
TRANSFORMER-BASED IMAGE GENERATION FROM SCENE GRAPHS

 **Renato Sortino**
PeRCeVe Lab
University of Catania
Catania, Italy
renato.sortino@phd.unict.it

 **Simone Palazzo**
PeRCeVe Lab
University of Catania
Catania, Italy
simone.palazzo@unict.it

 **Concetto Spampinato**
PeRCeVe Lab
University of Catania
Catania, Italy
concetto.spampinato@unict.it

ABSTRACT

Graph-structured scene descriptions can be efficiently used in generative models to control the composition of the generated image. Previous approaches are based on the combination of graph convolutional networks and adversarial methods for layout prediction and image generation, respectively. In this work, we show how employing multi-head attention to encode the graph information, as well as using a transformer-based model in the latent space for image generation can improve the quality of the sampled data, without the need to employ adversarial models with the subsequent advantage in terms of training stability.

The proposed approach, specifically, is entirely based on transformer architectures both for *encoding* scene graphs into intermediate object layouts and for *decoding* these layouts into images, passing through a lower dimensional space learned by a vector-quantized variational autoencoder. Our approach shows an improved image quality with respect to state-of-the-art methods as well as a higher degree of diversity among multiple generations from the same scene graph. We evaluate our approach on three public datasets: Visual Genome, COCO, and CLEVR. We achieve an Inception Score of 13.7 and 12.8, and an FID of 52.3 and 60.3, on COCO and Visual Genome, respectively. We perform ablation studies on our contributions to assess the impact of each component. Code is available at <https://github.com/perceivelab/trf-sg2im>

Keywords scene graphs · transformers · generative models · conditional image generation

1 Introduction

Conditional image generation has recently shown promising results in generating high-resolution images, especially in recent formulations involving text-to-image approaches with transformers [Ramesh et al., 2021, Crowson et al., 2022] or diffusion models [Ramesh et al., 2022, Saharia et al., 2022, Rombach et al., 2022]. The evolution of the generative model architectures has significantly contributed to make conditional image generation possible. Generative Adversarial Networks (GAN) [Goodfellow et al., 2014] have quickly evolved in less than a decade, reaching high sample quality through architectural improvements [Karras et al., 2020, Zhu et al., 2017, Kim et al., 2017] and conditioning information [Mirza and Osindero, 2014, Reed et al., 2016, Isola et al., 2017]. Recent approaches, particularly the ones based on transformers [Vaswani et al., 2017] and diffusion models [Ho et al., 2020], achieve even better results and avoid the training instability typical of adversarial approaches, but at the cost of an increase in computing and data requirements.

While generating data from a random noise distribution, possibly partitioned for conditional generation, can have interesting applications, using human-readable input to control the generation process makes these approaches more suitable for addressing specific problems by limiting the generated samples to a particular sub-domain. Text, in particular, is a flexible modality that allows unlimited combinations of conditioning inputs, including complex and abstract concepts, and can potentially lead to the generation of high-quality and high-quantity images [Ho and Salimans]. However, using text as a conditioning signal comes with several drawbacks: natural language sentences may be long and loosely structured; semantics depend strongly on syntax and language; it inherently contains ambiguity as many

different sentences can express the same concept, which can lead to unstable training, especially with low model capacity. Thus, in contexts where description constraints are important — for instance, highly-specialized domains (e.g., medicine, astronomy, agriculture) — text representations of a specific scene might not suffice. *Scene graphs* [Johnson et al., 2018] offer a valid alternative to structure the description of visual data using graphical syntax, where nodes represent objects in the scene and edges model the spatial/semantic relationships between them. In particular, scene graphs provide a comprehensive and flexible way of describing objects and their relations in a scene. Nevertheless, a major challenge in employing scene graphs to control image generation lies in the complexity of matching visual and graph features. A common effective strategy that has been employed to overcome this limitation is to estimate, from the graph, an intermediate scene layout representation that, in turn, is used to guide the generation process [Johnson et al., 2018, Li et al., 2019]. However, the choice of using adversarial models for the image generation process, adopted by most existing approaches, may limit the potentiality of this conditional generation, as adversarial-based methods are characterized by unstable training.

To overcome this limitation, we propose a fully transformer-based approach for scene-graph-to-image generation. More specifically, we exploit the generalization capabilities of transformers in processing arbitrary graphs through multi-head attention mechanisms and in autoregressively composing long sequences. The devised model is shown in Fig. 1: we first predict a scene layout from scene graphs through the *SGTransformer* module, that computes attention on neighboring nodes of a graph and the respective edge features, and predicts the layout, a set of bounding boxes and corresponding labels. We then map the layout information to a discrete space, by aggregating the coordinates of each box and the respective object labels into triplets, and use this information to condition another transformer, *Image Transformer* (*ImT*) which learns the joint distribution between images and layouts. Generation in the Image Transformer is posed as a classification task on a sequence of image tokens and leverages the transformer decoder’s capability to generate long sequences without losing context information. The sequence of predicted tokens is finally translated into an image through a pre-trained vector quantized autoencoder, i.e., VQVAE [Van Den Oord et al., 2017], that learns to project data from the pixel space \mathcal{X} to a latent space \mathcal{Z} of quantized feature vectors, and vice versa.

