RetrieveGAN: Image Synthesis via Differentiable Patch Retrieval

Hung-Yu Tseng[*2], Hsin-Ying Lee[*2], Lu Jiang[1], Ming-Hsuan Yang[1,2,3], Weilong Yang[1]

[1]Google Research   [2]University of California, Merced   [3]Yonsei University
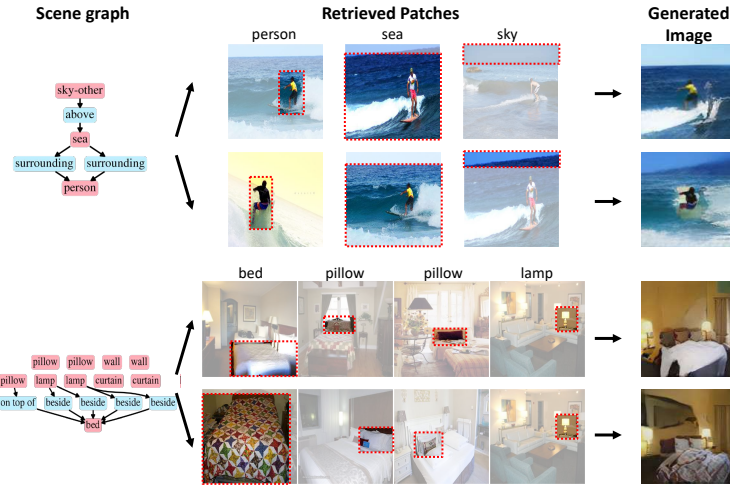
Fig. 1: **Image synthesize from retrieved examples.** We propose the RetrieveGAN model that takes as input the scene graph description and learns to 1) select mutually compatible image patches via a differentiable retrieval process and 2) synthesize the output image from the retrieved patches.

Abstract. Image generation from scene description is a cornerstone technique for the controlled generation, which is beneficial to applications such as content creation and image editing. In this work, we aim to synthesize images from scene description with retrieved patches as reference. We propose a differentiable retrieval module. With the differentiable retrieval module, we can (1) make the entire pipeline end-to-end trainable, enabling the learning of better feature embedding for retrieval; (2) encourage the selection of mutually compatible patches with additional objective functions. We conduct extensive quantitative and qualitative experiments to demonstrate that the proposed method can generate realistic and diverse images, where the retrieved patches are reasonable and mutually compatible.

---

[*] Equal contribution. Work done during their internships at Google Research.

1   Introduction

Image generation from scene descriptions has received considerable attention. Since the description often requests multiple objects in a scene with complicated relationships between objects, it remains challenging to synthesize images from scene descriptions. The task requires not only the ability to generate realistic images but also the understanding of the mutual relationships among different objects in the same scene. The usage of the scene description provides flexible user-control over the generation process and enables a wide range of applications in content creation [18] and image editing [24].

Taking advantage of generative adversarial networks (GANs) [5], recent research employs conditional GAN for the image generation task. Various conditional signals have been studied, such as scene graph [13], bounding box [40], semantic segmentation map [24], audio [19], and text [36]. A stream of work has been driven by parametric models that rely on the deep neural network to capture and model the appearance of objects [13,36]. Another stream of work has recently emerged to explore the semi-parametric model that leverages a memory bank to retrieve the objects for synthesizing the image [25,37].

In this work, we focus on the semi-parametric model in which a memory bank is provided for the retrieval purpose. Despite the promising results, existing retrieval-based image synthesis methods face two issues. First, the current models require pre-defined embeddings since the retrieval process is non-differentiable. The pre-defined embeddings are independent of the generation process and thus cannot guarantee the retrieved objects are suitable for the surrogate generation task. Second, oftentimes there are multiple objects to be retrieved given a scene description. However, the conventional retrieval process selects each patch independently and thus neglect the subtle mutual relationship between objects.

We propose *RetrieveGAN*, a conditional image generation framework with a differentiable retrieval process to address the issues. First, we adopt the Gumbel-softmax [11] trick to make the retrieval process differentiable, thus enable optimizing the embedding through the end-to-end training. Second, we design an iterative retrieval process to select a set of compatible patches (i.e., objects) for synthesizing a single image. Specifically, the retrieval process operates iteratively to retrieve the image patch that is most compatible with the already selected patches. We propose a co-occurrence loss function to boost the mutual compatibility between the selected patches. With the proposed differentiable retrieval design, the proposed RetrieveGAN is capable of retrieving image patches that 1) considers the surrogate image generation quality, and 2) are mutually compatible for synthesizing a single image.

We evaluate the proposed method through extensive experiments conducted on the COCO-stuff [2] and Visual Genome [16] datasets. We use three metrics, Fréchet Inception Distance (FID) [9], Inception Score (IS) [26], and the Learned Perceptual Image Patch Similarity (LPIPS) [39], to measure the realism and diversity of the generated images. Moreover, we conduct the user study to validate the proposed method's effectiveness in selecting mutually compatible patches.

To summarize, we make the following contributions in this work:

- We propose a novel semi-parametric model to synthesize images from the scene description. The proposed model takes advantage of the complementary strength of the parametric and non-parametric techniques.
- We demonstrate the usefulness of the proposed differentiable retrieval module. The differentiable retrieval process can be jointly trained with the image synthesis module to capture the relationships among the objects in an image.
- Extensive qualitative and quantitative experiments demonstrate the efficacy of the proposed method to generate realistic and diverse images where retrieved objects are mutually compatible.

