Journal of King Saud University – Computer and Information Sciences 35 (2023) 101543

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Image generation models from scene graphs and layouts: A comparative analysis

Muhammad Umair Hassan *, Saleh Alaliyat, Ibrahim A. Hameed

Department of ICT and Natural Sciences, Norwegian University of Science and Technology (NTNU), Ålesund, Norway

ARTICLE INFO

Article history:
Received 30 November 2022
Revised 21 March 2023
Accepted 27 March 2023
Available online 6 April 2023

Keywords:
Image generation
Image translation
Generative adversarial networks
Graph convolutional networks
Comparative analysis

ABSTRACT

An image is the abstraction of a thousand words. The meaning and essence of complex topics, ideas, and concepts can be easily and effectively conveyed visually by a single image rather than a lengthy verbal description. It is not only essential to teach computers how to recognize and classify images but also how to generate them. Controlled image generation depicting complex and multiple objects is a challenging task in computer vision despite the significant advancements in generative modeling. Among the core challenges, scene graph-based and scene layout-based image generation is a significant problem in computer vision and requires generative models to reason about object relationships and compositionality. Due to its ease of use, less time cost, and labor needs, image generation/synthesizing models from scene graphs and layouts are proliferating. In the case of a more significant number of scene graphs and layout to image generation models, a unique experimental evaluation methodology is required to evaluate the controlled image generation. To this extent, we, in this work, present a standard methodology to evaluate the performance of scene graph and scene layout-based image generation models. We perform a comparative analysis of image generation models to evaluate image generation models' complexity from scene graphs and scene layouts. We analyze the different components of these models on Visual Genome and COCO-Stuff datasets. The experimental results show that the scene layout-based image generation outperforms its graph-based counterpart in most quantitative and qualitative evaluations.

1. Introduction

Image generation from texts and natural language descriptions is a complex problem in computer vision. However, several attempts have been made to synthesize images from texts by integrating generative adversarial networks (GAN) with recurrent neural networks (Reed et al., 2016; Reed et al., 2016; Reed et al., 2017; Zhang et al., 2017). These methods yield impressive results on limited domains such as fine-grained descriptions of birds or flowers etc. Usually, the "text to image synthesis" based methods often confuse in generating images while using complex sentences containing several objects (Zhang et al., 2017).

The information conveyed by a sentence is represented in a linear structure in which one word follows another, and the complex sentences can often explicitly be represented as scene graphs. Scene graph is a higher-level understanding of relationships between objects. It is the deep representation of a scene that is very conducive to many visual tasks such as visual question answering (VQA) (Antol et al., 2015), image retrieval (Johnson et al., 2015), image/video captioning (Yang et al., 2019; Zhao et al., 2022), 3D scene understanding (Kim et al., 2019), image manipulation (Dhamo et al., 2020) and image generation (Johnson et al., 2018). In VQA, the scene graphs are derived from images for visual feature learning and applied to graph networks (Zhang et al., 2019a) to perform reasoning about questions provided. With the help of scene graphs, the model can accurately describe the image semantics without the help of unstructured text and retrieve the related images more interpretably (Schuster et al., 2015). To make a full use of semantic relationships between objects, the image captioning methods mostly rely on natural language reasoning models such as recurrent neural network (RNN) or long short-term memory (LSTM). These language models result in inaccurate image descriptions, which is why the scene graph-based

* Corresponding author.
E-mail addresses: muhammad.u.hassan@ntnu.no (M.U. Hassan), alaliyat.a.saleh@ntnu.no (S. Alaliyat), ibib@ntnu.no (I.A. Hameed).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.03.021
1319-1578/

image captioning methods are used (Gao et al., 2018). In 3D scene understanding, the scene graph-based methods can provide the numerically accurate analysis of object relationships in a 3D scene. A 3D scene graph helps in understanding the complex indoor environments and other tasks (Armeni et al., 2019).

The image generation based on scene graphs can better deal with complex scenes containing multiple objects and desired layouts. Object representations, attributes, and pairwise image relationships mainly reflect semantically structured information in the scene graphs. Thus, scene graphs help provide favorable reasoning and information about vision-text tasks such as image generation. With such contemporary tasks comes complex challenges. The scene graph-to-image generation (scene graph to image – SG2I) is a crucial problem in the computer vision and computer graphics community. The images generated by existing SG2I-based algorithms are blurred, and the appearance of objects in the generated images is hardly understandable, making the SG2I task challenging.

The general process of image generation from scene graphs is based on two-step: first, a layout is generated from the scene graph, and then bounding boxes are constructed to convert the layout into images. Nevertheless, sometimes, it is hard for someone to design vocabulary-based scene graphs. To bridle this issue, a direct scene layout to image generation (scene layout to image – SL2I) method was developed by Zhao et al. (2019b). The proposed method is based on the core concept of scene graph to image (SG2I) generation process. As mentioned earlier, traditional SG2I methods first generate the layouts from scene graphs, but, in the case of SL2I, the user only needs to define the bounding boxes with object categories which are used to generate the expected images.

Image generation works based on scene graphs, and layouts are overgrowing. Since SG2I and SL2I are becoming powerful enough, they can one day replace the work of scene-generation artists. There is a clear motivation to design a standardized methodology and analyze state-of-the-art methods to provide comprehensible insights on recent developments in SG2I and SL2I generation models. The standard input types for all such methods are scene graphs. SL2I methods also incorporate a scene graph-based strategy to synthesize images. The generation of images by scene graphs and layouts is more controllable, but it remains a one-to-many problem. In order to evaluate the existing proposed works, there is a great demand to propose a standard methodology to evaluate SG2I and SL2I methods. We, therefore, propose to compare the SG2I and SL2I methods in a unified pipeline to generate images. In this study, we aim to conduct a comparative analysis of image generation algorithms that are based on scene graphs and layouts. These methods, if improved, can assist the work of artists, graphic designers and can help crime scene investigators to learn more about the evidences, visually. In the algorithms addressed in this analysis, it is only required to define some objects and how they interact with each other, then one can generate an image based on provided characteristics and relationships. Moreover, the automatic image generation process is so vigorous that one day it might replace the image and video search engines with customized image and video generation algorithms based on individual user preferences. We investigated four SG2I and SL2I based algorithms, which are built upon the pioneer method (Johnson et al., 2018). We used the identical parameters to test the proposed methods. The different components of image generation methods are discussed and compared in this work.

2. Motivation and paper organization

Different surveys of text-based image synthesis using GANs have been conducted recently (He et al., 2021; Zhou et al., 2021; Shamsolmoali et al., 2021). However, to the best of our knowledge, no study has been conducted to provide a comprehensive comparative analysis of scene graph-based and scene layout-based image generation methods. There is lack of comprehensive comparison of image generation evaluation metrics, remedies for diverse image synthesis, and information about stable training. This paper reports an experimental comparison of the state-of-the-art scene graphs and scene layout-based image generation methods and provide a comprehensive knowledge about training of SG2I and SL2I algorithms. The main contributions of this work are as follows:

- A standard methodology is proposed to conduct the comparative analysis of SG2I and SL2I methods is proposed in this work. To this end, we apply the identical configurations for training the SG2I and SL2I models from scratch on the Visual Genome Krishna et al. (2017) and COCO-Stuff Caesar et al. (2018) datasets and tested the methods on different hyperparameters.
- A theoretical comparison of different components of SG2I and SL2I based methods is presented to help better understand the complexity of image generation methods from scene graphs and layouts for implementation purposes.

The organization of the paper is as follows. Section 3 presents the background knowledge of the basic methods used in image generation. We provide a comprehensive overview of current state-of-the-art methods for image generation from texts, scene graphs, and scene layouts in Section 4. Section 5 provides the methodological description of SG2I and SL2I based methods. Section 7 is about implementation details of comparison methods used in this study. The results of the compared methods are discussed in Section 8 while concluding remarks are presented in Section 10.

3. Background

This section overviews relevant concepts often employed in constructing image generation pipelines. This unified pipeline consists of three main components; scene graphs, graph convolutional networks, and generative adversarial networks, as shown in Fig. 2.

3.1. Scene graphs

A *scene graph* is a graph data structure that encapsulates information related to objects and their relationships in a scene. It was initially proposed for the image retrieval task to search images containing particular semantic contents (Johnson et al., 2015). As illustrated in Fig. 1, a complete scene graph includes objects and relationships and represents the semantics of the scene. Scene graphs are powerful enough to encode the 2D (Johnson et al., 2015) and 3D (Armeni et al., 2019) representations of images into their semantic elements without having any constraints on object types, attributes, and relationships.

