CONSISTENT AND MULTI-SCALE SCENE GRAPH TRANSFORMER FOR SEMANTIC-GUIDED IMAGE OUTPAINTING

Chiao-An Yang[1] *Meng-Lin Wu*[2] *Raymond A. Yeh*[1] *Yu-Chiang Frank Wang*[3]

[1]Purdue University [2]Qualcomm Technologies Inc [3]National Taiwan University

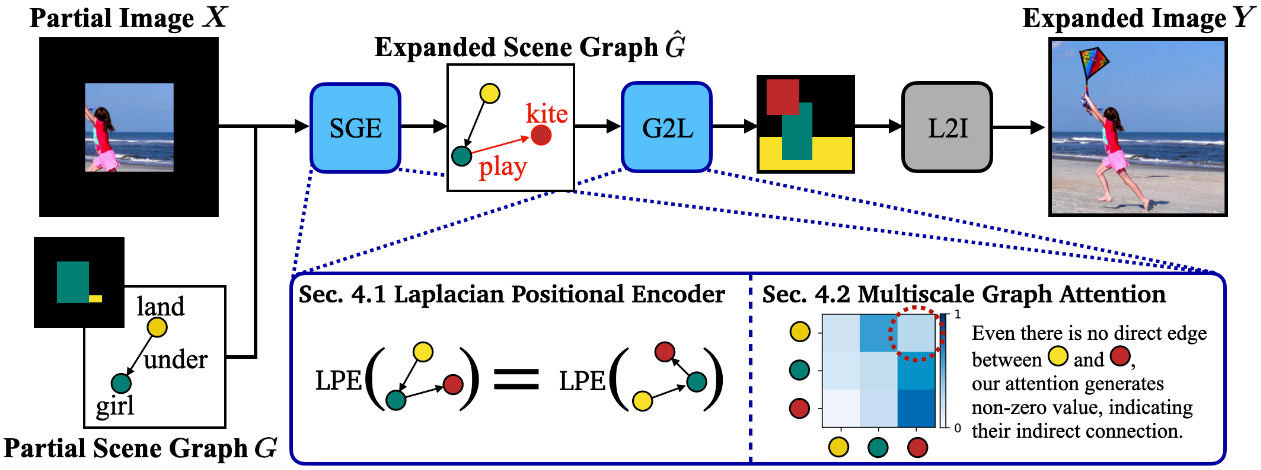
Partial Image X
Expanded Scene Graph $\hat{G}$
Expanded Image Y
SGE
kite
play
G2L
L2I
land
under
girl
Partial Scene Graph G
Sec. 4.1 Laplacian Positional Encoder
LPE
=
LPE
Sec. 4.2 Multiscale Graph Attention
1
0
Even there is no direct edge between ● and ●, our attention generates non-zero value, indicating their indirect connection.

Fig. 1. **Model Pipeline**. Our methods include three separate stages, (i) SGE, (ii) G2L, and (iii) L2I. We propose a Laplacian positional encoder that generates positional encoding features that are order-invariant and reflect the intrinsic structural information of a graph. We also introduce multiscale graph attention that can quickly increase the "receptive field" of the model.

ABSTRACT

The task of image outpainting extends an image beyond its boundaries with semantically plausible content. Recently, Scene Graph Transformer (SGT) introduced a transformer architecture to leverage scene graph guidance for image outpainting. Despite its success, we identified two shortcomings: (a) SGT uses a positional encoding that was originally proposed for 1D signal; (b) SGT uses a scene graph attention layer that propagates information between neighboring nodes which limited the model to learning local graph features. To address these issues, we propose incorporating Laplacian positional encoding and introducing a multiscale scene graph attention into SGT. Extensive results on MS-COCO and Visual Genome show that our proposed approach generates more plausible outpainted images with higher quality.

***Index Terms*—** Image outpainting, Scene-graph

1. INTRODUCTION

Image outpainting refers to the task of completing a partial image beyond its original boundary. Numerous computer vision and image processing applications benefit from advancement in image outpainting, e.g., view expansion [1], image panorama creation [2], and object removal editing [3]. While image inpainting (*a.k.a.* image completion) [4, 5, 6, 7, 8, 9, 10, 11] has received more interest, image outpainting is more challenging. Image inpainting requires only the recovery of the missing regions within the boundaries, i.e., interpolating. On the other hand, the missing regions of image outpainting can be infinitely outward, i.e., extrapolation.

Recently, Yang et al. [12] have shown the effectiveness of the three-stage scene graph guided image outpainting method which composes of three steps: (i) Scene Graph Expansion, (ii) Scene Graph to Layout, and (iii) Layout to Image, where each step is accomplished by a Scene Graph Transformer (SGT) model. Despite their success, we observe key shortcomings in SGT. First, SGT consists of standard sinusoidal position encoding which is originally designed for 1-dimensional input [13]. Second, SGT uses scene graph attention modules [12] which only propagates information between neighboring nodes in a scene graph. This led to SGT taking at least N attention layers to extract features of a graph with diameter N.

