R-GAN: Exploring Human-like Way for Reasonable Text-to-Image Synthesis via Generative Adversarial Networks

Yanyuan Qiao[1], Qi Chen[1], Chaorui Deng[1], Ning Ding[2], Yuankai Qi[1], Mingkui Tan[2], Xincheng Ren[3], Qi Wu[1]*
[1]University of Adelaide [2] South China University of Technology [3] Yanan University
{yanyuan.qiao,qi.chen04,chaorui.deng,qi.wu01}@adelaide.edu.au,seningding@mail.scut.edu.cn
qykshr@gmail.com,mingkuitan@scut.edu.cn,xchren@yau.edu.cn

ABSTRACT

Despite recent significant progress on generative models, context-rich text-to-image synthesis depicting multiple complex objects is still non-trivial. The main challenges lie in the ambiguous semantic of a complex description and the intricate scene of an image with various objects, different positional relationship and diverse appearances. To address these challenges, we propose R-GAN, which can generate reasonable images according to the given text in a human-like way. Specifically, just like humans will first find and settle the essential elements to create a simple sketch, we first capture a monolithic-structural text representation by building a scene graph to find the essential semantic elements. Then, based on this representation, we design a bounding box generator to estimate the layout with position and size of target objects, and a following shape generator, which draws a fine-detailed shape for each object. Different from previous work only generating coarse shapes blindly, we introduce a coarse-to-fine shape generator based on a shape knowledge base. At last, to finish the final image synthesis, we propose a multi-modal geometry-aware spatially-adaptive generator conditioned on the monolithic-structural text representation and the geometry-aware map of the shapes. Extensive experiments on the real-world dataset MSCOCO show the superiority of our method in terms of both quantitative and qualitative metrics.

CCS CONCEPTS

• **Computing methodologies** → *Image representations.*

KEYWORDS

Text-to-image Synthesis, Generative Adversarial Networks

ACM Reference Format:
Yanyuan Qiao, Qi Chen, Chaorui Deng, Ning Ding, Yuankai Qi, Mingkui Tan, Xincheng Ren, Qi Wu. 2021. R-GAN: Exploring Human-like Way for Reasonable Text-to-Image Synthesis via Generative Adversarial Networks. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475363

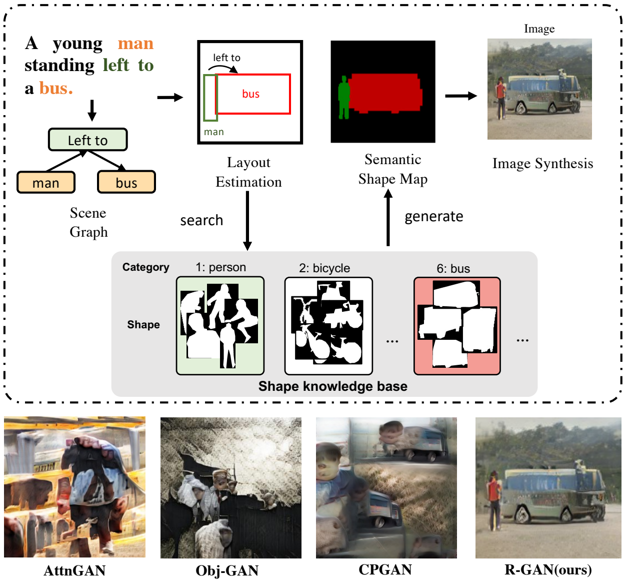


Figure 1: Our human-like way of reasonable text-to-image synthesis process.

1 INTRODUCTION

Generating images from text has aroused extensive research interest since it connects the research fields of computer vision and natural language processing. It requires capturing semantic information from a description and then generating a semantically aligned image according to the description. Recently, many researches [17, 19, 36, 39] seek to synthesise a photo-realistic image from textual description and have achieved promising performance. Since it is difficult to learn a direct mapping from text to image, some hierarchical methods [10, 17] have been proposed, which construct a semantic layout as intermediate representations to bridge the gap between text and image.

However, the success of these approaches still remains several limitations in the following aspects. First, not only the generated image should be realistic, but also its objects should match the semantic relationships of the text. When handling a complex description, most of existing methods [10, 39] just feed it into a simple LSTM model and then obtain a holistic text embedding, in which many important semantic information, *e.g.* , spatial and semantic relations between objects, has not been explicitly excavated. Second,

*Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10...$15.00
https://doi.org/10.1145/3474085.3475363

unlike simple datasets that involves only a single object such as birds [33] or flowers [24] that require learning one class of objects, the context-rich dataset MSCOCO [20] often contain more complex scenes and a wide variety of objects, *e.g.* , person, elephant, car. Due to lack of the knowledge of object shapes to guide image generation, many methods [10, 17] can only blindly generate coarse shapes. As a result, shapes of the objects generated in this way are often unrecognisable and unreasonable. Third, when generating target images, many existing methods [17] simply use an encoder-decoder structure, the semantic information of the generated shapes from previous steps may be washed away in the process of normalisation, which is crucial for generating meaningful and reasonable images.

To address the above mentioned issues, in this paper we propose R-GAN to synthesise a context-rich and reasonable image from the given intricate description in a human-like way. As shown in Figure 1, our approach imitates the drawing process of human that is usually hard to finish in one step, and decomposes the text-to-image generation into a multi-step task. Similar to what people usually do, given the text description "A young man standing left to a bus.", the drawing starts with a scene graph parser which finds the essential semantic elements('man', 'bus' and 'left to') in the text and a rough but reasonable layout as a simple sketch, taking into account the position and size of the man and bus in the image. Then, our approach learns the shapes of a standing man and the bus from the prior knowledge base, which are used to generate the semantic shape map as a preliminary drawing. When it is complete, we color the draft with reasonable details of the man, bus and background.

The R-GAN consists of four components: text representation, layout estimation, coarse-to-fine shape generation and image synthesis. We first build a monolithic-structural text representation to handle the complex description. The representation combines both the holistic feature from the whole sentence and the structural feature by building a scene graph from the text. In this way, it can capture the information of objects and their relationships well. Based on the text representation, we then predict a layout of objects using a bounding box generator. To address the second issue, we design a shape knowledge base and propose a coarse-to-fine shape generator to generate shapes. Unlike existing methods that often yield incomplete shapes, we generate a sharper and more complete shape via a shape gradient-sensitive loss. Last, we devise a multi-modal geometry-aware spatially-adaptive generator with a multi-scale discriminator to color the semantic shape map, which preserves the semantics by feeding multi-scale shapes into different generative layers. In addition, during the whole generation process, our human-like method keeps the textual information in mind at each stage, avoiding the deviation between the intermediate generation process and the original text, helping a lot to generate reasonable images. To evaluate the performance of R-GAN, we conduct experiments on the context-rich dataset MSCOCO [20] in both quantitative and qualitative metrics.