We show the effectiveness and flexibility of our scene-graph-to-image approach on standard benchmarks such as Visual Genome (VG) and COCO, where we report visual examples showing its robustness to (both minor and major) perturbations in the input scene graph, while ensuring significant variability and diversity in the synthesized images. We also compare our method with current state-of-the-art approaches, yielding generally better qualitative and quantitative results in terms of both generated layouts and images. We also carry out extensive ablation studies that substantiate our design choices. We finally conclude by concretely showing how the strategy to explicitly generate intermediate layout representations yields better performance than carrying out cross-attention between encoder and decoder modules of the transformer, thus substantiating again our choice.

2 Related Work

Transformers as generative models. Transformers [Vaswani et al., 2017] have shown to perform better than convolutional approaches in several computer vision tasks, such as object detection [Carion et al., 2020, Fang et al., 2021] and classification [Dosovitskiy et al., 2020, Touvron et al., 2021]. Vision transformers have been employed for generative models as well, trained both in an adversarial [Jiang et al., 2021, Hudson and Zitnick, 2021, Arad Hudson and Zitnick, 2021] and a supervised way [Esser et al., 2021, Ramesh et al., 2021]. TransGAN [Jiang et al., 2021] employs two transformer encoders that are trained adversarially and act as generator and discriminator in traditional GANs. GANformer [Hudson and Zitnick, 2021] introduces a bipartite attention mechanism between the image and a latent noise vector used to start the generation, which helps to compute attention between latent features and image features and couple parts of the latent vector with image regions. GANformer2 [Arad Hudson and Zitnick, 2021] evolves this work by adding conditioning information to control the generation process. Transformers are better than CNNs at modeling long-range sequences, which makes them suitable for autoregressive generation tasks. In particular, if images can be represented as a sequence of integers, using codebooks of quantized image features, transformers can be efficiently used for generating images at high resolution, as shown in VQGAN [Esser et al., 2021]. This method has been improved to perform also text-to-image synthesis by integrating CLIP [Radford et al., 2021] and using it to guide the model towards generating images that match a text description.

Scene graph to image. Research related to the task of image generation conditioned on scene graphs has seen interesting developments since the launch of the pioneering work of SG2IM [Johnson et al., 2018]. This work demonstrated how image generation can effectively be conditioned on scene graphs with good results, employing a Graph Convolutional Network (GCN) to generate a scene layout, and then a Generative Adversarial Network conditioned on this layout to generate the image. Since SG2IM, other approaches improved this task in several ways. PasteGAN [Li et al., 2019] enhances the GAN by conditioning it on image crops that contain single objects corresponding to scene graph nodes. The work proposed by [Ivgi et al., 2021] builds on SG2IM by adding context information to the image generator.

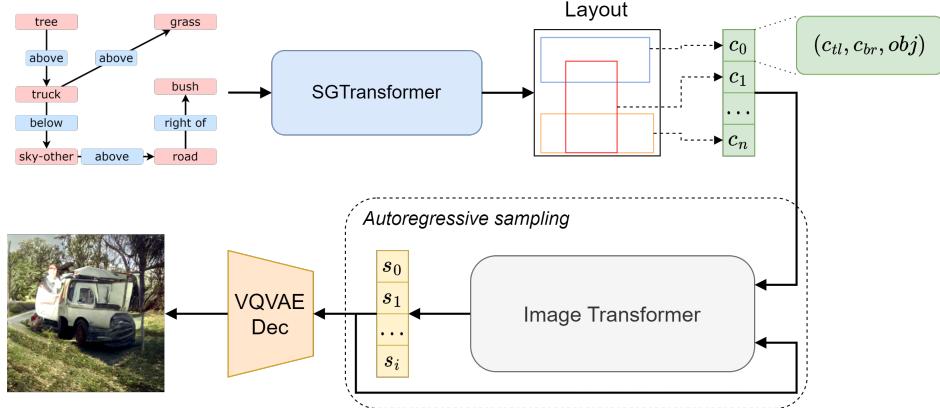


Figure 1: Overview of our framework. The scene graph is fed to our SGTransformer module, which updates the node feature vectors with multi-head attention on the neighboring nodes and the edge feature vectors to predict the layout. This is treated as a sequence of discrete elements c_i , which are triplets where the first two elements are bounding box coordinates, top left and bottom right, respectively, and the last one is the object category. The Image Transformer predicts a sequence of codebook indices s_i that are then reshaped and projected back onto the pixel space using a pre-trained VQVAE decoder.

CanonicalSG2IM [Herzig et al., 2020] adds transitive and converse relationships with a certain probability, to enrich the graph with additional information. Other methods use different approaches to improve the generation process, for instance by processing *subject-predicate-object* relations individually [Vo and Sugimoto, 2020], by posing the task as a meta-learning problem [Farshad et al., 2021], or by iteratively modifying the scene graph while retaining previously-generated content [Mittal et al., 2019].

All of these works employ a two-stage process to generate the image. Most of them generate a scene layout first, and then use it to condition a GAN generator. Our work follows the same two-stage pipeline, but adopts a fully transformer-based architecture, trained in a standard supervised way, thus avoiding the instability and mode collapse problems of adversarial training. This allows us to synthesize higher-quality samples compared to existing models, while increasing model flexibility, sensitivity to small graph changes, and diversity of generated samples.

3 Method

In this section, we discuss in detail the architectural elements that compose our framework. Our goal is to exploit the high capacity of transformer-based architectures to extract meaningful information from the scene graph, with a twofold objective: 1) to produce a plausible and coherent layout of the scene using information from the Laplacian matrix and the graph’s edge features; and 2) to exploit the capability of modeling long sequences to generate the output image as a sequence of codebook indices. We thus propose a two-stage approach, which first generates a layout from the scene graph and then conditions the image generation process on the predicted layout.

