18	0.14 ± 0.10	0.56
	VG	0.50 ± 0.24	0.26 ± 0.17	0.54

6.2. Mask Clarity

We report mask entropy values of raw and adapted masks generated by our model and adaption strength α in Tab. 4. The entropy reduces consistently after adaption. For references, a tainted mask in Sec. 3 has an entropy of 0.48 when $\tau = 0.1$ and 0.12 when $\tau = 0.02$. Examples of raw and adapted masks are shown in the last two rows in Fig. 5. Object masks before adaption seem blurry and overlapped, while after adaption they have clear individual areas and boundaries. These exemplify LAMA generates semantically more clear masks as expected.

6.3. Ablation Study

We conduct ablation and control experiments on variants of our proposed model and LostGAN (Tab. 5). We conduct control experiments on a variant of our model without LAMA and a LostGAN-V1 variant with LAMA. Without the LAMA module, our proposed model suffers from a performance decay, while LAMA also boosts LostGAN-V1's performance without other modifications. This shows LAMA contributes to the improvement of performance. Besides, we also test both models trained with ground-truth masks, as done in Sec. 3. In this case, the performances exceed those of models with LAMA. This supports our hypothesis on the importance of mask quality again.

Furthermore, we investigate the contribution of other components in our model. Firstly, we test whether locality-awareness is important in LAMA by matching the pixel query with the object key in (Eq. 3). In this case, adaption is only based on the pixelwise object distribution rather than the local mask configuration. Secondly, we test the contribution of BGN and noise injection (Sec. 4.1). Thirdly we train the model with an extra reconstruction loss. Results in Tab. 5 show the performance decreases in most case with these changes and supports our current design.

7. Conclusion

In this paper, we hypothesize the importance of semantically clear masks in the layout-to-image generation. The hypothesis is supported by that a LostGAN [32] model trained with real masks has decayed performances when given tainted masks. Based on the hypothesis, we propose Locality-Aware Mask Adaption module. Experimental results show LAMA substantially improves mask clarity and contributes to the improvement of performance.

Acknowledgements

This paper is supported by National Science and Technology Innovation 2030 Major Project (2018AAA0100703) of the Ministry of Science and Technology of China, the National Natural Science Foundation of China (61773336, 62006208) and the Provincial Key Research and Development Plan of Zhejiang Province (2019C03137). Zejian Li would like to thank Pei Chen, Yongxing He in Zhejiang University for helpful comments and Wei Sun for the kindness to answer questions regarding LostGANs.

References

[1] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4561–4569, 2019. 2

[2] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence (UAI)*. Proceedings of Machine Learning Research, 2021. 3

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020. 7

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 6, 7

[5] Arantxa Casanova, Michal Drozdzal, and Adriana Romero-Soriano. Generating unseen complex scenes: are we there yet? *ArXiv*, abs/2012.04027, 2020. 3

[6] 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 (NeurIPS)*, pages 2672–2680. Curran Associates, Inc., 2014. 5

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 1, 5

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5, 7

[9] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pages 630–646. Springer, 2016. 3

[10] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6626–6637. Curran Associates, Inc., 2017. 2, 3, 7

[11] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Generating multiple objects at spatially distinct locations. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[12] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 2

[13] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and H. Lee. Inferring semantic layout for hierarchical text-to-image synthesis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7986–7994, 2018. 2

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456. PMLR, 2015. 3

[15] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1219–1228, 2018. 2, 6

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 3

[17] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014. 6

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 123(1):32–73, 2017. 6, 7, 8

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 1097–1105. Curran Associates Inc., 2012. 7

[20] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12166–12174, 2019. 2, 5

[21] Yandong Li, Y. Cheng, Zhe Gan, Licheng Yu, Liqiang Wang, and Jing jing Liu. BachGAN: High-resolution image synthesis from salient object layout. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8362–8371, 2020. 2, 6

[22] Ke Ma, Bo Zhao, and Leonid Sigal. Attribute-guided image generation from layout. In *British Machine Vision Virtual Conference (BMVC)*, pages 0384:1–13. British Machine Vision Virtual Association, 2020. 2, 6

[23] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. 5

[24] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations (ICLR)*, 2018. 5

[25] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2558–2567. Curran Associates Inc., 2018. 3

[26] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2346, 2019. 3

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit

Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019. 2

[28] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12247–12258. Curran Associates, Inc., 2019. 7

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 7

[30] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2234–2242. Curran Associates, Inc., 2016. 7

[31] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 1–9. Curran Associates, Inc., 2015. 2, 3, 5

[32] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *IEEE International Conference on Computer Vision (ICCV)*, pages 10531–10540, 2019. 1, 2, 3, 5, 6, 7, 8

[33] Wei Sun and Tianfu Wu. Deep consensus learning. *ArXiv*, abs/2103.08475, 2021. 2

[34] Wei Sun and Tianfu Wu. Learning layout and style reconfigurable GANs for controllable image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 2, 3, 5, 6, 7, 8

[35] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R. Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 3, 5, 6, 7

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 7

[37] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8584–8593, 2019. 1, 2, 5, 6, 7

[38] Bo Zhao, Weidong Yin, Lili Meng, and Leonid Sigal. Layout2image: Image generation from layout. *International Journal of Computer Vision (IJCV)*, pages 1–18, 2020. 1, 2, 3, 5, 6, 7