Hierarchical Image Generation via Transformer-Based Sequential Patch Selection

Xiaogang Xu,[1] **Ning Xu** [2]

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2] Adobe Research
xgxu@cse.cuhk.edu.hk, nxu@adobe.com

Abstract

To synthesize images with preferred objects and interactions, a controllable way is to generate the image from a scene graph and a large pool of object crops, where the spatial arrangements of the objects in the image are defined by the scene graph while their appearances are determined by the retrieved crops from the pool. In this paper, we propose a novel framework with such a semi-parametric generation strategy. First, to encourage the retrieval of mutually compatible crops, we design a sequential selection strategy where the crop selection for each object is determined by the contents and locations of all object crops that have been chosen previously. Such process is implemented via a transformer trained with contrastive losses. Second, to generate the final image, our hierarchical generation strategy leverages hierarchical gated convolutions which are employed to synthesize areas not covered by any image crops, and a patch-guided spatially adaptive normalization module which is proposed to guarantee the final generated images complying with the crop appearance and the scene graph. Evaluated on the challenging Visual Genome and COCO-Stuff dataset, our experimental results demonstrate the superiority of our proposed method over existing state-of-the-art methods.

Introduction

It is challenging to generate an image from a scene graph consists of several objects with sophisticated interactions. With such a framework, users just need to provide flexible scene descriptions to define the objects as well as their interactions, and the framework would synthesize images, achieving a user-controllable generation process. Current frameworks for generating images from scene descriptions take advantage of generative adversarial networks (GANs) (Goodfellow et al. 2014). Compared with parametric models (Johnson, Gupta, and Fei-Fei 2018) that solely lean upon the networks to model the appearance of objects, semi-parametric approaches (Li et al. 2019c; Tseng et al. 2020) have recently been proposed and shown superior performance. Such methods first leverage a memory bank to retrieve image crops for objects in the scene graph (called retrieve stage), and then synthesize realistic images from scene graphs and retrieved crops (called generation stage). In this work, we focus on the amelioration of the semi-parametric model, improving the retrieve stage as well as the generation stage.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The retrieved crops from the retrieve stage should be mutually compatible for synthesizing an image and consistent with the corresponding scene graph. To complete the retrieve stage, existing retrieval-based image synthesis methods either employ pre-defined embeddings for retrieval (Li et al. 2019c) or proposes a differentiable retrieval process (Tseng et al. 2020) to retrieve the image crop that is compatible with the previously selected ones. However, they all neglect the usage of the crops' spatial information during the optimizing of retrieve stage. In this paper, we reformulate the retrieve stage and complete it as a novel sequential process. In our process, the selection of the crop for each object in the scene graph would be determined by spatial, style, and content features of crops that have already been chosen. To implement such sequential selection, we propose to adopt a transformer (Vaswani et al. 2017) structure that is trained with contrastive learning (Oord, Li, and Vinyals 2018; Chen et al. 2020). In the transformer, the candidate image crops for selection and previously selected crops are embedded with two specific heads and incorporated with spatial information via position embedding. Iterative operations on the transformer can retrieve the image crops that are mutually compatible. This is the first successful attempt to complete the crop retrieval with self-supervised contrastive learning for image synthesis, and experiments on public datasets demonstrate the superiority of our sequential selection strategy over existing retrieve approaches.

Moreover, in this paper, we further propose a novel generator that generates realistic images from a scene graph with selected image crops as the guidance. To synthesize the realistic and high-resolution images, we design the generator with a hierarchical generation strategy, using hierarchical gated convolutions and proposing patch-guided spatially adaptive normalization module. The patch-guided spatially adaptive normalization module is designed to guarantee the synthesized images highly respecting the selected crops. The generator is trained with crops selected by our transformer, boosting the performance of the generator in the inference stage, and the generator boosts the mutual compatibility between the selected crops in the output image. Evaluated on Visual Genome and COCO-Stuff dataset (including objective analyses and a user study for subjective evaluation) , our proposed method significantly outperforms the state-of-the-art (SOTA) generation approaches.

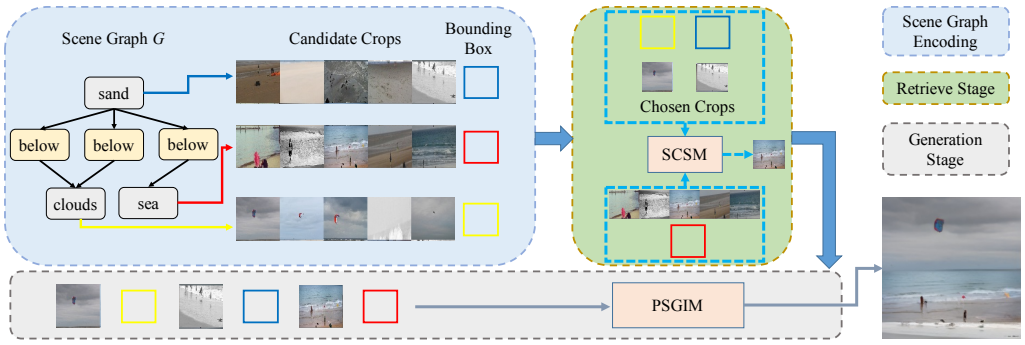


Figure 1: Illustration of our framework for synthesizing images from scene graphs, including the scene graph encoding, retrieve stage and the generation stage.