To address these issues, we propose to replace the sinusoidal encoding with a learnable Laplacian encoding (Sec. 4.1). Different from sinusoidal encoding which captures the node ordering, which is arbitrary during graph construction, Laplacian encoding captures the adjacency matrix of a graph. This allows Laplacian encoding to capture features of the graph structure in the input scene graph. Next, we propose a multiscale scene graph attention (Sec. 4.2). Different from the attention in SGT which uses the adjacency matrix of the input scene graph to capture local features, our multiscale attention views the transformers as a series of multiple message-passing layers and integrates the overall effects of multiple attention computation into one single matrix.

Empirically, we demonstrate that our method outperforms SGT [12] on VG-MSDN and COCO-stuff. In terms of accuracy, our approach significantly improves scene graph expansion and layout prediction. The resulting scene graphs and layouts can be used to generate more natural images.

Our contributions: We identify two architecture improvements over SGT [12]. Specifically, we introduce Laplacian encoding and multiscale Attention which are suitable for modeling scene-graph data. We show that our solution is effective and is able to generate results with realistic object placements and higher image quality.

2. RELATED WORK

Sabini and Rusak [11] is the first to apply GAN [14] on image outpainting which focuses on expanding along horizontal directions. Several extensions Yang et al. [1], Lu et al. [15], Teterwak et al. [16], Wang et al. [10], based on the formulation explore other directions of image outpainting such as expanding along vertical directions or through intermediate gaps. Tan et al. [17] is able to perform image outpainting without pre-specifying any directions by implementing a margin prediction module. More recently, Khurana et al. [18] and Yang et al. [12] both propose to leverage semantic information, e.g., scene graphs and segmentation maps, to outpaint images with more concrete guidance.

3. PRELIMINARIES

Problem formulation. The goal of image outpainting is to generate an outpainted image $\boldsymbol{Y}$ of $h_2 \times w_2$ pixels which extends a given partial image $\boldsymbol{X}$ of $h_1 \times w_1$ pixels. We follow the three-stage semantic-guided image outpainting [12] that partitions the outpainting task into scene graph expansion (SGE), scene graph to layout (G2L), and layout to image (L2I). An overview of the method is shown in Fig. 1.

The SGE stage deploys an SGT to expand the partial scene graph extracted from the input incomplete image. Next, in the G2L stage, another SGT is used to convert the expanded scene graph into an expanded layout. Finally, the L2I stage transforms a layout into an outpainted image using a GAN model. Please refer to Yang et al. [12] for details. This paper focuses on the architecture of SGT, which we review next.

Scene Graph Transformer. SGT [12] is a transformer architecture designed to handle scene-graph data. At a high level, a scene-graph is represented by a graph $\boldsymbol{G}$ consisting of a set of nodes $\mathcal{O} = \{o_i\}$ and a set of relationship (edges) $\mathcal{R} = \{r_{ij}\}$.

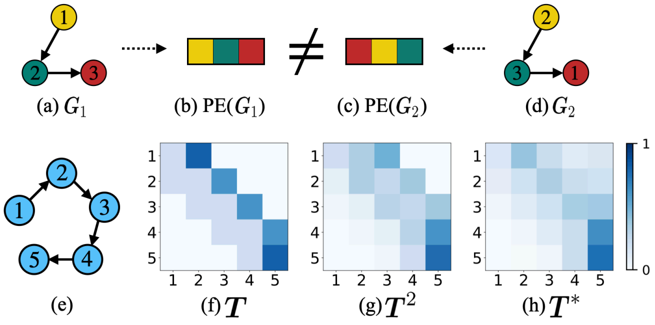

Fig. 2. **(a-d) The problem of standard positional encoding.** For the same graph, i.e. (a) $\boldsymbol{G}_1$ and (b) $\boldsymbol{G}_2$, a 1D sinusoidal positional encoding would extract a different representation, i.e. (c) PE($\boldsymbol{G}_1$) and (d) PE($\boldsymbol{G}_2$), depending on how the nodes are "ordered". **(e-f) Multiscale graph attention.** (e) Here is a simple chain-like scene graph with 5 nodes $\{n_1, n_2, \cdots, n_5\}$. (f) Many weights in $\boldsymbol{T}$ are zero or negligible since there is no direct edge between (n_1, n_3), (n_1, n_4), $\cdots$, and so on. (g) By passing the message for an additional iteration with matrix multiplication in $\boldsymbol{T}^2$, some of the weights, e.g.,(n_1, n_3) have positive values now. (h) Our attention matrix $\boldsymbol{T}^*$ has positive weights between nodes in a larger neighborhood.

