Learning Object Consistency and Interaction in Image Generation from Scene Graphs

Yangkang Zhang\(^1\), Chenye Meng\(^2\), Zejian Li\(^2\*) , Pei Chen\(^1\), Guang Yang\(^3\), Changyuan Yang\(^3\) and Lingyun Sun\(^1\)

\(^1\)College of Computer Science and Technology, Zhejiang University, China
\(^2\)School of Software Technology, Zhejiang University, China
\(^3\)Alibaba Group

\{yangkz, zejianlee, chenpei, sunly\}@zju.edu.cn, mengcye@stu.jiangnan.edu.cn, qingyun@taobao.com, changyuan.yangcy@alibaba-inc.com

Abstract

This paper is concerned with synthesizing images conditioned on a scene graph (SG), a set of object nodes and their edges of interactive relations. We divide existing works into image-oriented and code-oriented methods. In our analysis, the image-oriented methods do not consider object interaction in spatial hidden feature. On the other hand, in empirical study, the code-oriented methods lose object consistency as their generated images omit certain objects in the input scene graph. To alleviate these two issues, we propose Learning Object Consistency and Interaction (LOCI). To preserve object consistency, we design a consistency module with a weighted augmentation strategy for objects easy to be ignored and a matching loss between scene graphs and image codes. To learn object interaction, we design an interaction module consisting of three kinds of message propagation between the input scene graph and the learned image code. Experiments on COCO-stuff and Visual Genome datasets show our proposed method alleviates the ignorance of objects and outperforms the state-of-the-art on visual fidelity of generated images and objects.

1 Introduction

Conditional Image Synthesis is to generate images based on a given condition such as a segmentation mask [Park et al., 2019; Luo et al., 2021], text prompt [Ramesh et al., 2021; Schaldenbrand et al., 2022], a layout [Jahn et al., 2021; Li et al., 2021; Yang et al., 2022] or a scene graph [Johnson et al., 2018; Zhao et al., 2022]. These conditions enable humans to control the content, layout or style of synthesized results.

This paper is concerned with image generation from scene graphs (SG), a specific task of conditional image synthesis. Scene graphs are compact semantic representations of images. Nodes in scene graphs represent semantic objects and edges describe objects’ interactive relations. The SG-to-image generation task is to convert the multiple interacting objects to a photorealistic image without additional conditions such as segmentation masks [Park et al., 2019] or bounding boxes [Zhao et al., 2019]. This is a reversed task of scene graph generation from images [He et al., 2020; Yu et al., 2021]. As is underspecified, this task remains challenging but provides a wide range of applications such as image manipulation [Dhamo et al., 2020], drawing [Zhang et al., 2022] and computer-aided design [Zhang et al., 2021].

Existing SG-to-image generation methods can be divided into two types, image-oriented methods and code-oriented methods. The image-oriented methods apply a graph neural network (GNN) to learn object embeddings from the given scene graph and then generate new images directly with learned object embeddings [Johnson et al., 2018; Ashual and Wolf, 2019; Herzig et al., 2020; Hua et al., 2021; Xu and Xu, 2022]. The code-oriented methods [Zhao et al., 2022; Fan et al., 2022] have a pretrained autoencoder to map images to a learned code space. During generation, these meth-
ods first infer a latent code with encoded tokens of the scene graph and then decode the latent code to a generated image. In our analysis, Code-oriented methods implicitly model objects’ interaction in the spatial latent code, generally enjoying a better generative quality (Sec. 2).

However, we empirically find code-oriented methods suffer from an issue of ignoring objects in scene graphs and fail to preserve object consistency (Sec. 3). We examine existing code-oriented methods with Object Occurrence Ratio, a proposed metric to measure whether objects in a scene graph are present in the generated image. The code-oriented methods fail to generate objects which are with small area or rare in the dataset. Such ignorance is also observed in a user study. A further control trial between original and processed scene graphs reveals that objects are ignored in the composition mapping from scene graphs to latent codes (Sec. S6).

Based on the discussion and experimental results, we propose to Learn Object Consistency and Interaction (LOCI) for scene graph to image generation (Sec. 4). We adopt a code-oriented method [Esser et al., 2021] as a backbone and propose a consistency module to alleviate the object missing issue. Besides, it augments training scene graphs by removing nodes of objects which tend to be ignored.