In summary, we make the following contributions in this work:

- We propose a novel Generative Adversarial Network called R-GAN to produce photo-realistic and reasonable images from the corresponding intricate descriptions by imitating the drawing process of human.
- To well capture the information from an intricate description, we design a monolithic-structural text representation, which seeks to sort out the objects and their relationships by using a scene graph. To generate a recognisable, complete and reasonable shape, we propose a coarse-to-fine shape generator based on shape knowledge base. To generate a fine-grained and reasonable image with semantic alignment, we devise a multi-modal spatially-adaptive geometry-aware generator conditioned on a geometry-aware map, which avoids common normalisation and can preserve the semantic information of the given shape.
- Extensive quantitative and qualitative evaluations are conducted on MSCOCO dataset, which demonstrates the effectiveness of our model compared with SOTA methods.

2 RELATED WORK

Text-to-Image Synthesis. There have been many studies for the text-to-image synthesis task, such as Variational Autoencoders (VAE) [16, 29] and Autoregressive models [32]. To generate sharper images, many GAN-based methods [14, 18, 26, 28, 36–39, 41] have been proposed. Reed *et al.* [28] first attempted to apply GAN to text-to-image synthesis. Zhang *et al.* [39] propose a hierarchical network, namely StackGAN, which generates images of different sizes. Based on StackGAN, Xu *et al.* [36] develop an attention mechanism, which ensures the alignment between generated fine-grained images and the corresponding word-level conditions. More recently, to preserve the semantic consistency, Qiao *et al.* [26] consider both text-to-image and image-to-text problems jointly. Hinz *et al.* [8] introduce an additional object pathway to the generator and the discriminator to control the location of objects within images. Liang *et al.* [19] propose CPGAN, which concentrates on content-oriented parsing on both the text descriptions and generated images to learn the consistency of text and image from semantic level.

To bridge the gap between text and image, Hong *et al.* [10] design a hierarchical approach that first constructs a semantic layout as an intermediate representation. The layout generator constructs a semantic label map from text and then convert the layout to an image. Following the generation process of [10], Li *et al.* [17] propose a multi-stage Object-driven Attentive Generative Adversarial Networks (Obj-GAN), which contains an object-driven attentive image generator and an object-wise discriminator. We follow a similar pipeline with [17] but focus more on details. To be specific, we design a monolithic-structural text representation to predict the reasonable layout of objects. Then a coarse-to-fine shape generator is devised to learn the knowledge of object to generate more realistic and reasonable shapes. In addition, a gradient-sensitive loss is utilized on shape generation process, which aims to generate a sharp, complete and recognisable shapes. Furthermore, different from all the above methods, we propose a semantic image synthesis method to the draft when we have the semantic shape maps.

Semantic Image Synthesis. Given a semantic layout, there are many approaches to generate photo-realistic images [2, 12, 21, 25, 35]. Isola *et al.* [12] propose Pix2pix, which is an image-to-image translation framework with image-conditional GANs. Following [12]'s framework, Wang *et al.* [35] propose Pix2pixHD, which contains a coarse-to-fine generator and a multi-scale discriminator to generate high-resolution images. Park *et al.* [25] propose SPADE, which uses semantic label mapping to predict affine transformation

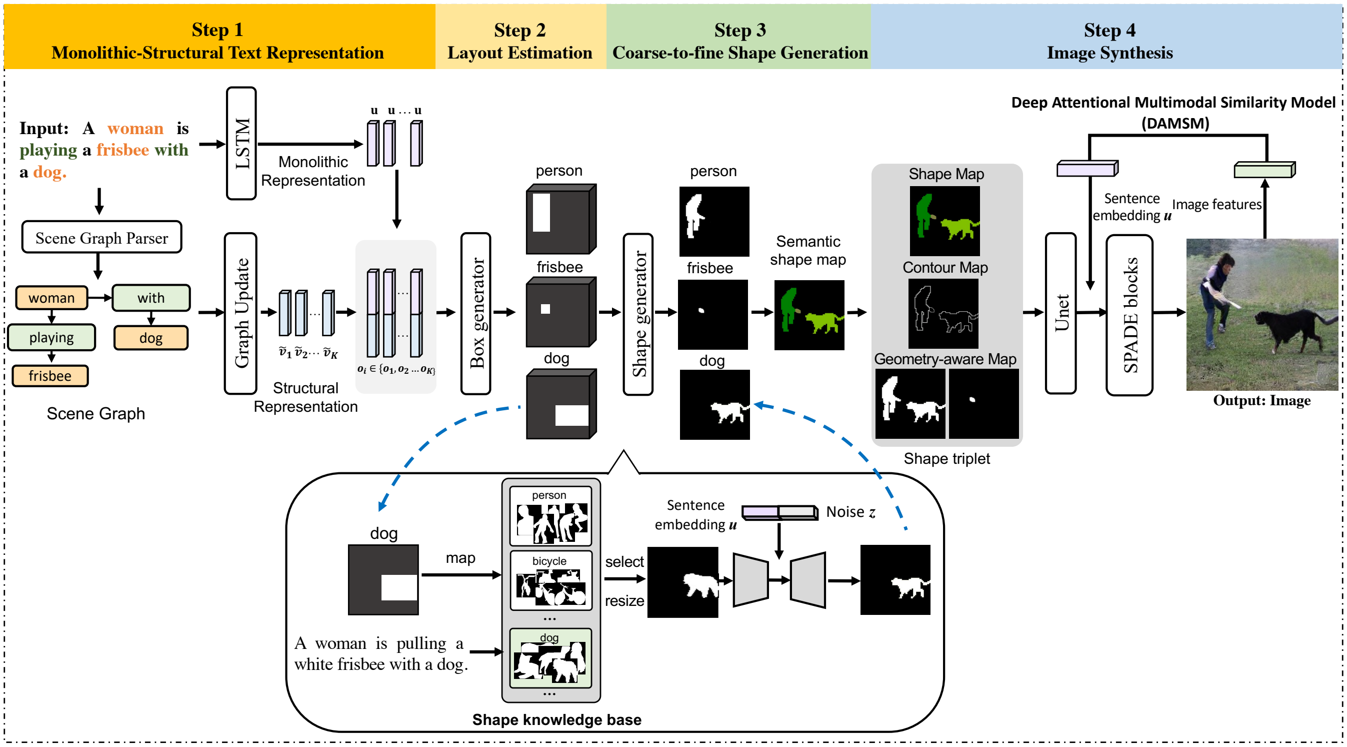

Step 1
Monolithic-Structural Text Representation
Step 2
Layout Estimation
Step 3
Coarse-to-fine Shape Generation
Step 4
Image Synthesis
Input: A woman is playing a frisbee with a dog.
LSTM
Monolithic Representation
u u … u
Scene Graph Parser
woman
with
playing
dog
frisbee
Scene Graph
Graph Update
$\widetilde{v}_1\, \widetilde{v}_2 \cdots \widetilde{v}_K$
Structural Representation
$o_i \in \{o_1, o_2 \ldots o_K\}$
Box generator
person
frisbee
dog
Shape generator
person
frisbee
dog
Semantic shape map
Shape Map
Contour Map
Geometry-aware Map
Shape triplet
Deep Attentional Multimodal Similarity Model (DAMSM)
Sentence embedding $\boldsymbol{u}$
Image features
Unet
SPADE blocks
Output: Image
dog
map
person
bicycle
dog
A woman is pulling a white frisbee with a dog.
select
resize
Sentence embedding $\boldsymbol{u}$
Noise z
Shape knowledge base

Figure 2: Main architecture of the proposed R-GAN, which is composed of four major parts:Monolithic-Structural Text Representation (Section 3.1), Layout Estimation (Section 3.2), Coarse-to-fine Shape Generation(Section 3.3) and Image Synthesis (Section 3.4).

parameters to modulate activation in the normalisation layer. In our work, based on the SPADE, we redesign the generator architecture. Specifically, we propose a multi-modal geometry-aware spatially-adaptive module conditioned on a multi-scale shape map to synthesise images that retain semantic information.