3.1 Scene graph to layout

Scene graphs describe the composition of an image by defining the relationships between its objects in the form of a directed graph $G(V, E)$. Objects are represented as graph nodes containing the object category $o \in \{0, \dots, N_o\}$, while edges encode the type of relationship $p \in \{0, \dots, N_p\}$ that occurs between two nodes, with N_o and N_p being the number of object categories and relationship types, respectively. Nodes and edges are projected into a higher-dimensional space, thus each node i is characterized by a feature vector $h_i \in \mathbb{R}^{d_h}$; for each node j in its neighborhood \mathcal{N}_i , the corresponding edge is described by a feature vector $e_{ij} \in \mathbb{R}^{d_e}$. State-of-the-art scene-graph-to-image approaches use Graph Convolutional Networks (GCNs) to encode the graph representation by updating each node’s feature vector with information from neighboring nodes and edges. Dwivedi and Bresson [2021] point out how GCNs, and particularly Graph Attention Networks [Veličković et al., 2017], can be interpreted as a generalized case of transformers, where the attention mechanism is computed only on neighboring nodes. Motivated by the high versatility of transformers in this context, we employ such an architecture to encode our scene graph. A first challenge in this formulation lies in the definition of a *positional encoding* that preserves the structure of the input. Typically, the positional encoding helps to maintain the relative ordering of the elements within the sequence. When dealing with graphs, the concept of “ordering”

of nodes has no meaning, as graphs are unordered by definition. However, the Laplacian matrix provides geometric information on the graph, and Dwivedi and Bresson [2021] show that it can be used as a positional encoding to enrich the encoded graph representation. We employ the architecture proposed by Dwivedi and Bresson [2021] as our scene graph encoder, which we call *SGTransformer*. An overview of this module’s architecture is shown in Fig. 2.

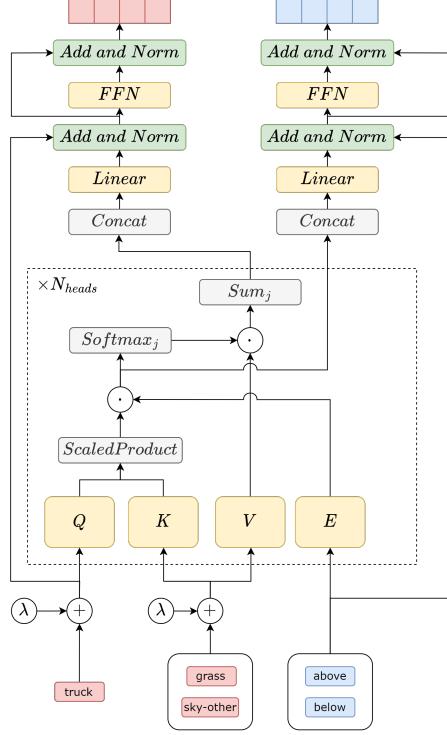


Figure 2: Detail of a SGTransformer layer. Each node is used as query, while its neighboring nodes j are mapped to key-value pairs. Edge features are multiplied before the softmax operations with the product of Q and K , and the output of this operation is used to update the edge features.

The graph Laplacian $\Delta \in \mathbb{R}^{n \times n}$ is defined, in its symmetrically normalized form, in the following way:

$$\Delta = I - D^{-1/2} A D^{-1/2} \quad (1)$$

where $D \in \mathbb{R}^{n \times n}$ represents the degree matrix and $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix. As a positional encoding, we use the l smallest non-trivial eigenvectors of Δ , which we indicate with λ , and randomly flip their sign during training, due to their multiplicity given by arbitrary sign. These eigenvectors are projected, via a learnable matrix, onto a space with the same dimensionality of the node embeddings and summed to them.

The SGTransformer architecture is composed of a stack of blocks, each of which computes multi-head attention between each node h_i and its neighbors $h_j \in \mathcal{N}_i$. This attention mechanism can be interpreted as a masked self-attention where, for each node, all the non-neighboring nodes are masked out. This reduces the complexity of the attention operation from $O(n^2)$ to $O(nm)$ where n is the number of nodes and m the maximum number of edges per node. In addition, when computing attention, we take into account the edge features e_{ij} , as they contain important information to describe the scene structure.

For each node h_i , the multi-head attention mechanism is defined as follows:

$$\text{Attention}(Q, K, V, E) = \text{softmax} \left(\frac{h_i W_k^Q \cdot h_j W_k^K}{\sqrt{d_k}} \right) \cdot h_j W_k^V \cdot e_{ij} W_k^E. \quad (2)$$

$W_k^Q, W_k^K, W_k^V, W_k^E \in \mathbb{R}^{d_k \times d}$ are learnable projection matrices, $k \in \{0, \dots, N_{\text{heads}}\}$, and $d_k = d/N_{\text{heads}}$. Details on the layer of the SGTransformer are shown in Figure 2.

The output of SGTransformer is then passed to two MLP heads that predict the bounding box and class label for each node vector. The set of bounding boxes and labels for a graph defines a layout. We train the SGTransformer module with a compounded layout loss \mathcal{L}_{layout} , defined as follows:

$$\mathcal{L}_{layout} = \mathcal{L}_{box} + \mathcal{L}_{label} + \mathcal{L}_{iou}, \quad (3)$$

where \mathcal{L}_{box} is a L_2 loss on the bounding box coordinates, \mathcal{L}_{label} is a cross-entropy on the labels, and \mathcal{L}_{iou} is a distance IoU loss [Zheng et al., 2020] on the bounding boxes.