2   Related Work

Conditional image synthesis. The goal of the generative models is to model a data distribution given a set of samples from that distribution. The data distribution is either modeled explicitly (e.g., variational autoencoder [15]) or implicitly (e.g., generative adversarial networks [5]). On the basis of unconditional generative models, conditional generative models target synthesizing images according to additional context such as image [4,17,23,30,41], segmentation mask [10,24,33,43], and text. The text conditions are often expressed in two formats: natural language sentences [36,38] or scene graphs [13]. Particularly, the scene graph description is in a well-structured format (i.e., a graph with a node representing objects and edges describing their relationship), which mitigates the ambiguity in natural language sentences. In this work, we focus on using the scene graph description as our input for the conditional image synthesis.

Image synthesis from scene descriptions. Most existing methods employ parametric generative models to tackle this task. The appearance of objects and relationships among objects are captured via a graph convolution network [13,21] or a text embedding network [20,29,36,38,42], then images are synthesized with the conditional generative approach. However, current parametric models synthesize objects at pixel-level, thus failing to generate realistic images for complicated scene descriptions. More recent frameworks [25,37] adopt semi-parametric models to perform generation at patch-level based on reference object patches. These schemes retrieve reference patches from an external bank and use them to synthesize the final images. Although the retrieval module is a crucial component, existing works all use predefined retrieval modules that cannot be optimized during the training stage. In contrast, we propose a novel semi-parametric model with a differentiable retrieval process that is end-to-end trainable with the conditional generative model.

Image Retrieval. Image retrieval has been a classical vision problem with numerous applications such as product search [22,6,1], multimodal image retrieval [32,3], image geolocalization [7], event detection [12], among others. Solutions based on deep metric learning use the triplet loss [8,31] or softmax cross-entropy objective [32] to learn a joint embedding space between the query (e.g., text, image, or audio) and the target images. However, there is no prior work studying learning retrieval models for the image synthesis task. Different from the existing semi-parametric generative models [29,37] that use the

(a) RetrieveGAN

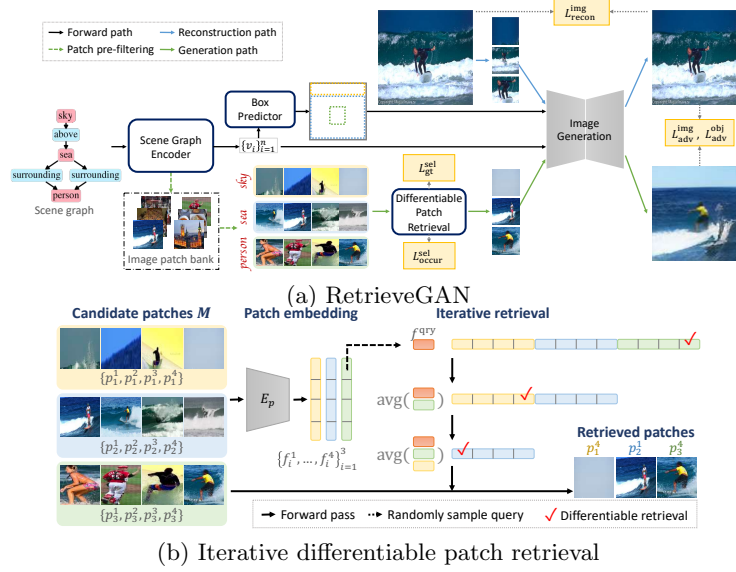(b) Iterative differentiable patch retrieval

Fig. 2: **Method overview.** (a) Our model takes as input the scene graph description and sequentially performs scene graph encoding, patch retrieval, and image generation to synthesize the desired image. (b) Given a set of candidate patches, we first extract the features using the patch embedding function. We then randomly select a patch feature as the query feature for the iterative retrieval process. At each step of the iterative procedure, we select the patch that is most compatible with the already selected patches. The iteration ends as all the objects are assigned with a selected patch.

pre-defined (or fixed) embedding to retrieve image patches, we propose a differentiable retrieval process that can be jointly optimized with the conditional generative model.

3 Methodology

3.1 Preliminaries

Our goal is to synthesize an image $x \in \mathbb{R}^{H \times W \times 3}$ from the input scene graph g by compositing appropriate image patches retrieved from the image patch bank. As the overview shown in Figure 2, the proposed RetrieveGAN framework consists of three stages: scene graph encoding, patch retrieval, and image generation. The scene graph encoding module processes the input scene graph g, extracts features, and predicts bounding box coordinates for each object o_i defined in the scene graph. The patch retrieval module then retrieves an image patch for each object o_i from the image patch bank. The goal of the retrieval module is to maximize the compatibility of all retrieved patches, thus improving the quality of the image synthesized by the subsequent image generation module. Finally,

the image generation module takes as input the selected patches along with the predicted bounding boxes to synthesize the final image.

Scene graph. Serving as the input data to our framework, the scene graph representation [14] describes the objects in a scene and the relationships between these objects. We denote a set of object categories as $\mathcal{C}$ and relation categories as $\mathcal{R}$. A scene graph g is then defined as a tuple $(\{o_i\}_{i=1}^n, \{e_i\}_{i=1}^m)$, where $\{o_i | o_i \in \mathcal{C}\}_{i=1}^n$ is a set of objects in the scene. The notation $\{e_i\}_{i=1}^m$ denotes a set of direct edges in the form of $e_i = (o_j, r_k, o_t)$ where $o_j, o_t \in \mathcal{C}$ and $r_k \in \mathcal{R}$.

Image patch bank. The second input to our model is the memory bank consisting of all available real image patches for synthesizing the output image. Following PasteGAN [37], we use the ground-truth bounding box to extract the images patches $M = \{p_i \in \mathbb{R}^{h \times w \times 3}\}$ from the training set. Note that we relax the assumption in PasteGAN and do not use the ground-truth mask to segment the image patches in the COCO-Stuff [2] dataset.