According to Johnson et al. (2018), a scene graph data structure $\mathscr{G}$ contains a set of object categories $\mathscr{C}$ and relationship categories $\mathscr{R}$. The scene graph $\mathscr{G}$ can be defined as a tuple consisting of (O, E), where $O = \{o_1, \ldots, o_n\}$ is a set of objects, which may be, for example (See Fig. 1), persons (man, boy), places (patio), things (frisbee), and other parts (arms, legs) with each $o_i \in \mathscr{C}$. E is a set of directed edges $E \subseteq O \times \mathscr{R} \times O$, which are the relationships between objects, i.e., geometry (boy on the patio) and actions (man throwing frisbee) in the form of (o_i, r, o_j) where $o_i, o_j \in O$ and $r \in \mathscr{R}$.

3.2. Graph convolutional networks

Graphs are commonly used to represent the relationships between data points in a vector space ($\boldsymbol{x} \in \mathbb{R}^n$), where $x(i)$ is the

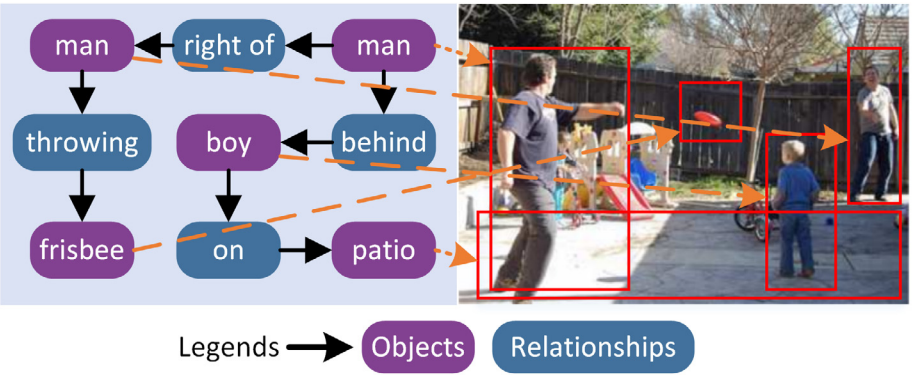

Fig. 1. The scene graph on the left contains objects (in blue) and their relationships (in purple). The objects of the scene graph refer to regions of the image on the right.

value of the signal of node i. The graph neural network (GNN) was initially proposed to process data representation in a graph domain (Scarselli et al., 2008), which can handle various types of graphs, such as cyclic, acyclic, directed, and undirected. A special type of GNN, called the graph convolutional network (GCN), uses convolutional aggregation. The convolutional layers in a GCN function similarly to the traditional 2D convolutional layers in a convolutional neural network (CNN).

In a GCN, a graph G is defined as a pair (V, E), where V denotes the set of nodes, and E represents the set of edges. A spatial grid of feature vectors is used as input, and a new spatial grid of vectors is produced through convolutional aggregates, where the *weights* are shared across all neighborhoods. For all objects $o_i \in O$ and edges $(o_i, r, o_j) \in E$, the input vectors v_i and v_r for each node and edge, respectively, are given as $\mathbb{R}^{\mathbb{D}}_{\text{in}}$, and the output vectors v'_i and v'_r for all nodes and edges are $\mathbb{R}^{\mathbb{D}}_{\text{out}}$.

In the typical pipeline for scene graph to image generation, scene graphs are processed end-to-end using GCNs, which comprise multiple graph convolutional layers. At each node V and edge E, the input graph with dimension vector D_{in} computes the new dimension vectors D_{out} for each incoming node and edge. The graph convolution theory applied in GCN can be represented as v_i and $v_r, \in \mathbb{R}^{\mathbb{D}}_{\text{in}}$, representing the given input vectors with objects $o_i \in O$ and edges $(o_i, r, o_j) \in E$. The output vector in Eq. 1 can be computed for all nodes and edges using three functions g_s, g_p, and g_o, which take in triple vectors v_i, v_r, and v_j, respectively. The three functions g_s, g_p, and g_o are implemented using a single network by concatenating the three input vectors, which are then fed to a multilayer perceptron (MLP). This results in new vectors for the subject o_i, predicate r, and object o_j. Therefore, the formulation can be stated as follows:

$$v'_i, v'_r \in \mathbb{R}^{\mathbb{D}}_{\text{out}} \tag{1}$$

$$v'_r = g_p(v_i, v_r, v_j) \tag{2}$$

where v'_i and v'_r are the output vectors for object o_i, and v'_γ is the output vector for edges presented in Eq. 2. A candidate vector for edges starting at o_i is computed by collecting all candidates in a set V^S_i, and for all edges terminating at o_j, a candidate vector is computed by V^o_i. All methods use a similar GCN formulation for scene graph to image generation reported in this study.

3.3. Generative adversarial networks

A generative adversarial network (GAN) is a deep neural network proposed by Goodfellow et al. (2014) to solve the generative modeling problem. The GAN consists of two adversarial models, i.e., generator and discriminator. The generator network G receives a collection of training examples as input and learns the probability distribution to produce data, whereas the discriminator network D distinguishes between real and fake data. Both networks (generator and discriminator) are trained against each other in a min–max game strategy, in which D divides the input into two classes (real or fake), and G tries to fool the discriminator. The most typical loss function for training a GAN is defined as follows:

$$\begin{aligned} &\min_G \max_D V(D,G) \\ &= E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \end{aligned} \tag{3}$$

where a min–max strategy is played between generator G and discriminator D, V is the value function, $p_{\text{data}}(x)$ is the actual data distribution drawn from data x, $p_z(z)$ is the input noise variable, $D(x)$ is the probability of x where D is trained to maximize the probability of assigning correct labels to training examples and samples whereas G is trained to minimize $\log(1 - D(G(z)))$.

4. Literature review

As of today, few attempts have been made regarding image generation from scene graphs and layouts (Johnson et al., 2018; Li et al., 2019; Zhao et al., 2019b; Zhao et al., 2020). That makes image generation from scene graphs and layouts a bespoke area to work on. This section provides an overview of related work concerning image generation from text, scene graphs, and layouts. The goal of this section is to review image synthesis from different modalities.

4.1. GANs for image synthesis

Since the inception of GANs, image synthesis has become crucial for many real-world applications that generate synthetic data to represent different entities. Usually, a generator synthesizes an image, and a discriminator differentiates between a real and a fake image. Promising results have been achieved using GANs in different fields, such as image generation (Johnson et al., 2018; Isola et al., 2017), video prediction (Vondrick et al., 2016), texture synthesis (Zhao et al., 2021), natural language processing (Li et al., 2018), and image style transfer (Karras et al., 2019). GANs can apply the conditions on category labels by providing the category labels as an additional input to the generator G and discriminator D resulting in conditional image synthesis (Gauthier, 2014; Mirza and Osindero, 2014). However, the discriminator can also be forced to predict the labels (Odena et al., 2017). Mainly, the GANs are used in image synthesis, which can produce better synthetic images than previous state-of-the-art approaches (Salimans et al., 2016; Mao et al., 2019).

Although the state-of-the-art performance of GANs is visible on the front end of image generation, the training process of GANs is often unstable. To overcome this, Wassertein GAN (Arjovsky et al., 2017) that is an alternative of traditional GAN (Goodfellow et al., 2014) was proposed for improving the learning stability and getting rid of model collapsing issues during the training process. Another approach, Unrolled GAN (Metz et al., 2016), was also proposed to address the same problem. Wasserstein GAN (Arjovsky et al., 2017) uses the Wasserstein distance metric to improve the learning stability, and Unrolled GAN (Metz et al., 2016) addresses the same problem by unrolling the optimization process. Image generation requires high resolution and high fidelity images. For these reasons, Progressive GAN (Karras et al., 2017) and BigGAN (Brock et al., 2018) are also a good fit, where Progressive GAN keeps adding new layers starting from a low-resolution until a high-resolution fine detail is achieved, and BigGAN is trained on a very large scale on ImageNet to achieve the high-resolution fine details of images. In summary, Progressive GAN (Karras et al., 2017) generates high-resolution images by adding new layers progressively, and BigGAN (Brock et al., 2018) is trained on a large scale to achieve high-resolution fine details in the generated images.

