Layout Free Scene Graph to Image Generation

Rameshwar Mishra
rameshwarm@iiitd.ac.in
A V Subramanyam
subramanyam@iiitd.ac.in
Indraprastha Institute of Infromation Techonology
Delhi, India

Abstract

Advancements in generative models have sparked significant interest in generating images while adhering to specific structural guidelines. Scene graph to image generation is one such task of generating images which are consistent with the given scene graph. However, the complexity of visual scenes poses a challenge in accurately aligning objects based on specified relations within the scene graph. Existing methods approach this task by first predicting a scene layout and generating images from these layouts using adversarial training. In this work, we introduce a novel approach to generate images from scene graphs which eliminates the need of predicting intermediate layouts. We leverage pre-trained text-to-image diffusion models and CLIP guidance to translate graph knowledge into images. Towards this, we first pre-train our graph encoder to align graph features with CLIP features of corresponding images using a GAN based training. Further, we fuse the graph features with CLIP embedding of object labels present in the given scene graph to create a graph consistent CLIP guided conditioning signal. In the conditioning input, object embeddings provide coarse structure of the image and graph features provide structural alignment based on relationships among objects. Finally, we fine tune a pre-trained diffusion model with the graph consistent conditioning signal with reconstruction and CLIP alignment loss. Elaborate experiments reveal that our method outperforms existing methods on standard benchmarks of COCO-stuff and Visual Genome dataset. Our code, and instructions to reproduce the results can be found in https://anonymous.4open.science/r/GANDiffuCLIP-D9E8.

1 Introduction

Scene graph represents a visual scene as a graph where nodes correspond to objects and edges represent relationships or interactions between these objects. Improved generative models now allow users to generate high quality images where they can control the style, structure or layout of the synthesised images. Such conditional image generation allow users to guide the generation using text [23, 26], segmentation mask [20], class labels [3], scene layout [10, 38], sketches, stroke paintings [18], and such more conditional signals. In particular, use of text as a conditioning modality offers a versatile approach, allowing for diverse combinations of inputs, encompassing intricate and abstract concepts. However, leveraging text for conditioning is not without challenges. Natural language sentences tend to be lengthy

and loosely structured, relying heavily on syntax for semantic interpretation. The inherent ambiguity in language, where different sentences may convey the same concept, poses a risk of instability during training. This becomes particularly apparent in scenarios where precise description constraints are crucial. In this context, relying solely on text representations for a specific scene may prove to be insufficient.

Motivated by promising results of conditional generation and limitation of text as a conditional signal, in this work we propose a novel method to generate images from scene graphs. First introduced by [12], scene graph to image generation is a task of generating images using a set of semantic object labels and underlying semantic relationships among these objects. Most of the existing works follow a two stage architecture where they first generate a scene layout and use GAN to synthesize realistic images from these scene layouts [1, 9, 12]. Object nodes of scene graph is mapped to bounding boxes in the layout and the relationships are signified by the spatial structure of the layout. While these scene layouts can be effective in representing spatial relationships in the scenes, they fail to capture non-spatial complex relationships among objects. Translating scene graphs to accurate layouts and limiting representation capabilities of these layouts results in images inconsistent with the input scene graph. To overcome the limitations of existing methodologies, we propose to learn an optimized intermediate graph representation while eliminating the need of predicting scene layouts.

We first employ a GAN-based CLIP [22] alignment module to train our graph encoder. This module instructs the graph encoder to generate graph embeddings that closely resemble the visual features of corresponding images in the CLIP latent space. To construct an effective conditioning signal to harness the strong semantic understanding offered by diffusion model, we fuse output of graph encoder with semantic label embedding of objects present in the scene graph. We demonstrate the effectiveness of our method using established benchmarks like Visual Genome[14] and COCO-stuff [4]. Comparisons with current state-of-the-art methods reveal superior quantitative and qualitative results. We can summarise our contributions as follows:

- Our work for the first time leverages a pre-trained diffusion model for the task of image generation from scene graph without requiring an additional text prompt from the user.
- We propose a novel methodology to learn an effective graph representation, eliminating the need of predicting intermediate layouts to synthesize images. We use this graph representation to construct a suitable conditioning signal for text-to-image diffusion model.
- We propose a training strategy that optimizes our scene graph input for diffusion models. We propose a GAN-based CLIP alignment module to guide our scene graph embeddings to leverage the semantic knowledge of text-to-image diffusion model.

