

Learning Object Consistency and Interaction in Image Generation from Scene Graphs

Yangkang Zhang¹, Chenye Meng², Zejian Li^{2*}, Pei Chen¹, Guang Yang³, Changyuan Yang³ and Lingyun Sun¹

¹College of Computer Science and Technology, Zhejiang University, China

²School of Software Technology, Zhejiang University, China

³Alibaba Group

{yangkz, zejianlee, chenpei, sunly}@zju.edu.cn, mengcy@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 interact-

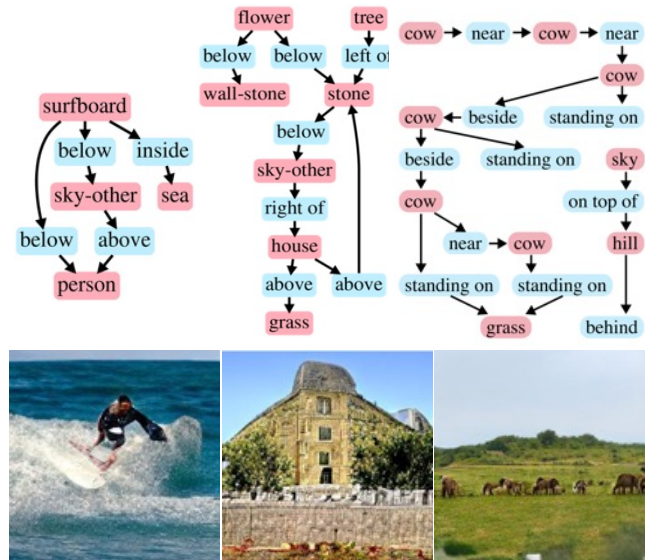


Figure 1: Example images in 256×256 generated by the proposed LOCI from scene graphs (first row) on COCO-Stuff (first two columns) and Visual Genome dataset (last column).

ing 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-

*Corresponding author

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 rela-

tionships. 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.

Methods	OOD [†]	ER [†]
GT images	89.37%	95.25%
HCSS [Jahn <i>et al.</i> , 2021]	77.39%	81.27%
IGSGWT [Zhao <i>et al.</i> , 2022]	75.48%	78.73%
Specifying [Ashual and Wolf, 2019]	70.84%	83.04%

Table 1: Quantitative results in pre-experiments. OOR and ER measure whether objects in the input graph are preserved.

Methods	Object area	# Objects in the dataset
GT images	0.41	0.37
HCSS	0.59	0.54
IGSGWT	0.61	0.57

Table 2: Pearson Correlation Coefficients between OOR and object area or the sample number of objects in the dataset. For each kind of object, its individual OOR is calculated for different methods, and its average area or the number of samples is summarized from the training set. The correlation is computed on all kinds of objects. A high coefficient implies when a kind of object has a large area on average, the object is more likely to be generated. This is similar to the object’s number of samples.

3 Pre-Experiment on Code-Oriented Methods

Existing code-oriented methods show superiority in generation, but they ignore those objects which have a small area or limited image samples. This issue is exemplified by the following experiments on COCO-stuff [Caesar *et al.*, 2018].

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 indicate 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 but has a lower generative quality.

Further statistical analysis shows OOR is correlated with object area and the number of objects in the dataset (Tab. 2). In detail, we compute the Pearson Correlation Coefficient between OOR and the area (height times width of bounding boxes) of each object category in the training set. That between OOR and the number of objects present in training images is also computed. Both correlation coefficients are