4.2. Image synthesis from scene graphs

Image generation based on scene graphs is a challenging task that requires a substantial effort to produce recognizable objects in complex scenes. Most of the methods proposed for image synthesis from scene graphs rely on the use of GCN. One of the first attempts in this area considered the creation of an image retrieval framework based on a scene graph formulation (Johnson et al., 2015). Later, Johnson et al. (Johnson et al., 2018) proposed sg2im, the pioneer method for image generation using scene graphs. The sg2im method aims to solve the challenges of image generation faced by natural language/textual description methods regarding semantic entity information.

The image generation from scene graphs is also referred to as conditional image generation, in which the expected image is conditioned on some additional information. The seminal work of Johnson et al. (2018) focused on scene graphs that contain information of multiple objects in the foreground. They used a GCN which passes the scene graph information along its graph edges. A scene layout is constructed by predicting the bounding boxes and segmentation masks, and finally, a cascaded refinement network (CRN) (Chen and Koltun, 2017) is used to convert the predicted layout into the expected image. Previous approaches typically involve encoding the scene graph into a vector representation and then decoding the vector to generate an image having several drawbacks, such as the loss of spatial information and the inability to handle complex relationships between objects. To address these issues, the authors propose a new model that generates images directly from the scene graph.

To incorporate spatial information into the model, the authors (Johnson et al., 2018) introduce a new type of layer called a spatial feature transform (SFT) layer. This layer uses the spatial positions of objects in the scene graph to transform the feature maps generated by the generator. The SFT layer enables the model to generate images that accurately reflect the spatial relationships between objects in the scene.

Basically, the GCNs are of two types: (i) Spectral GCN and (ii) Spatial GCN. The former one was proposed by Henaff et al. (2015) to construct a deep architecture with a slight learning complexity by incorporating a graph estimation procedure for the classification problem. The latter is built upon classic CNN and propagation models Zhang et al. (2019). The classic CNNs are extended to spatial GCNs by mapping the graph data into structure-aware convolution operations in both Euclidean and non-Euclidean spaces. Li et al. (2019) proposed a method that uses an external object crop that acts as an anchor to control the generation task. Their proposed method is different from Johnson et al. (2018) in three ways. First, external object crops are used; secondly, they used a Crop Refining Network to convert layouts masks into images; third, a Crop Selector is introduced to choose the most-suitable crops from the objects database automatically.

An interactive approach to generate images from scene graphs using recurrent neural networks by preserving image content and modifying cumulative images was proposed by Mittal et al. (Mittal et al., 2019). The method works in three stages, with increasing levels of complexity. At each stage, more nodes and edges are added to the scene graph to give more information to the GCN to generate layout. They used a Scene Layout Network (SLN) on top of the architecture proposed in Johnson et al. (2018) that generates the layouts for predicting the bounding boxes. Their proposed method utilizes the GCN and adversarial image translation method to generate images in an unsupervised manner. The generated images still need improvements, such as the images are blurry and the objects are not generated according to the input scene graphs. Another work Tripathi et al. (2019) also improved upon (Johnson et al., 2018) by introducing scene context network. That work added a context-aware loss for a higher image matching and introduced two new metrics for measuring the compliance of generated images with scene graphs: (i) relation score and (ii) mean opinion relation score.

GCN-based methods present some limitations, i.e., the GCN sometimes gets confused over relations among attributes, and finding the correct relation is also laborious. For example, (Man, right, Woman) and (Woman, left, Man) are always true, but it will typically result in different illustrations for most cases. Herzig et al. (2020) proposed a canonical representation based on a method for scene graph to image generation that respects the relations of attributes by keeping the graph's information in a canonicalization process.

4.3. Image synthesis from layouts

Scene layouts are the intermediate states when generating images from scene graphs. However, Zhao et al. (2019b) proposed an explicit framework of directly generating images from layouts without the need to define scene graphs manually. The bounding boxes and object categories are specified at the beginning. Then, a diverse set of images is generated based on the defined coarse layouts. They also made an extension (Zhao et al., 2020) to their already proposed work (Zhao et al., 2019b) by explicitly defining the loss functions and extending the object feature map module by adding the object-wise attention to their proposed framework.

Sylvain et al. (2020) proposed an object-centric method to generate images from layouts. However, their proposed method incorporates scene graph-based retrieval to increase the fidelity of layouts. Therefore, their proposed method is a hybrid of scene graph-based and layout-based image generation mechanisms. They proposed an Object-Centric GAN (OC-GAN), which integrates the scene graphs similarity module to learn the spatial representations of the objects in a scene layout. One of the limitations of their method is that a distant look of images generated by most SL2I generation methods appears to be adhering to the input layouts and look realistic. However, a closer inspection of these images reveals that there is a lack of context awareness and location sensitiveness. To overcome these limitations, a final work to date is proposed by He et al. (2021) by introducing the context-aware feature transformation module. The generated features are updated for each object in their proposed method while computing the

Gram matrix for feature maps to capture the inter-feature correlations, which respect the location sensitiveness of objects.

5. Methods comparison

This section highlights the methodological descriptions of SG2I and SL2I-based methods used in this analysis. We selected four methods that are built upon the pioneer work of SG2I method Johnson et al. (2018) and have identical input and training data.

Creating complex scene images from real-world objects requires a high-level understanding of computer vision and computer graphics. The goal of image generation from scene graphs is to take a *scene graph* as an input and generate a realistic image corresponding to the described objects and their relationships in a graph.

We propose following the typical pipeline to train all SG2I and SL2I generation methods for synthesizing images. For a scene-graph-based image generation method, the proposed framework should be able to move from the graph domain to the image domain. The typical workflow employed for scene-graph-based image generation is illustrated in Fig. 2 where a GCN-based graph is constructed based on the input scene graph. The coarse 2D structure of scene layouts is predicted using an object layout network based on the embedding vectors. Finally, an adversarial network generates the output image.

As a pioneer of scene graph to image generation works, Johnson et al. (2018) proposed to generate images from scene graph using GCNs and a cascaded refinement network (CRN) Chen and Koltun (2017). The GCN processes the input graphs to generate a scene layout. The generated scene layout is based on the prediction of bounding boxes and segmentation masks of objects. In the final stage, the CRN generates the output image based on the predicted scene layout. During the generation of images, Johnson et al. (2018) experienced three primary challenges: (i) the development of a method that can process graph-structured inputs; (ii) the generated images must comply with the objects and the relationship between objects specified in a graph; and (iii) the images generated using GCN and CRN must be realistic.

5.1. sg2im

This section describes in detail the sg2im method. The image generation network f takes the input scene graph G and noise z to generate the output image $I = f(G, z)$. The processing of G takes place along with a GCN, which generates the embedding vectors for each object. The embedding vectors of GCN respect the relationships between objects in a scene graph by predicting the bounding box and segmentation masks for each object in a scene graph. After this step, a layout is generated, which acts as an intermediate element between a scene graph and an output image. The generation of output image I is based on the CRN, and the realistic images based on the scene graph are generated by adversarial training of network f against a pair of network discriminators D_{img} and D_{obj}. The D_{img} encourages image I to appear realistic while D_{obj} contains the information of realistic and recognizable objects. Each node and edge of the input scene graph is converted to a dense vector from a categorical label through a learned embedding layer.

Cascaded Refinement Network. To synthesize an image with respect to a given layout, it is necessary to respect the object positions available in the given layout. A cascaded refinement network (CRN) introduced works on this pattern and consists of a series of convolutional refinement modules. In CRN, the modules are concatenated to each other channel-wise and are passed to a pair of 3×3 convolutional layers. Each module receives the inputs from the scene layout, which is down-sampled to a specific input resolution and the output of the previous module.

5.2. PasteGAN

To achieve a more robust control of the image generation process in a more fine-grained manner, a crop selection-based strategy, named PasteGAN, was developed by Li et al. (2019). They made a threefold contribution: (i) the objects of scene graph work as crops that use external object crops bank to guide image generation process; (ii) to better generate an image, a Crop Refining Network and an Object-Image Fuser were designed with the goal of making object crops appear in a fine-grained way in the final image generation process; and (iii) to automatically select the most compatible crop, a Crop Selector module is also devised in PasteGAN. Basically, the PasteGAN encodes the input scene graph and object crops to generate the corresponding images.