Related Work

Conditional Image Generation

Current conditional generative models can synthesize images according to additional conditions such as image (Choi et al. 2020; Lee et al. 2020; Liu et al. 2021), label (Chen et al. 2019; Yang et al. 2021), segmentation mask (Park et al. 2019; Huang et al. 2020; Tan et al. 2021), text (Xu et al. 2018; Qiao et al. 2019; Li et al. 2019a; Zhu et al. 2019; Li et al. 2019b) and layout (Sun and Wu 2019; Ashual and Wolf 2019; Zhao et al. 2019). The text conditions can either be natural language sentences or scene graphs (Li et al. 2019c; Tseng et al. 2020). Compared with sentences, the scene graph description is more well-structured, since the nodes in a scene graph can represent objects and the edges can denote their relationship. Therefore, the synthesis from scene graphs allows better controllability. In this work, we focus on employing the scene graph description as the conditions.

Image Generation from Scene Descriptions

With the development of deep generative models, especially the adversarial generative networks, the synthesis from scene descriptions becomes feasible. Existing methods for such synthesis can be divided into two categories. The first kind of approach employs parametric generative models to tackle this task (Johnson, Gupta, and Fei-Fei 2018). The feature of objects and the relationships among objects are captured via a graph convolution network from a scene graph, then images are synthesized based on the extracted feature with the conditional generative models (Mirza and Osindero 2014). However, these methods often fail to generate realistic images for complicated scene descriptions, due to various objects and complex interactions in the scene. To this, semi-parametric approaches (Li et al. 2019c; Tseng et al. 2020) are recently proposed and they perform generation based on reference object crops. The reference crops are retrieved from an external bank and help to synthesize the final images. The retrieval module is a crucial component. PasteGAN (Li et al. 2019c) employed predefined retrieval modules that cannot be optimized during the training. RetrieveGAN (Tseng et al. 2020) later designed a differentiable retrieval process, thus enable optimizing the retrieve stage through the end-to-end training. However, they ignore the employing of image crops' spatial features in the optimization of retrieve stage.

Contrastive Learning

Contrastive learning has recently become a prominent approach in unsupervised representation learning. These methods learn representations by a "contrastive loss" which pushes apart dissimilar pairs (called negative pairs) while pulling together similar pairs (called positive pairs). The main difference between different approaches to complete contrastive learning lies in their strategies for obtaining positive and negative pairs. For the semantic computer vision task, e.g., image classification, different types of data augmentation are employed for obtaining positive pairs (Chen et al. 2020; Oord, Li, and Vinyals 2018; Khosla et al. 2020), including random cropping and flipping. For the language task, Logeswaran et al. (Logeswaran and Lee 2018) treat the context sentences as positive samples to learn representations. However, no existing works have analyzed the effect of applying contrastive learning in the crop retrieval for image synthesis.

Method

Given a scene graph $\mathcal{G}$ which contains a set of n objects $O = \{o_1, ..., o_n\}$ and their pairwise relations $R = \{r_1, ..., r_m\}$, our goal is to synthesize an image $x \in \mathbb{R}^{H \times W \times 3}$ from the scene graph. In addition, our method leverages an external pool of object crops (can be either segmented out or not) to facilitate the generation process. The overall framework is shown in Fig. 1 which consists of three stages. In the first stage, we leverage the scene graph to extract semantic features which are useful for crop retrieval and location prediction for each object. In the second stage, the sequential crop selection module (SCSM) sequentially selects a most compatible crop for each object given all previously chosen crops, which will be used for the image synthesis. Finally, the progressive scene graph to image module (PSGIM) synthesizes the target image based on the scene graph features and selected crops.

Scene Graph Encoding

Following (Li et al. 2019c), our method first processes the input scene graph to extracts text embeddings for all the

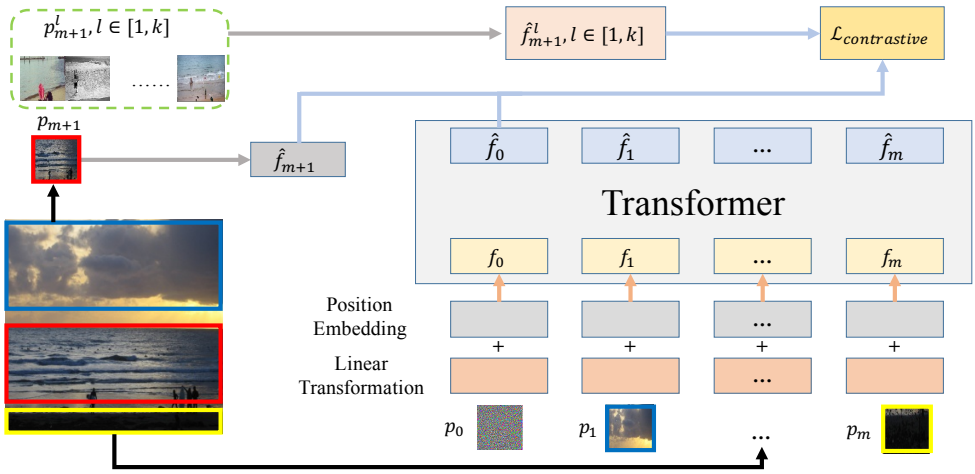


Figure 2: The illustration for the SCSM during training.

n objects as $\{t_1, ..., t_n\} = E_g(O, R)$ via a graph convolution network E_g, where $t_i \in \mathbb{R}^{C_t}$ is the text embedding for object o_i. For each object o_i, we match t_i with the text embeddings of other object crops in the external object pool to retrieve a set of its candidate crops $M(o_i) = \{p_i^1, ...p_i^k | p_i^j \in \mathbb{R}^{H_p \times W_p \times 3}, j \in [1,k]\}$ with a fixed size k. In addition, t_i is further used to predict a bounding box $b_i \in \mathbb{R}^4$ for object o_i. Please refer to (Li et al. 2019c) for details of these steps.