2 Related Works

Diffusion as generative model. The introduction of diffusion models by [30] marked a notable approach to image generation. These models operate by learning the reverse process of the forward diffusion, where input is transformed into Gaussian noise. The denoising process is implemented using U-net [19] or transformer [34] based models. In order to reduce the computation and training complexity, [27, 33] introduced diffusion models which operate in latent space. [8] proposed conditional generation by diffusion using classifier guidance.

Recent advancements in these latent diffusion models have enabled users to produce diverse and realistic high quality images conditioned on various factors such as text [23], artistic style, sketch, pose, and class labels [6]. Diffusion can also be used to generate text [16, 17], videos [6, 35] and graphs [11]. The conditioning is applied using cross attention mechanism between output of individual layers of denoising U-net and given conditional signals. Similar techniques are employed in text-to-image latent diffusion models. [24] uses CLIP latent of text to condition high quality image generation. [28] uses latent from large language models such as T5 to condition latent diffusion models. While there has been notable progress in the diffusion-based conditional image synthesis, there exists a notable gap in the exploration of image generation from graph-structured data. In this work we explore the capabilities of text-to-image diffusion models in the task of image generation conditioned on scene graphs.

Image generation from scene graphs. Scene graph represents an image using set of nodes and edges. Nodes represent objects present in the image and their underlying relationships are captured by edges. Conventional scene graph to image methods tackle this task by following two stage architecture. At first a scene layout is predicted from graph. Scene layout represents an image using bounding boxes of corresponding objects present in the image. Scene layout is then translated into an image using convolution neural network based image synthesis models such as SPADE [21], OC-GAN [32]. This task was first introduced by [12]. They employ a multi layer graph convolution network [13] to get graph representation. This graph representation is used to predict object bounding boxes. The boxes are then used to generate images using cascaded refinement network. Generation is guided by GAN-based setup where a discriminator is employed to generate realistic images. Following [12] subsequent works adopt the two stage approach combined with GAN-based generation. [1] provides a way to control style of generated objects by providing a module to capture the style information of objects. [7] use canonicalization for scene graph representation before translating it into scene layouts. This enhances the graph representation by incorporating supplementary information for semantic equivalence. [9] introduces an overlap loss to eliminate object overlapping. [31] use transformers for image generation. They learn layout representation using graph transformer. Further an image transformer coupled with VQ-VAE [25] is used to sample images from these layouts. [8] uses scene layout and segmentation masks at sampling time of diffusion to generate graph aligned images. [36] introduce a consistency module to overcome negligence of smaller object in the generated images.

Most of the existing works utilise a layout based representation of graphs and GAN-based image generation. In this work we propose to use a graph representation which aligns well with the semantic prior of diffusion models. We use this aligned graph representation as a conditioning signal for diffusion based image generation. Notably, we eliminate the need for layout generation and convert the two stage to single stage generation.

3 Method

In this section, we present our proposed methodology with a detailed description of each component. We first give a brief overview of the conditional diffusion model. Subsequently, we explain the functionality of the graph encoder, and the process of obtaining optimized graph representations. We note that the diffusion models are conditioned on text embeddings obtained from CLIP. However, the scene graph encodings are not aligned with the CLIP latent space. We propose an alginment module to overcome this challenge. An overview of