We also propose an interaction module to strengthen object interaction. The interaction module performs three kinds of message propagation on a supergraph of the input SG containing image latent codes as nodes. Each latent code represents an image patch, therefore termed patch nodes. The first propagation is from object nodes to patch nodes allowing the input SG to directly control the image generation process. The second one is among patch nodes of each single object enabling the patches to be locally aware for better generative quality. The third one is among patch nodes of different objects with direct relations in the scene graph, and it models the relationship of objects in image level.

Based on the experiments and user study on COCO-stuff [Caesar et al., 2018] and Visual Genome [Krishna et al., 2017] datasets, our approach is superior to prior work with improved quality and consistency of generated images (Sec. 5). Our main contributions are summarized as follows:

1. We observe code-oriented methods ignore conditional objects with a proposed consistency metric and show that small and rare objects tend to be ignored.
2. We propose a consistency module to mitigate the object ignorance issue. The contribution is orthogonal to existing works and the module can be integrated with other methods.
3. We propose an interaction module to learn object interaction explicitly. It performs three kinds of message propagation to enhance spatial and relational appearance.

This paper also publishes a dataset containing about 1 million art images with basic attribute annotations detailed in supplementary. Source code, dataset and supplementary file are available at https://github.com/yangkzz/LOCI.

2 Review and Analysis of Existing Works

Scene Graph is a directed graph describing the relationships among objects in a scene [Xu et al., 2017]. The nodes in a scene graph represent objects and the edges denote their relationships. SG2Im [Johnson et al., 2018] is the first framework to achieve scene graph to image. It first computes a scene layout from input graphs and then generates images.

Image-oriented methods are based on SG2Im’s framework Ashual and Wolf [2019] which adopt a dual embedding scheme to generate multiple images per scene graph. Herzig et al. [2020] design a canonical representation of scene graphs to capture semantic equivalence, thus obtaining stronger invariance properties. Hua et al. [2021] introduce a novel model, which contains a pair-wise spatial constraint module, a relation-guided appearance generator and a scene graph discriminator. Ivgi et al. [2021] propose an architecture for generating instance segmentation layouts directly from scene graphs. Xu and Xu [2022] propose a semi-parametric generation strategy to retrieve image crops from datasets and then synthesize realistic images with the crops.

These methods boast object consistency between generated images and scene graphs. In the generation pipeline of image-oriented methods, object embeddings learned from GNN and the accordingly predicted layout are fed into an image generator to synthesize an image. Because this pipeline processes and supervises the position and shape of an individual object, objects are well-preserved in generated images.

However, object interaction is only considered in GNN but implicitly ignored in the image generation stage. Specifically, interactive relations described by edges are only processed in GNN and represented in the learned object embeddings. The embeddings must include information of the whole graph and precisely plan objects’ position, shape, texture and all other details in the generated image with limited capacity as vectors. Therefore, the generative quality depends on the embeddings to capture the whole graph’s information, especially objects’ interaction. Previous methods improves graph embedding learning with dual embeddings [Ashual and Wolf, 2019], canonical graph representation [Herzig et al., 2020] and the inclusion of relative scale and distance [Hua et al., 2021]. On the other hand, objects’ position and other details based on their interaction are more easily represented in the spatial latent code in image generation stage.

Code-oriented methods implicitly model objects’ interaction in the spatial latent code, generally enjoying a better generative quality. The code-oriented methods first encode images into a latent code space with an autoencoder [Esser et al., 2021]. Jahn et al. [2021] train a transformer [Vaswani et al., 2017; Radford et al., 2019] to convert the tokens of a layout to image latent codes, which implements controlled image generation from layouts. Similarly, Zhao et al. [2022] propose IGSGWT to generate images from latent codes converted from the tokens of a scene graph. Fan et al. [2022] propose Frido, which introduces scene graph information in a multi-scale denoising process for image synthesis. During the generation of most methods, new image latent codes are inferred in an auto-regressive way, so newly generated codes are aware of already generated ones. Therefore, different from image-oriented methods, object interaction is implicitly modeled among the generated latent codes spatially.
We first introduce Object Occurrence Ratio (OOR) to estimate the degree of preserving objects. Specifically, OOR evaluates the fraction of objects correctly recognized (recall) by YOLOv7 [Wang et al., 2022] in generated images according to input scene graphs in the whole COCO-stuff. A higher value of OOR indicates more objects are successfully generated. Furthermore, we perform user studies to summarize the ratio of objects recognized by human, termed Existing Ratio (ER) of objects. We randomly pick 100 scene graphs from COCO-stuff. In each trial, a user is given a scene graph as well as the generated image and chooses objects which he/she thinks are present on the image. Each trial is evaluated by 5 males and 5 females aged from 20 to 35 having different backgrounds in computer science, management and design. They are given unlimited time. The generated images from HCSS [Jahn et al., 2021] and IGSGWT [Zhao et al., 2022] are examined. Specifying [Ashual and Wolf, 2019] is also included for reference. Both results are in Tab. 1. The performances of both code-oriented methods are largely lower than that of GT images, which indicates objects are ignored. As an image-based method, Specifying has a higher ER than the code-oriented methods but a lower OOR; it preserves more objects. Empirically, the changes of inferred codes are significantly positive for the two methods. This indicates that objects with smaller areas or limited samples are more likely to be ignored.