Sequential Crop Selection Module (SCSM)

To synthesize the final image, we first need to select only one crop from every object's k candidate crops. Our method performs the crop selection operation in an iterative fashion. Specifically, suppose there are already $m \in [1, n)$ crops $p_1, ..., p_m$ selected for object $o_1, ..., o_m$. Let us define the set $\{p_1, ..., p_m\}$ as the chosen crop set P. Given a new object o_{m+1}, our SCSM aims to select one crop from its k candidate crops which is most compatible with all the crops in P, and thus can improve the realism of the final synthesized image. To achieve this goal, we propose a novel contrastive learning framework, i.e. given a chosen crop set P from the same image, the compatible score between P and a new crop from the same image should be higher than the compatible score between P and a new crop from a different image. Such learning objective helps our model to select object crops likely belonging to the same image, and thus improves the compatibility among the selected crops.

We leverage a novel transformer to implement the idea, as shown in Fig. 2. Specifically, for every crop $p_i \in P$ (the shape is $H_p \times W_p \times 3$) with its predicted bounding box location b_i, we embed both its appearance and position information as an input token to the transformer as $f_i = W_1 \cdot p_i + E_b(b_i)$, where W_1 is a trainable linear transformation matrix to convert p_i into a 1-D embedding with shape $\mathbb{R}^{C_p}$. E_b is a position encoder with three nonlinear layers, the output shape of which is also $\mathbb{R}^{C_p}$. In addition, we add a learnable start token f_0 to represent the overall compatible feature of all the input tokens. Its appearance input p_0 is randomly initialized with the normal distribution while its position input is initialized with b_{m+1}, which is the predicted bounding box location of the new object o_{m+1}. A notable difference between previous methods (Li et al. 2019c; Tseng et al. 2020) and ours is that we explicitly leverage the position information of each crop, which is proven to be effective in our ablation study.

Following recent transformer structures (Vaswani et al. 2017), our transformer has a total of six layers, each of which consists of multi-head self-attention as well as MLP Let us denote the output embedding of the start token as $\widehat{f_0} \in \mathbb{R}^{C_p}$. Given a new candidate crop $p^l_{m+1}, l \in [1,k]$, we first apply another trainable linear matrix W_2 with the same shape as W_1 to obtain its appearance features as $\widehat{f}^l_{m+1} = W_2 \cdot p^l_{m+1}$. Then its compatible score with the chosen crop set P is computed as the cosine similarity between their embeddings, i.e. $\widehat{f_0} \cdot \widehat{f}^l_{m+1}$. Note that both the embeddings are normalized to the unit hypersphere before matching.

During training, given an image with its paired scene graph, our method randomly selects a crop set $P = \{p_1, ..., p_m | m \in [1, n)\}$ from the image. The size m is randomly determined to mimic the iterative selection process for inference. Then for a new object o_{m+1}, its crop p_{m+1} from the original image is treated as the positive crop while its retrieved candidate crops $\{p^1_{m+1}, ..., p^k_{m+1}\}$ from different images are treated as negative crops. Then the contrastive loss for this training image can be defined as

$$\mathcal{L}_{contrastive} = -\log \frac{\exp(\widehat{f_0} \cdot \widehat{f}_{m+1}/\tau)}{\exp(\widehat{f_0} \cdot \widehat{f}_{m+1}/\tau) + \sum_{l=1}^{k} \exp(\widehat{f_0} \cdot \widehat{f}^l_{m+1}/\tau)}, \quad (1)$$

where $\widehat{f}_{m+1}$ and $\widehat{f}^l_{m+1}$ are the embeddings of the positive crop p_{m+1} and a negative crop p^l_{m+1}. τ is a positive scalar temperature parameter.

During inference, given a scene graph and the predicted bounding boxes, since initially there is no crop selected for any object, our method simply randomly samples one object and randomly set one of its candidate crops as the chosen crop. Then, for each remaining object $o_i (i \in (2, n])$, we apply the trained SCSM once to find its candidate crop with