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

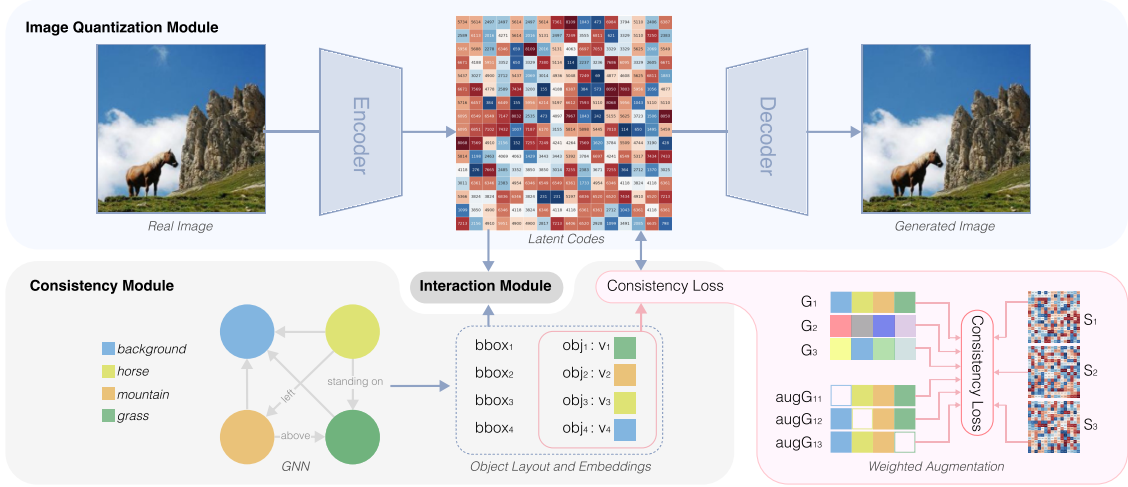


Figure 2: The overview of LOCI structure. In the training phase, we firstly train an encoder of VQGAN that quantize images into latent codes and a decoder to synthesize images from codes. Secondly, we train a consistency module based on a graph neural network that predicts bounding box of each object and obtains object embeddings. The training is supervised with consistency loss that help preserve object occurrence in the generated image. A weighted augmentation of SGs is adopted to draw the module’s attention to objects easy to be ignored. Thirdly, we train an interaction module that builds a bridge between object embeddings and image latent codes which form a supergraph. We apply three kinds of message propagation upon the supergraph to enhance spatial and relational information of latent codes. During testing, we generate latent codes from the input SG in an auto-regressive manner and synthesize the final image by decoding the latent codes.

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 $B: \mathbb{R}^d \mapsto [0, 1]^4$ 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

$$P(G_i | I_i) = \frac{\exp(\gamma R(I_i, G_i))}{\sum_{j=1}^m \exp(\gamma R(I_i, G_j))} \quad (1)$$

Here γ is a smoothing factor, set as 10. $P(I_i | G_i)$ 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:

$$\mathcal{L}_{con} = -\mathbb{E}_{I, G} [\log P(G | I) + \log P(I | G)] \quad (2)$$

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 $augG$ 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

$$P(G_i | I_i) = \frac{\exp(\gamma R(I_i, G_i))}{\sum_{j=1}^m \exp(\gamma R(I_i, G_j)) + \sum_{l=1}^m \exp(\gamma R(I_i, augG_{i,l}))} \quad (3)$$

As the scene graphs with or without possibly ignored objects are compared directly in (3), $P(G | I)$ and the consistency loss \mathcal{L}_{con} become sensitive to the existence of objects. When ignorance happens, both $R(I, G)$ and $R(I, augG)$ are large. This decreases $P(G | I)$ and $P(I | G)$ but enlarges \mathcal{L}_{con} . Thus, Minimizing \mathcal{L}_{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 $V = \{v_1, \dots, v_n, v_{n+1}, \dots, v_{n+|S|}\}$, with n object embeddings and

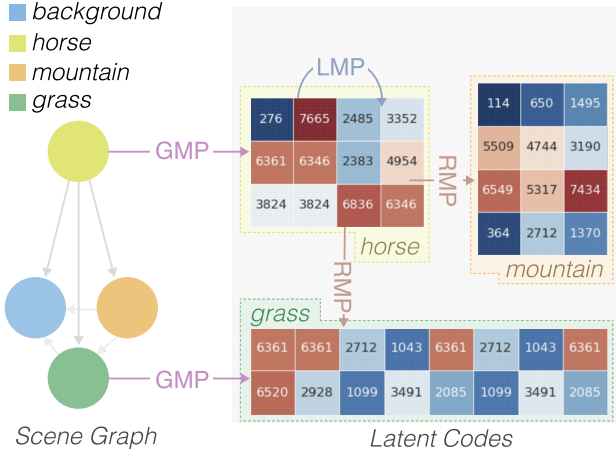


Figure 3: The proposed three kinds of message propagation on the supergraph in the interaction module. (1) Global message propagation (GMP) between object nodes and patch nodes. Message from each object node propagates to all the patch nodes overlapped with the object node’s bounding box. (2) Local message propagation (LMP) among patch nodes in each object. Message from each patch node propagates to all other patch nodes in the same bounding box. (3) Relational message propagation (RMP) among patch nodes of different objects. Here messages from patch nodes of “horse” propagate to those of “grass” and “mountain”.