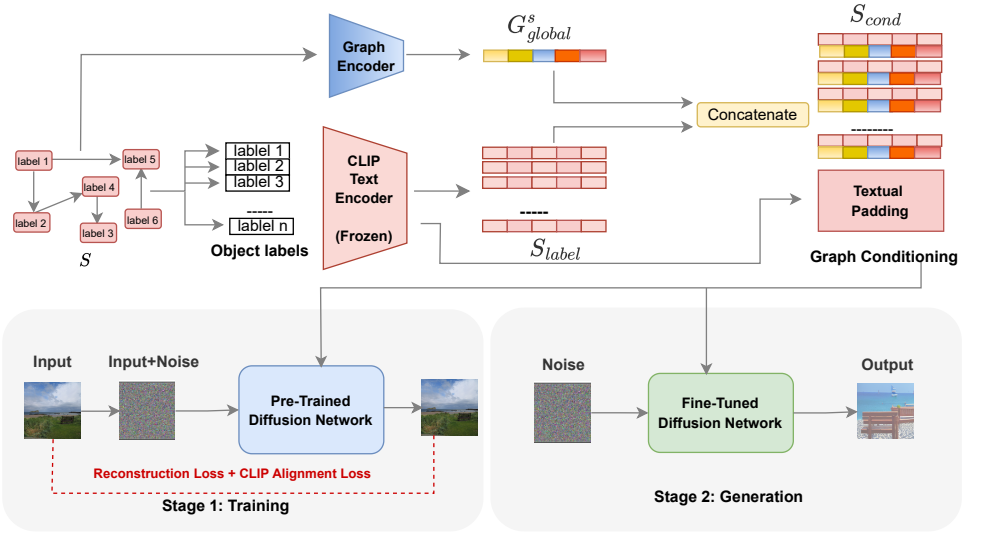

Figure 1: Overview of the proposed architecture. Graph encoder gives a CLIP aligned graph embedding. This embedding is fused with semantic label embedding of objects present in the scene graph. The fused embedding forms a conditioning signal for diffusion model.

our methodology is given in Figure 1.

3.1 Background for diffusion

Diffusion models form a class of generative models designed to simulate the process of data generation through a series of diffusion steps. They constitute probabilistic generative models trained to comprehend data distributions through the sequential denoising of a variable sampled from a given distribution, mainly Gaussian. In the context of conditional generation, the goal is to generate data conditioned on some input information. For our case, we are concerned with pre-trained text-to-image diffusion model $\hat{x}_\theta$. Given noise $\varepsilon \sim \mathcal{N}(0,I)$ and conditioning signal S_{cond}, the model generates an image x_{gen} as follows:

$$x_{gen} = \hat{x}_\theta(\varepsilon, S_{cond}). \tag{1}$$

U-net based architecture is used to predict the added noise. The training is guided by a squared error loss to denoise an image or latent code with variable levels of noise. It is given by,

$$\mathbb{E}_{(x,\varepsilon,t)} \|x - \hat{x}_\theta(x_t, S_{cond})\|_2^2, \tag{2}$$

where x is the reference image, $x_t = \alpha_t x + \sigma_t \varepsilon$, is the noisy sample at diffusion time step t. α_t and σ_t control the noise schedule. For latent diffusion models, latent embedding of input image is generated using VQGAN [9] or KL-Autoencoder [27]. All diffusion steps are applied in latent space, and then the final latent is decoded into an image.

3.2 Graph Encoder

We use multi layer graph convolution network [12, 13] to generate graph features from scene graph. We follow existing architecture of graph encoder for fair comparison with existing

methodologies. Scene graph S contains a set of objects S_o and a set of relationships S_r. S is represented using relationship triplets (o_i, r_{ij}, o_j) where $o_i \in S_o$ and $o_j \in S_o$ are two objects from object set S_o, and $r_{ij} \in S_r$ is the relationship between i^{th} and j^{th} object. Graph encoder fuses individual object embedding and individual relationship embedding to give a global scene graph embedding. For object o_i, we take a set $Out(o_i)$ to be the set of object to which o_i has an outgoing directed edge. Set $In(o_i)$ denotes the set of objects where o_i has an incoming directed edge from these objects. We find embedding for object o_i as follows:

$$G_{o_{out}} = F_o^{out}(G_{o_i}, G_{r_{ij}}, G_{o_j})_{j \in Out(o_i)}, G_{o_{in}} = F_o^{in}(G_{o_j}, G_{r_{ji}}, G_{o_i})_{j \in In(o_i)},$$

$$G_{o_i} = F^{pool}((G_{o_{out}}) \cup (G_{o_{in}})),$$

where $G_{o_i}, G_{o_j} \in R^{d_o}$ are the embeddings of object o_i and o_j respectively. $G_{r_{ij}}, G_{r_{ji}} \in R^{d_r}$ are the embeddings for relationship r_{ij} and r_{ji} respectively. F_o^{out}, F_o^{in} are graph convolution layers and F^{pool} is an average pooling layer. Similar to this, we find relationship embedding as follows:

$$G_{r_{ij}} = F^{rel}(G_{o_i}, G_{r_{ij}}, G_{o_j}), \tag{3}$$

where F^{rel} is a graph convolution layer. After getting these individual object and relationship embedding, we calculate a global graph feature G^s_{global} as follows. First we map each object and relationship embedding to same dimension d_g using $F^o_{d_g}$ and $F^r_{d_g}$. $F^o_{d_g}$ and $F^r_{d_g}$ are 2 layer MLP's. They map d_o dimensional object embedding and d_r dimension relationship embedding to d_g dimensional embedding respectively. After getting same dimension embedding, we find an embedding to represent each individual triplet of scene graph as follows:

$$G_{triplet_{ij}} = F^o_{d_g}(G_{o_i}) + F^r_{d_g}(G_{r_{ij}}) + F^o_{d_g}(G_{o_j}). \tag{4}$$