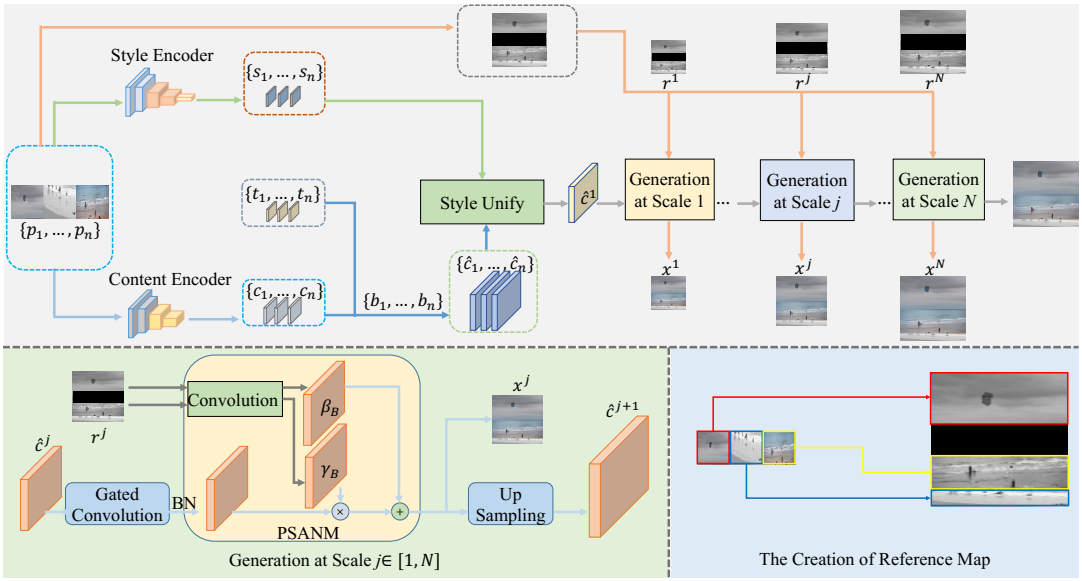


Figure 3: Our overall framework for PSGIM.

the highest compatible score and add it to the chosen set P. The process is repeated until every object has selected its own crop.

Progressive Scene Graph to Image Module

For every object o_i in the given scene graph, we now have its scene-graph embedding t_i, bounding box prediction b_i as well as a selected object crop p_i. Our PSGIM leverages these inputs to generate the final image $x \in \mathbb{R}^{H \times W \times 3}$. The framework is shown in Fig. 3.

Our generation module first leverages a content encoder E_c and a style encoder E_s to extract different features from the object crop p_i. The output content feature of E_c is used as the generator input to provide the structural information of object o_i in the generated image. The output style feature of E_s is used to modulate the content feature to have a unified style with the other object crops, and thus improves the overall realism of the final generated image. It is worth noting that although our SCSM is able to select mutually compatible object crops, it is still possible that the initially retrieved candidate crop sets do not have enough compatible crops to choose. Therefore we propose a style unifier in the generator to handle the limitation.

Specifically, we extract the content feature as $c_i = E_c(p_i), c_i \in \mathbb{R}^{H_c \times W_c \times C_c}$. We further combine the spatially-expanded scene-graph feature t_i with c_i to include more semantics. The new feature is further interpolated and pasted onto a specific region of a zero feature map with the shape $\mathbb{R}^{\frac{H}{2^{N-1}} \times \frac{W}{2^{N-1}} \times (C_c + C_t)}$, where the location of the region is determined by the location b_i scaled by $\frac{1}{2^{N-1}}$ and N is the number of output scales of the generator. Let $\hat{c}_i$ denote the final pasted feature.

The style feature is extracted as $s_i = E_s(p_i), s_i \in \mathbb{R}^{C_s}$. Our style unifier takes as input the averaged style features of all object crops and produces as output the modulation parameters which are applied to normalize the channels of each content feature $\hat{c}_i$ independently.

$$\begin{aligned} &\gamma_s, \beta_s = \text{StyleUnifier}(\tfrac{s_1 + \ldots + s_n}{n}), \quad \gamma_s, \beta_s \in \mathbb{R}^{C_c + C_t}, \\ &\hat{c}_i = \gamma_s \tfrac{\hat{c}_i - \mu_i}{\sqrt{\sigma_i + \epsilon}} + \beta_s, \quad \hat{c}_i \in \mathbb{R}^{\frac{H}{2^{N-1}} \times \frac{W}{2^{N-1}} \times (C_c + C_t)}, \end{aligned} \tag{2}$$

where the style unifier is implemented via a MLP with several nonlinear layers. μ_i and σ_i are the mean and variance of the content feature $\hat{c}_i$ and ϵ is a small positive constant for numerical stability. Finally, the normalized content features of all crops are aggregated together to represent the generator input at the coarsest level, i.e. $\hat{c}^1 = \sum_{i=1}^{n} \hat{c}_i$.

Our generator has a hierarchical structure with N output scales. At every scale $j \in [1, N]$, the generator takes as input $\hat{c}^j$ and produces an output image $x^j \in \mathbb{R}^{\frac{H}{2^{N-j}} \times \frac{W}{2^{N-j}} \times 3}$. There are two important network components at every scale of the generator. The first component is the gated convolutions employed from (Yu et al. 2019) which aims to inpaint the missing areas uncovered by any object crops. The second component is the patch-guided spatially adaptive normalization module (PSANM) that is inspired from the SPADE (Park et al. 2019) module, where we first copy and paste the object crops into a reference image and then use it to guide the structure and content of the generated image. Specifically, the reference image $r^j \in \mathbb{R}^{\frac{H}{2^{N-j}} \times \frac{W}{2^{N-j}} \times 1}$ is generated by pasting the gray scales of all object crops p_i onto an empty canvas based on their location b_i scaled by a factor $\frac{1}{2^{N-j}}$. The crops are turned into gray scale to eliminate the negative effects of possible inconsistent color styles, which is already handled by our style unifier. Then similar to the semantic mask input to SPADE, our reference image r^j is employed to predict spatially adaptive normalization parameters. Please refer to Fig. 3 for more details about PSANM.

Experiments demonstrate the superiority of our PSGIM to existing generators for the synthesis from scene graphs.