3 METHODS

As shown in Figure 2, the proposed a Generative Adversarial Network R-GAN consists of four components: 1) text representation, 2) layout estimation, 3) coarse-to-fine shape generation and 4) image synthesis. Specifically, to sort out the objects and their relationships from a complex description, we design a *monolithic-structural text representation*, which combines a holistic feature from the whole sentence and a structural feature by building a scene graph. As for drawing the sketch, we first generate a rough but reasonable layout containing both position and size for each object by using bounding boxes and then outline different shapes for different boxes according to their categories and semantic context. To generate a fine-detailed shape for each object, we devise a coarse-to-fine shape generator based on a pre-constructed shape knowledge base. Besides, we use a *shape gradient-sensitive loss* to generate sharper and more complete shapes, which focuses on the contour of shapes. Conditioned on such a multi-scale shape map, we propose a *multi-modal geometry-aware spatially-adaptive generator* to produce the target image, which can well preserve the semantic information of the given shape and can generate realistic and reasonable images. We will depict more details in the following sections.

3.1 Monolithic-Structural Text Representation

The monolithic-structural text representation consists of two components: structural representation of objects, and monolithic representation of sentences.

Structural Representation To capture the information of objects and their relationships, we parse the linguistic description into a scene graph [15]. We employ the Standford Scene Graph Parser [30] to convert an text description t to a scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which is a Directed Acyclic Graph (DAG). Here, $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K\}$ refers to a set of objects which have been mentioned in the text description t, and K is the number of objects. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is a set of directed edges, where $\mathcal{R}$ is a set of relationships among objects. Each edge $\mathbf{e}_{ij} \in \mathcal{E}$ can be defined as a triplet $\mathbf{e}_{ij} = (\mathbf{v}_i, \mathbf{r}_{ij}, \mathbf{v}_j)$, where $\mathbf{r}_{ij} \in \mathcal{R}$ is a relationship label from $\mathbf{v}_i$ to $\mathbf{v}_j$. After obtaining the graph, we use a learned embedding layer to convert each node and relationship of the graph to a dense vector. For the simplicity, we reuse notations of $\mathbf{v}$ and $\mathbf{r}$ to represent the node and relationship feature vectors in the following sections.

Inspired by [14], we process and update the objects and their relationships in graph $\mathcal{G}$ with three functions f_s, f_r and f_e. They take as input a triple of vectors $(\mathbf{v}_i, \mathbf{r}_{ij}, \mathbf{v}_j)$ of an edge, and output new vectors $\widetilde{\mathbf{v}}_i$ and $\widetilde{\mathbf{r}}_{ij}$ as embeddings for the objects and relationships, respectively. For the embedding of relationship, we simply obtain the output vector via $\widetilde{\mathbf{r}}_{ij} = f_r(\mathbf{e}_{ij}) = f_r(\mathbf{v}_i, \mathbf{r}_{ij}, \mathbf{v}_j)$. Updating the object vector is more complex since an object may contain multiple relationships. Hence, for each edge starting at $\mathbf{v}_i$, we use f_s to calculate a candidate vector and collect all these candidates in a set $\mathcal{V}_i^s$. Formally,

$$\mathcal{V}_i^s = \{f_s(\mathbf{e}_{ij}) \mid \mathbf{e}_{ij} = (\mathbf{v}_i, \widetilde{\mathbf{r}}_{ij}, \mathbf{v}_j) \in \mathcal{E}\}. \tag{1}$$

Similarly, we use f_e to compute a set of candidate vectors $\mathcal{V}_i^e$ for all the edges ending at $\mathbf{v}_i$. Mathematically,

$$\mathcal{V}_i^e = \{f_e(\mathbf{e}_{ji}) \mid \mathbf{e}_{ji} = (\mathbf{v}_j, \widetilde{\mathbf{r}}_{ji}, \mathbf{v}_i) \in \mathcal{E}\}. \tag{2}$$

We finally obtain the updated object vector $\widetilde{\mathbf{v}}_i = g(\mathcal{V}_i^s \cup \mathcal{V}_i^e)$, where $g(\cdot)$ denotes an element-wise average pooling function. For the functions f_s, f_r and f_e, we follow [14] to use a network which concatenates three input vectors and feeds them to a multilayer perceptron (MLP).

Monolithic-structural Representation In addition to the structural features, we also consider the semantics from the whole sentence since a global understanding of the given description is necessary. To this end, we use an LSTM [9] to extract the vector from the text description t via $\mathbf{u} = LSTM(t)$ as monolithic representation.

To combine the structural and monolithic information, we concatenate the global vector $\mathbf{u}$ with each object vector $\widetilde{\mathbf{v}}_i \in \{\widetilde{\mathbf{v}}_1, \widetilde{\mathbf{v}}_2 ... \widetilde{\mathbf{v}}_K\}$ and then get the target monolithic-structural representation $\mathbf{o}_i \in \{\mathbf{o}_1, \mathbf{o}_2 ... \mathbf{o}_K\}$, where $\mathbf{o}_i = [\widetilde{\mathbf{v}}_i; \mathbf{u}]$.

3.2 Layout Estimation

Based on the monolithic-structural representation $\mathbf{o}_i \in \{\mathbf{o}_1, \mathbf{o}_2 ... \mathbf{o}_K\}$, in this part, we aim to generate a coarse layout by predicting a labelled bounding box $B_i \in \{B_1, B_2, ... B_K\}$ for each object.

Specifically, we define each box $B_i = (\mathbf{b}_i, \mathbf{l}_i)$, where $\mathbf{b}_i = (b_{i,x}, b_{i,y}, b_{i,w}, b_{i,h}) \in \mathbb{R}^4$ denotes the coordinate of the predicted box while $\mathbf{l}_i$ denotes the predicted category for the i-th object by using an one-hot label. Inspired by [10], we use an LSTM model as the box generator to produce the bounding box incrementally. To generate the i-th labelled bounding box B_i, we first predict the label $\mathbf{l}_i$ for the i-th object and then predict the corresponding coordinate $\mathbf{b}_i$. Mathematically, we define the conditional probability as $p(B_i|\cdot) = p(\mathbf{b}_i, \mathbf{l}_i|\cdot) = p(\mathbf{l}_i|\cdot)p(\mathbf{b}_i|\mathbf{l}_i, \cdot)$.