The PasteGAN mainly used the external memory tank to find objects given in the input scene graphs to generate images. The training process of PasteGAN consists of two stages in which, the first stage aims to reconstruct the ground-truth images using original crops m_i^{ori}, whereas the second stage focuses on generating images with selected images crops m_i^{sel} from the external memory tank.

The scene graph is processed with GCN to obtain a latent vector z containing contextual information of each entity. The PasteGAN processes scene graphs using GCN and then a crop selector selects the good crop for objects that are relevant for generating realistic images. The good crop selector should be able to recognize not only the accurate objects, but it should also match with the similarity of scenes. The PasteGAN used the pretrained sg2im based GCN to process scene graphs.

The crop refining network adopted by PasteGAN is based on two steps: (i) a crop encoder, which aims at extracting main visual features of objects, and (ii) an object refiner, consisting of two 2D graph convolutional layers. It fuses the visual appearance of

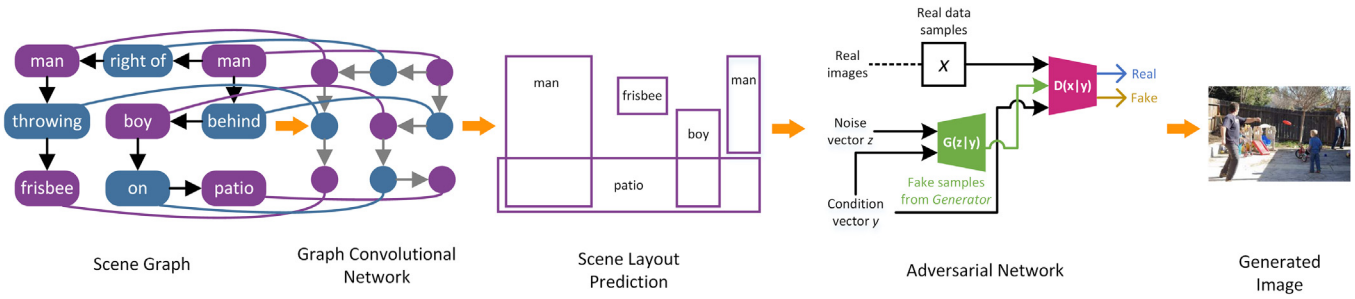

Fig. 2. Typical pipeline employed for scene graph and layout to image generation. A *scene graph* is given as input and a *graph convolutional network* processes it to convert into an image domain. This step produces scene layout predictions. Finally, a conditional GAN further synthesizes the image as final output.

cropped object series. Furthermore, an object-image fuser is used to combine all object crops into a latent space-based canvas.

An image decoder based on a cascaded refinement network is used to take the input of latent canvas and generate an image by respecting the object positions in latent scene canvas. After this, a discriminator based on a pair of image discriminators and object discriminators generates realistic and recognizable images by training the image generation network adversarially.

5.3. WSGC

In SG2I, typically, there are two steps involved; the first step is to generate a layout based on the input scene graphs, and in the second step, the pixels are generated out of the layout as a final image. The transformation of layouts to images relies on geometric properties (for example (man, right of, boy). Since humans primarily generate the scene graphs, there always remains the possibility of error, such as there may not always be all correct relations in the data. We can elaborate on this with Fig. 3. It can be seen that the scene graphs (man, right of, boy) and (boy, left of, man) both are semantically equivalent, but existing SG2I methods do not consider them semantically equivalent, which is the limitation of previous SG2I based methods.

In order to overcome the semantic equivalence difficulty, Herzig et al. (2020) proposed a canonical scene graph-based image generation method by replacing traditional scene graphs. Logically equivalent conventional scene graphs are replaced with canonical scene graphs that are used to generate the layouts. The advantage of their method Herzig et al. (2020) is its capacity of learning more compact models by distributing the information across the graph with only a few parameters. By using the canonicalization process, the robustness and noise of graphs can be improved.

The scene graph canonicalization is performed in two different stages. At first, the scene graph canonicalization is calculated by assuming the transitive relations and converse relations as an instance of inference. Secondly, a weighted scene graph canonicalization is calculated based on an exactly weighted scene graph canonicalization (WSGC-E) and a sampling weighted scene graph canonicalization (WSGC-S). However, a scene graph to image canonicalization is carried in two steps. Initially, a layout is predicted using a weighted scene graph. The weighted scene graph used the GCN to predict the layout bounding boxes. WSGC used the same methodology of generating images from layouts as proposed in Sun and Wu (2019); Zhao et al. (2019b). The work of Zhao et al. (2019b) is extended as layout2image Zhao et al. (2020) by proposing new loss functions and is used to generate images from layouts, but an extension to Zhao et al. (2019b) was proposed by introducing the CLEVR dataset where attributes of objects can be specified.

5.4. Layout2im

As described earlier, Zhao et al. (Zhao et al., 2019b) proposed a method to generate images from layouts using the word embedding methodology. Following a similar strategy, layout2image (Zhao et al., 2020) was proposed by defining the new loss functions explicitly. An extension to the object feature map composition module is also added. Thanks to the sg2im method, which is used as a core model for differentiable bilinear cropping of images, these crops are fed to object discriminator in the image generation process of layout2image. However, layout2image (Zhao et al., 2020) is similar to Zhao et al. (2019b) in many aspects. The image generation training phases from layout are divided into seven parts:

1. The object-latent code estimations are first calculated from ground-truth images to sample-out object latent codes.
2. A latent object code is then sampled to construct a feature map using the object feature map composition strategy. This step produces object feature maps by simplifying the regions with their bounding boxes.
3. An object-wise attention module is added to object feature map composition to implicitly alleviate the need to add different classes of objects in fusion layers.
4. A resulting image is decoded after the fusion module which is added after an object feature map.
5. An image decoder is tasked to generate images from the hidden feature maps. Furthermore, an explicit object latent code regression is introduced to encourage the consistency between latent codes and outputs.
6. Finally, an image and object discriminator, similar to the sg2im method (Section: sg2im), is introduced for classifying an input image or object as real or fake.

6. Loss functions

This section highlights the loss function used in the comparison method of this comparative analysis. We briefly explain all relevant loss functions used in scene graph and scene layout based models.

The network f of the sg2im method is jointly trained using two discriminators D_{obj} and D_{img}. The image generation network f is trained by minimizing the weighted sum of six loss functions which are as follows:

1. Box Loss, calculated by penalizing the L_1 difference between ground-truth and predicted bounding boxes.
2. Mask Loss, calculated by finding the difference between ground-truth mask and the predicted mask.
3. Pixel Loss, calculated by finding the L_1 difference between ground-truth and generated images.

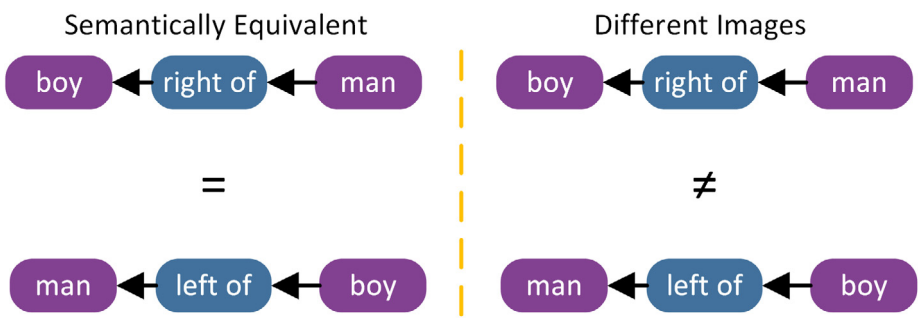

Fig. 3. Limitation of existing SG2I methods.

4. Image adversarial loss, which encodes if the generated image patches to appear realistic.
5. Object Adversarial Loss, which used for ensuring that the generated objects to appear realistic.
6. Auxiliary Classifier Loss, which ensures that object discriminators should classify all the generated objects in an image.

PasteGAN generator is based on the minimized weighted sum of six loss functions that were available in the provided source code; however, the reported study has mentioned to use eight loss functions in their proposed method.

1. Image Reconstruction Loss, which finds the L_1 difference between ground-truth image and reconstructed image.
2. Crop Matching Loss, which calculates the L_1 difference between object crop feature maps and re-extracted object feature maps from generated images.
3. Adversarial Loss, which is similar to object adversarial loss in sg2im (Johnson et al., 2018), i.e., its use aims to ensure that the objects to appear realistic.
4. Auxiliar Classifier Loss, which is used to ensure that object discriminators should classify all the generated objects in an image.
5. Perceptual Loss, which calculates the L_1 difference between ground-truth images and reconstructed images in the global feature space.
6. Box Regression Loss, which is used to calculate L_1 difference between ground truth and the prediction boxes.