Training losses. Given the generated image output at the j-th scale, we propose several losses to train our generator. First, for a training image y with its paired scene graph, we can use ground truth crops and bounding boxes for each object, and thus the generated output x^j should reconstruct $(y)^{\downarrow j}$, where $(\cdot)^{\downarrow j}$ denotes the operation of downsampling of an image to the j-th scale.

$$\mathcal{L}_r^j = \mathbb{E}[\|(y)^{\downarrow j} - x^j\|_2]. \quad (3)$$

We also leverage the perceptual loss (Johnson, Alahi, and Fei-Fei 2016; Chen, Xu, and Jia 2020) to compare x^j with $(y)^{\downarrow j}$ using ImageNet (Deng et al. 2009) pretrained VGG (Simonyan and Zisserman 2015) features Φ_k, as

$$\mathcal{L}_p^j = \sum_l \mathbb{E}[\|\Phi_l((y)^{\downarrow j}) - \Phi_l(x^j)\|_2]. \quad (4)$$

In addition, we apply the adversarial loss (Mao et al. 2017) with a discriminator D^j at the j-the scale, as

$$\mathcal{L}_d^j = \mathbb{E}[(D^j((y)^{\downarrow j})-1)^2+(D^j(x^j))^2], \quad \mathcal{L}_g^j = \mathbb{E}[(D^j(x^j)-1)^2]. \quad (5)$$

Furthermore, we propose a consistency loss term to encourage the similarity between the generated outputs at different scales, as

$$\mathcal{L}_c = \sum_{j=1}^{N-1} \mathbb{E}[\|(x^N)^{\downarrow j} - x^j\|]. \quad (6)$$

Besides using ground truth crops and bounding boxes, we can also use retrieved crops and predicted bounding boxes to generate a new image $\bar{x}^j$ which does not have corresponding ground truth image. Therefore, we can only apply the adversarial loss and consistency loss for it.

In summary, the total loss to train our generator at all scales can be written as

$$\mathcal{L} = \lambda_1 \sum_{x^j,j} \mathcal{L}_r^j + \lambda_2 \sum_{x^j,j} \mathcal{L}_p^j + \lambda_3 \sum_{x^j,\bar{x}^j,j} \mathcal{L}_g^j + \lambda_4 \sum_{x^j,\bar{x}^j} \mathcal{L}_c, \quad (7)$$

where λ_1 to λ_4 are parameters to balance various losses.

Experiments

Datasets

The COCO-Stuff (Caesar, Uijlings, and Ferrari 2018) and Visual Genome (Krishna et al. 2017) datasets are standard benchmark datasets for evaluating scene-graph-to-image generation models. Our framework can synthesize images with arbitrary resolution. However, due to the computation limitation, we use the image resolution of 256 $\times$ 256 for all the experiments. We follow the protocol in sg2im (Johnson, Gupta, and Fei-Fei 2018) to pre-process the dataset and complete the train-test split.

Implementation Details

We implement with PyTorch (Paszke et al. 2017) and train SCSM and PSGIM with 90 epochs on both the COCO-Stuff and Visual Genome datasets. In addition, we use the Adam optimizer (Kingma and Ba 2015) with a batch size of 16. The learning rates for the generator and discriminator are both 0.0001, and the exponential decay rates (β_1, β_2) are set to be (0, 0.9). We set the hyper-parameters as follows: $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $\lambda_3 = 0.02$, $\lambda_4 = 1.0$. For the training of SCSM, the proportion between positive samples and negative samples is 1:10. The number of candidate crops for each object during inference is 5. To implement the perceptual loss term, we use the ReLU1_2, ReLU2_2, ReLU3_3, ReLU4_3, ReLU5_3 layers of an ImageNet-pretrained VGG-16 network (Simonyan and Zisserman 2015). The crop size is set to 64×64 and 32×32 for COCO-Stuff and Visual Genome.

Baselines

The baselines for the comparison include four categories: 1) the parametric generative models for mapping scene graphs to images sg2im (Johnson, Gupta, and Fei-Fei 2018); 2) the semi-parametric approaches PasteGAN and RetrieveGAN (Li et al. 2019c; Tseng et al. 2020); 3) the text-to-image methods AttnGAN (Xu et al. 2018), MirrorGAN (Qiao et al. 2019),ControlGAN (Li et al. 2019a), DM-GAN (Zhu et al. 2019),Obj-GAN (Li et al. 2019b); 4) the layout-to-image methods Reconfigurable (Sun and Wu 2019), Specifying (Ashual and Wolf 2019),Layout2im (Zhao et al. 2019). For the text-to-image methods, we follow the strategy in RetrieveGAN (Tseng et al. 2020) for comparison: we convert the scene graph to the corresponding text description. Specifically, we convert each relationship in the graph into a sentence, and link every sentence via the conjunction word "and". The layout-to-image methods take input as the ground-truth bounding boxes. For a fair comparison, all baselines are trained to synthesize with resolution of 256×256.

Metrics

We employ three metrics. 1) Inception Score (IS) (Salimans et al. 2016): IS uses the Inception V3 (Szegedy et al. 2016) model to measure the visual quality of the generated images. 2) Fr´echet Inception Distance (FID) (Heusel et al. 2017): FID measures the visual quality and diversity of the synthesized images. 3) Diversity (DS): we follow the setting of (Li et al. 2019c; Tseng et al. 2020) to evaluate the diversity by measuring distances among features of synthesized images (these images are synthesized with same input scene graph while different crops) using the Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018) metric.

Comparison with Existing Approaches

Quantitative evaluation. To have a fair comparison with different methods, we conduct the evaluation using two different settings. First, bounding boxes of objects are predicted by models. Second, ground-truth bounding boxes are given as inputs in addition to the scene graph. The results of these two settings are shown in Table 1. Since our approach has an optimized crop retrieval process and better generator structure, our approach performs favorably against the other algorithms. And our results are also superior over SOTA text-to-image and layout-to-image methods on both COCO-Stuff and Visual Genome datasets. Moreover, since our framework can have different crops for the initialization of the chosen crops in SCSM, our framework synthesizes comparably diverse images compared to others.