To get probability $p(B_i|\cdot)$, we first use the LSTM-based generator to approximate the probability $p(\mathbf{l}_i|\cdot)$. For the i-th step of LSTM, based on the previously generated box $B_{1:i-1}$, we feed the i-th object vector $\mathbf{o}_i$ into the LSTM and output an embedding $\mathbf{e}_i$ followed by a Softmax function. Mathematically, we define $p(\mathbf{l}_i|\cdot)$ as $p(\mathbf{l}_i|B_{1:i-1}, \mathbf{o}_i) = \text{Softmax}(\mathbf{e}_i)$.

Based on the predicted label $\mathbf{l}_i$, we further approximate the probability $p(\mathbf{b}_i|\mathbf{l}_i, \cdot)$ by a Gaussian Mixture Model (GMM) with multiple normal distributions as in [5]. Formally, the function is defined as

$$p(\mathbf{b}_i|\mathbf{l}_i, B_{1:i-1}, \mathbf{o}_i) = \sum_{n=1}^{N} \pi_{i,n} \mathcal{N}\left(\mathbf{b}_i; \mu_{i,n}, \Sigma_{i,n}\right), \tag{3}$$

where $\mathcal{N}\left(\mathbf{b}_i|\mu_{i,n}, \Sigma_{i,n}\right)$ denotes the n-th normal distribution at i-th step, and they are totally have N normal distributions at each step. The parameters $\pi_{i,n}$, $\mu_{i,n}$ and $\Sigma_{i,n}$ are also generated by the above LSTM.

Loss Function To optimise the box generator model, we use a negative log-likelihood loss

$$\mathcal{L}_{box} = -\frac{1}{K}\sum_{i=1}^{K} \mathbf{l}_i^* \log p(\mathbf{l}_i) - \lambda \frac{1}{K}\sum_{i=1}^{K} \log p(\mathbf{b}_i^*), \tag{4}$$

where K refers to the number of objects in an image while λ is a hyper-parameter to balance these two terms, we set $\lambda = 0.25$. $\mathbf{b}_i^*$ and $\mathbf{l}_i^*$ denote the coordinates of the ground-truth bounding box and the corresponding ground-truth label for the i-th object.

3.3 Coarse-to-Fine Shape Generation

This part consists of three steps: shape knowledge base construction, text-relevant shape selection, and shape editing.

Shape Knowledge Base Construction We collect shapes from MSCOCO instance-wise annotations, which consists of 597,701 shape images from 80 categories (*e.g.* , person, bicycle, car, *etc.*). Then, on accounting of performance and computation efficiency, we use ResNet50 [6] to extract shape feature for each shape image. Next, we use K-means [22] method to cluster the extracted features of each category into 10 groups, and each group has 3 shape images which are closest to the cluster center. This gives us a shape knowledge base that we can choose relevant shapes corresponding to the objects mentioned in the textual description.

Text-relevant Shape Selection To select a text-relevant shape, we utilize CLIP [27], a model pre-trained on large-scale image-text description pairs, which learns a multi-modal embedding space that can be used to calculate the semantic similarity between the text description and the image.

Specifically, we first use the object category obtained from the previous stage (see Section 3.2) to find the corresponding shape candidates $\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n\}$.

Then, we use the CLIP image encoder to encode shape candidates $\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n\}$ to shape features $\{f(\mathbf{s}_1), f(\mathbf{s}_2), ... f(\mathbf{s}_n)\}$, and use CLIP text encoder to encode text description t to text feature $g(\mathbf{t})$. The similarity score between the text description $g(\mathbf{t})$ and shape candidates is computed via

$$Sim_n(\mathbf{s}_n, \mathbf{t}) = \text{Softmax}(g(\mathbf{t}) \cdot f(\mathbf{s_n})^T). \tag{5}$$

We select the maximum similarity shape as input to the shape generator.

Shape Editor Shape generator G_{shape} takes both the selected shape and text information as inputs. The goal of G_{shape} is to generate the shape according to text and selected shape. We first encode the selected shape by down-sampling layers and then use bi-directional Convolutional LSTM to capture the feature from it. After concatenating the feature with i-th random noise $\mathbf{z}_i$ and the monolithic representation $\mathbf{u}$, the concatenated feature is fed into several residual blocks, and mapped to a binary shape $\mathbf{M}_i \in \mathbb{R}^{H \times W}$. Given K selected shape tensors $\mathrm{S}_{1:K} = \{\mathrm{S}_1, \mathrm{S}_2, ... \mathrm{S}_K\}$, we define the process of shapes edition as

$$\mathbf{M}_{1:K} = G_{shape}\left(\mathrm{S}_{1:K}, \mathbf{z}_{1:K}, \mathbf{u}\right), \tag{6}$$

where $\mathbf{z}_{1:K} = \{\mathbf{z}_1, \mathbf{z}_2 ... \mathbf{z}_K\} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a series of random noise vectors, $\mathbf{u}$ refers to the monolithic text representations (See Section 3.1), $\mathbf{M}_{1:K}$ denotes the generated shapes.

Loss Function To encourage the generator to generate better shapes, we use two discriminators at both object and global (*i.e.* , aggregated image) levels respectively: object discriminator D_{object} and global discriminator D_{global}. Here, we simply use the CNN model as the encoder in both object and global discriminators. The encoder takes a concatenation of box and shape as input and outputs the probability of the shape being real.

For the i-th object, we optimise the discriminator D_{object} via minimising $\mathcal{L}^{(i)}_{D_{object}} = -\log D_{object}(\mathbf{B}^*_i, \mathbf{M}^*_i) - \log(1 - D_{object}(\mathbf{B}^*_i, \mathbf{M}_i))$, where $\mathbf{B}^*_i$ is the ground-truth box tensor for i-th object. And $\mathbf{M}_i$ and $\mathbf{M}^*_i$ denote the generated shape and ground-truth shape of the i-th object, respectively. In this sense, the loss $\mathcal{L}_{D_{object}}$ can be defined as $\mathcal{L}_{D_{object}} = \frac{1}{K}\sum_{i=1}^{K}\mathcal{L}^{(i)}_{D_{object}}$. Thus, the corresponding generator loss can be defined as

$$\mathcal{L}^{(i)}_{object} = -\log(D_{object}(\mathbf{B}^*_i, \mathbf{M}_i)). \tag{7}$$

Similarly, we define the loss $\mathcal{L}_{object}$ for generator G_{shape} as $\mathcal{L}_{object} = \frac{1}{K}\sum_{i=1}^{K}\mathcal{L}^{(i)}_{object}$. Similar to the object discriminator, we optimise the global discriminator D_{global} by minimising

$$\begin{aligned}\mathcal{L}_{D_{global}} = &-\log D_{global}(\mathbf{B}^*_{global}, \mathbf{M}^*_{global}) \\ &-\log(1 - D_{global}(\mathbf{B}^*_{global}, \mathbf{M}_{global})),\end{aligned} \tag{8}$$

where $\mathbf{M}^*_{global}$ and $\mathbf{M}_{global}$ denote the ground-truth and generated shapes, respectively, which are aggregated by element-wise addition over $\mathbf{M}^*_{1:K}$ and $\mathbf{M}_{1:K}$. The $\mathbf{B}^*_{global}$ is an integrated ground-truth box tensor obtained by taking element-wise maximum over $\mathbf{B}^*_{1:K}$. Similarly, we can optimise the generator G_{shape} by

$$\mathcal{L}_{global} = -\log(D_{global}(\mathbf{B}^*_{global}, \mathbf{M}_{global}). \tag{9}$$

To ensure the generated shapes can semantically align with the ground-truth ones, we adopt a widely used reconstruction loss [3, 13, 34] $\mathcal{L}_{rec} = \|\Phi(\mathbf{M}_{global}) - \Phi(\mathbf{M}^*_{global})\|_2$, where $\Phi(\cdot)$ refers to the pretrained VGG-19 model [31], which is used to extract the feature from the given shape image.