3.2   Scene Graph Encoding

The scene graph encoding module aims to process the input scene graph and provides necessary information for the later patch retrieval and image generation stages. We detail the process of scene graph encoding as follows:

Scene graph encoder. Given an input scene graph $g = (\{o_i\}_{i=1}^n, \{e_i\}_{i=1}^m)$, the scene graph encoder E_g extracts the object features, namely $\{v_i\}_{i=1}^n = E((\{o_i\}_{i=1}^n, \{e_i\}_{i=1}^m))$. Adopting the strategy in sg2im [13], we construct the scene graph encoder with a series of graph convolutional networks (GCNs). We further discuss the detail of the scene graph encoder in the supplementary document.

Bounding box predictor. For each object o_i, the bounding box predictor learns to predict the bounding box coordinates $\hat{b}_i = (x_0, y_0, x_1, y_1)$ from the object features v_i. We use a series of fully-connected layers to build the predictor.

Patch pre-filtering. Since there are a large number of image patches in the image patch bank, performing the retrieval on the entire bank online is intractable in practice due to the memory limitation. We address this problem by pre-filtering a set of k candidate patches $M(o_i) = \{p_i^1, p_i^2, \cdots, p_i^k\}$ for each object o_i. And the later patch retrieval process is conducted on the pre-filtered candidate patches as opposed to the entire patch bank. To be more specific, we use the pre-trained GCN in sg2im [13] to obtain the candidate patches for each object. We use the corresponding scene graph to compute the GCN feature. The computed GCN feature is used to select similar candidate patches $M(o_i)$ with respect to the negative ℓ_2 distance.

3.3   Patch Retrieval

The patch retrieval aims to select a number of mutually compatible patches for synthesizing the output image. We illustrate the overall process on the bottom side of Figure 2. Given the pre-filtered candidate patches $\{M(o_i)\}_{i=1}^n$, we first use a patch embedding function E_p to extract the patch features. Starting with a

randomly sampled patch feature as a query, we propose an iterative retrieval process to select compatible patches for all objects. In the following, we 1) describe how a single retrieval is operated, 2) introduce the proposed iterative retrieval process, and 3) discuss the objective functions used to facilitate the training of the patch retrieval module.

Differentiable retrieval for a single object. Given the query feature f^{qry}, we aim to sample a single patch from the candidate set $M(o) = \{p^1, p^2, \cdots, p^k\}$ for object o. Let $\pi \in \mathbb{R}^k_{>0}$ be the categorical variable with probabilities $P(x = i) \propto \pi_i$ which indicates the probability of selecting the i-th patch from the bank. To compute π_i, we calculate the ℓ_2 distance between the query feature and the corresponding patch feature, namely $\pi_i = e^{-\|f_{\mathrm{qry}} - E_p(p^i; \theta_{E_p})\|_2}$, where E_p is the embedding function and θ_{E_p} is the learnable mode parameter. The intuition is that the candidate patch with smaller feature distance to the query feature should be sampled with higher probability. By optimizing θ_{E_p} with our loss functions, we hope our model is capable of retrieving compatible patches. As we are sampling from a categorical distribution, we use the Gumbel-Max trick [11] to sample a single patch:

$$\arg\max_i[P(x = i)] = \arg\max_i[g_i + \log \pi_i] = \arg\max_i[\hat{\pi}_i], \tag{1}$$

where $g_i = -\log(-\log(u_i))$ is the re-parameterization term and $u_i \sim \mathrm{Uniform}(0, 1)$. To make the above process differentiable, the argmax operation is approximated with the continuous softmax operation:

$$s = \mathrm{softmax}(\hat{\pi}) = \frac{\exp(\hat{\pi}_i / \tau)}{\sum_{q=1}^k \exp(\hat{\pi}_q / \tau)}, \tag{2}$$

where τ is the temperature controlling the degree of the approximation.[1]

Iterative differentiable retrieval for multiple objects. Rather than retrieving only a single image patch, the proposed framework needs to select a subset of n patches for the n objects defined in the input scene graph. Therefore, we adopt the weighted reservoir sampling strategy [35] to perform the subset sampling from the candidate patch sets. Let $M = \{p_i | i = 1, \ldots, n \times k\}$ denote a multiset (with possible duplicated elements) consisting of all candidates patches in which n is the number of objects and k is the size of each candidate patch set. We leave the preliminaries on weighted reservoir sampling in the supplementary materials. In our problem, we first compute the vector $\hat{\pi}_i$ defined in (1) for all patches. We then iteratively apply n softmax operations over $\hat{\pi}$ to approximate the top-k selection. Let $\hat{\pi}_i^{(j)}$ denote the probability of sampling patch p_i at iteration j and $\hat{\pi}_i^{(1)} \leftarrow \hat{\pi}_i$. The probability is iteratively updated by:

$$\hat{\pi}_i^{(j+1)} \leftarrow \hat{\pi}_i^{(j)} + \log(1 - s_i^{(j)}), \tag{3}$$

---

[1] When τ is small, we found it is useful to make the selection variable s uni-modal. This can also be achieved by post-processing (e.g., thresholding) the softmax outputs.

where $s_i^{(j)} = \mathrm{softmax}(\hat{\pi}^{(j)})_i$ computed by (2). Essentially, (3) sets the probability of the selected patch to negative infinity, thus ensures this patch will not be chosen again. After n iterations, we compute the relaxed n-hot vector $s = \sum_{j=1}^{n} s^{(j)}$, where $s_i \in [0, 1]$ indicates the score of selecting the i-th patch and $\sum_{i=1}^{|M|} s_i = n$. The entire process is differentiable with respect to the model parameters.