Each of the nodes and edges has their corresponding features $\mathcal{F}^O = \{\boldsymbol{f}_i^o\}$ and $\mathcal{F}^R = \{\boldsymbol{f}_{ij}^r\}$, where $\boldsymbol{f}_i^o$ and $\boldsymbol{f}_{ij}^r$ are the features extracted from each object and relationship.

The main building block of SGT is the self-attention layer, which updates the node features as follows:

$$\hat{\boldsymbol{f}}_i^o = \sum_j \underbrace{\left(\frac{s(q(\boldsymbol{f}_i^o), k(\boldsymbol{f}_j^o), \boldsymbol{f}_{ij}^r)}{Z_i}\right)}_{T_{ij}} v(\boldsymbol{f}_j^o), \qquad (1)$$

where $s(\cdot)$ is the similarity function proposed in node-level attention [12]; $q(\cdot), k(\cdot), v(\cdot)$ are the query, key, value vector function respectively; Z_i is the normalization factor such that $\sum_j T_{ij} = 1$.

We can also rewrite Eq. (1) as a matrix multiplication operation and view the transformers as a sequence of multiple message-passing layers,

$$\hat{\boldsymbol{F}} = \boldsymbol{T} v(\boldsymbol{F}), \qquad (2)$$

where the attention weight matrix $\boldsymbol{T} \in \mathbb{R}^{N \times N}$ consists all T_{ij}. Here, we illustrate the formulation using node features, we noted that the same formulation is also applied to edge-level attention [12].

4. APPROACH

There are two main shortcomings that lie in the design of scene graph transformers. First, SGT's scene-graph attention uses the standard sinusoidal positional encoding [13], which

	VG-MSDN				COCO-stuff			
	Object		Relation		Object		Relation	
	rAVG ↓	Hit@ 1 / 5 ↑	rAVG ↓	Hit@ 1 / 5 ↑	rAVG ↓	Hit@ 1 / 5 ↑	rAVG ↓	Hit@ 1 / 3 ↑
LTNet [19]	24.45	13.9 / 34.8	4.70	34.8 / 74.6	17.22	20.1 / 45.8	2.36	29.1 / 78.4
GTwE [20]	11.91	27.0 / 57.2	5.36	35.8 / 72.5	11.81	28.4 / 57.2	2.89	20.4 / 63.3
SGT [12]	8.38	39.7 / 68.9	3.43	55.3 / 84.3	11.03	29.6 / 59.0	2.19	45.5 / 82.2
Ours	**6.77**	**44.2 / 70.2**	**3.22**	**59.8 / 85.8**	**10.12**	**32.1 / 61.3**	**1.92**	**47.3 / 83.4**

Table 1. Quantitative evaluation on scene graph expansion.

is one-dimensional, i.e., order-dependent; See Fig. 2 (a-d) for illustration. We propose to use the Laplacian positional encoder to generate positional encoding features that are able to reflect the structural position of a given scene graph. Second, the scene graph attention solely focuses on edges with relationship annotations. This causes the transformer to take multiple layers to understand a whole graph. We introduce depth-aware scene graph attention so that our transformer can learn a whole graph in each and every single layer.

4.1. Laplacian Positional Encoder

The Laplacian Positional Encoder utilizes Laplacian positional encoding (LPE) [20, 21] to generate the position encoding feature from its input tokens, including node tokens $\{n_i : i = 1 : O\}$ and edge tokens $\{e_{ij} : i, j = 1 : O\}$.

For the positional encoding of each node token, we first compute the normalized Laplacian matrix:

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}, \tag{3}$$

where $\boldsymbol{A}$ and $\boldsymbol{D}$ denote adjacency and degree matrix of the input scene graph $\boldsymbol{G} = (\boldsymbol{O}, \boldsymbol{R})$.

We denote $\Lambda \in \mathbb{R}^{N\times N}$, i.e., the eigenvectors of L, as a function of input graph $\Lambda = \Lambda(\boldsymbol{O}, \boldsymbol{R})$. We noted that N is usually ≤ 30, which is insignificant in comparison to the channel size typically used in conventional neural models. Hence, we extract higher dimensional features from the Laplacian positional encoding (LPE) by adding an MLP on top of it, i.e.,

$$\text{LPE}(n_i) \triangleq \text{MLP}_\text{N}(\Lambda_i). \tag{4}$$

As for the position encoding of each edge token e_{ij}, we aggregate the positional encoding of the associated nodes, i.e.,

$$\text{LPE}(e_{ij}) \triangleq \text{MLP}_\text{E}(\Lambda_i \oplus \Lambda_j), \tag{5}$$

where $\oplus$ denotes the concatenation operation.

4.2. Multiscale Graph Attention

For each layer in our Transformer, the attention module takes in the input tokens, computes their self-attention, and updates the tokens, which are used in the next layer.