3.2 Images as sequences of discrete tokens

There are several methods that perform image generation conditioned on a layout [Sun and Wu, 2019, Yang et al., 2022, Fan et al., 2022] and some of them have been employed as the second stage of scene-graph-to-image approaches. State-of-the-art approaches [Johnson et al., 2018, Herzig et al., 2020, Li et al., 2019] rely on adversarially trained networks for image generation. We propose a transformer-based image generator, which we refer to as *Image Transformer (ImT)*. The goal is to use the affirmed capability of transformers to efficiently process long sequences, widely explored in NLP, for image generation. To accomplish this, we need to express image data as a sequence of tokens in a similar way as is done for words in a sentence. Vector-Quantized Variational Autoencoders (VQVAE) [Van Den Oord et al., 2017] encode the input RGB image $x \in \mathbb{R}^{H \times W \times 3}$ into a latent representation $\hat{z} \in \mathbb{R}^{h \times w \times n_z}$, where h and w are the number of feature vectors along the H and W sizes of the input image, and n_z represents each vector's size. Each vector is then used to retrieve the nearest feature vector z_q from a codebook:

$$z_q = \left(\arg \min_{z_k \in \mathcal{Z}} \| \hat{z}_i - z_k \| \right) \in \mathbb{R}^{N \times n_z} \quad (4)$$

Here, z_k are the codebook vectors and z_i the i -th feature vector of the encoded input, with $i \in \{0, \dots, hw\}$. The VQVAE is trained to reconstruct \hat{x} from input data x while defining a meaningful codebook using a reconstruction loss \mathcal{L}_r , a quantization loss \mathcal{L}_q , and a commitment loss \mathcal{L}_c , defined as follows:

$$\mathcal{L}_{vqvae}(x) = \mathcal{L}_r(x) + \mathcal{L}_q(x) + \mathcal{L}_c(x) \quad (5)$$

$$\mathcal{L}_r(x) = \|x - \hat{x}\| \quad (6)$$

$$\mathcal{L}_q(x) = \|\text{sg}[\mathcal{E}(x)] - z_k\|_2^2 \quad (7)$$

$$\mathcal{L}_c(x) = \|\mathcal{E}(x) - \text{sg}[z_k]\|_2^2 \quad (8)$$

Here, sg refers to the stop-gradient operator. The quantization loss \mathcal{L}_q is used to minimize the distance between the encoded vectors and the codebook vectors, while the commitment loss \mathcal{L}_c is employed as a form of regularization of the latent space to prevent it from growing indefinitely.

The quantization operation causes high-frequency information loss in the reconstructed image. Therefore, in addition to these losses, we add a patch discriminator loss \mathcal{L}_{disc} and a perceptual loss \mathcal{L}_{VGG} , as suggested by Esser et al. [2021].

3.3 Layout to Image

Once we learned the feature codebook, we can use the VQVAE to efficiently encode the images from the pixel space \mathcal{X} to the latent space \mathcal{Z} and vice versa. To generate new samples, we need to estimate the prior distribution of the latent codebook vectors. The Image Transformer is an autoregressive transformer that models the conditional distribution $p(s_i | s_{<i})$, where s represents the codebook indices (image tokens) and i is the step within the iterative process. It follows the architecture of GPT [Brown et al., 2020], employing multiple masked self-attention blocks to process the input sequence. The transformer attends only the previous blocks in the sequence by masking out the next tokens in the sequence, $\{s_j\}_{j>i}$. We condition the generation on the layout predicted from the SGTransformer. To this purpose, we encode the layout by mapping each object into a triplet of the form $o = (c_y, tl_y, br_y)$, as done by [Jahn et al., 2021]. Here, c_y represents the class index, tl_y and br_y the bounding box coordinates on a diagonal, top left and bottom right, respectively. The sequence of encoded layout objects $c \in \{0, \dots, N_o\}$ is prepended to the sequence of image tokens, and self-attention is computed on the entire sequence, combining information from the layout and the generated indices. Thus, the transformer learns to model the joint distribution of the condition c and the sequence of tokens s : $p(s_i | s_{<i}, c)$.

The Image Transformer learns this prior by optimizing the following log-likelihood:

$$\mathcal{L}_{\text{Transformer}} = \mathbb{E}_{x,c \sim p(x,c)} [-\log p(s | c)] \quad (9)$$

4 Experimental Results

We hereby present the experimental protocol used to evaluate our model’s performance and compare it to the state of the art, describing in details the employed datasets, the training procedure and our results, both quantitatively and qualitatively.

4.1 Dataset

For comparability with state-of-the-art methods, we trained and evaluated our model on three public image datasets that provide annotations in the form of object-to-object relations, which can be expressed as scene graphs: Visual Genome [Krishna et al., 2017], COCO-Stuff [Caesar et al., 2018], and CLEVR-Dialog [Kottur et al., 2019].