There is only one loss function used in this method which is based on the single scene graph and its ground-truth layout.

1. L_1 Loss, which is used for minimizing the error between the difference of ground truth mask and the predicted mask.

Different from other SG2I approaches, this method introduces two loss functions. For example, a KL (Kullback–Leibler) loss function is proposed to compute the KL-divergence between a distribution and normal distribution, and an object latent code reconstruction loss strengthens the connection of specific object appearance and latent codes to be invertible. However, other loss functions, such as image reconstruction loss, object adversarial loss, auxiliary classification loss, and adversarial image losses, are the same as introduced in the sg2im method Johnson et al. (2018).

7. Experimental details

The methods which we used for comparison in this study are selected based on three keywords, i.e., (i) scene graph, (ii) layout generation, and (iii) image generation. This section provides in depth details of experimental setup employed to perform the analysis.

7.1. Datasets

The experiments in all four methods are mainly performed on Visual Genome (VG) and COCO-Stuff datasets. We used the same settings and the datasets for the comparison methods. Both datasets contain the varying size of images. For a fair evaluation of all methods, we resize all images with size 64×64. Table 1 shows the attributes of the datasets used for methods comparison.

Visual Genome. This dataset is composed of 108,077 scene graph annotated images with seven main components such as objects, attributes, relationships, scene graphs, region descriptions, region graphs, and question–answer (QA) pairs. Each image consists of an average of 35 objects, and the relationships between two objects can be actions, e.g., *jumping over, wear, behind, drive on* etc. and 26 attributes, e.g., color (red), states (sitting/standing), etc. The scene graphs in this dataset are the localized representations of an image and are combined to construct an entire image. The region descriptions are the natural descriptions in a sentence format to describe a region of the scene. The objects, attributes, and relationships are combined through a directed graph to construct region graphs in the VG dataset. Furthermore, two types of QA pairs are associated with each image: (i) Freeform QA and (ii) Region-based QA. The dataset is preprocessed at the beginning and then divided into training (80%), validation (10%), and test (10%) sets.

Table 1
Statistics of Visual Genome and COCO-Stuff datasets.

Dataset	Visual Genome	COCO-Stuff
Training set	62,565	24,972
Validation set	5506	1024
Test set	5088	2048
Total number of objects	178	171
No. of objects in an image	3–30	3–8
Min. number of relationships between objects	1	6

COCO-Stuff. This dataset consists of 164 K complex scene images Caesar et al. (2018). It contains 172 classes comprising 80 things, 91 stuff, and 1 class as *unlabeled*. An expert annotator curated the 91 stuff classes. Additionally, the class *unlabeled* is used in two scenarios. First, when the label is not listed in any of 171 predefined classes, and second when the annotator is unable to infer the pixel label. However, this dataset contains 40 K training images and 5 K validation images from scene graphs and layouts for the image generation task. Dense pixel-level annotations in COCO-Stuff are augmented from the COCO (Lin et al., 2014) dataset.

After a thorough study of preceding SG2I and SL2I methods, we come to the evaluation of these methods. The evaluation protocols and implementation details of all methods used for comparative analysis are defined in the evaluation metrics and implementation details sections, respectively.

7.2. Evaluation metrics

The image quality of all generated images by the four methods needs to be quantitatively measured, and for this reason, different image quality evaluation metrics have been used by different methods. Usually, the Inception score is the primary image quality measure that uses ImageNet to encourage recognizable objects within the generated images. This analysis reported four main evaluation metrics for a comparison of all methods, namely: Inception Score (IS), Frechet Inception Distance (FID), Diversity Score (DS), and Classification Accuracy (CA).

Inception Score: The inception score (↑) is implemented for the evaluation of the generated image quality, specifically for synthetic images, the higher the value is, the better the inception score is (Salimans et al., 2016). It involves using a pre-trained deep neural network for the classification of generated images. It has two main objectives: (i) image quality, are the generated images look like a specific object?, and (ii) image diversity, are the generated objects lie in a wide range?. A pre-trained VGGNet Simonyan and Zisserman (2014) is used to implement and compute the IS for all the methods in this analysis.

Fréchet Inception Distance: FID was proposed by Heusel et al. (2017), and is a metric that is used to embed a set of generated images into feature space which is given by a special layer of the

Inception network. The lower value of FID (↓) represents the higher quality of images generated by the generator compared to the real ones.

Diversity Score: This metric is used in the deep feature space to compute the perceptual similarity between two images. The DS is different from IS in the sense that it measures the difference between generated images and real images from the same input. The higher the metric value is, the better the DS (↑) is.

Classification Accuracy: This is a measure to quantify the capacity to create identifiable objects, a crucial criterion for evaluating the SG2I and SL2I works. We initially train a ResNet-101 object classification model He et al. (2016). This is accomplished by using the actual objects cropped and downsized from ground truth images inside the training set of each dataset. Afterwards, we calculate and report the object classification precision for the generated images. The higher value of CA (↑) represents the best score is achieved.

7.3. Implementation details

We implemented the identical parameters for all four methods in this analysis using python version 3.5, PyTorch 0.4, and Linux (Ubuntu) 20.04. With the updates of all libraries, we used a virtual environment for a fair comparison. The training learning rate was set at 10^{-4} and a batch size of 32, 16, 8 and 16 for all methods, respectively. Table 2 highlights the hyperparameter details of all the methods used in this work. All scene graphs are augmented with a special image object, and a specific in image relationship is connected to each true object in the SG2I and SL2I methods, through which all scene graphs are connected.

To generate the images of size 64×64 on the VG and COCO-Stuff datasets, we used RTX 3090 GPU, and it took days to finish training with a million iterations on each dataset. The scene graphs are available in a human-readable format in a JSON file. After installing the GraphViz library, it is also possible to visualize the input scene graph as a graph. The program used the Pytorch library.

8. Experimental results

Table 3 shows the performance of comparison methods on four metrics, i.e., IS, FID, DS, and CA. Each dataset is split into 3 groups and we report mean and standard deviation for IS, and DS for all methods. The samples are generated on a full model with image size 64×64 by defining different synthetic scene graphs. The models' abilities are evaluated through generating complex scenes. The best IS and DS is achieved by Layout2im Zhao et al. (2019b), and the best FID is achieved by PasteGAN Li et al. (2019).

The methods reported in this study used different loss functions, which we have reported in earlier methodological descriptions. However, for the sake of fair comparison, we evaluated the generator and discriminator loss functions of all methods, which are the two main components of a GAN. Fig. 4 illustrates the comparison of two loss functions for 90 epochs, where we report the loss for every ten epochs. Fig. 4a is the representation of generator loss, where we can see that the loss for sg2im (Johnson et al., 2018) and PasteGAN (Li et al., 2019) is relatively lower than WSGC (Herzig et al., 2020) and Layout2im (Zhao et al., 2019b). In the case of discriminator loss (Fig. 4b), all four methods performed comparatively better. However, we can observe that WSGC Herzig et al. (2020) performed best in the case of generator loss on the COCO dataset, whereas Layout2im Zhao et al. (2019b) outperformed in the case of discriminator loss on all datasets. The object classification accuracy is also reported for all four comparison methods. It can be observed that the objected generated by the SL2I method Zhao et al. (2019b) can be accurately classified for real images. It is also observed during the experimentation that the object classification's upper bound limit does not necessarily confirm the difficulty of distinguishing the generated images.

Fig. 5 shows the statistics of bounding box predictions for all methods on two datasets. R@t is used to predict the accuracy of predicted bounding boxes, and from the experiments, it can be seen that using the SL2I mechanism improves the prediction performance. Fig. 5a presents the R@3 while Fig. 5b is the demonstration of R@5.

Fig. 6 shows the generated images using comparison methods. From the set of images, it can be seen that Layout2im Zhao et al. (2019b) performed relatively better than other scene graph-based methods in the qualitative evaluation. According to their input labels, the objects are more recognizable and consistent. The images generated by SG2I-based methods also respect the compositionality of objects. However, the blurriness and object overlapping is still a major problem of object synthesis in these methods.