We make two modifications to the above iterative process based on practical consideration. First, our candidate multiset $M = \{p_i\}_{i=1}^{n \times k}$ is formed by n groups of pre-filtered patches where every object has a group k patches. Since we are only allowed to retrieve a single patch from a group, we modify (3) by:

$$\hat{\pi}_i^{(j+1)} \leftarrow \hat{\pi}_i^{(j)} + \log(1 - \max_t[s_t^{(j)}]) \quad \forall t \text{ such that } m^{-1}(p_i) = t, \qquad (4)$$

where we denote $m^{-1}(p_j) = i$ if patch p_j in M is pre-fetched by the object o_i. (4) uses max pooling to disable selecting multiple patches from the same group. Second, to incorporate the prior knowledge that compatible images patches tend to lie closer in the embedding space, we use a greedy strategy to encourage selecting image patches that are compatible with the already selected ones. We detail this process in Figure 2(b). To be more specific, at each iteration, the features of the selected patches are aggregated by average pooling to update the query f^{qry}. π and $\hat{\pi}$ is also recomputed accordingly after the query update. This leads to a greedy strategy encouraging the selected patches to be visually or semantically similar in the feature space. We summarize the overall retrieval process in Algorithm 1.

As the retrieval process is differentiable, we can optimize the retrieval module (i.e., patch embedding function E_p) with the loss functions (e.g., adversarial loss) applied to the following image generation module. Moreover, we incorporate two additional objectives to facilitate the training of iterative retrieval process: ground-truth selection loss $L_{\mathrm{gt}}^{\mathrm{sel}}$ and co-occurrence loss $L_{\mathrm{occur}}^{\mathrm{sel}}$.

Ground-truth selection loss. As the ground-truth patches are available at the training stage, we add them to the candidate set M. Given one of the ground-truth patch features as the query feature f^{qry}, the ground-truth selection loss $L_{\mathrm{gt}}^{\mathrm{sel}}$ encourages the retrieval process to select the ground-truth patches for the other objects in the input scene graph.

Co-occurrence penalty. We design a co-occurrence loss to ensure the mutual compatibility between the retrieved patches. The core idea is to minimize the distances between the retrieved patches in a co-occurrence embedding space. Specifically, we first train a co-occurrence embedding function F_{occur} using the patches cropped from the training images with the triplet loss [34]. The distance on the co-occurrence embedding space between the patches sampled from the same image is minimized, while the distance between the patches cropped from the different images is maximized. Then the proposed co-occurrence loss is the pairwise distance between the retrieved patches on the co-occurrence embedding space:

$$L_{\mathrm{occur}}^{\mathrm{sel}} = \sum_{i,j} d(F_{\mathrm{occur}}(p_i), F_{\mathrm{occur}}(p_j)), \qquad (5)$$

Algorithm 1: Iterative Differential Retrieval

Input : Candidate patches $M = \{p_i\}_{i=1}^{n \times k}$ for n objects and each object has k pre-filtered patches.

Output: relaxed n-hot vector s where $\sum_{i=1}^{|M|} s_i = n$ and $0 \leq s_i \leq 1$.

1 **for** $i = 1, \ldots, |M|$ **do** $f_i = E_p(p_i)$ `// Get patch features`

2 Randomly select a patch feature to initialize the query f^{qry}

`// Iterative patch retrieval`

3 **for** $t = 1, \ldots, n$ **do**

4   **for** $i = 1, \ldots, |M|$ **do**

5 $\pi_i = e^{-\|f_i - f^{\mathrm{qry}}\|_2}$ `// Calculate` π `according to the query`

   `// Gumbel-Max trick`

6 $u_i \leftarrow \mathrm{Uniform}(0, 1)$

7 $\hat{\pi}_i \leftarrow -\log(-\log(u_i)) + \log(\pi_i)$

   `// Disable other patches in the selected group`

8 $\hat{\pi}_i \leftarrow \hat{\pi}_i + \log(1 - \max_j[s_j^{(t-1)}])$ $\forall j$ such that $m^{-1}(p_i) = j$

9   **end**

10   **for** $i = 1, \ldots, |M|$ **do** $s_i^{(t)} = \frac{\exp(\hat{\pi}_i / \tau)}{\sum_{q=1}^{|M|} \exp(\hat{\pi}_q / \tau)}$ `// Softmax operation`

11 $f^{\mathrm{qry}} = \mathrm{avg}(f^{\mathrm{qry}}, \sum_{i=1}^{|M|} s_i^{(t)} f_i)$ `// Update the query`

12 **end**

13 **return** the relaxed n-hot vector $s^{(n)}$

---

where p_i and p_j are the patches retrieved by the iterative retrieval process.

Limitations vs. advantages. The size of the candidate patches considered by the proposed retrieval process is currently limited by the GPU memory. Therefore, we cannot perform the differentiable retrieval over the entire memory bank. Nonetheless, the differentiable mechanism and iterative design enable us to train the retrieval process using the abovementioned loss functions that maximize the mutual compatibility of the selected patches.