Visual Genome (VG) contains 108,077 images, with annotations in JSON format specifying bounding boxes, object categories, attributes, and relationships. We filter the annotations and generate the scene graphs as done by [Johnson et al., 2018], keeping object and relationship instances that occur at least 2000 and 500 times respectively. This results in 178 objects categories and 45 relationship types, describing actions and spatial relationships. The COCO-Stuff dataset extends COCO [Lin et al., 2014] by adding 91 object categories with bounding box annotations. This augmented dataset has a total of 45,000 images and 171 object categories. Scene graphs are generated following [Johnson et al., 2018], computing the relative positions of a subset of all the possible bounding box pairs and inferring the spatial relation among the contained objects and generating a total of 6 relationship types: *above*, *below*, *left of*, *right of*, *inside*, and *surrounding*. For VG and COCO-Stuff, we use the same train/test split defined by [Johnson et al., 2018]. CLEVR-Dialog [Kottur et al., 2019] is a procedurally generated dataset with associated scene descriptions, expressed as question-answers pairs describing the scene. It includes a total of 70,000 images for the training set and 10,000 in the validation set. Scene graphs from CLEVR Dialog have been generated following the procedure shown in [Herzig et al., 2020]. We normalize to zero mean and unitary standard deviation (per channel) and resize the images to 128×128 for all datasets.

4.2 Training procedure

We hereby describe the training procedure we followed to lead our framework to convergence.

The latent codebook contains $K = 8192$ vectors, each of which has a dimensionality n_z of 256. The VQVAE encoder compresses each image with a compression factor of $f = 8$, meaning that the input image of size $H \times W$ is compressed into a feature map of dimension $h \times w$, where $h = \frac{H}{f}$ and $w = \frac{W}{f}$. We chose this value based on the analysis of the compression factors reported by [Rombach et al., 2022], as it shows the best compromise between compression rate and information loss.

The SGTransformer is composed of 12 transformer layers, each with 12 attention heads, and an embedding size of 768. For the Laplacian positional encoding, we use the first $l = 8$ non-trivial eigenvectors and, as the number of nodes N_o in a scene graph is variable and we have at most N_o eigenvectors, in the case where $N_o < l$, we pad the eigenvector matrix with $l - N_o$ zero vectors.

The Image Transformer is a stack of 40 layers, with an embedding size of 1408 and 16 attention heads. Image data shows a higher correlation between neighboring pixels with respect to further ones. As the VQVAE is CNN-based, we maintain this correlation in the latent space. To exploit such a correlation, we employ a convolutional attention mask, as shown in Fig. 3, which attends only the previous elements of the sequence that fall within a 2D kernel. We use a kernel of size 7. Additional high-level information on scene objects is provided by the layout encoding obtained from the SGTransformer output.

We train our model for 300 epochs with a learning rate of 10^{-4} , using a One Cycle learning rate scheduler [Smith and Topin, 2019] and the Adam optimizer [Kingma and Ba, 2014].

4.3 Results

We evaluate our approach on the following tasks: 1) *scene graph to layout*, where we evaluate the performance of our SGTransformer module and compare with state-of-the-art models; 2) *layout to image*, evaluating the quality of

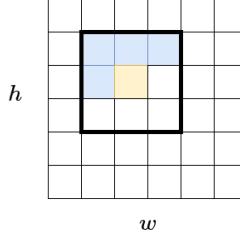


Figure 3: Representation of the decoder self-attention mask on the latent space. Elements in blue are attended by the element in yellow.

the generated images feeding the decoder with layouts predicted by the SGTransformer and ground-truth ones; 3) *scene graph to image*, assessing the performance of the model when tested end-to-end. We compare our approach with the following state-of-the-art methods: SG2IM [Johnson et al., 2018], which is the first approach performing image generation from scene graph; CanonicalSG2IM [Herzig et al., 2020], which enhances graph relationships by learning converse and transitive relationships; and PasteGAN [Li et al., 2019]

We use FID [Heusel et al., 2017] and Inception Score (IS) [Salimans et al., 2016] to compare the performance of our model with the state-of-the-art methods that employ intermediate layout generation for controlling image generation. In Table 1 we report a comparison, in terms of image generation performance, with state-of-the-art models evaluated on Visual Genome and COCO. We computed FID and IS when image generation is controlled either by the intermediate generated layout or by the ground-truth layout (indicated with * in Table 1). On COCO, our method outperforms previous approaches in terms of both FID and IS, while on VG it reaches state-of-the-art performance in terms of Inception Score, but not FID. This can be explained by the lower capacity of the VQVAE which, compressing the image, cuts high-frequencies in the output images, thus yielding a higher FID score. A better Inception Score, instead, suggests a higher visual appearance diversity among the generated images, which translates into a higher capacity of our model to model the joint graph/image distribution. These considerations can be also appreciated when comparing qualitatively our model against existing methods. Fig 4 shows how our model yields higher fidelity in the image details with respect to state-of-the-art methods. The VQVAE, trained using a perceptual loss and a discriminator loss, contributes to retrieve high-frequency components, resulting in a sharper image. Fig. 5, instead, shows the capability of our *Image Transformer* in synthesizing different images, given the same layout. The generated samples show a significant high degree of diversity among them (varying in color, lighting and object shapes), while being coherent with the input layout. Its capability to synthesize diverse samples, given the same scene graph, is even more evident when comparing the generated samples with those of existing methods.

We also assess qualitatively how the proposed approach is able to cope with perturbations in the input scene graph. Examples are reported in Fig. 6: from the first input graph we replaced "truck" with the "stop sign" and "grass" with "river", and the model inferred correctly to replace the truck in the scene with a stop sign in front of a river. These examples show the flexibility by our model in handling graph changes.

Table 1: Results in terms of image quality evaluated on COCO-Stuff and Visual Genome datasets. * refers to using ground-truth layouts as input to the Image Transformer. In bold best performance, in italic second best performance.