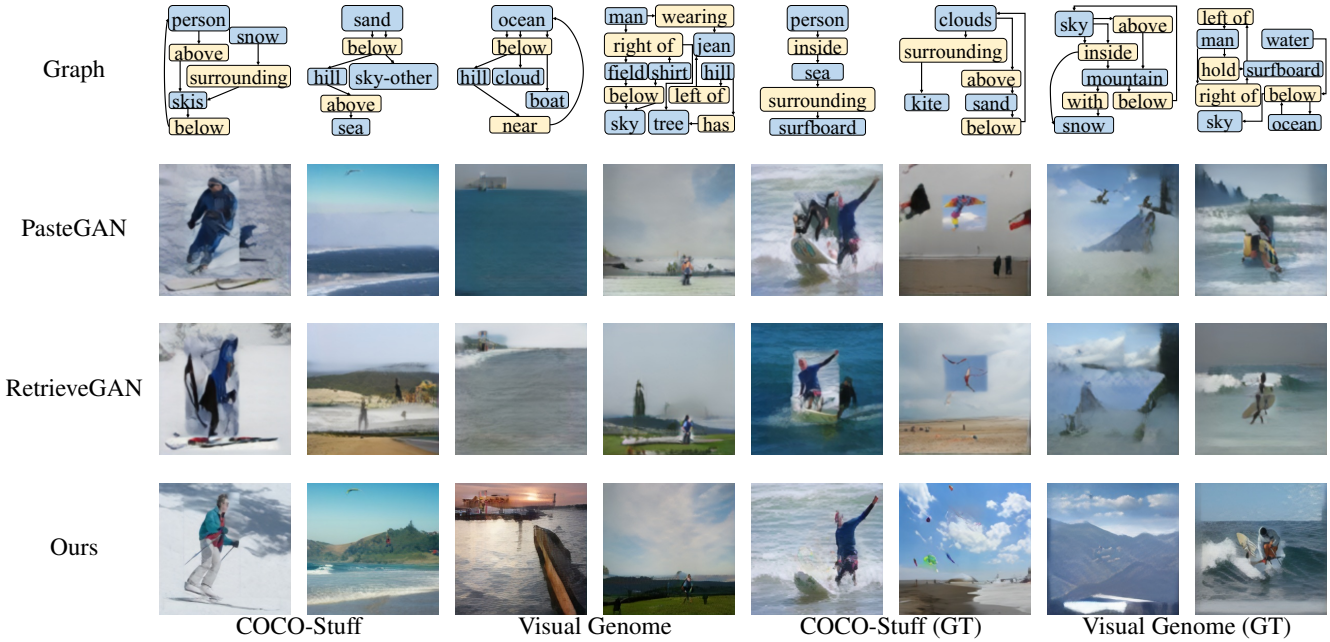


Figure 4: The synthesis on COCO-Stuff (left four columns), Visual Genome (right four columns).

Datasets	COCO-Stuff			Visual Genome		
	FID ↓	IS ↑	DS ↑	FID ↓	IS ↑	DS ↑
sg2im	226.3	3.8	0.02	210.0	4.7	0.10
AttnGAN	80.1	9.2	0.15	126.1	8.4	0.30
MirrorGAN	78.3	9.5	0.15	123.3	8.7	0.31
ControlGAN	86.1	8.6	0.14	135.6	7.7	0.28
DM-GAN	68.0	10.9	0.18	107.0	10.5	0.34
Obj-GAN	68.4	10.8	0.19	107.8	10.4	0.35
PasteGAN	78.8	8.5	0.60	131.6	6.5	0.38
RetrieveGAN	56.9	10.2	0.47	113.1	7.5	0.30
Ours	**51.6**	**15.2**	**0.63**	**63.7**	**10.8**	**0.59**
sg2im (GT)	100.9	9.9	0.02	141.3	6.8	0.14
Layout2im	50.6	11.4	0.55	60.6	10.7	0.53
Reconfigurable	48.6	14.0	0.56	57.6	10.1	0.54
Specifying	65.2	12.4	0.62	63.3	11.0	**0.61**
PasteGAN (GT)	70.2	11.0	0.45	114.3	9.5	0.27
RetrieveGAN (GT)	54.6	12.3	0.25	77.7	10.8	0.22
Ours (GT)	**46.9**	**15.5**	**0.64**	**56.7**	**11.4**	0.59

Table 1: ↑ means the higher the better, ↓ means the lower the better. The top part shows results of employing the predicted bounding boxes during the inference, and the bottom part displays results of using the ground-truth bounding boxes.