To further identify the potential causal factors of object ignorance, we conduct control experiments (Sec. S6). We first examine whether objects are ignored by the autoencoder of HCSS and IGSGWT as a bijection between images and latent codes. Experiments are conducted on the original training images and processed images with intentionally added objects which are with smaller areas or limited samples. There is no significant difference on reconstruction error with Wilcoxon signed-rank test ($p = 0.129$). We then examine the composition mapping from scene graph tokens to image latent codes. We design two cases when an object with a small area or with a large area is removed in an input scene graph. Empirically, the changes of inferred codes are significantly less in the former case than that in the latter for both HCSS ($p = 5.07 \times 10^{-9}$) and IGSGWT ($p = 7.72 \times 10^{-10}$). It seems plausible to change a limited number of latent codes when the object area is small, but the adopted autoencoder [Esser et al., 2021] has a global receptive field with an attention module [Xu et al., 2018]. Local change should be visible to all codes. Therefore, it is tentatively concluded that the ignorance happens in the mapping part and is caused by the object area and the number of objects. The conclusion motivates us to design a consistency module to regularize the mapping detailed in the next section.

### 4 Method

#### 4.1 Method Overview

The proposed method aims to generate new images based on an input SG describing objects and their relationships. To mitigate the issue of object ignorance discussed in Sec. 3 and to enhance object interaction, we propose to Learn Object Consistency and Interaction (LOCI) simultaneously in the synthesis. The whole-generation model has three modules. Firstly, the image quantization module encodes images into latent codes with an accompanied decoder to synthesize images backward (Sec. 4.2). Secondly, a consistency module is proposed to regularize the learning of object embeddings and the mapping from embeddings to image latent codes (Sec. 4.3). Thirdly, an interaction module is designed to model object interaction explicitly and enhance local appearance in the mapping (Sec. 4.4).

#### 4.2 Image Quantization Module

The image quantization module transforms images into discretized codes in the latent space. Each latent code represents an image patch, and learning on latent codes has a smaller computation complexity compared with learning on image pixels. Formally, given an image $I$, the module has an encoder to form discretized image codes $S$ accordingly and a decoder to recover the image backward. We utilize the autoencoder of VQGAN [Esser et al., 2021].

#### 4.3 Consistency Module

The consistency module aims at preserving objects within the input scene graphs. It has a component to regress bounding boxes, a consistency loss term, and a weighted augmentation
strategy. This module is used in learning object embeddings and training the mapping from scene graphs to latent codes.

Following image-oriented methods, we adopt a graph neural network [Ye et al., 2019] to learn object embeddings of an input scene graph \( G \) with \( n \) object nodes. Each object node \( o \) has a learnable embedding \( v \in \mathbb{R}^d \). A fully-connected neural network \( B: \mathbb{R}^d \rightarrow [0, 1]^4 \) predicts the object’s bounding box \( \hat{b} = B(v) \). Here \( \hat{b} \) determines latent codes’ affiliation with objects; all latent codes located in \( \hat{b} \) are viewed as the object’s latent codes. As each object has its codes explicitly, its existence in the generated image is basically secured.