Method	Inception Score (\uparrow)		FID (\downarrow)	
	COCO	VG	COCO	VG
Real images	23.0 ± 0.4	22.8 ± 1.7	-	-
SG2IM	9.4 ± 0.2	8.3 ± 0.3	81.6	73.4
CanonicalSG2IM	10.8 ± 0.5	10.0 ± 0.7	73.8	46.4
PasteGAN	12.5 ± 0.8	8.5 ± 0.9	70.5	62.8
Ours	13.7 ± 0.9	12.8 ± 0.7	52.3	60.3
SG2IM*	11.4 ± 0.5	10.8 ± 0.4	74.2	62.1
CanonicalSG2IM*	15.6 ± 0.5	11.7 ± 0.8	54.7	36.4
PasteGAN*	14.3 ± 0.7	10.9 ± 0.6	59.7	50.9
Ours*	17.3 ± 0.9	15.5 ± 0.6	44.6	42.6

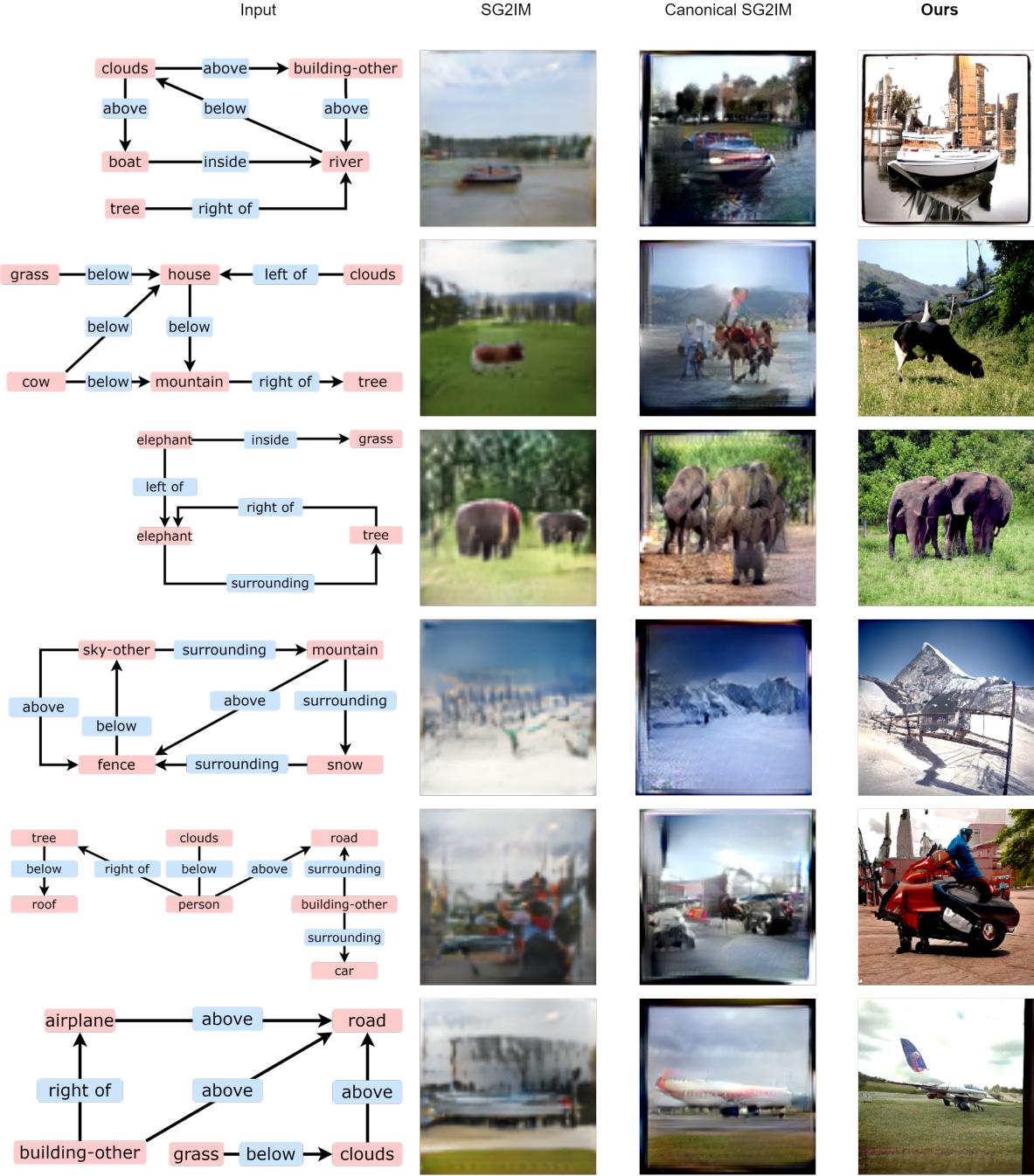


Figure 4: Visual comparison between our approach and SG2IM [Johnson et al., 2018] and CanonicalSG2IM [Herzig et al., 2020] on the COCO dataset

We further evaluate the capability of our *SGTransformer* to generate meaningful layouts (which then control the image generation process). Thus, we measure the mean Intersection over Union (mIoU) between the predicted layout and the ground truth layout associated to the input scene graph (available in the COCO and VG datasets). Table 2 reports the comparison between our model and state-of-the-art approaches in terms of accuracy on the generate layout, showing that the *SGTransformer* outperforms state-of-the-art models on the scene-graph-to-layout task. This can be explained with a better capabilities of *SGTransformer* in learning the geometry of the input scene graph to better encode scene information, resulting in a more coherent layout, as also shown in Fig. 6.