$|S|$ patch codes’ embeddings. Secondly, given the predicted bounding boxes B in the consistency module, each object node is connected to patch nodes within its bounding box. Thirdly, messages are propagated from object nodes to patch nodes. With $M: \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^d$, $M(v_j, v_i)$ means message propagation from v_j to v_i (Sec. S4). $\mathcal{N}_i^{(G)}$ is the set of object nodes whose predicted bounding box includes patch node v_i .

Local message propagation (LMP) is to preserve patch nodes’ locality. It allows each patch to be aware of how its neighbors are generated already to improve fine-grained details. Specifically, all patch nodes inside an object’s bounding box form a clique (complete subgraph). $\mathcal{N}_i^{(L)}$ denotes the set of patch nodes in the same bounding box of patch node v_i .

Relational message propagation (RMP) is to model objects’ interaction on patch nodes. When two objects are directly connected in the input scene graph, patch nodes in their predicted bounding boxes are connected by RMP. Formally, when object o_i and o_j are connected in the scene graph G , patch nodes in bounding box \hat{b}_i and in \hat{b}_j are connected to form a complete bipartite graph. These connections strengthen the interaction between o_i and o_j on the image level. $\mathcal{N}_i^{(R)}$ denotes the patch nodes within other objects’ bounding box but connected to v_i .

We summarize the message propagation of a patch node v_i :

$$v'_i = \sum_{j \in \mathcal{N}_i} \alpha_{ji} M(v_j, v_i) \quad (4)$$

$$\text{where } \mathcal{N}_i = \mathcal{N}_i^{(G)} \cup \mathcal{N}_i^{(L)} \cup \mathcal{N}_i^{(R)}$$

Here α_{ji} is the attention from v_j to v_i [Velickovic *et al.*, 2018] and the message propagation is done on all patch nodes.

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].

$$u_i = \text{LayerNorm}(v'_i + v_i) \quad (5)$$

$$v''_i = \sigma(u_i \mathbf{W}_1) \mathbf{W}_2 + u_i.$$

Here $u_i, v'_i, v''_i \in \mathbb{R}^d$ and $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$. $\sigma(\cdot)$ 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 b and an embedding v given by the GNN, the training minimizes

$$\mathcal{L}_2 = \mathcal{L}_{bbox} + \mathcal{L}_{con} \quad (6)$$

where $\mathcal{L}_{bbox} = \mathbb{E}_{b,v} \|b - B(v)\|_2$

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

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 $\hat{s}_{<i}^{\mathcal{N}_i}$. $\hat{s}_{<i}^{\mathcal{N}_i}$ are those codes which are already predicted and whose patch node is connected to v_i based on \mathcal{N}_i in (4). Embeddings in V' and $\hat{s}_{<i}^{\mathcal{N}_i}$ are aggregated with GAT [Velickovic *et al.*, 2018]. The training loss is

$$\mathcal{L}_3 = \lambda_1 \mathcal{L}_{con} + \lambda_2 \mathcal{L}_{ce} \quad (7)$$

where $\mathcal{L}_{ce} = -\mathbb{E}_{s_i} \log P(s_i | \hat{s}_{<i}^{\mathcal{N}_i}, V')$

\mathcal{L}_{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].