9. Discussions

9.1. Limitations

Since GANs are a powerful class of deep learning models widely used in image generation methods. While this work also leverages GANs-based SG2I and SL2I generation methodology. There are several limitations associated with both using GANs and scene graphs to synthesize images.

For example, GANs can sometimes suffer from mode collapse, where the generator network produces limited types of output that fail to represent the full diversity of the target distribution. This happens when the generator network produces similar or identical outputs for different input values, resulting in a loss of variety in the generated images (Wang et al., 2020). GANs can be challenging to train, and their training can be unstable. The generator and discriminator networks can get stuck in a suboptimal state, resulting in poor-quality output images. GANs require careful tuning of their hyperparameters, such as learning rates, batch sizes, and regularization terms. The choice of these hyperparameters can significantly impact the quality of the generated images (Meshry, 2022). GANs require large datasets for training, and the quality of the generated images can depend on the quality and quantity of the training data. GAN-based image synthesis methods often cannot provide fine-grained control over the generated images. For example, it may be challenging to generate images with specific attributes or to generate images that match a given textual

Table 2
Hyperparameter details.

Methods	Dataset	Input Size	BS	LR	Ep.	Iterations	Training Time
sg2im Johnson et al. (2018)	VG, COCO-Stuff	64×64	32	$1e^{-4}$	90	1,000,000	5 Days/dataset
PasteGAN Li et al. (2019)	VG, COCO-Stuff	64×64	16	$1e^{-4}$	90	1,000,000	5 Days/dataset
WSGC Herzig et al. (2020)	VG, COCO-Stuff	64×64	8	$1e^{-4}$	90	1,000,000	10 Days/dataset
Layout2im Zhao et al. (2019b)	VG. COCO-Stuff	64×64	16	$1e^{-4}$	90	300,000	3 Days/dataset

Table 3
Performance evaluation of IS, FID, DS, and CA of all methods on two datasets.

Methods	IS ↑		FID ↓		DS ↑		CA ↑	
	VG	COCO	VG	COCO	VG	COCO	VG	COCO
sg2im Johnson et al. (2018)	5.3 ± 0.1	6.1 ± 0.2	65.13	74.42	0.07 ± 0.05	0.02 ± 0.01	42.31	38.52
PasteGAN Li et al. (2019)	6.5 ± 0.1	7.1 ± 0.1	32.43	40.35	0.05 ± 0.02	0.04 ± 0.01	44.52	45.96
WSGC Herzig et al. (2020)	7.3 ± 0.1	5.1 ± 0.2	55.31	62.23	0.12 ± 0.09	0.07 ± 0.03	48.74	49.63
Layout2im Zhao et al. (2019b)	7.9 ± 0.2	8.7 ± 0.3	39.68	44.19	0.17 ± 0.08	0.13 ± 0.05	51.85	50.36

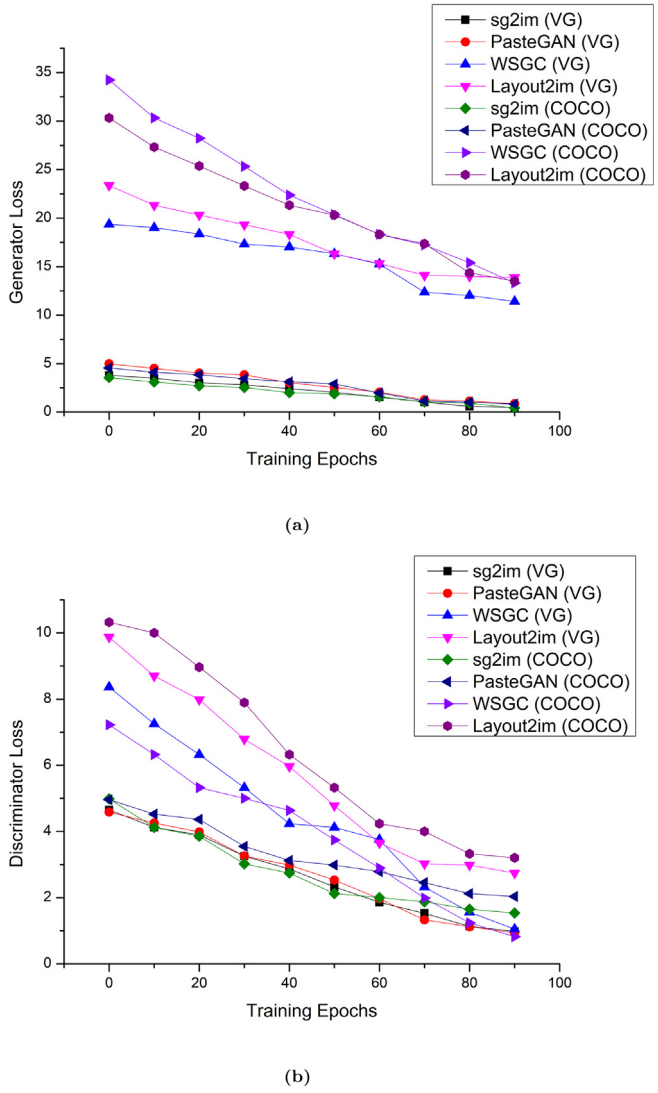

Fig. 4. Performance evaluation of comparative methods based on their training **(a)** generator loss and **(b)** discriminator loss.

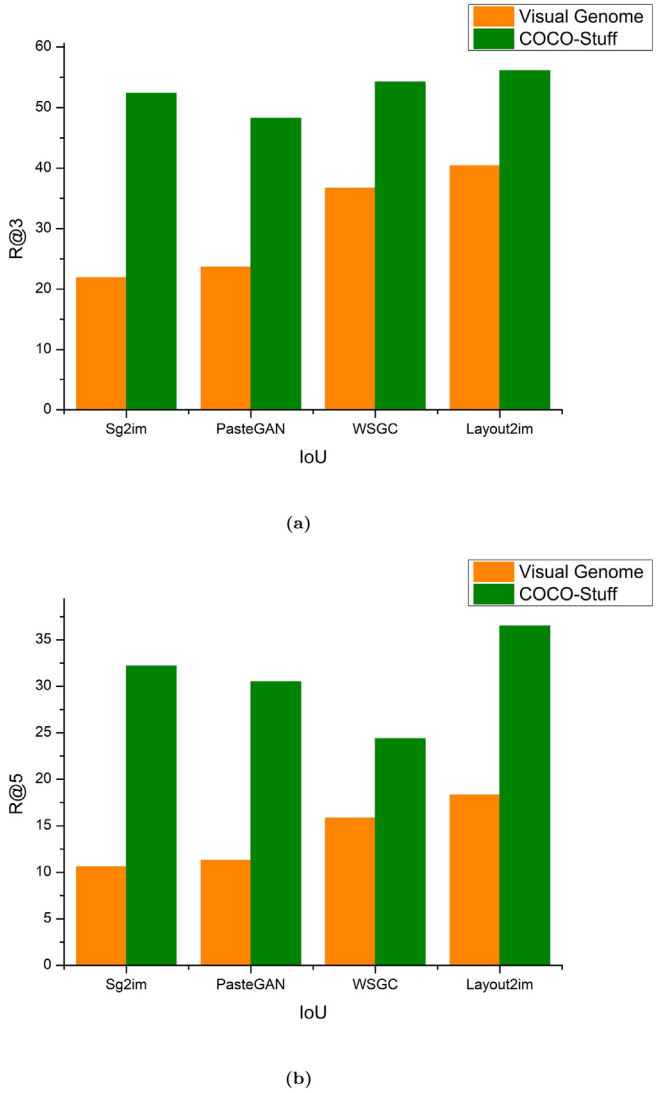

Fig. 5. Intersection over union comparison of all four methods on two datasets. **(a)** R@3 and **(b)** R@5.

description. GANs are often limited in their ability to generate complex scenes with multiple objects or large-scale contexts and can struggle to maintain spatial coherence and realistic object interactions. Overall, while GAN-based image synthesis methods have made significant progress, they still have limitations that must be addressed to enable their wider adoption and enhance their ability to generate high-quality and diverse images.