To sharpen the shape for each object, inspired by [23], we design a shape gradient-sensitive loss, which pays attention to the outline of shape. The loss between generated shape $\mathbf{M}_{global}$ and ground-truth shape $\mathbf{M}^*_{global}$ can be defined as

$$\mathcal{L}_{grad} = \|\nabla_x\mathbf{M}_{global} - \nabla_x\mathbf{M}^*_{global}\|_2 + \|\nabla_y\mathbf{M}_{global} - \nabla_y\mathbf{M}^*_{global}\|_2, \tag{10}$$

where $\nabla_x\mathbf{M}_{global}$ ($\nabla_x\mathbf{M}^*_{global}$) and $\nabla_y\mathbf{M}_{global}$ ($\nabla_y\mathbf{M}^*_{global}$) refer to the directional gradients of $\mathbf{M}_{global}$ ($\mathbf{M}^*_{global}$) along the horizontal (denoted by x) and vertical (denoted by y) directions, respectively.

Finally, we define the objective loss function as

$$\mathcal{L}_{G_{shape}} = \lambda_1\mathcal{L}_{object} + \lambda_2\mathcal{L}_{global} + \lambda_3\mathcal{L}_{rec} + \lambda_4\mathcal{L}_{grad}, \tag{11}$$

We set $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 10, \lambda_4 = 1$.

Semantic Shape Map Generation Previous works use a 3-D (H, W, L) mask tensor M with binary value (0,1) to represent the shape, where H represents height, W represents width and L represents all categories of objects. Different from these methods, R-GAN generates a semantic shape map. To be specific, we transform the 3-D mask tensor to a 2-D (H, W) map tensor m with a set of integer values L where each integer value represents a category of objects. In other words, each pixel in the map m will be allocated with an integer value according to their label categories. For example, as shown in Figure 3, the pixel value of “clock” is 85, and the pixel value of “cup” is 47.

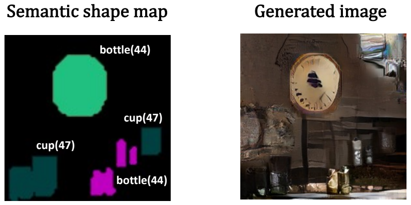


Figure 3: Illustration of semantic shape map, of which labels and corresponding pixel values are displayed.

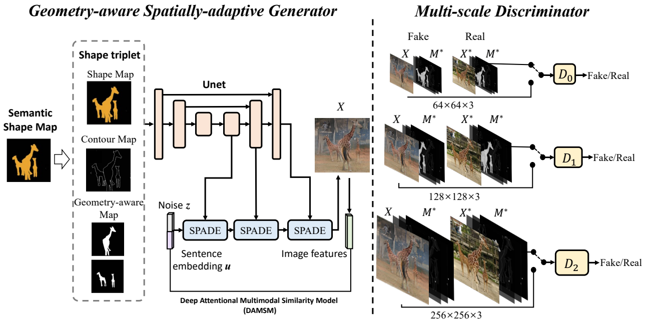


Figure 4: Overview of image synthesis. The input shape M contains shape map, contour map and geometry-aware map. We first feed the shape into a Unet and obtain a series of features from different layers. By combining the features with noise z and monolithic text representation u, we produce a target image via a geometry-aware spatially-adaptive generator. To enable the generator to yield a photo-realistic image, we propose a multi-scale discriminator to estimate images in different resolutions.

3.4 Image Synthesis

In this part, we seek to synthesise a real-world image from the given semantic shape map and the textual context. To this end, we first build a shape triplet, and then we extract a multi-scale shape embedding. Based on the shape embedding and monolithic text representation, we propose a multi-modal geometry-aware spatially-adaptive generator to produce the target image semantically aligned with them.

Shape Information Extraction Before generating images, we need to extract the semantic information and build a shape triplet from the given shapes. The shape triplet consists of a shape map, a contour map and a geometry-aware map. The shape map contains shape and category information of the object, the contour map contains the contour information of the object, and the geometry-aware map contains geometry information.

In practice, even in the same category, the appearance of different instances may still vary dramatically due to their various geometry properties such as the object scale. As shown in Figure 4, the large giraffe in the middle has an obviously different texture from the two small giraffes around it. Therefore, images should be generated differently for objects of different scales.

To capture this geometry information, we design a geometry-aware map $\widetilde{\mathbf{M}}_{1:T}$, which divides the object shapes into different scales. Specifically, the scale of an object is defined as the long side of its bounding box. Suppose we have T scale ranges, for objects that fall into the t-th ($0 < t < T$) scale range, all the pixels inside those objects will be assigned with the value 1 and the background pixel values are set to zero in $\widetilde{\mathbf{M}}_t$. We set T=3, which is the number of scale groups [32, 64), [64,128), [128, 256].

Unlike the previous methods that compress the shape information into a normalised single-scale embedding, we instead learn multi-scale embeddings (see Figure 4) through a Unet module:

$$\mathbf{s}_{0:N-1} = \text{Unet}(M), \tag{12}$$

where $\mathbf{s}_{0:N-1}$ refer to N shape embeddings from different layers of a Unet while $M = (\mathbf{M}_{global}, \nabla_{xy}\mathbf{M}_{global}, \widetilde{\mathbf{M}}_{1:T})$ denotes a shape triplet. Here, $\mathbf{M}_{global}$ denotes the shape map generated from the Section 3.3. $\nabla_{xy}\mathbf{M}_{global}$ and $\widetilde{\mathbf{M}}_{1:T}$ are the contour map and the geometry-aware map, respectively. Note that $\nabla_{xy}\mathbf{M}_{global}$ is an aggregated shape gradient combining by $\nabla_{x}\mathbf{M}_{global}$ and $\nabla_{y}\mathbf{M}_{global}$. In practice, we concatenate $\mathbf{M}_{global}$, $\nabla_{xy}\mathbf{M}_{global}$ and $\widetilde{\mathbf{M}}_{1:T}$ together as the input of a Unet model.