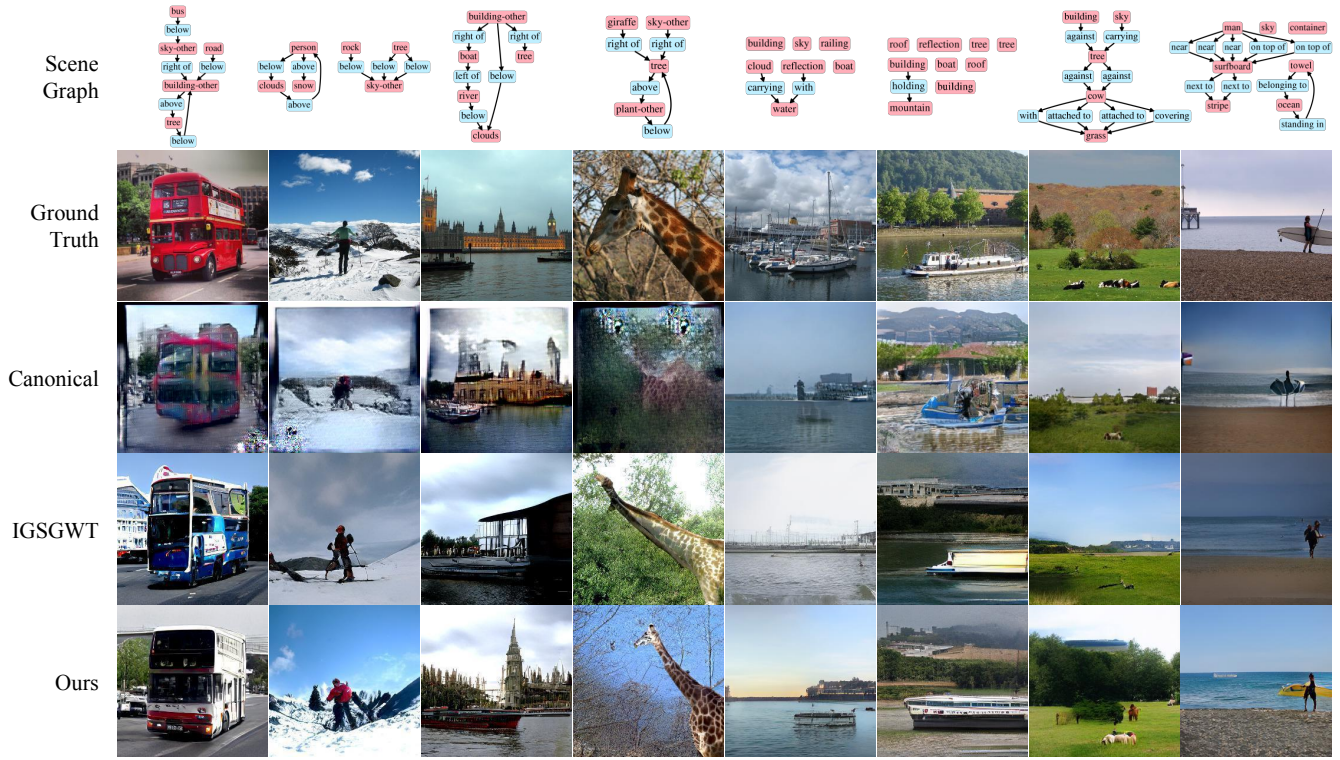


Figure 4: Visual comparison experiments on COCO-stuff (first 4 columns) and Visual Genome (last 4 columns) datasets. Methods for comparison include Canonical (3rd row), IGSGWT (4th row). The first two rows are the input scene graphs and the ground truth images. The last row demonstrates our generated images.

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

Type	Methods	COCO-Stuff					Visual Genome				
		FID \downarrow	IS \uparrow	DS \uparrow	SFID \downarrow	OOR \uparrow	FID \downarrow	IS \uparrow	DS \uparrow	SFID \downarrow	OOR \uparrow
Image-oriented	SG2IM	226.3	3.8 \pm 0.1	0.02 \pm 0.0	-	-	210.0	4.7 \pm 0.1	0.10 \pm 0.1	-	-
	Specifying	81.0	14.5 \pm 0.7	0.67 \pm 0.1	35.9	70.84	-	-	-	-	-
	Canonical \dagger	119.1	13.9 \pm 0.3	0.70 \pm 0.1	52.7	73.77	45.7	16.5 \pm 0.7	0.68 \pm 0.1	25.2	72.83
	Canonical \circ	119.1	13.9 \pm 0.3	0.70 \pm 0.1	52.7	73.77	77.8	9.0 \pm 0.5	0.64 \pm 0.1	33.7	70.25
	ERCIG	-	-	-	-	-	85.7	10.8 \pm 0.9	-	-	-
	PasteGAN *	78.8	8.5 \pm 0.3	0.60 \pm 0.1	-	-	131.6	6.5 \pm 0.3	0.38 \pm 0.1	-	-
	RetrieveGAN *	56.9	10.2 \pm 0.4	0.47 \pm 0.1	-	-	113.1	7.5 \pm 0.1	0.30 \pm 0.1	-	-
	SCSM *	51.6	15.2 \pm 0.1	0.63 \pm 0.1	-	-	63.7	10.8 \pm 0.2	0.59 \pm 0.1	-	-
Code-oriented	HCSS (GT)	56.6	14.2 \pm 0.3	0.66 \pm 0.1	24.1	77.39	28.1	13.8 \pm 0.4	0.63 \pm 0.1	11.4	75.47
	HCSS (GT) + <i>con</i>	57.8	13.7 \pm 0.5	0.65 \pm 0.1	24.8	80.97	27.6	13.9 \pm 0.6	0.60 \pm 0.1	11.2	78.23
	IGSGWT	61.4	12.6 \pm 0.6	0.64 \pm 0.1	25.8	75.48	52.8	12.1 \pm 0.7	0.60 \pm 0.1	27.7	73.07
	IGSGWT + <i>con</i>	59.6	12.9 \pm 0.4	0.63 \pm 0.1	24.3	79.45	53.6	12.0 \pm 0.5	0.57 \pm 0.1	26.4	76.92
	LOCI (ours)	49.8	15.7 \pm 0.5	0.65 \pm 0.1	22.0	81.26	44.9	14.6 \pm 0.4	0.62 \pm 0.1	20.8	79.04