Datasets	COCO-Stuff			Visual Genome		
	Other	Same	Ours	Other	Same	Ours
sg2im	15.3	8.7	**76.0**	17.3	2.7	**80.0**
AttnGAN	24.0	4.7	**71.3**	16.7	8.0	**75.3**
MirrorGAN	20.7	6.0	**73.3**	8.0	10.0	**82.0**
ControlGAN	18.0	13.3	**68.7**	16.6	16.7	**66.7**
DM-GAN	20.7	12.7	**66.6**	15.3	28.7	**56.0**
Obj-GAN	17.3	10.7	**72.0**	10.0	14.0	**76.0**
PasteGAN	8.0	17.3	**74.7**	3.4	19.3	**77.3**
RetrieveGAN	12.7	5.3	**82.0**	6.0	6.0	**88.0**
sg2im (GT)	8.7	9.3	**82.0**	2.6	6.7	**90.7**
Layout2im	19.3	10.0	**70.7**	14.0	12.7	**73.3**
Reconfigurable	25.3	7.3	**67.4**	26.6	10.7	**62.7**
Specifying	22.7	12.7	**64.6**	14.0	18.0	**68.0**
PasteGAN (GT)	18.7	11.3	**70.0**	9.3	14.7	**76.0**
RetrieveGAN (GT)	28.7	16.0	**55.3**	26.6	12.7	**60.7**

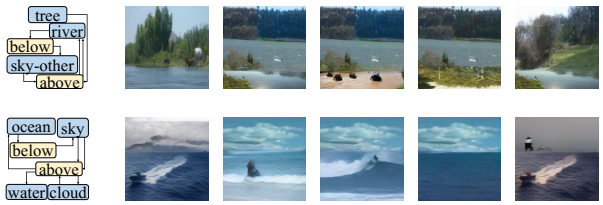
Table 2: User preference in the user study for the synthesis. "Ours" is the percentage (%) that our result is preferred, "Other" is the percentage that other method is selected, "Same" means the percentage that the users can not decide.

Qualitative evaluation. Furthermore, we qualitatively compare the visual results generated by different methods in Fig. 4. We show the results on the COCO-Stuff and the Visual Genome datasets under two settings of using predicted and ground-truth bounding boxes. Moreover, our model synthesizes comparably diverse images compared to the other schemes. Our framework can synthesize diverse images by setting different crops to initialize the set of chosen crops in SCSM, and the diverse results can be viewed in Fig. 5.

User study. We synthesize images with our framework and all other baselines, and conduct a user study. We invite 30 participants to see one scene graph and two images synthesized by our framework and one of the baselines. And they will choose which one is better (they can also select that they have no preference). The criteria include the consistency with the input scene graph and the quality of synthesized images. Each participant is required to complete 70 pairs of AB-test. The results are shown in Table 2, demonstrating that our framework can better implement the synthesis.

Datasets		COCO-Stuff			Visual Genome		
Retrieve	Generation	FID ↓	IS ↑	DS ↑	FID ↓	IS ↑	DS ↑
PasteGAN	PSGIM	58.9	12.9	0.64	83.8	8.1	0.56
RetrieveGAN	PSGIM	55.3	13.2	**0.65**	80.2	8.5	0.57
SCSM w/o P	PSGIM	54.7	13.5	0.62	81.5	9.2	0.53
SCSM	PasteGAN	53.5	11.7	0.58	98.7	8.0	0.54
SCSM	PSGIM	**51.6**	**15.2**	0.63	**63.7**	**10.8**	**0.59**

Table 3: Ablation study's results. "w/o P" denotes the setting without using the position encoding of bounding boxes.



Input Graph Diverse Synthesized Images

Figure 5: Diverse synthesis on COCO-Stuff (first row) and Visual Genome (bottom row).

Ablation Study

The effects of SCSM and PSGIM. First, we prove the effectiveness of our proposed SCSM. We replace SCSM with the pre-trained embedding function in PasteGAN (Li et al. 2019c), and the differentiable retrieval module in RetrieveGAN (Tseng et al. 2020). Second, we validate the effect of our PSGIM. We replace PSGIM with the generator in PasteGAN and RetrieveGAN while keeping our trained SCSM. We conduct the ablation studies on both COCO-Stuff dataset and Visual Genome dataset, and the results are shown in Table 3. Obviously, our retrieve process is better than the retrieve strategy proposed by PasteGAN and RetrieveGAN. Moreover, the ablation study also proves the superiority of our generator compared with the generator proposed by PasteGAN.

The effect of the position encoding. We also validate the effect of using the position encoding in our SCSM, by deleting the position encoding of bounding boxes in SCSM (the results are denoted as "SCSM w/o P"). As shown in Table 3, the deletion of position encoding causes the decrease of the performance.

The influence of candidate crops' number. Moreover, we also analyze the impact from the number of candidate crops for each object. The candidate crops are pre-retrieved by the method of PasteGAN (Li et al. 2019c) and the number of candidate crops is 5 in the above experiments. In this section, we conduct experiments with the number of candidate crops as 50 and 100. As shown in Table 4, the performance is improved a little since our retrieve module will preferentially choose the candidate crop that has a higher rank after the pre-retrieve. We expect the improvement will be more clear if a large-scale object bank (e.g., Google search engine) is available. However, if we set the number of candidate crops high, the time cost for retrieve is increased a lot. Thus, the choice of 5 in this paper is rational and practical.

Datasets	COCO-Stuff			Visual Genome		
	FID ↓	IS ↑	DS ↑	FID ↓	IS ↑	DS ↑
Ours (50)	50.8	15.8	0.66	62.5	11.2	0.62
Ours (100)	50.5	16.3	0.68	62.1	11.7	0.64
Ours (5)	51.6	15.2	0.63	63.7	10.8	0.59

Table 4: "Ours (xx)" means the results with the number of candidate crops as xx.

Datasets	COCO-Stuff			Visual Genome		
Retrieve	FID ↓	IS ↑	DS ↑	FID ↓	IS ↑	DS ↑
RetrieveGAN	65.1	8.5	0.59	126.5	4.2	0.45
SCSM w/o P	64.9	8.6	0.58	126.0	4.3	0.44
SCSM	**63.5**	**8.9**	**0.59**	**125.6**	**4.5**	**0.46**

Table 5: Comparison among retrieve strategies.