We adopt a consistency loss based on a matching score \( R(I, G) \) between an input SG \( G \) and its paired image \( I \) [Sylvain et al., 2021]. Specifically, \( R(I, G) \) is the multivariable softplus of cosine similarity between each object embedding and the object’s latent codes with attention mechanism [Xu et al., 2018] (Sec. S4). Our goal is to minimize \( R(I, G) \) when \( I, G \) are paired and maximize when unpaired. Given a batch of \( m \) pairs \( \{(I_i, G_i)\}_{i=1}^m \), the posterior probability that \( G_i \) is paired with \( I_i \) is defined as

\[
P(G_i \mid I_i) = \frac{\exp(\gamma R(I_i, G_i))}{\sum_{j=1}^m \exp(\gamma R(I_i, G_j))}
\]

(1)

Here \( \gamma \) is a smoothing factor, set as 10. \( P(I_i \mid G_i) \) is defined similarly. The consistency loss is defined as the negative log-posterior that the images are matched with their paired SGs and vice versa:

\[
\mathcal{L}_{\text{con}} = -E_{I,G} \left[ \log P(G \mid I) + \log P(I \mid G) \right]
\]

(2)

Accompanied by the loss, a weighted augmentation strategy is proposed to emphasize the existence of ignored objects. For a training scene graph \( G \), a counterfactual scene graph \( \text{aug}G \) is produced by randomly removing objects easy to be ignored and their edges. Based on the result in Sec. 3, the removing probability is determined by the object area or the number of samples. The probability in (1) becomes

\[
P(G_i \mid I_i) = \frac{\exp(\gamma R(I_i, G_i))}{\sum_{j=1}^m \exp(\gamma R(I_i, G_j)) + \sum_{l=1}^m \exp(\gamma R(I_i, \text{aug}G_{i,l}))}
\]

(3)

As the scene graphs with or without possibly ignored objects are compared directly in (3), \( P(G \mid I) \) and the consistency loss \( \mathcal{L}_{\text{con}} \) become sensitive to the existence of objects. When ignorance happens, both \( R(I, G) \) and \( R(I, \text{aug}G) \) are large. This decreases \( P(G \mid I) \) and \( P(I \mid G) \) but enlarges \( \mathcal{L}_{\text{con}} \). Thus, Minimizing \( \mathcal{L}_{\text{con}} \) penalize ignorance of objects.

4.4 Interaction Module

In this section, we introduce the interaction module to learn image latent codes. Although interaction is also represented in object embeddings on graphs, detailed information like position and shape is easier represented spatially by latent codes. In particular, the interaction among objects is better learned by considering the influence of object nodes on latent codes, the organization of latent codes within objects, and the interaction of codes between related objects. Thus, we propose three message propagation on a constructed graph.

Global message propagation (GMP) models the interactions between objects and image patches. Firstly, we construct a supergraph of the input scene graph including image latent codes as extra nodes, dubbed patch nodes. The supergraph’s node embeddings is denoted as \( \mathcal{V} = \{v_1, \ldots, v_n, v_{n+1}, \ldots, v_{n+|S|} \} \), with \( n \) object embeddings and
After message propagation, we utilize feed-forward network (FFN) with layer normalization [Ba et al., 2016] on each node. The FFN improves the feature transformation capacity and alleviates over-smoothing [Han et al., 2022].

\[
\begin{align*}
  u_i &= \text{LayerNorm}(v_i + v_i') \\
  v_i'' &= \sigma(u_i W_1) W_2 + u_i.
\end{align*}
\]

Here \( u_i, v_i', v_i'' \in \mathbb{R}^d \) and \( W_1, W_2 \in \mathbb{R}^{d \times d} \). \( \sigma(\cdot) \) is a GeLU activation [Hendrycks and Gimpel, 2016].

### 4.5 Training and Sampling

The training is supervised by images annotated with scene graphs and bounding boxes. It has three phases. The first phase trains a VQGAN [Esser et al., 2021] autoencoder and a code book of the image quantization module with a reconstruction and an adversarial loss. It offers image latent codes for the second and third phase.

The second phase trains the regression of bounding boxes \( B \) and a GNN model with the consistency module. For a sampled object with a GT bounding box \( \hat{b} \) and an embedding \( v \) given by the GNN, the training minimizes

\[
L_2 = L_{\text{bbox}} + L_{\text{con}} \quad \text{where} \quad L_{\text{bbox}} = \| \mathbf{b} - B(v) \|_2
\]

The trained GNN gives object embeddings \( V' = \{v_1, ..., v_n\} \).