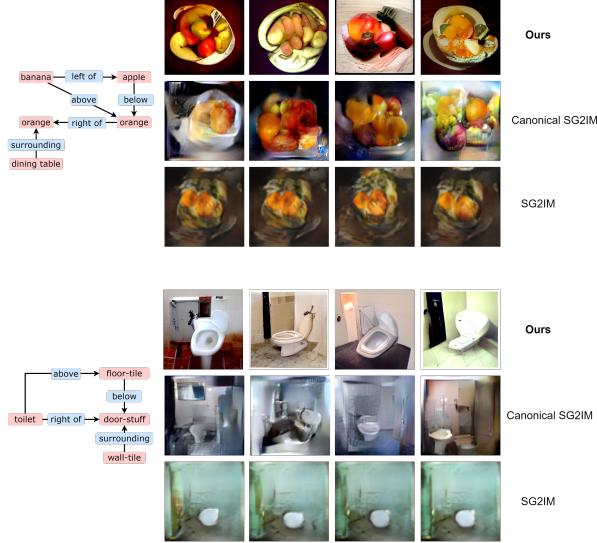


Figure 5: Diversity of several images sampled by the Image Transformer using the same layout. We compare our method with [Herzig et al., 2020] and [Johnson et al., 2018]

Table 2: Mean IoU on the bounding boxes predicted from the scene graph.

Model	COCO	VG
SG2IM	41.7	16.9
WSGC	41.9	18.0
Ours	42.0	20.6

5 Ablation study

The core components of our proposed approach rely on the multi-head attention mechanism to encode the graph nodes and to conditionally generate the image. We perform ablation studies on the components that constitute our main contribution in this work: the SGTransformer module and the Image Transformer. We first evaluate the impact of edge features E in the graph self-attention and the Laplacian positional encoding on our *SGTransformer*. Table 3 reports the results in terms of mIoU of the predicted layout with respect to the ground truth. It shows that taking edge features into account when computing attention in the graph is crucial to predict a more structured layout, as edge features contain useful information on the type of relations among the nodes. Adding the Laplacian positional encoding contributes to enrich the knowledge encoded by the transformer with the implicit geometric information of the graph.

Table 3: Ablation study on the SGTransformer architecture, analyzing the contribution of the self-attention using the edge features as in Equation and the Laplacian positional encoding. Results are expressed in mIoU.

Model	COCO	CLEVR	VG
SGTransformer	21.24	35.48	9.45
+ E	39.33	37.26	14.62
+ <i>LapPE</i>	42.03	40.63	20.63

We then evaluate the contribution of the chosen attention strategy in our *Image Transformer* to correctly predict autoregressively the sequence of codebook indices, which are then used for image synthesis. We specifically test two architectural variants of the *Image Transformer*:

- *CrossAtt-ImT* that uses latent scene graph representation, learned by our *SGTransformer*, for cross-attention with the sequence predicted by the decoder. This follows the original decoder architecture proposed in [Vaswani et al., 2017].