Finally, a global embedding G^s_{global}, for a scene graph S is calculated by concatenating individual triplet embedding and then mapping it to a d_g dimensional feature.

$$G^s_{global} = F^{global}_{d_g}(concat(G_{triplet_{ij}}))_{triplet_{ij} \in S}. \tag{5}$$

We use G^s_{global} while creating a conditioning signal for fine-tuning text-to-image diffusion model. The diffusion models are conditioned on CLIP embeddings of text prompts. Initially the latent space of the graph encoder output differs substantially from the CLIP feature space. To bridge this gap, we propose G^s_{global} and CLIP Alignment module (GCA) to pre-train graph encoder. We discuss this next.

3.3 G^s_{global} and CLIP Alignment

We use GAN-based pre-training to align G^s_{global} with CLIP features. Figure 2 provides an overview of our GAN-based CLIP alignment module. We consider graph encoder as our generator and G^s_{global} as our generated data. CLIP visual features, c, form real data. Discriminator is trained to predict whether the input is from real or generated data. It guides the output of our generator to align with CLIP features. Training of graph encoder

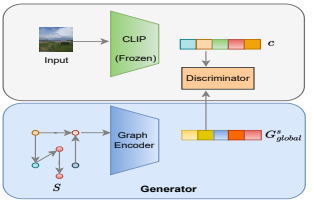

Figure 2: GCA. Graph embedding is aligned with CLIP features.

is guided by $\mathcal{L}_{\text{graph}}$, where $\mathcal{G}(S)$ is graph encoder output
when given a scene graph S as an input. $\mathcal{L}_{\text{graph}}$ is a standard generator loss for GAN-based architectures, $\mathcal{L}_{\text{graph}} = \mathbb{E}_{S\sim p(S)}\left[-\log \mathcal{D}(\mathcal{G}(S))\right]$.

Training of discriminator is guided by $\mathcal{L}_{\text{disc}}$, a standard discriminator loss for GAN-based architectures. Let $\mathcal{D}(c)$ be the predicted probability by discriminator when the input is from real distribution (CLIP features, c), $\mathcal{L}_{\text{disc}} = -\mathbb{E}_{c\sim p(c)}\left[\log \mathcal{D}(c)\right] - \mathbb{E}_{S\sim p(S)}\left[\log(1-\mathcal{D}(\mathcal{G}(S)))\right]$.

3.4 Training diffusion model with CLIP guided graph conditioning

G^S_{global} captures the overall structure and interaction between the entities of scene-graph. However, this is global in nature. We hypothesize to use object label embeddings via CLIP with G^S_{global} in our conditioning signal. The object encodings via CLIP can provide fine-grained semantic label details which can complement the global scene graph encodings. In Figure 1, the semantic labels of objects present in the scene graph are passed through CLIP text encoder to generate object label embedding S_{label}. Then scene graph S is passed through a graph encoder to generate embedding G^s_{global} using Eq.5. The object label embedding captures the entity level information of the image, while G^S_{global} captures the interaction between these entities. Let $label_i$ be the semantic label of i^{th} object present in S. Then,

$$S_{label_i} = CLIPtext(label_i).$$

Finally we fuse G^s_{global} and S_{label_i} for all the labels to generate conditioning signal for fine-tuning the diffusion model.

$$S_{cond} = concat(G^s_{global}, S_{label_i})\ \forall i \in S.$$

We add textual padding to generate conditional signals of same dimension irrespective of the number of objects present in the scene graph. However, we note that pre-trained diffuion model we used is trained for text and image pairs. Thus it is necessary to design a training strategy to generate images conditioned on S_{cond}. Our training strategy is centered around optimizing the scene graph input for the diffusion model. This ensures that the scene graph aligns effectively with the input space of the diffusion model and weights of diffusion models are optimized for our conditioning signal. Experimentally, we verified that, when we simply pass our designed conditioning signal without fine-tuning the diffusion models, it results in the generation of low-quality images. These images lack coherence with the input scene graphs. Ablation results supporting the use of S_{label} and the impact of fine-tuning are present in the supplementary material.