Table 3: Quantitative results on COCO-Stuff and Visual Genome. * means semi-parametric approaches. \dagger means that Canonical filters 10 objects per image at most on Visual Genome. \circ 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.

Methods	FID \downarrow	IS \uparrow	DS \uparrow	SFID \downarrow	OOR \uparrow
GMP	90.2	12.1 \pm 0.2	0.67 \pm 0.1	34.2	42.75
+ LMP	65.8	13.9 \pm 0.3	0.65 \pm 0.1	26.8	69.54
+ RMP	53.1	15.5 \pm 0.4	0.68 \pm 0.1	23.9	74.23
+ <i>con</i> \circ	50.3	16.0 \pm 0.3	0.66 \pm 0.1	23.2	77.47
+ <i>con</i> \dagger	51.7	14.9 \pm 0.3	0.66 \pm 0.1	23.7	79.83
+ <i>con</i>	49.8	15.7 \pm 0.5	0.65 \pm 0.1	22.0	81.26

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

Ours vs. Baselines	Quality	Fidelity to SG	ER \uparrow
Specifying	93.26%	90.35%	83.04%
Canonical	87.77%	84.36%	85.54%
IGSGWT	78.45%	75.12%	78.73%
IGSGWT + <i>con</i>	71.16%	63.32%	86.82%

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

components of our interaction module. The results show the performance boosts with LMP and RMP due to the local and relational interactions. Then, we add the consistency module with uniform, area-weighted or occurrence-weighted augmentation. The results show that object ignorance problem is mitigated by focusing on the consistency between the image latent codes and the input objects especially with small area or rarely seen. This exemplify the consistency module’s efficacy. We adopt the version with augmentation weighted by object area as our full model.

5.6 User Study

The same users in the user study of Sec. 3 are invited again. One experiment is to evaluate the preference between synthe-

sized images. The users are given one scene graph and two images generated by LOCI and a baseline. They are asked to select the better one with the criteria of image quality and fidelity to the scene graph. LOCI is preferred in all case of both criteria (Tab. 5). The other experiment is to measure Existing Ratio in Sec. 3. ER of our model is 88.94%, showing superiority to other baselines. Specifically, if IGSGWT equips with our consistency module, its ER improves by 8.09%. Also, the users’ preference to LOCI decreases largely when compared with this IGSGWT variant; the consistency module earns users’ preference for IGSGWT. Both results indicate the contribution of the proposed consistency module.

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 (2018AAA0100703) and the National Natural Science Foundation of China (No. 62006208 and No. 62107035).