3.4 Image Generation

Given selected patches after the differentiable patch retrieval process, the image generation module synthesizes the realistic image with the selected patches as reference. We adopt a similar architecture to PasteGAN [37] as our image generation module. Please refer to the supplementary materials for details regarding the image generation module. We use two discriminators D_{img} and D_{obj} to encourage the realism of the generated images on the image-level and object-level, respectively. Specifically, the adversarial loss can be expressed as:

$$
\begin{aligned}
L_{\mathrm{adv}}^{\mathrm{img}} &= \mathbb{E}_x[\log D_{\mathrm{img}}(x)] + \mathbb{E}_{\hat{x}}[\log(1 - D_{\mathrm{img}}(\hat{x}))], \\
L_{\mathrm{adv}}^{\mathrm{obj}} &= \mathbb{E}_p[\log D_{\mathrm{obj}}(p)] + \mathbb{E}_{\hat{p}}[\log(1 - D_{\mathrm{obj}}(\hat{p}))],
\end{aligned}
\tag{6}
$$

where x and p denote the real image and patch, whereas $\hat{x}$ and $\hat{p}$ represent the generated image and the patch crop from the generated image, respectively.

3.5   Training objective functions

In addition to the abovementioned loss functions, we use the following loss functions during the training phase:

Bounding box regression loss. We penalize the prediction of the bounding box coordinates with $\ell 1$ distance $L_{\mathrm{bbx}} = \sum_{i=1}^{n} \|b_i - \hat{b_i}\|_1$.

Image reconstruction loss. Given the ground-truth patches and the ground-truth bounding box coordinates, the image generation module should recover the ground-truth image. The loss $L_{\mathrm{recon}}^{\mathrm{img}}$ is an $\ell 1$ distance measuring the difference between the recovered and ground-truth images.

Auxiliary classification loss. We adopt the auxiliary classification loss $L_{\mathrm{ac}}^{\mathrm{obj}}$ to encourage the generated patches to be correctly classified by the object discriminator D_{obj}.

Perceptual loss. The perceptual loss is computed as the distance in the pre-trained VGG [27] feature space. We apply the perceptual losses $L_{\mathrm{p}}^{\mathrm{img}}, L_{\mathrm{p}}^{\mathrm{obj}}$ on both image and object levels to stabilize the training procedure.

The full loss functions for training our model is:

$$
\begin{aligned}
L = & \lambda_{\mathrm{gt}}^{\mathrm{sel}} L_{\mathrm{gt}}^{\mathrm{sel}} + \lambda_{\mathrm{occur}}^{\mathrm{sel}} L_{\mathrm{occur}}^{\mathrm{sel}} + \lambda_{\mathrm{adv}}^{\mathrm{img}} L_{\mathrm{adv}}^{\mathrm{img}} + \lambda_{\mathrm{recon}}^{\mathrm{img}} L_{\mathrm{recon}}^{\mathrm{img}} + \lambda_{\mathrm{p}}^{\mathrm{img}} L_{\mathrm{p}}^{\mathrm{img}} + \\
& \lambda_{\mathrm{adv}}^{\mathrm{obj}} L_{\mathrm{adv}}^{\mathrm{obj}} + \lambda_{\mathrm{ac}}^{\mathrm{obj}} L_{\mathrm{ac}}^{\mathrm{obj}} + \lambda_{\mathrm{p}}^{\mathrm{obj}} L_{\mathrm{p}}^{\mathrm{obj}} + \lambda_{\mathrm{bbx}} L_{\mathrm{bbx}},
\end{aligned}
\tag{7}
$$

where λ controls the importance of each loss term. We describe the implementation detail of the proposed approach in the supplementary document.

4   Experimental Results

Datasets. The COCO-Stuff [2] and Visual Genome [16] datasets are standard benchmark datasets for evaluating scene generation models [13,37,40]. We use the image resolution of 128×128 for all the experiments. Except for the image resolution, we follow the protocol in sg2im [13] to pre-process and split the dataset. Different from the PasteGAN [37] approach, we do not access the ground-truth mask for segmenting the image patches.

Evaluated methods. We compare the proposed approach to three parametric generation models and one semi-parametric model in the experiments:

- **sg2im** [13]: The sg2im framework takes as input a scene graph and learns to synthesize the corresponding image.
- **AttnGAN** [36]: As the AttnGAN method synthesizes the images from text, 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". We train the AttnGAN model on these converted sentences.

Table 1: **Quantitative comparisons.** We evaluate all methods on the COCO-Stuff and Visual Genome datasets using the FID, IS, and DS metrics. The first row shows the results of models that predict bounding boxes during the inference time. The second row shows the results of models that take ground-truth bounding as inputs during the inference time.

Datasets	COCO-Stuff			Visual Genome		
	FID $\downarrow$	IS $\uparrow$	DS $\uparrow$	FID $\downarrow$	IS $\uparrow$	DS $\uparrow$
sg2im [13]	136.8	$4.1_{\pm0.1}$	$0.02_{\pm0.0}$	126.9	$5.1_{\pm0.1}$	$0.11_{\pm0.1}$
AttnGAN [36]	72.8	$8.4_{\pm0.2}$	$0.14_{\pm0.1}$	114.6	$\mathbf{10.4}_{\pm0.2}$	$\underline{0.27}_{\pm0.2}$
PasteGAN [37]	$\underline{59.8}$	$8.8_{\pm0.3}$	$\mathbf{0.43}_{\pm0.1}$	$\underline{81.8}$	$6.7_{\pm0.2}$	$\mathbf{0.30}_{\pm0.1}$
RetrieveGAN (Ours)	$\mathbf{43.2}$	$\mathbf{10.6}_{\pm0.6}$	$\underline{0.34}_{\pm0.1}$	$\mathbf{70.3}$	$\underline{7.7}_{\pm0.1}$	$0.24_{\pm0.1}$
sg2im (GT)	79.9	$8.5_{\pm0.1}$	$0.02_{\pm0.0}$	111.9	$5.8_{\pm0.1}$	$0.13_{\pm0.1}$
layout2im [40]	$\underline{45.3}$	$\underline{10.2}_{\pm0.6}$	$\underline{0.29}_{\pm0.1}$	$\mathbf{44.0}$	$\mathbf{9.3}_{\pm0.4}$	$\mathbf{0.29}_{\pm0.1}$
PasteGAN (GT)	54.9	$9.6_{\pm0.2}$	$\mathbf{0.38}_{\pm0.1}$	68.1	$6.7_{\pm0.1}$	$\underline{0.28}_{\pm0.1}$
RetrieveGAN (GT)	$\mathbf{42.7}$	$\mathbf{10.7}_{\pm0.1}$	$0.21_{\pm0.1}$	$\underline{46.3}$	$\underline{9.1}_{\pm0.1}$	$0.23_{\pm0.1}$
Real data	6.8	$24.3_{\pm0.3}$	-	6.9	$24.1_{\pm0.4}$	-