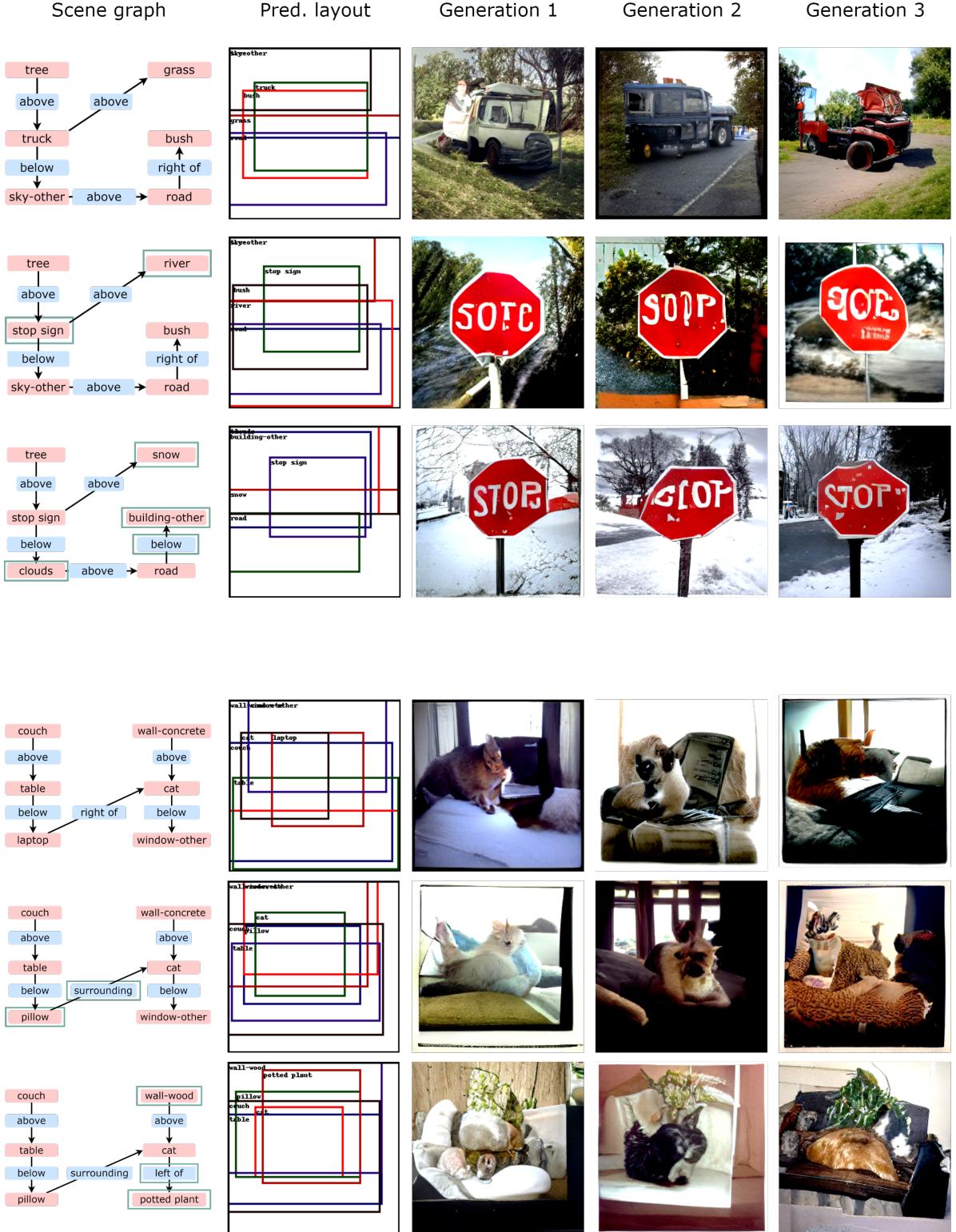


Figure 6: Manipulation of input scene graphs on the COCO-Stuff dataset. For each block of images: we start from an initial scene graph (top line) and as we move to the bottom, at each line, we modify some concepts/relations (highlighted in green) in the graph.

- *SelfAtt-ImT* that employs a GPT [Brown et al., 2020] architecture and computes only self-attention on the input sequence, using the intermediate layout (estimated by *SGTransformer*) as conditioning information. This is the approach described in Sect. 3.3.

Table 4: Ablation study on the Image Transformer architecture. *CrossAtt-ImT* uses cross-attention between the *SGTransformer* output and the codebook index sequence. *SelfAtt-ImT*, instead, concatenates the predicted layout with the decoder input and performs self-attention on the sequence.

Model	FID (↓)		Inception Score (↑)	
	COCO	CLEVR	COCO	CLEVR
<i>CrossAtt-ImT</i>	98.19	37.99	4.48 ± 0.6	2.57 ± 0.3
<i>SelfAtt-ImT</i>	52.34	32.49	9.82 ± 0.4	2.85 ± 0.7

Due to the elevated computational requirements the full model requires to be trained, we evaluate a light-weight version of the models, where the *SGTransformer* keeps the same configuration but the number of layers in the Image Transformer is reduced to 12, with 12 attention heads and an embedding size of 768. As shown in Table 4, *SelfAtt-ImT* yields a higher image quality, showing that encoding the layout into a discrete representation and prepending it to the sequence of tokens is more effective than using the original cross-attention mechanism.

6 Conclusion

In this paper, we propose a fully transformer-based scene graph to image approach, which exploits a multi-head attention for graph geometry learning to generate an intermediate layout representation. This representation is subsequently translated into a sequence of codebook indices by a GPT-based transformer that conditions autoregressive predictions through object location constraints. Using a lower-dimensional latent space, represented by a learned codebook of feature vectors, allows for an effective training of a high-capacity transformer model, whereas training on the pixel space would be computationally unfeasible. Both quantitative and qualitative evaluations on standard benchmarks in the field (such as Visual Genome, COCO-Stuff, and CLEVR) show a better capability of our approach in generating high quality diverse images as well as an extensive flexibility in handling minor and major changes in the input scene graph.

7 Acknowledgements

We acknowledge financial support from: PNRR MUR project PE0000013-FAIR. This work has been also supported by the research project titled “Safe and Smart Farming with Artificial Intelligence and Robotics”, Piano di Incentivi per la Ricerca di Ateneo 2020/2022, University of Catania (Italy).

References

- 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.
- Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*, pages 88–105. Springer, 2022.
- Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014.

- Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning*, pages 1857–1865. PMLR, 2017.
- Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.
- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 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, 2017.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.
- Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018.
- 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, 2019.
- Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.
- Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.
- Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34:14745–14758, 2021.
- Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International conference on machine learning*, pages 4487–4499. PMLR, 2021.
- Dor Arad Hudson and Larry Zitnick. Compositional transformers for scene generation. *Advances in Neural Information Processing Systems*, 34:9506–9520, 2021.
- Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.

- 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.
- 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.
- Duc Minh Vo and Akihiro Sugimoto. Visual-relation conscious image generation from structured-text. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 290–306. Springer, 2020.
- Azade Farshad, Sabrina Musatian, Helisa Dhamo, and Nassir Navab. Migs: Meta image generation from scene graphs. *arXiv preprint arXiv:2110.11918*, 2021.
- Gaurav Mittal, Shubham Agrawal, Anuva Agarwal, Sushant Mehta, and Tanya Marwah. Interactive image generation using scene graphs. *arXiv preprint arXiv:1905.03743*, 2019.
- Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.
- Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10531–10540, 2019.
- Zuopeng Yang, Daqing Liu, Chaoyue Wang, Jie Yang, and Dacheng Tao. Modeling image composition for complex scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7764–7773, 2022.
- Wan-Cyuan 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 preprint arXiv:2208.13753*, 2022.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Manuel Jahn, Robin Rombach, and Björn Ommer. High-resolution complex scene synthesis with transformers. *arXiv preprint arXiv:2105.06458*, 2021.
- 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*, 123:32–73, 2017.
- 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.
- Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. *arXiv preprint arXiv:1903.03166*, 2019.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- 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, 2017.
- 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, 2016.