The third phase trains the mapping from object embeddings to image latent codes with our consistency loss and the interaction module. The mapping infers latent code in an auto-regressive manner; a latent code \( s_i \in S \) is inferred with \( V' \) and \( \hat{s}_{c_i}^{N_i}, \hat{s}_{c_j}^{N_j} \) are those codes which are already predicted and whose patch node is connected to \( v_i \) based on \( N_i \) in (4). Embeddings in \( V' \) and \( \hat{s}_{c_i}^{N_i} \) are aggregated with GAT [Velickovic et al., 2018]. The training loss is

\[
L_3 = \lambda_1 L_{\text{con}} + \lambda_2 L_{\text{ce}} \quad \text{where} \quad L_{\text{ce}} = -\mathbb{E}_{s_i} \log P \left( s_i \mid \hat{s}_{c_i}^{N_i}, V' \right)
\]

\( L_{\text{ce}} \) is a cross-entropy loss. \( \lambda_1 = 0.6 \) and \( \lambda_2 = 0.4 \).

For an unseen scene graph during sampling, the trained GNN in the second phase gives new object embeddings, and the mapping in the third phase infers new latent codes auto-regressively. Accordingly, the decoder in the first phase generates new images. We leverage the multinomial resampling strategy [Jahn et al., 2021] to improve generative diversity.

### 5 Experiment

#### 5.1 Datasets and Baselines

We validate the proposed LOCI on the COCO-stuff and Visual Genome dataset using the same split of datasets as previous works [Johnson et al., 2018; Zhao et al., 2022].

Both image-oriented and code-oriented methods are compared. The former includes leading methods such as Specifying [Ashual and Wolf, 2019], Canonical [Herzig et al., 2020] and ERCIG [Hua et al., 2021]. The semi-parametric approaches PasteGAN [Li et al., 2019], RetrieveGAN [Tseng et al., 2020] and SCSM [Xu and Xu, 2022] are also compared. Code-oriented methods include IGSGWT [Zhao et al., 2022] and a layout-to-image method HCSS [Jahn et al., 2021].
5.2 Evaluation Metrics

Models are evaluated from three aspects including: the overall visual quality and diversity of generated images, the fidelity of generated objects, and the consistency of generated images and input SGs.

Inception Score (IS) [Salimans et al., 2016] and Fréchet Inception Distance (FID) [Heusel et al., 2017] measure overall visual quality. Specifically, we compute IS of generated images and FID between generated images and test images. Diversity Score (DS) [Zhang et al., 2018] estimates generative diversity. It is the perceptual similarity of deep features extracted from two generated images of the same SG. We also adopt SceneFID [Sylvain et al., 2021] which computes the Fréchet Inception Distance (FID) on the crops of all objects instead of the whole image to evaluate generated object fidelity. The Object Occurrence Ratio (OOR) introduced in Sec. 3 is also used to measure preservation of objects.

5.3 Qualitative Result

Fig. 4 presents generated 256 × 256 images on COCO-Stuff and Visual Genome. Each column shows a scene graph, the associated ground-truth image, the baseline results of [Herzig et al., 2020] and [Zhao et al., 2022] and our result. Our model is more likely to generate realistic and visually appealing images and objects. Moreover, our generated images are more consistent with the input SG than other methods.

5.4 Quantitative Result

Tab. 3 reports the quantitative results of the baseline models and ours. Our model outperforms the existing methods in most cases. In terms of image and object quality, our model has lower FID, SceneFID values and higher IS scores. For consistency, our model has higher OOR values than other code-oriented methods do. Notice that the HCSS and IGSGWT variants equipped with our consistency module have markedly improved OOR values, and other metrics change slightly. In COCO-stuff, OOR of HCSS and IGSGWT increases by 3.58% and 3.97%.

OOR values of image-oriented methods are lower than those of code-oriented methods. We attribute this to the methods’ difficulty to generate objects of high quality rather than object ignorance. One reason is that OOR is based on the recognition YOLOv7 [Wang et al., 2022], which measures both existence and quality. The other is based on results in the following user study. Existing Ratio of image-oriented methods is higher (Tab. 5), so objects in the scene graph are already generated and identified by humans.

5.5 Ablation Study

We conduct ablation studies of LOCI to show the positive role of the consistency and the interaction module on COCO-stuff (Tab. 4). Image quantization module with global message propagation is treated as our baseline model which maps the object embeddings at graph-level to image latent codes at image-level as existing works do. We first gradually add three
### Table 3: Quantitative results on COCO-Stuff and Visual Genome. * means semi-parametric approaches. † means that Canonical filters 10 objects per image at most on Visual Genome. o means Canonical adopt the filtering strategy as existing methods. GT means using ground truth layouts instead of scene graphs. SFID is SceneFID. + con means applying our consistency module with weighted augmentation.