Table 2: **Ablation studies.** We conduct ablation studies on two loss functions added upon the proposed retrieval module.

Datasets			COCO-Stuff		
	$\mathcal{L}_{\mathrm{gt}}^{\mathrm{sel}}$	$\mathcal{L}_{\mathrm{occur}}^{\mathrm{sel}}$	FID $\downarrow$	IS $\uparrow$	DS $\uparrow$
RetrieveGAN	-	-	56.8	$8.8_{\pm0.3}$	$0.30_{\pm0.1}$
RetrieveGAN	$\checkmark$	-	47.8	$9.7_{\pm0.2}$	$\mathbf{0.36}_{\pm0.1}$
RetrieveGAN	-	$\checkmark$	52.8	$9.8_{\pm0.2}$	$0.29_{\pm0.1}$
RetrieveGAN	$\checkmark$	$\checkmark$	$\mathbf{43.2}$	$\mathbf{10.6}_{\pm0.6}$	$0.34_{\pm0.1}$
Real data			6.8	$24.3_{\pm0.3}$	-

- **layout2im** [40]: The layout2im scheme takes as input the ground-truth bounding boxes to perform the generation. For a fair comparison, we use the ground-truth bounding box coordinate as the input data for other methods, which we denote GT in the experimental results.
- **PasteGAN** [37]: The PasteGAN approach is most related to our work as it uses the pre-trained embedding function to retrieve candidate patches.

Evaluation Metrics. We use the following metrics to measure the realism and diversity of the generated images:

- **Inception Score (IS).** Inception Score [26] uses the Inception V3 [28] model to measure the visual quality of the generated images.
- **Fréchet Inception Distance (FID).** Fréchet Inception Distance [9] measures the visual quality and diversity of the synthesized images. We use the Inception V3 model as the feature extractor.
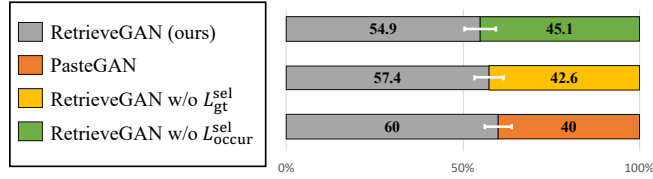
Fig. 3: **User study.** We conduct the user study to evaluate the mutual compatibility of the selected patches.

- **Diversity (DS).** We use the AlexNet model to explicitly evaluate the diversity by measuring the distances between the features of the images using the Learned Perceptual Image Patch Similarity (LPIPS) [39] metric.

4.1   Quantitative Evaluation

Realism and diversity. We evaluate the realism and diversity of all methods using the IS, FID, and DS metrics. 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 the first and second row of Table 1, respectively. Since the patch retrieval process is optimized to consider the generation quality during the training stage, our approach performs favorably against the other algorithms in terms of realism. On the other hand, as we can sample different query features for the proposed retrieval process, our model synthesizes comparably diverse images compared to the other schemes.

Moreover, there are two noteworthy observations. First, the proposed RetrieveGAN has similar performance in both settings on the COCO-Stuff dataset, but has significant improvement using ground-truth bounding boxes on the Visual Genome dataset. The reason for the inferior performance on the Visual Genome dataset without using ground-truth bounding boxes is due to the existence of lots of isolated objects (i.e., objects that have no relationships to other objects) in the scene graph annotation (e.g., the last scene graph in Figure 6), which greatly increase the difficult of predicting reasonable bounding boxes. Second, on the Visual Genome dataset, AttnGAN outperforms the proposed method on the IS and DS metrics, while performs significantly worse than the proposed method on the FID metric. Compared to the FID metric, the IS score has the limitation that it is less sensitive to the mode collapse problem. The DS metric only measures the feature distance without considering visual quality. The results from AttnGAN shown in Figure 4 also support our observation.

Patch compatibility. The proposed differentiable retrieval process aims to improve the mutual compatibility among the selected patches. We conduct a user study to evaluate the patch compatibility. For each scene graph, we present
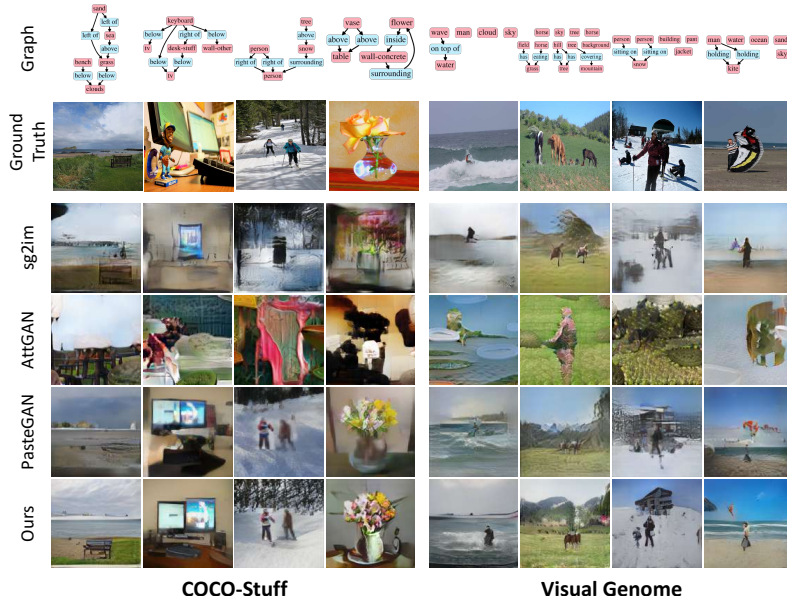
Fig. 4: **Sample generation results.** We show example results on the COCO-Stuff (*left*) and Visual Genome (*right*) datasets. The object locations in each image are predicted by models.