References

- [Ashual and Wolf, 2019] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4560–4568, 2019.
- [Ba *et al.*, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [Caesar *et al.*, 2018] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018.
- [Dhamo *et al.*, 2020] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory Hager, Federico Tombari, and Christian Rupprecht. Semantic image manipulation using scene graphs. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5212–5221, 2020.
- [Esser *et al.*, 2021] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12868–12878, 2021.
- [Fan *et al.*, 2022] Wanshu Fan, Yen-Chun Chen, Dongdong Chen, Yu Cheng, Lu Yuan, and Yu-Chiang Frank Wang. Frido: Feature pyramid diffusion for complex scene image synthesis. *ArXiv*, abs/2208.13753, 2022.
- [Han *et al.*, 2022] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision GNN: An image is worth graph of nodes. *arXiv preprint arXiv:2206.00272*, 2022.
- [He *et al.*, 2020] Tao He, Lianli Gao, Jingkuan Song, Jianfei Cai, and Yuan-Fang Li. Learning from the scene and borrowing from the rich: Tackling the long tail in scene graph generation. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 587–593. International Joint Conferences on Artificial Intelligence Organization, 7 2020.
- [Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.
- [Herzig *et al.*, 2020] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *European Conference on Computer Vision*, pages 210–227. Springer, 2020.
- [Heusel *et al.*, 2017] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30:6626–6637, 2017.
- [Hua *et al.*, 2021] Tianyu Hua, Hongdong Zheng, Yalong Bai, Wei Zhang, Xiao-Ping Zhang, and Tao Mei. Exploiting relationship for complex-scene image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1584–1592, 2021.
- [Ivgi *et al.*, 2021] Maor Ivgi, Yaniv Benny, Avichai Ben-David, Jonathan Berant, and Lior Wolf. Scene graph to image generation with contextualized object layout refinement. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2428–2432. IEEE, 2021.
- [Jahn *et al.*, 2021] Manuel Jahn, Robin Rombach, and Björn Ommer. High-resolution complex scene synthesis with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2021.
- [Johnson *et al.*, 2018] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018.
- [Krishna *et al.*, 2017] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.
- [Li *et al.*, 2019] Yikang Li, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. PasteGAN: A semi-parametric method to generate image from scene graph. *Advances in Neural Information Processing Systems*, 32:3950–3960, 2019.
- [Li *et al.*, 2021] Zejian Li, Jingyu Wu, Immanuel Koh, Yongchuan Tang, and Lingyun Sun. Image synthesis from layout with locality-aware mask adaption. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13819–13828, 2021.
- [Luo *et al.*, 2021] Ziwei Luo, Jing Hu, Xin Wang, Siwei Lyu, Bin Kong, Youbing Yin, Qi Song, and Xi Wu. Stochastic actor-executor-critic for image-to-image translation. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2775–2781. International Joint Conferences on Artificial Intelligence Organization, 8 2021.
- [Park *et al.*, 2019] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341, 2019.
- [Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, last accessed on 18th Jan, 2023.
- [Ramesh *et al.*, 2021] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

- [Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, 29:2226–2234, 2016.
- [Schaldenbrand *et al.*, 2022] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. StyleCLIPDraw: Coupling content and style in text-to-drawing translation. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4966–4972. International Joint Conferences on Artificial Intelligence Organization, 7 2022.
- [Sylvain *et al.*, 2021] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2647–2655, 2021.
- [Tseng *et al.*, 2020] Hung-Yu Tseng, Hsin-Ying Lee, Lu Jiang, Ming-Hsuan Yang, and Weilong Yang. RetrieveGAN: Image synthesis via differentiable patch retrieval. In *European Conference on Computer Vision*, pages 242–257. Springer, 2020.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.
- [Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.
- [Wang *et al.*, 2022] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.
- [Xu and Xu, 2022] Xiaogang Xu and Ning Xu. Hierarchical image generation via transformer-based sequential patch selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2938–2945, 2022.
- [Xu *et al.*, 2017] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5419, 2017.
- [Xu *et al.*, 2018] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018.
- [Yang *et al.*, 2022] Zuopeng Yang, Daqing Liu, Chaoyue Wang, Jie Yang, and Dacheng Tao. Modeling image composition for complex scene generation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7754–7763, 2022.
- [Ye *et al.*, 2019] Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. A vectorized relational graph convolutional network for multi-relational network alignment. In *International Joint Conferences on Artificial Intelligence*, pages 4135–4141, 2019.
- [Yu *et al.*, 2021] Jing Yu, Yuan Chai, Yujing Wang, Yue Hu, and Qi Wu. CogTree: Cognition tree loss for unbiased scene graph generation. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1274–1280. International Joint Conferences on Artificial Intelligence Organization, 8 2021.
- [Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- [Zhang *et al.*, 2021] Wensheng Zhang, Yan Zheng, Taiga Miyazono, Seiichi Uchida, and Brian Kenji Iwana. Towards book cover design via layout graphs. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, *International Conference on Document Analysis and Recognition*, pages 642–657. Cham, 2021. Springer International Publishing.
- [Zhang *et al.*, 2022] Tianyu Zhang, Xu Du, Chia-Ming Chang, Xi Yang, and Haoran Xie. Interactive drawing interface for editing scene graph. *International Conference on Cyberworlds (CW)*, pages 171–172, 2022.
- [Zhao *et al.*, 2019] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8584–8593, 2019.
- [Zhao *et al.*, 2022] Xin Zhao, Lei Wu, Xu Chen, and Bin Gong. High-quality image generation from scene graphs with transformer. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.