Multi-modal Geometry-aware Spatially-adaptive Generator To generate the target image, we propose a multi-modal geometry-aware spatially-adaptive generator G_{image} (see Figure 4). The generator takes a random noise $\mathbf{z}$, the monopolistic text representation $\mathbf{u}$[1] (extracted in Section 3.1) and the shape embeddings $\mathbf{s}_{0:N-1}$ as inputs, and outputs a target image semantically and spatially aligned with them. Formally, we define the process as

$$X = G_{image}(\mathbf{s}_{0:N-1}, \mathbf{z}, \mathbf{u}). \tag{13}$$

where X refers to the generated image, and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The generator G_{image} consists of several ResNet blocks [6] with upsampling layers. To effectively propagate the semantic information throughout the network, rather than using the traditional Batch Normalisation (BN) [11], we adopt the famous SPADE [25] for all the normalisation layers inside the ResNet blocks. Specifically, for the i-th SPADE, we first compute two spatially-adaptive affine modulation parameters $\boldsymbol{\gamma}_i \in \mathbb{R}^{h\times w\times c}$ and $\boldsymbol{\beta}_i \in \mathbb{R}^{h\times w\times c}$ from $\mathbf{s}_i$ by two single-layer convolutional networks:

$$\boldsymbol{\gamma}_i = \text{Conv}_{\gamma}(\mathbf{s}_i) \text{ and } \boldsymbol{\beta}_i = \text{Conv}_{\beta}(\mathbf{s}_i). \tag{14}$$

Then, we transform the i-th intermediate feature[2] $\mathbf{x}_i \in \mathbb{R}^{h\times w\times c}$ from the previous layers by

$$\mathbf{x}_i \longleftarrow \text{Norm}(\mathbf{x}_i) \times \boldsymbol{\gamma}_i + \boldsymbol{\beta}_i, \tag{15}$$

where Norm($\cdot$) normalises $\mathbf{x}_i$ using batch statistics like BN while h, w and c denote height, width and depth of the intermediate feature $\mathbf{x}_i$, respectively. Note that, SPADE only normalises $\mathbf{x}_i$ so that the semantic information from $\mathbf{s}_i$ is still preserved.

Multi-scale Discriminator To ensure the quality of generated objects in different scales and sizes, we propose a multi-scale discriminator, which considers both large and small objects jointly. We resize the ground-truth image X^* and generated image X with N different scales, *i.e.* $X^* = \{X_i^*\}_{i=0}^{N-1}$ and $X = \{X_i\}_{i=0}^{N-1}$. For the ground-truth shape triplet M^*, we also resize it as the same way, *i.e.* , $M^* = \{M_i^*\}_{i=0}^{N-1}$. Our multi-scale discriminator D_{image} (see Figure 4) consists of several sub-discriminators $D_{image} = \{D_i\}_{i=0}^{N-1}$, which are actually N CNN encoders with the different number of convolutional layers. We feed a concatenation of ground-truth shape triplet M_i^* and real image X_i^* (or fake image X_i) into the i-th discriminator D_i, and the discriminator outputs a probability to indicate whether the input image is real or not.

Loss Function To optimise discriminators $D_{image} = \{D_i\}_{i=0}^{N-1}$ and generator G_{image}, we use an adversarial loss, which can be defined in two parts. First, we optimise each discriminator D_i by

$$\mathcal{L}_{D_i} = -\log(D_i(M_i^*, X_i^*)) - \log(1 - D_i(M_i^*, X_i)), \tag{16}$$

where X_i^* and X_i denote the ground-true and generated images in the i-th scale, respectively. In this way, the total loss for $D_{image} = \{D_i\}_{i=0}^{N-1}$ can be defined as $\mathcal{L}_{D_{image}} = \frac{1}{N}\sum_{i=0}^{N-1}\mathcal{L}_{D_i}$. Then, we optimise the generator G_{image} by

$$\mathcal{L}_{G_{image}} = -\frac{1}{N}\sum_{i=0}^{N-1}\log(D_i(M_i^*, X_i)). \tag{17}$$

To make generated images better conditioned on text descriptions, we utilize the deep multi-modal attentive similarity model (DAMSM) [36] loss which measures the matching degree between images and text descriptions.

Finally, the complete loss is

$$\mathcal{L}_G = \mathcal{L}_{G_{image}} + \lambda_{DAMSM}\mathcal{L}_{DAMSM}. \tag{18}$$

Following previous work, we set $\lambda_{DAMSM} = 50$.

4 EXPERIMENTS

4.1 Datasets and Implementation Details

For context-rich text-to-image generation, we conduct all the experiments on the MSCOCO [20] dataset, where the images contain intricate scenes with linguistic descriptions. We use the official train and validation splits, *i.e.* , the training set contains 80k images while the validation set contains 40k images from 80 object categories.

We generate images with size 256 × 256 and optimise the R-GAN model step-by-step. For training, we optimise the box generator with 1 batch size for 10 epochs, trained on the provided objects bounding boxes in COCO. As for the shape generator, we train the model with 16 batch size and 30 epochs, with the objects segmentation mask. As for image generator (*i.e.* , multi-modal geometry-aware spatially-adaptive generator), we set batch size = 32 and epoch = 100. Note that we use the same annotation (including bounding boxes and segmentation masks) as Obj-GAN [17], thus the comparison with Obj-GAN is fair.

4.2 Evaluation Metrics

We evaluate the image quality by commonly used Fréchet Inception Distance (FID) [7] while test the consistency between generated image and text description via R-precision [36]. As noticed by [17], the IS can be saturated, even over-fitted and thus fails to evaluate the semantic layout of the generated images. We thus propose a Patch Inception Score (PIS). Specifically, rather than using the Inception network, we use a DeepLab-V2 [1] model trained on MSCOCO to ensure the alignment of classes during training and testing. The DeepLab-V2 model generates the feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, where

[1] Here we use the monopolistic representation only since the final image generation stage relies more on the global context.

[2] We generate the 0-th intermediate feature $\mathbf{x}_0$ using a single-layer convolutional network from a concatenation of $\mathbf{z}$ and $\mathbf{u}$.

Table 1: Comparison against State-of-the-art Method. Methods marked with 0, 1, 2 respectively generate images using the predicted boxes & predicted shapes, the ground-truth boxes & predicted shapes, and the ground-truth boxes & ground-truth shapes. ↑ (↓) means that the higher (lower) value is better. The results of methods marked with † were calculated with a pre-trained model provided by the authors.

Methods	PIS ↑	FID ↓	R -precision↑	#Parameters
StackGAN-V2 [38]	–	81.59	–	466M
StackGAN [39]	–	74.05	–	996M
HDGAN [40]	–	71.27	–	–
AttnGAN [36]†	41.28 ± 0.39	31.90	82.98	956M
DM-GAN [42]	–	32.64	88.56	223M
OPGAN [8]†	43.71 ± 0.74	26.55	87.90	1019M
CPGAN† [19]	44.30 ± 0.74	53.84	93.59	318M
Obj-GAN [17]^0 †	34.70 ± 0.38	32.04	91.05	–
Obj-GAN [17]^1 †	37.23 ± 0.38	30.25	92.54	–
Obj-GAN [17]^2 †	39.99 ± 0.44	25.01	93.39	–
R-GAN^0 (Ours)	**45.16 ± 0.34**	**24.60**	**94.08**	171M
R-GAN^1 (Ours)	47.39 ±0.57	22.64	94.67	–
R-GAN^2 (Ours)	49.70± 0.65	17.57	95.25	–

Table 2: The results of human study.

	Obj-GAN	CPGAN	R-GAN (Ours)
Choice(%)	19.56	26.11	**54.33**

Table 3: Performance of layout and shape.