two sets of patches selected by different methods, and ask users "which set of patches are more mutually compatible and more likely to coexist in the same image?". Figure 3 presents the results of the user study. The proposed method outperforms PasteGAN, which uses a pre-defined patch embedding function for retrieval. The results also validate the benefits of the proposed ground-truth selection loss and co-occurrence loss.

Ablation study. We conduct an ablation study on the COCO-Stuff dataset to understand the impact of each component in the proposed design. The results are shown in Table 2. As the ground-truth selection loss and the co-occurrence penalty maximize the mutual compatibility of the selected patches, they both improve the visual quality of the generated images.

4.2 Qualitative Evaluation

Image generation. We qualitatively compare the visual results generated by different methods. We show the results on the COCO-Stuff (left column) and the Visual Genome (right column) datasets under two settings of using predicted (Figure 4) and ground-truth (Figure 5) bounding boxes. The sg2im and layout2im methods can roughly capture the appearance of objects and mutual relationships among objects. However, the quality of generated images in complicated scenes is limited. Similarly, the AttnGAN model cannot handle scenes
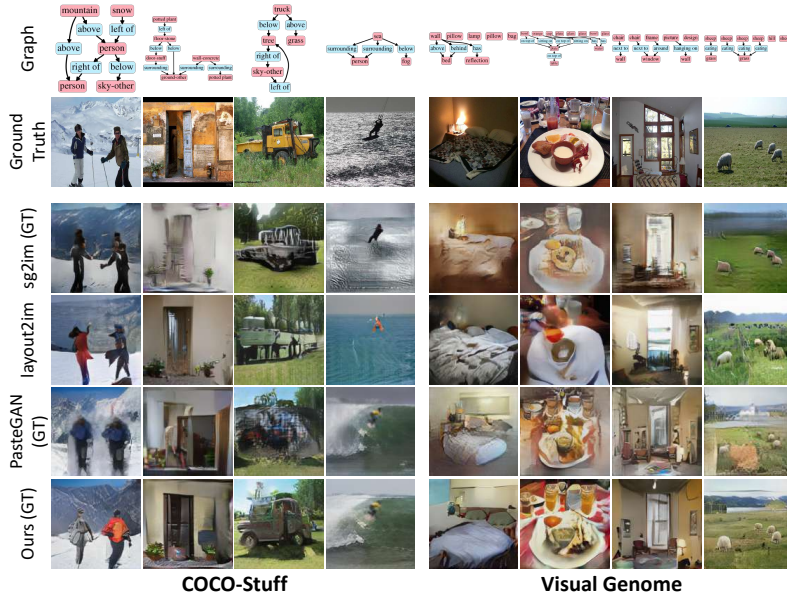
Fig. 5: **Sample generation results.** We show example results on the COCO-Stuff (*left*) and Visual Genome (*right*) datasets. The object locations in each image are given as additional inputs.

with complex relationships well. The overall image quality generated by the PasteGAN scheme is similar to that by the proposed approach, yet the quality is affected by the compatibility of the selected patches (e.g., the third result on COCO-Stuff in Figure 5).

Patch retrieval. To better visualize the source of retrieved patches, we present the generated images as well as the original images of selected patches in Figure 6. The proposed method can tackle complex scenes where multiple objects are present. With the help of selected patches, each object in the generated images has a clear and reasonable appearance (e.g., the boat in the second row and the food in the third row). Most importantly, the retrieved patches are mutually compatible, thanks to the proposed iterative and differentiable retrieval process. As shown in the first example in Figure 6, the selected patches are all related to baseball, while the PasteGAN method, which uses random selection, has chances to select irrelevant patches (i.e., the boy on the soccer court).

5   Conclusions and Future Work

In this work, we present a differentiable retrieval module to aid the image synthesis from the scene description. Qualitative and quantitative evaluations validate that the synthesized images are realistic and diverse, while the retrieved patches are reasonable and compatible. The proposed approach points out a new di-
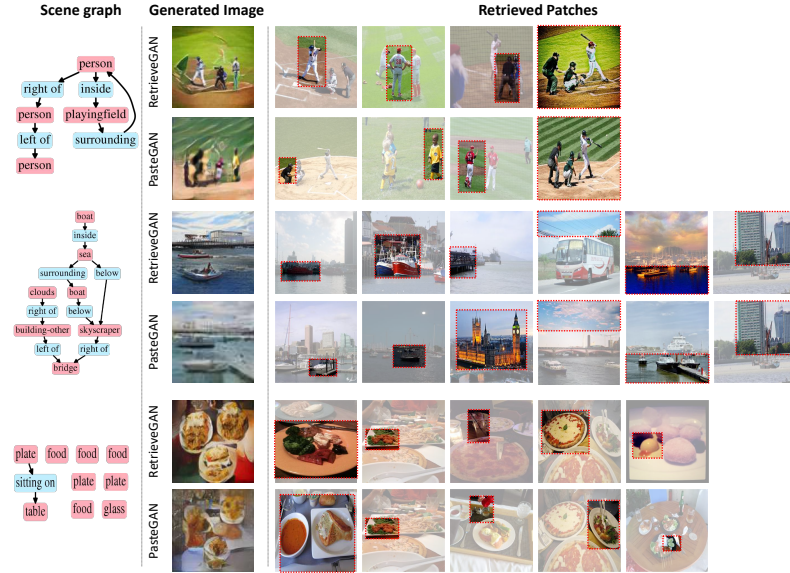
Fig. 6: **Retrieved patches.** For each sample, we show the retrieved patches which are used to guide the following image generation process. We also show the original image of each selected patch for more clear visualization.