The average number of external images. For the generation from one scene graph, our SCSM would obtain a set of patches for synthesis. These crops are extracted from a set of external images, and we calculate the average number. For COCO-stuff, retrieved patches for one scene graph are cropped from 5.46 external images averagely; for Visual Genome, the average number is 4.18. These results effectively demonstrate that the good results of our method are not simply because it retrieves a real image matching the given scene graph. Instead, it is because of its ability to search mutually compatible patches from different image sources and strong generation capability.

Evaluation of crop selection. We can paste retrieved crops from different methods onto one image, building a canvas. And we sort the crops according to their area and put the crops with a larger size to the back. To further demonstrate the superiority of our retrieve strategy directly, we compute the FID, IS, and DS for the canvas formed by different retrieve methods. The results are shown in Table 5. The results demonstrate three important conclusions: 1) our retrieved crops are more mutually compatible than other baselines; 2) our generator (PSGIM) can further enhance the mutual compatibility, producing more realistic images than the canvas; 3) position encoding is useful to improve the results while previous methods fail to leverage such information.

Conclusion

We in this paper propose a novel Sequential Crop Selection Module (SCSM) and Progressive Scene Graph to Image Module (PSGIM). In SCSM, the selection of the image crop for each object would be determined with the contents and locations of image crops that have been chosen previously. Such sequential selection is implemented with a transformer that is trained with contrastive learning. Hierarchical gated convolutions in the generator are employed to enhance the areas that are not covered by any image crops; a patch-guided spatially adaptive normalization module is also proposed to guarantee the generated images highly respecting the crops. Evaluated on Visual Genome and COCO-Stuff, the results demonstrate the superiority of our proposed over SOTA methods.

References

Ashual, O.; and Wolf, L. 2019. Specifying object attributes and relations in interactive scene generation. In *ICCV*.

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. Coco-stuff: Thing and stuff classes in context. In *CVPR*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

Chen, Y.-C.; Xu, X.; and Jia, J. 2020. Domain adaptive image-to-image translation. In *CVPR*.

Chen, Y.-C.; Xu, X.; Tian, Z.; and Jia, J. 2019. Homomorphic latent space interpolation for unpaired image-to-image translation. In *CVPR*.

Choi, Y.; Uh, Y.; Yoo, J.; and Ha, J.-W. 2020. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial networks. In *NIPS*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*.

Huang, H.-P.; Tseng, H.-Y.; Lee, H.-Y.; and Huang, J.-B. 2020. Semantic view synthesis. In *ECCV*.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*.

Johnson, J.; Gupta, A.; and Fei-Fei, L. 2018. Image generation from scene graphs. In *CVPR*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. In *NIPS*.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*.

Lee, H.-Y.; Tseng, H.-Y.; Mao, Q.; Huang, J.-B.; Lu, Y.-D.; Singh, M.; and Yang, M.-H. 2020. Drit++: Diverse image-to-image translation via disentangled representations. *IJCV*.

Li, B.; Qi, X.; Lukasiewicz, T.; and Torr, P. H. 2019a. Controllable text-to-image generation. In *NIPS*.

Li, W.; Zhang, P.; Zhang, L.; Huang, Q.; He, X.; Lyu, S.; and Gao, J. 2019b. Object-driven text-to-image synthesis via adversarial training. In *CVPR*.

Li, Y.; Ma, T.; Bai, Y.; Duan, N.; Wei, S.; and Wang, X. 2019c. Pastegan: A semi-parametric method to generate image from scene graph. In *NIPS*.

Liu, B.; Zhu, Y.; Song, K.; and Elgammal, A. 2021. Self-Supervised Sketch-to-Image Synthesis. In *AAAI*.

Logeswaran, L.; and Lee, H. 2018. An efficient framework for learning sentence representations. In *ICLR*.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; and Paul Smolley, S. 2017. Least squares generative adversarial networks. In *ICCV*.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv:1411.1784*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv:1807.03748*.

Park, T.; Liu, M.-Y.; Wang, T.-C.; and Zhu, J.-Y. 2019. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Qiao, T.; Zhang, J.; Xu, D.; and Tao, D. 2019. Mirrorgan: Learning text-to-image generation by redescription. In *CVPR*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *NIPS*.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Sun, W.; and Wu, T. 2019. Image synthesis from reconfigurable layout and style. In *ICCV*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *CVPR*.

Tan, Z.; Chai, M.; Chen, D.; Liao, J.; Chu, Q.; Liu, B.; Hua, G.; and Yu, N. 2021. Diverse Semantic Image Synthesis via Probability Distribution Modeling. In *CVPR*.

Tseng, H.-Y.; Lee, H.-Y.; Jiang, L.; Yang, M.-H.; and Yang, W. 2020. Retrievegan: Image synthesis via differentiable patch retrieval. In *ECCV*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; and He, X. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*.

Yang, G.; Fei, N.; Ding, M.; Liu, G.; Lu, Z.; and Xiang, T. 2021. L2M-GAN: Learning To Manipulate Latent Space Semantics for Facial Attribute Editing. In *CVPR*.

Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2019. Free-form image inpainting with gated convolution. In *ICCV*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

Zhao, B.; Meng, L.; Yin, W.; and Sigal, L. 2019. Image generation from layout. In *CVPR*.

Zhu, M.; Pan, P.; Chen, W.; and Yang, Y. 2019. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *CVPR*.