Methods	Accuracy(%)	mIoU (%)
Obj-GAN	43.67	62.10
R-GAN (Ours)	**48.99**	**67.92**

Table 4: Effect of structural representation. ↑ (↓) means that the higher (lower) value is better.

R-GAN (Ours)	PIS↑	FID ↓	R -precision↑
w/o structural representation	40.06 ± 0.23	28.91	92.43
w structural representation	**45.16 ± 0.34**	**24.60**	**94.08**

each super-pixel $\mathbf{x}_{ij} \in \mathbb{R}^C$ indicates a patch of image, which can be regarded as a simple image with a single object. Here, C denotes the number of classes while H and W are the height and width of the feature map, respectively. Formally,

$$\mathrm{PIS}(\mathcal{X}) = \exp\left(\frac{1}{NHW}\sum_{k=1}^{N}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathrm{KL}\big(p(y|\mathbf{x}_{ij}^{(k)})\,||\,p(y)\big)\right), \tag{19}$$

where $\mathcal{X} = \{\mathrm{X}_k\}_{k=1}^N$ is a set of feature maps (images), KL denotes Kullback–Leibler divergence and N is the number of feature maps. To avoid the misleading from unsuitable marginal distribution $p(y)$, we simply set $p(y)$ as a Uniform distribution with C dimensions.

4.3 Quantitative Evaluation

Comparison against State-of-the-art Methods To prove the superiority of our method, we consider several state-of-the-art methods, such as OPGAN [8], Obj-GAN [17], CPGAN [19].

From Table 1, we can see compared to SoTA methods (*e.g.* , OPGAN and CPGAN), our model has only about $\frac{1}{6}$ parameters of OPGAN, $\frac{1}{2}$ parameters of CPGAN. Though our R-GAN has the minimum parameters, it achieves the best performance compared to other state-of-the-art methods in all metrics. Especially compared with Obj-GAN, which has a similar multi-step pipeline, our model achieves a large improvement by 30.14% in PIS and 23.22% in FID metircs.

Table 5: Impact of different losses in shape generation.

$\mathcal{L}_{global}$	$\mathcal{L}_{object}$	$\mathcal{L}_{rec}$	$\mathcal{L}_{grad}$	mIoU
√				19.63
√	√			50.24
√	√	√		64.32
√	√	√	√	**67.92**

Table 6: Influence of geometry-aware map when generating images from ground-truth boxes and shapes.

R-GAN (Ours)	PIS↑	FID↓
w/o geometry-aware map	46.08 ± 0.49	23.45
w geometry-aware map	**49.70 ± 0.65**	**17.57**

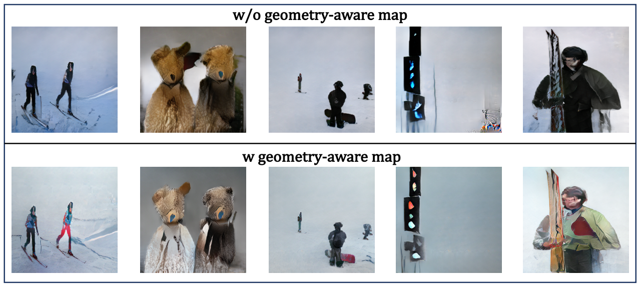


Figure 5: Visual comparison of the generated images with and without geometry-aware map.

We also randomly sampled 600 descriptions and corresponding generated images from both R-GAN, Obj-GAN and CPGAN, to ask human subjects to choose which one is more reasonable and more relevant to the given text. Table 2 shows R-GAN achieves the highest score of 54.33%, which outperforms both CP-GAN and Obj-GAN with a large margin.

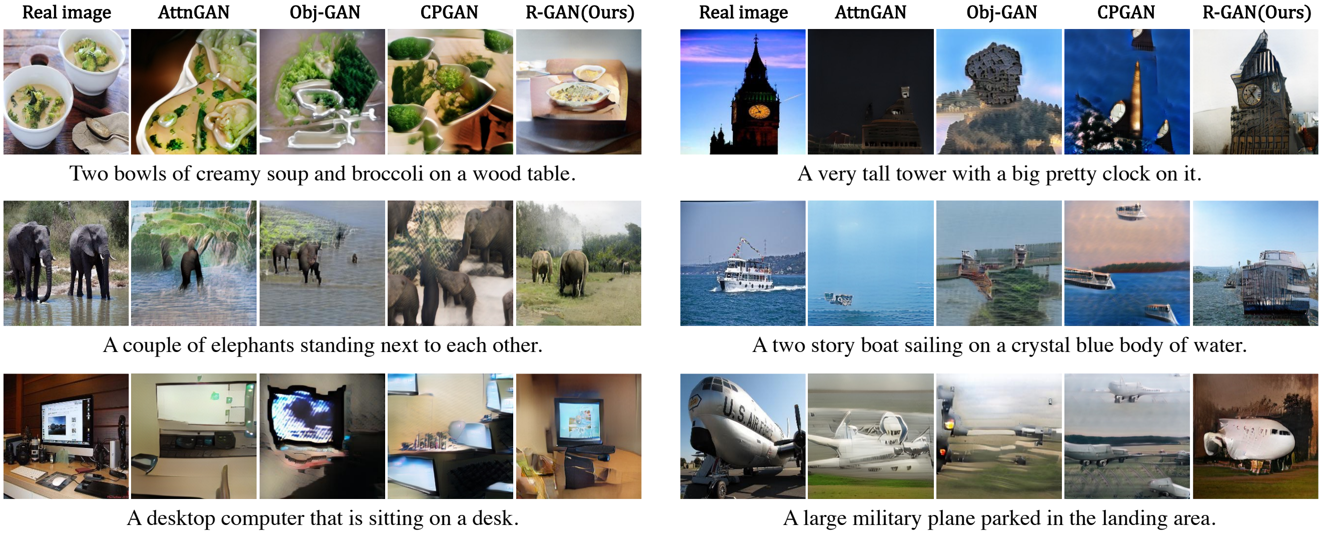
Comparison of Layout and Shape To measure the intermediate results, *i.e.* , layouts and shapes, we compute the accuracy for the predicted labels of generated bounding boxes (*i.e.* , object) within layouts. We then adopt the widely-used mean Intersection-over-Union (mIoU) [4] (predicted shape against ground truth) to evaluate shape generator. From Table 3, compared to Obj-GAN, our method achieves higher accuracy on both label accuracy and mIoU, which shows the superiority of our layout and shape generator.

4.4 Ablation Study

In this section, we conduct several ablation studies in terms of 1) our monolithic-structural text representation, 2) losses in shape generation and 3) geometry-aware map in image generation.

Impact of Structural Representation To test the effect of structural representation in Section 3.1, we conduct an ablation study to compare the results with or without it. Table 4 shows that the model with structural representation performs better on all metrics with large margins.

Impact of Losses in Shape Generation Table 5 shows both of the global loss $\mathcal{L}_{global}$ and object loss $\mathcal{L}_{object}$ are important, demonstrating that the shape discriminator should be considered at both global and object levels. We also find that the reconstruction loss



Two bowls of creamy soup and broccoli on a wood table.