rection in the content creation research field. It can be trained with the image generation or manipulation models to learn to select real reference patches that improves the generation or manipulation quality.

Acknowledgements

References

1. Ak, K.E., Kassim, A.A., Hwee Lim, J., Yew Tham, J.: Learning attribute representations with localization for flexible fashion search. In: CVPR (2018) 3
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018) 2, 5, 9
3. Chen, Y., Gong, S., Bazzani, L.: Image search with text feedback by visiolinguistic attention learning. In: CVPR (2020) 3
4. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: CVPR (2020) 3
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) 2, 3
6. Han, X., Wu, Z., Huang, P.X., Zhang, X., Zhu, M., Li, Y., Zhao, Y., Davis, L.S.: Automatic spatially-aware fashion concept discovery. In: ICCV (2017) 3

7. Hays, J., Efros, A.A.: Im2gps: estimating geographic information from a single image. In: CVPR (2008) 3
8. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017) 3
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017) 2, 10
10. Huang, H.P., Tseng, H.Y., Lee, H.Y., Huang, J.B.: Semantic view synthesis. In: ECCV (2020) 3
11. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: ICLR (2017) 2, 6
12. Jiang, L., Meng, D., Mitamura, T., Hauptmann, A.G.: Easy samples first: Self-paced reranking for zero-example multimedia search. In: ACM MM (2014) 3
13. Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: CVPR (2018) 2, 3, 5, 9, 10
14. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: CVPR (2015) 5
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014) 3
16. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV **123**(1), 32–73 (2017) 2, 9
17. Lee, H.Y., Tseng, H.Y., Mao, Q., Huang, J.B., Lu, Y.D., Singh, M.K., Yang, M.H.: Drit++: Diverse image-to-image translation viadisentangled representations. IJCV (2020) 3
18. Lee, H.Y., Yang, W., Jiang, L., Le, M., Essa, I., Gong, H., Yang, M.H.: Neural design network: Graphic layout generation with constraints. In: ECCV (2020) 2
19. Lee, H.Y., Yang, X., Liu, M.Y., Wang, T.C., Lu, Y.D., Yang, M.H., Kautz, J.: Dancing to music. In: NeurIPS (2019) 2
20. Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., Gao, J.: Object-driven text-to-image synthesis via adversarial training. In: CVPR (2019) 3
21. Li, Y., Jiang, L., Yang, M.H.: Controllable and progressive image extrapolation. arXiv preprint arXiv:1912.11711 (2019) 3
22. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: CVPR (2016) 3
23. Mao, Q., Lee, H.Y., Tseng, H.Y., Ma, S., Yang, M.H.: Mode seeking generative adversarial networks for diverse image synthesis. In: CVPR (2019) 3
24. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR (2019) 2, 3
25. Qi, X., Chen, Q., Jia, J., Koltun, V.: Semi-parametric image synthesis. In: CVPR (2018) 2, 3
26. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NeurIPS (2016) 2, 10
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2015) 9
28. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016) 10
29. Tan, F., Feng, S., Ordonez, V.: Text2scene: Generating compositional scenes from textual descriptions. In: CVPR (2019) 3, 4
30. Tseng, H.Y., Fisher, M., Lu, J., Li, Y., Kim, V., Yang, M.H.: Modeling artistic workflows for image generation and editing. In: ECCV (2020) 3

31. Vo, N., Hays, J.: Localizing and orienting street views using overhead imagery. In: ECCV (2016) 3
32. Vo, N., Jiang, L., Sun, C., Murphy, K., Li, L.J., Fei-Fei, L., Hays, J.: Composing text and image for image retrieval-an empirical odyssey. In: CVPR (2019) 3
33. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: CVPR (2018) 3
34. Wang, X., He, K., Gupta, A.: Transitive invariance for self-supervised visual representation learning. In: ICCV (2017) 7
35. Xie, S.M., Ermon, S.: Reparameterizable subset sampling via continuous relaxations. In: IJCAI (2019) 6
36. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: CVPR (2018) 2, 3, 9, 10
37. Yikang, L., Ma, T., Bai, Y., Duan, N., Wei, S., Wang, X.: Pastegan: A semi-parametric method to generate image from scene graph. In: NeurIPS (2019) 2, 3, 4, 5, 8, 9, 10
38. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV (2017) 3
39. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 2, 11
40. Zhao, B., Meng, L., Yin, W., Sigal, L.: Image generation from layout. In: CVPR (2019) 2, 9, 10
41. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: ICCV (2017) 3
42. Zhu, M., Pan, P., Chen, W., Yang, Y.: Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In: ICCV (2019) 3
43. Zhu, P., Abdal, R., Qin, Y., Wonka, P.: Sean: Image synthesis with semantic region-adaptive normalization. In: CVPR (2020) 3