<table>
<thead>
<tr>
<th>Type</th>
<th>Methods</th>
<th>COCO-Stuff</th>
<th>Visual Genome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID†</td>
<td>IS†</td>
<td>DS†</td>
</tr>
<tr>
<td>Image-oriented</td>
<td>SG2IM</td>
<td>226.3</td>
<td>3.8±0.1</td>
</tr>
<tr>
<td></td>
<td>Specifying</td>
<td>81.0</td>
<td>14.5±0.7</td>
</tr>
<tr>
<td></td>
<td>Canonical †</td>
<td>119.1</td>
<td>13.9±0.3</td>
</tr>
<tr>
<td></td>
<td>Canonical ‡</td>
<td>119.1</td>
<td>13.9±0.3</td>
</tr>
<tr>
<td></td>
<td>ERCIG</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>PasteGAN *</td>
<td>78.8</td>
<td>8.5±0.3</td>
</tr>
<tr>
<td></td>
<td>RetrieveGAN *</td>
<td>56.9</td>
<td>10.2±0.4</td>
</tr>
<tr>
<td></td>
<td>SCSM *</td>
<td>51.6</td>
<td>15.2±0.1</td>
</tr>
<tr>
<td>Code-oriented</td>
<td>HCSS (GT)</td>
<td>56.6</td>
<td>14.2±0.3</td>
</tr>
<tr>
<td></td>
<td>HCSS (GT) + con</td>
<td>57.8</td>
<td>13.7±0.5</td>
</tr>
<tr>
<td></td>
<td>IGSSWT</td>
<td>61.4</td>
<td>12.6±0.6</td>
</tr>
<tr>
<td></td>
<td>IGSSWT + con</td>
<td>59.6</td>
<td>12.9±0.4</td>
</tr>
<tr>
<td></td>
<td>LOCI (ours)</td>
<td>49.8</td>
<td>15.7±0.5</td>
</tr>
</tbody>
</table>

### Table 4: Ablation studies on COCO-Stuff. + con means adding consistency module without augmentation. + con † means adding consistency module with augmentation weighted on rarely seen objects and small objects respectively.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FID†</th>
<th>IS†</th>
<th>DS†</th>
<th>SFID†</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMP</td>
<td>90.2</td>
<td>12.1±0.2</td>
<td>0.67±0.1</td>
<td>34.2</td>
</tr>
<tr>
<td>+ LMP</td>
<td>65.8</td>
<td>13.9±0.3</td>
<td>0.65±0.1</td>
<td>26.8</td>
</tr>
<tr>
<td>+ RMP</td>
<td>53.1</td>
<td>15.5±0.4</td>
<td>0.68±0.1</td>
<td>23.9</td>
</tr>
<tr>
<td>+ con †</td>
<td>50.3</td>
<td>16.0±0.3</td>
<td>0.66±0.1</td>
<td>23.2</td>
</tr>
<tr>
<td>+ con †</td>
<td>51.7</td>
<td>14.9±0.3</td>
<td>0.66±0.1</td>
<td>23.7</td>
</tr>
<tr>
<td>+ con †</td>
<td>49.8</td>
<td>15.7±0.5</td>
<td>0.65±0.1</td>
<td>22.0</td>
</tr>
</tbody>
</table>

**Table 5: User study results. + con means applying our consistency module. ER of LOCI is 88.94% and of GT images is 95.25%**.

**6 Conclusion**

In this paper, we study image generation from scene graphs. We observe object ignorance of code-based method with the proposed OOR metric and propose the consistency module to alleviate this problem. We also propose the interaction module to strengthen objects’ interaction when inferring latent codes. The consistency module helps to preserve objects and both modules improve the generation performance in our ablation studies. We also provide more details of LOCI (Sec. S1-5), detail our experiments (S6-8), discuss the limitation based on failure examples (S9), and present an ethical statement (S10) in the supplementary material.

Our work discusses a fundamental topic in conditional image synthesis, the fidelity to conditions. Object ignorance is observed in image generation from scene graphs and layouts (Sec. 3), and similar issues may exist in other generation tasks. Behind this phenomenon is the missing mode problem of generative models, which deserves further exploration.

**Acknowledgements**

This paper is funded by National Key R&D Program of China (2018YAA0100703) and the National Natural Science Foundation of China (No. 62006208 and No. 62107035).
References