A very tall tower with a big pretty clock on it.

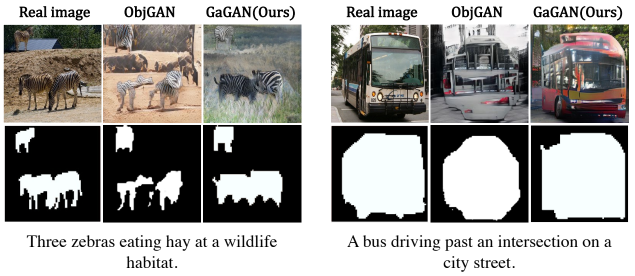
A couple of elephants standing next to each other.

A two story boat sailing on a crystal blue body of water.

A desktop computer that is sitting on a desk.

A large military plane parked in the landing area.

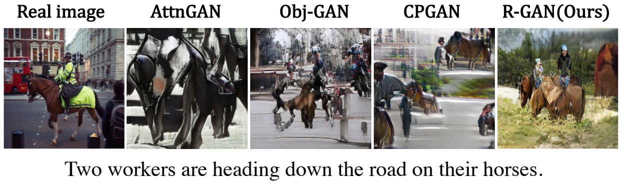
Figure 6: Comparisons of generated images with other methods.



Three zebras eating hay at a wildlife habitat.

A bus driving past an intersection on a city street.

Figure 7: Comparisons between Obj-GAN and our method in shape generation. Results are generated based on the ground-truth box & predicted shape.



Two workers are heading down the road on their horses.

Figure 8: Challenging examples

$\mathcal{L}_{rec}$ further improves the performance, with a gain of 14.08%. Adding the shape gradient-sensitive loss $\mathcal{L}_{grad}$ contributes to the performance boost with 3.60%, shows the effect of sharping shape.

Impact of Geometry-aware Map From Table 6, we can see that our method conditioned on geometry-aware map consistently outperforms the counterpart without this map. Besides, as shown in Figure 5, based on the geometry-aware map, the appearances of the generated objects are more diverse even they belong to the same category.

4.5 Qualitative Analysis

Figure 6 shows our generation results compared to other methods. It is obvious that our method is able to generate more photo-realistic and reasonable images.

In order to compare with Obj-GAN [17] that has a similar multi-step pipeline, we also visualise the generated shapes compared to the Obj-GAN in Figure 7. We can see that shapes from our R-GAN are more natural and complete with sharper contours. It helps our R-GAN generate more complete and recognisable images, while images generated by Obj-GAN are often incomplete and broken.

Specifically, we compare our method with CPGAN, which has a competitive performance on quantitative evaluation metrics. We can see that images generated by CPGAN tend to have the pattern which has a similar background with ground-truth images and repeated objects. This pattern makes up a messy and unreasonable image. On the contrary, R-GAN can generate images with a boat on water rather than in the sky, a desktop computer on a desk rather than a room filled with computer screens.

As shown Figure 8, in a very complicated scene, R-GAN is the only one that can generate a complete and reasonable image which contains all the semantic elements in the given description.

5 CONCLUSION

In this paper, we propose a Generative Adversarial Network called R-GAN to synthesise reasonable context-rich images from complex descriptions via a human-like drawing process, *i.e.* , from reasonable text representation to reasonable layouts, then to reasonable shapes and finally to reasonable images. Specifically, several innovative modules such as the monolithic-structural text representation, shape gradient-sensitive loss and multi-modal geometry-aware spatially-adaptive generator are proposed.

Experimental results on MSCOCO dataset show the effectiveness of R-GAN in both quantitative and qualitative metrics, which can generate photo-realistic and reasonable images based on complex descriptions.

ACKNOWLEDGMENTS

This work was supported by ARC DE190100539.

REFERENCES

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 4 (2017), 834–848.

[2] Qifeng Chen and Vladlen Koltun. 2017. Photographic Image Synthesis with Cascaded Refinement Networks. In *Proc. IEEE Int. Conf. Comp. Vis.* 1520–1529.

[3] Qifeng Chen and Vladlen Koltun. 2017. Photographic image synthesis with cascaded refinement networks. In *Proc. IEEE Int. Conf. Comp. Vis.* 1511–1520.

[4] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vision* 111, 1 (2015), 98–136.

[5] David Ha and Douglas Eck. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477* (2017).

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 770–778.

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proc. Advances in Neural Inf. Process. Syst.* 6626–6637.

[8] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. 2020. Semantic object accuracy for generative Text-to-Image synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020).

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. 2018. Inferring Semantic Layout for Hierarchical Text-to-Image Synthesis. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 7986–7994.

[11] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. Int. Conf. Mach. Learn.* 448–456.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 5967–5976.

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proc. Eur. Conf. Comp. Vis.* Springer, 694–711.

[14] Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image generation from scene graphs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 1219–1228.

[15] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 3668–3678.

[16] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *Proc. Int. Conf. Learn. Representations.*

[17] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. 2019. Object-Driven Text-To-Image Synthesis via Adversarial Training. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 12174–12182.

[18] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. 2019. StoryGAN: A Sequential Conditional GAN for Story Visualization. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 6329–6338.

[19] Jiadong Liang, Wenjie Pei, and Feng Lu. 2020. CPGAN: Content-Parsing Generative Adversarial Networks for Text-to-Image Synthesis. In *Proc. Eur. Conf. Comp. Vis.*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). 491–508.

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proc. Eur. Conf. Comp. Vis.* 740–755.

[21] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and Hongsheng Li. 2019. Learning to Predict Layout-to-image Conditional Convolutions for Semantic Image Synthesis. In *Proc. Advances in Neural Inf. Process. Syst.* 568–578.

[22] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.

[23] Michael Mathieu, Camille Couprie, and Yann LeCun. 2016. Deep multi-scale video prediction beyond mean square error. *Proc. Int. Conf. Learn. Representations* (2016).

[24] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 722–729.

[25] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 2337–2346.

[26] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. 2019. MirrorGAN: Learning Text-to-image Generation by Redescription. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 1505–1514.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021).

[28] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. *Proc. Int. Conf. Mach. Learn.* (2016).

[29] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proc. Int. Conf. Mach. Learn.* 1278–1286.

[30] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*. 70–80.

[31] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. *Proc. Int. Conf. Learn. Representations* (2015).

[32] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel Recurrent Neural Networks. In *Proc. Int. Conf. Mach. Learn.* 1747–1756.

[33] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. (2011).

[34] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. 2018. Perceptual adversarial networks for image-to-image transformation. *IEEE Trans. Image Process.* 27, 8 (2018), 4066–4079.

[35] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 8798–8807.

[36] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

[37] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. 2019. Semantics Disentangling for Text-to-Image Generation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 2327–2336.

[38] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. 2018. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* (2018).

[39] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proc. IEEE Int. Conf. Comp. Vis.*

[40] Zizhao Zhang, Yuanpu Xie, and Lin Yang. 2018. Photographic Text-to-Image Synthesis With a Hierarchically-Nested Adversarial Network. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 6199–6208.

[41] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. 2019. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 5802–5810.

[42] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. 2019. DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-To-Image Synthesis. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* 5802–5810.