3.4.1 Training objective

Our learning objective is two-fold. First, diffusion model should learn the underlying distribution of image and scene-graph pairs. Second, we want to map output of graph encoder to a space where it aligns with prior semantic knowledge of text-to-image diffusion models. We achieve these goals in the following manner.

Reconstruction Loss: We use a reconstruction loss to guide the diffusion model to learn the underlying distribution of data. The loss $\mathcal{L}_{recon}$ is given by,

$$\mathcal{L}_{recon} = \mathbb{E}_{(x,S)\sim \text{data}} \|x - \hat{x}_\theta(x_t, S_{cond})\|_2^2, \tag{6}$$

where $\hat{x}_\theta$ is denoising network of diffusion, x_t is the noised sample at diffusion time step t, and S_{cond} is the conditioning signal we curated. (x,S) is the image-graph pair, sampled randomly from the data.

Alignment Loss: Towards the second goal of aligning G^s_{global} with CLIP space, we apply a mean squared error loss between G^s_{global} and CLIP visual features of the corresponding image. For an image-graph pair (x,S), the loss $\mathcal{L}_{CLIP}$ is given by,

$$\mathcal{L}_{CLIP} = \mathbb{E}_{(x,S)\sim \text{data}} \| G^s_{global} - CLIP(x) \|_2^2. \tag{7}$$

Additionally, we also use Maximum Mean Discrepancy (MMD) loss [15] to bridge the domain gap between G^s_{global} and CLIP visual features. In our experiments we observe that MMD loss makes the training stable, improves quantitative results and quality of images. MMD loss $\mathcal{L}^2_{\text{MMD}}$ is defined as follows:

$$\mathbb{E}_{c\sim p(c)}\left[\phi(c,c)\right] + \mathbb{E}_{G\sim p(\text{graph})}\left[\phi(G^s_{\text{global}}, G^s_{\text{global}})\right] - 2\mathbb{E}_{c\sim p(c), G\sim p(\text{graph})}\left[\phi(c, G^s_{\text{global}})\right],$$

where c is the CLIP feature of an input image and G^s_{global} is the output of graph encoder for the corresponding scene graph. ϕ is the kernel function. We combine both $\mathcal{L}_{CLIP}$ and $\mathcal{L}_{MMD}$ to define an alignment loss as,

$$\mathcal{L}_{align} = \beta \mathcal{L}_{CLIP} + (1-\beta)\mathcal{L}_{MMD}, \tag{8}$$

where β is a hyperparameter.

Total Loss: Now by considering both reconstruction loss and alignment loss we define our training objective as, $\mathcal{L}_{train} = \lambda \mathcal{L}_{recon} + (1-\lambda)\mathcal{L}_{align}$, where λ is a hyperparameter. We fine tune diffusion model and graph encoder using $\mathcal{L}_{train}$.

3.5 Sampling process

In Figure 2, stage 2 gives an overview for the sampling process. Once the diffusion network is trained, we can sample images from a latent noise ε. For a fine-tuned denoising U-net $\hat{x}_\theta$, we can sample latent conditioned on scene graph S as, $x_{latent} = \hat{x}_\theta(\varepsilon, S_{cond})$, where S_{cond} is the curated graph conditioning signal and $\varepsilon \sim \mathcal{N}(0,I)$. x_{latent} is then decoded using diffusion's latent decoder to get an image. The generated image aligns well with the input scene graph.

4 Experiments

In this section, due to the limited space, we briefly outline the implementation details of our approach. More details regarding implementation can be found in the supplementary material. We compare our results with existing state-of-the-art scene graph to image models. We verify effectiveness of each component of our training scheme by providing ablation results.

4.1 Experimental setup

We train and evaluate our model on COCO-stuff and Visual genome dataset. We follow existing works [7, 12] to filter out and divide the data into training and validation set for fair

comparison. To show effectiveness of our approach we evaluate our model using Inception Score (IS) [29], Fréchet Inception Distance (FID) [8], Diversity Score (DS) [37], and Object occurrence ratio (OOR) [36].