Scene graph-based methods can suffer from limited diversity, where the generated images tend to follow a small set of prototypical visual layouts, which can make them less visually interesting or less representative of the full range of possible scenes. These methods are often computationally intensive and require large amounts of memory to process and manipulate complex graph structures (Zhu et al., 2022). Similar to GAN-based methods, scene graph-based methods require large amounts of training data to achieve good results. SG2I and SL2I methods struggle to scale to more complex and diverse scenes. They rely on explicit modeling of object relationships and interactions, which can become computationally expensive as the number of objects and relationships increases. Although scene graphs can include textual information about objects and their relationships, incorporating more detailed textual descriptions or natural language instructions into SG2I and SL2I methods can be challenging. Like GAN-based methods, scene graph-based methods can lack fine-grained control over the

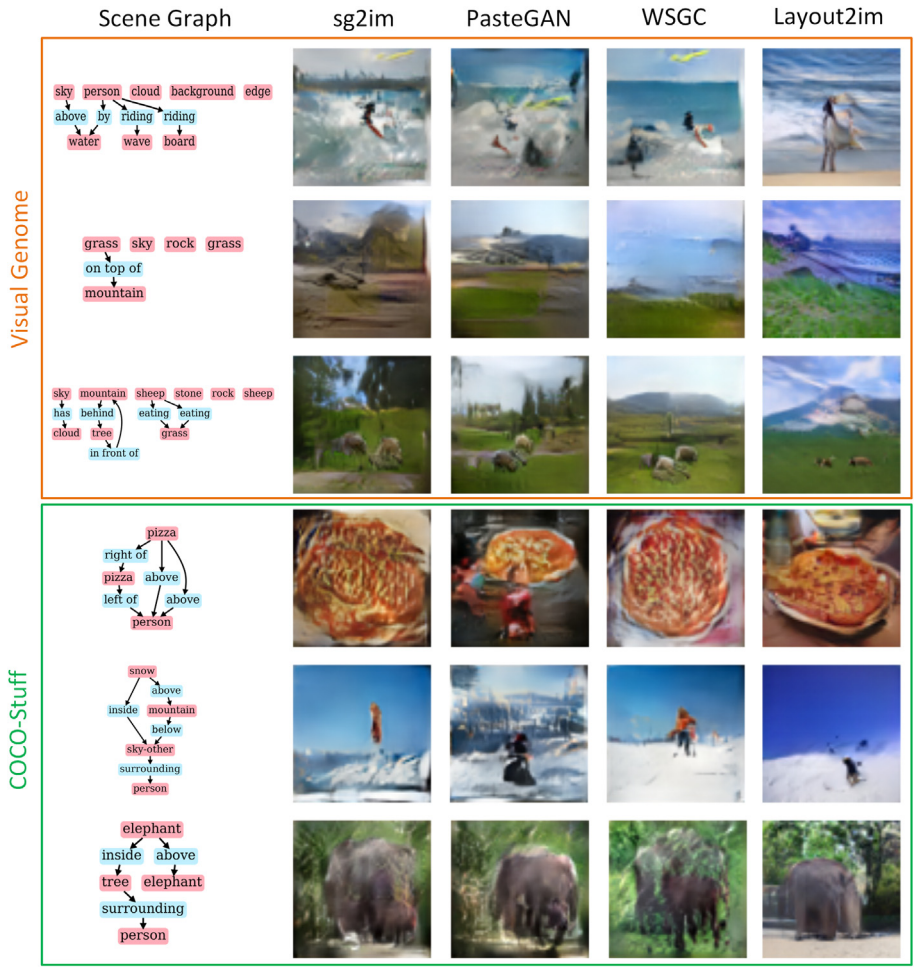

Fig. 6. A comparison of images generated by SG2I and SL2I based methods. The images of size 64 × 64 are generated using the same setting for all methods. (Orange) shows the generated results on VG dataset while (Green) illustrate the images generated on COCO-Stuff dataset.

generated images, such as generating images with specific attributes or modifying certain aspects of the generated scenes. In summary, while scene graph-based methods offer a promising direction for image synthesis, challenges remain to be addressed, such as improving scalability and diversity and finding ways to incorporate textual information better and provide more fine-grained control over the generated images.

9.2. Future directions

In this section, we discuss some potential future research directions for SG2I and SL2I generation methods. Scene graph-based image generation methods have recently gained popularity in the computer vision community for their ability to generate realistic and diverse images by leveraging the rich semantic information encoded in scene graphs.

Improving the quality and diversity of generated images is crucial for SG2I and SL2I generation methods. Although SG2I methods have achieved impressive results, there is still room for improvement in terms of the quality and diversity of generated images. Future research can explore novel techniques to enhance the realism and diversity of synthesized images, such as incorporating attention mechanisms (Kitada and Iyatomi, 2022), adversarial training, and semantic consistency regularization.

Exploring multi-modal and multi-task image generation is another potential research direction. SG2I generation can be extended to generate images with multiple modalities or to perform multiple tasks simultaneously. For example, a model could generate an image and its corresponding textual description or generate images with different styles or viewpoints. The researchers can investigate these multi-modal and multi-task scenarios to create more versatile and flexible image generation models.

Incorporating real-world constraints for generating high-quality images is intrinsic for SG2I and SL2I generation methods. For instance, in real-world scenarios, synthesized images must adhere to certain constraints, such as object occlusions, lighting conditions, and camera angles. There are possibilities to explore how to incorporate these constraints into the image synthesis process to generate more realistic images that better reflect the complexities of real-world scenes.

Time consumption is a major issue in synthesizing images from scene graphs. It happens due to the fine-tuning of larger parameters and complex hierarchical structures. To enhance the acceleration of SG2I and SL2I generation methods, deep neural networks can be constructed based on extreme learning machine (ELM) theory (Zhang et al., 2020). The method is faster and easy to implement, involving two stages: (i) randomly generating hidden layer parameters from a predefined specific interval and (ii) calculating the generalized inverse of the output weight matrix. The acceleration of SG2I and SL2I-based methods can significantly be increased by incorporating the ELM theory to fine-tune parameters.

One of the critical challenges for SG2I and SL2I methods is adapting to new domains and modalities. SG2I and SL2I-based models have mainly been applied to 2D images, but they can also

be extended to other domains, such as 3D scene synthesis or video synthesis. Future research can investigate how to adapt scene graph-based image generation methods to these new domains and modalities. The generation of scene graph-based 3D objects and scenes can be extended to 3D digital twins for smart cities creation in a virtual space. In order to create real-life scenarios, a user can provide the textual description, and a scene or object can be generated for use in smart city applications and further use in authoring tools (Hassan et al., 2022).

Finally, developing interpretable and explainable models is essential. Scene graph-based image generation models are typically black boxes that are difficult to interpret and explain. Future research can focus on developing more interpretable and explainable models, which can facilitate their use in real-world applications where interpretability and transparency are crucial.

10. Conclusion

A study on the analysis of different scene graphs and layout to image generation methods is presented in this work. The SL2I method showed state-of-the-art performance on all evaluation measures due to predefining the compositionality of object layouts during the training process. Our experiments of SG2I and SL2I generation models suggest that the image generation pipeline from scene graphs is still lagging in image quality, and a good layout prediction is necessary before generating an image from a predicted layout. All reported methods need at least three defined objects to generate an image. Moreover, experiments suggest that the complex scenes are tricky for current image generation frameworks because current state-of-the-art methods ignore much information of objects while synthesizing objects. There is a need to improve the overall visual quality of objects for scene graph-based image generation networks. The generation of high-resolution images is still in significant demand for image generation models, so introducing new loss functions and adding more experiments to current state-of-the-art methods can significantly improve this area.

Data Availability Statement

Data sharing not applicable.

CRediT authorship contribution statement

Muhammad Umair Hassan: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Saleh Alaliyat:** Data curation, Writing – review & editing, Supervision. **Ibrahim A. Hameed:** Conceptualization, Investigation, Supervision, Project administration, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We thank the anonymous reviewers for their valuable feedback, which improved the quality of this work. We also thank the Norwegian University of Science and Technology (NTNU), Norway, for providing the open access funding to publish this work.

References

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D., 2015. Vqa: Visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433.

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein generative adversarial networks. In: International Conference on Machine Learning, PMLR, pp. 214–223.

Armeni, I., He, Z.-Y., Gwak, J., Zamir, A.R., Fischer, M., Malik, J., Savarese, S., 2019. 3d scene graph: A structure for unified semantics, 3d space, and camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5664–5673.

Brock, A., Donahue, J., Simonyan, K., 2018. Large scale gan training for high fidelity natural image synthesis, arXiv preprint arXiv:1809.11096.

Caesar, H., Uijlings, J., Ferrari, V., 2018. Coco-stuff: Thing and stuff classes in context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1209–1218.