Methods (Reference)	COCO-Stuff				Visual Genome			
	FID↓	IS↑	DS↑	OOR↑	FID↓	IS↑	DS↑	OOR↑
SG2IM (CVPR'18) *	125.58	7.8	0.02	–	92.8	6.5	0.1	–
PasteGAN (NeurIPS'19)*	70.2	11.28	0.60	–	130	6.5	0.38	–
Specifying (ICCV'19)*	68.27	15.2	0.67	70.84	–	–	–	–
Canonical (ECCV'20)*	64.65	14.5	0.70	73.77	45.7	16.4	0.68	72.83
RetrieveGAN (ECCV'20)	56.9	10.2	0.47	–	113.1	7.5	0.30	–
SCSM (AAAI'22)	51.6	15.2	0.63	–	63.7	10.8	0.59	–
SGTransformer (CVIU'23)*	52.8	15.8	0.57	–	50.16	14.6	0.59	–
SceneGenie (ICCV'23)	63.27	22.16	–	–	42.21	20.25	–	–
LOCI (IJCAI'23)	49.8	15.7	0.65	81.26	44.9	14.6	0.62	79.04
Ours	**38.12**	**30.18**	**0.73**	**82.38**	**35.8**	**26.2**	**0.71**	**81.04**

Table 1: Quantitative results on Visual Genome and COCO-Stuff dataset. All the results are either reproduced (*) or taken directly from the original papers. Best results are shown in bold letters, and second best results are underlined. Results are for 256×256 images.

4.2 Results

Quantitative results: Following previous works [7, 12, 36], we have reported a comparison between our method and existing methods using FID score, IS, DS, OOR. Table 1 shows the effectiveness of our method based on these evaluation metric. On COCO-stuff benchmark, we are able to reduce FID score by 11.68, increase IS by 8.02 when compared to existing SOTA [36], [6] respectively. We also achieve the best results for DS score and OOR implying that the model generates diverse images yet contains the objects provided in the input scene graph. From Table 1 we can see that similar to COCO-stuff, there is significant improvement in terms of all the evaluation metrics for Visual Genome benchmark as well. Quantitative results show that we generate high quality and diverse images which are aligned with the given scene graph.

Qualitative results: Figure 3 show qualitative comparison between images generated by publicly available existing models and our method. We compare our results with SG2IM [12], canonicalization based model [7] and Transformer based model SGTransformer [31]. Qualitative comparison shows superior performance of our model. It can be seen that generated images align well with the input scene graph and conserve the relationship structure provided by the scene graph. For example, in row 5, images generated by canonical and SGTransformer contain trees, but fails to generate it's shadow. Similarly in row 3, SG2Im and canonical generate distorted images, whereas image generated by our model is most consistent with the input scene graph. More qualitative results are given in supplementary.

Ablation study on COCO-stuff: In this section, we illustrate the significance of each component in our training scheme. G^s_{global} and CLIP alignment (GCA) module aligns output of the graph encoder with CLIP features of the corresponding image. This alignment is important since text-to-image diffusion models have strong semantic prior of CLIP features.