Chen, Q., Koltun, V., 20107. Photographic image synthesis with cascaded refinement networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1511–1520.

Dhamo, H., Farshad, A., Laina, I., Navab, N., Hager, G.D., Tombari, F., Rupprecht, C., 2020. Semantic image manipulation using scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5213–5222.

Gao, L., Wang, B., Wang, W., 2018. Image captioning with scene-graph based semantic concepts. In: Proceedings of the 2018 10th International Conference on Machine Learning and Computing, pp. 225–229.

Gauthier, J., 2014. Conditional generative adversarial nets for convolutional face generation, Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition. Winter semester 2014 (5), 2.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, Adv. Neural Informat. Process. Syst. 27.

Hassan, M.U., Angelaki, S., Alfaro, C.V.L., Major, P., Styve, A., Alaliyat, S.A.-A., Hameed, I.A., Besenecker, U., da Silva Torres, R., 2022. Digital twins for lighting analysis: Literature review, challenges, and research opportunities. In: 36th International ECMS Conference on Modelling and Simulation, ECMS 2022, vol. 36, pp. 226–235.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

S. He, W. Liao, M.Y. Yang, Y. Yang, Y.-Z. Song, B. Rosenhahn, T. Xiang, Context-aware layout to image generation with enhanced object appearance, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15049–15058.

Henaff, M., Bruna, J., LeCun, Y., 2015. Deep convolutional networks on graph-structured data, arXiv preprint arXiv:1506.05163.

Herzig, R., Bar, A., Xu, H., Chechik, G., Darrell, T., Globerson, A., 2020. Learning canonical representations for scene graph to image generation. In: European Conference on Computer Vision. Springer, pp. 210–227.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Adv. Neural Informat. Process. Syst. 30.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M., Fei-Fei, L., 2015. Image retrieval using scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3668–3678.

Johnson, J., Gupta, A., Fei-Fei, L., 2018. Image generation from scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1219–1228.

Karras, T., Aila, T., Laine, S., Lehtinen, J., 2017. Progressive growing of gans for improved quality, stability, and variation, arXiv preprint arXiv:1710.10196.

Karras, T., Laine, S., Aila, T., 2019. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410.

Kim, U.-H., Park, J.-M., Song, T.-J., Kim, J.-H., 2019. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. IEEE Trans. Cybernet. 50 (12), 4921–4933.

Kitada, S., Iyatomi, H., 2022. Making attention mechanisms more robust and interpretable with virtual adversarial training. Appl. Intell., 1–16

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D.A., et al., 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. 123 (1), 32–73.

Li, C., Su, Y., Liu, W., 2018. Text-to-text generative adversarial networks. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–7.

Li, Y., Ma, T., Bai, Y., Duan, N., Wei, S., Wang, X., 2019. Pastegan: A semi-parametric method to generate image from scene graph. Adv. Neural Informat. Process. Syst. 32, 3948–3958.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: European Conference on Computer Vision. Springer, pp. 740–755.

Mao, Q., Lee, H.-Y., Tseng, H.-Y., Ma, S., Yang, M.-H., 2019. Mode seeking generative adversarial networks for diverse image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1429–1437.
Meshry, M., 2022. Neural rendering techniques for photo-realistic image generation and novel view synthesis, Ph.D. thesis.
Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J., 2016. Unrolled generative adversarial networks, arXiv preprint arXiv:1611.02163.
Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.
Mittal, G., Agrawal, S., Agarwal, A., Mehta, S., Marwah, T., 2019. Interactive image generation using scene graphs, arXiv preprint arXiv:1905.03743.
Odena, A., Olah, C., Shlens, J., 2017. Conditional image synthesis with auxiliary classifier gans. In: International Conference on Machine Learning, PMLR, pp. 2642–2651.
Reed, S.E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H., 2016. Learning what and where to draw. Adv. Neural Informat. Process. Syst. 29, 217–225.
Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H., 2016. Generative adversarial text to image synthesis. In: International Conference on Machine Learning, PMLR, pp. 1060–1069.
Reed, S., Oord, A., Kalchbrenner, N., Colmenarejo, S.G., Wang, Z., Chen, Y., Belov, D., Freitas, N., 2017. Parallel multiscale autoregressive density estimation. In: International Conference on Machine Learning, PMLR, pp. 2912–2921.
Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training gans. Adv. Neural Informat. Process. Syst. 29, 2234–2242.
Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2008. The graph neural network model. IEEE Trans. Neural Networks 20 (1), 61–80.
Schuster, S., Krishna, R., Chang, A., Fei-Fei, L., Manning, C.D., 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In: Proceedings of the Fourth Workshop on Vision and Language, pp. 70–80.
Shamsolmoali, P., Zareapoor, M., Granger, E., Zhou, H., Wang, R., Celebi, M.E., Yang, J., 2021. Image synthesis with adversarial networks: A comprehensive survey and case studies. Informat. Fusion.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
Sun, W., Wu, T., 2019. Image synthesis from reconfigurable layout and style. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10531–10540.
Sylvain, T., Zhang, P., Bengio, Y., Hjelm, R.D., Sharma, S., 2020. Object-centric image generation from layouts, arXiv preprint arXiv:2003.07449 1 (2), 4.
Tripathi, S., Bhiwandiwalla, A., Bastidas, A., Tang, H., 2019. Using scene graph context to improve image generation, arXiv preprint arXiv:1901.03762.
Vondrick, C., Pirsiavash, H., Torralba, A., 2016. Generating videos with scene dynamics. Adv. Neural Informat. Process. Syst. 29, 613–621.
Wang, Y., Gonzalez-Garcia, A., Berga, D., Herranz, L., Khan, F.S., Weijer, J.V.D., 2020. Minegan: effective knowledge transfer from gans to target domains with few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9332–9341.
Yang, X., Tang, K., Zhang, H., Cai, J., 2019. Auto-encoding scene graphs for image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10685–10694.
Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N., 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5907–5915.
Zhang, S., Tong, H., Xu, J., Maciejewski, R., 2019. Graph convolutional networks: a comprehensive review. Comput. Social Networks 6 (1), 1–23.
Zhang, C., Chao, W.-L., Xuan, D., 2019a. An empirical study on leveraging scene graphs for visual question answering, arXiv preprint arXiv:1907.12133.
Zhang, J., Li, Y., Xiao, W., Zhang, Z., 2020. Non-iterative and fast deep learning: Multilayer extreme learning machines. J. Franklin Inst. 357 (13), 8925–8955.
Zhao, B., Meng, L., Yin, W., Sigal, L., 2019b. Image generation from layout. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8584–8593.
Zhao, B., Yin, W., Meng, L., Sigal, L., 2020. Layout2image: Image generation from layout. Int. J. Comput. Vision 128.
Zhao, X., Wang, L., Guo, J., Yang, B., Zheng, J., Li, F., 2021. Solid texture synthesis using generative adversarial networks, arXiv preprint arXiv:2102.03973.
Zhao, S., Li, L., Peng, H., 2022. Aligned visual semantic scene graph for image captioning. Displays 74, 102210.
Zhou, R., Jiang, C., Xu, Q., 2021. A survey on generative adversarial network-based text-to-image synthesis. Neurocomputing 451, 316–336.
Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., Shen, P., Feng, M., Zhao, X., Miao, Q., Shah, S.A.A. et al., 2021. Scene graph generation: A comprehensive survey, arXiv preprint arXiv:2201.00443.

Muhammad Umair Hassan is a PhD candidate at the Norwegian University of Science and Technology, Ålesund, Norway. He obtained his master's from the University of Jinan, P.R. China, and a bachelor's from the University of the Punjab, Pakistan. His research interests include Computer Vision, Computer Graphics and Deep Learning. He has a soundtrack of publications in multidisciplinary areas, including journals and conference publications.

Saleh Alaliyat is currently an Associate Professor at the Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Ålesund, Norway. His research interests include Artificial Intelligence, Swarm Intelligence, and Computer Vision.

Ibrahim A. Hameed has a PhD in AI from Korea University, South Korea and PhD in field robotics from Aarhus University, Denmark. He is a Professor and Deputy Head of Research and Innovation within NTNU, IEEE senior member, elected chair of the IEEE Computational Intelligence Society (CIS) Norway section, Founder and Head of Social Robots Lab in Ålesund. His current research interests include Artificial Intelligence, Machine Learning, Optimization, and Robotics.