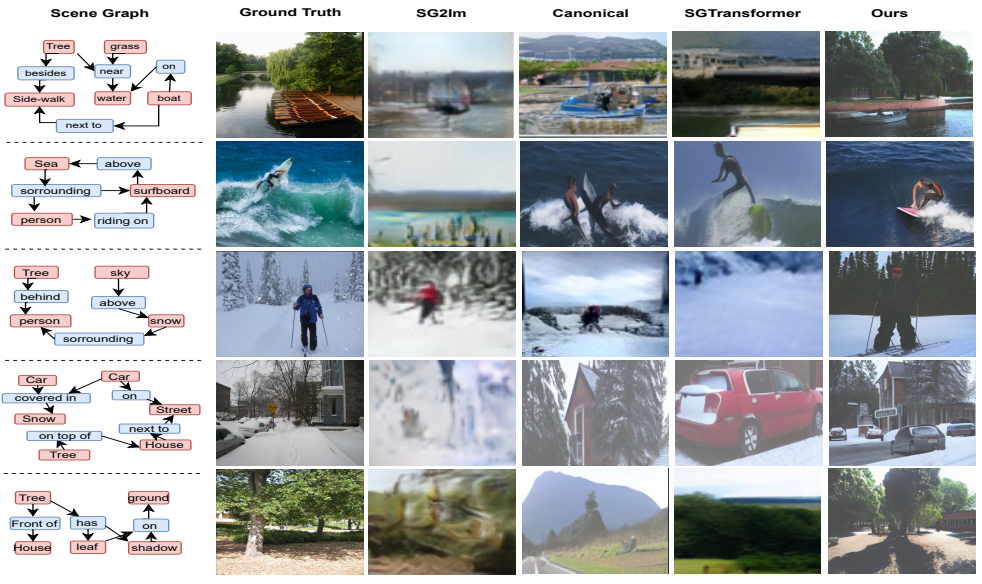

Figure 3: Qualitative comparison of (256×256) images generated by various publicly available scene graph to image models. All given input graphs corresponding to ground truth images are perturbed slightly to check effectiveness of each methods.

Model type	IS ↑	FID ↓
W/O GCA	27.6	43.28
W/O $\mathcal{L}_{\text{align}}$	28.72	41.17
W/O $\mathcal{L}_{\text{MMD}}$	29.24	39.4
Ours (λ=0.8,β=0.7)	29.74	39.28
Ours (λ=0.6,β=0.3)	28.2	40.12
Ours	30.18	38.12

Table 2: Ablation

In Table 2, W/O GCA shows the performance without GCA. It is clearly evident that the incorporation of this module prior to fine-tuning the diffusion model results in improvements in both IS and FID scores. For fine-tuning diffusion model, we introduce an alignment loss $\mathcal{L}_{align}$ with a standard reconstruction loss. After GCA, this loss further guides the graph encoder to generate graph embeddings aligned with CLIP latent spaces. We also take multiple combinations of hyperparameters λ and β to define our training loss $\mathcal{L}_{train}$. Finally, we show that our methodology containing GCA, $\mathcal{L}_{train}$ with $\lambda = 0.7, \beta = 0.5$ gives best results.

5 Conclusion

In this work, we propose a novel scene graph to image generation method. Our method eliminates the need of intermediate scene layouts for image synthesis. We use a pre-trained text-to-image model with CLIP guided graph conditioning signal to generate images conditioned on scene graph. We propose a GAN-based alignment module which aligns graph embedding with CLIP latent space to leverage the prior semantic understanding of text-to-image diffusion models. To further enhance the graph-conditioned generation, we introduce an alignment loss. Through comprehensive assessments using various metrics that measure the quality and diversity of generated images, our model showcases state-of-the-art performance in the task of scene graph to image generation.

References

[1] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4561–4569, 2019.

[2] 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.

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[4] 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.

[5] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023.

[6] Azade Farshad, Yousef Yeganeh, Yu Chi, Chengzhi Shen, Böjrn Ommer, and Nassir Navab. Scenegenie: Scene graph guided diffusion models for image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 88–98, 2023.

[7] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 210–227. Springer, 2020.

[8] 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.

[9] 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.

[10] Manuel Jahn, Robin Rombach, and Björn Ommer. High-resolution complex scene synthesis with transformers. *arXiv preprint arXiv:2105.06458*, 2021.

[11] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, pages 10362–10383. PMLR, 2022.

[12] 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.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[14] 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.

[15] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017.

[16] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11109–11119, 2021.

[17] Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning*, pages 21051–21064. PMLR, 2023.

[18] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[19] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[20] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[23] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. 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 8821–8831. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html.

[24] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[25] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[26] 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 (CVPR)*, pages 10684–10695, June 2022.

[27] 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.

[28] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[29] 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.

[30] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[31] Renato Sortino, Simone Palazzo, Francesco Rundo, and Concetto Spampinato. Transformer-based image generation from scene graphs. *Comput. Vis. Image Underst.*, 233(C), aug 2023. ISSN 1077-3142. doi: 10.1016/j.cviu.2023.103721. URL https://doi.org/10.1016/j.cviu.2023.103721.

[32] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2647–2655, 2021.

[33] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[34] 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.

[35] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18456–18466, 2023.

[36] Yangkang Zhang, Chenye Meng, Zejian Li, Pei Chen, Guang Yang, Changyuan Yang, and Lingyun Sun. Learning object consistency and interaction in image generation from scene graphs. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 1731–1739, 2023.

[37] Zijun Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pages 1–2. Ieee, 2018.

[38] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22490–22499, 2023.