	VG-MSDN	COCO-stuff
	mIoU	mIoU
GTwE [20]	12.3 / 79.9 / 62.1	21.3 / 73.2 / 64.8
SGT [12]	14.5 / 81.1 / 62.4	28.2 / 85.1 / 74.9
Ours	**16.1 / 83.2 / 63.5**	**30.7 / 87.2 / 77.1**

Table 2. Results on scene graph to layout.

The transformer can be viewed as a sequence of multiple message-passing layers. Each layer in our multiscale graph attention can be described as follows,

$$\hat{\boldsymbol{f}}_i^o = \Bigg(\underbrace{\sum_{s=1}^{M} \beta^{s-1}\boldsymbol{T}^s}_{\boldsymbol{T}^*}\Bigg)_{ij} v(\boldsymbol{f}_j^o). \tag{6}$$

Here we introduce the multi-scale attention weight matrix $\boldsymbol{T}^* \in \mathbb{R}^{N\times N}$ which is generated from our multiscale graph attention: where β and M are constants controlling the impact of non-neighboring nodes on the attention. Note, the standard graph attention $\boldsymbol{T}$ is simply a special case with $M = 1$.

In comparison to $\boldsymbol{T}$, the multiscale graph attention $\boldsymbol{T}^*$ is able to generate an attention matrix with a global view of the whole graph. To simulate the propagation of the information passing down through multiple transformer layers in a single layer, we take inspiration from the random walk Markov matrix. A simple example is demonstrated in Fig. 2 (e-h).

4.3. Implementation Details

For a fair comparison, we strictly follow the data preprocessing and training hyperparameters in [12]. For Laplacian Positional Encoders, each MLP has three layers with the output channel size of 256. For depth-aware scene graph attention, we set $M = 4$ and $\beta = 0.5$.

5. EXPERIMENTS

To show the effectiveness of the proposed methods fairly, we follow the experiments setting in [12]. We report the performance on SGE and L2I to LTNet [19], GTwE (Graph Transformer with Edge features [20]), SGT [12], ours. We compare our method to several baselines, including, Outpainting-SRN [10], Boundless [16], AttSpade [22], and SGT [12].

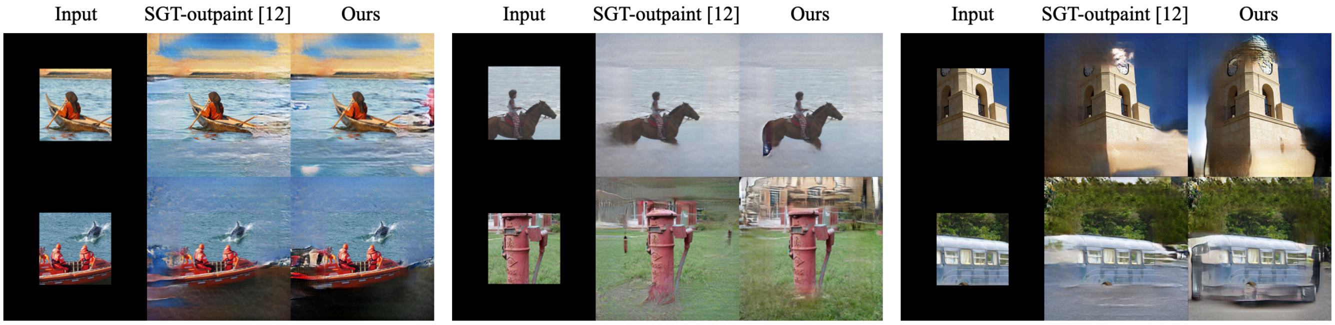

Fig. 3. **Visualization on VG-MSDN**. From left to right: input image $\boldsymbol{X}$, output image $\boldsymbol{Y}$ from [12], and $\boldsymbol{Y}$ from ours.

5.1. Datasets

VG-MSDN. The VG-MSDN [23, 24] dataset annotates images with scene graphs, including 150 object categories and 50 relationship categories. Following the standard split ratio, we use 45K images for training and 10k images for testing.

COCO-stuff. There are 171 object categories in the COCO-stuff [25, 26] dataset. The scene graph annotations are generated by following the rule-based preprocessing technique of [22, 19]. We use 118K images for training and 50k images for testing.

5.2. Scene Graph Predication

We conduct experiments on both scene graph prediction tasks, i.e., SGE and G2L, to show the performance gain of our improved transformer design.

For SGE: we report the top-k accuracy (Hit@k) and average ranking score (rAVG) of its scene graph prediction. **For G2L:** we report the mean intersection over union (mIoU) of the generated bounding boxes. Specifically, we report three mIoU scores. The first score only considers objects that are presented in the input image. The second score only considers those that are novel. The third score reports the total average over all objects.

In Tab. 1, we compare our approach to SGT [12] and other methods [19, 20] for scene graph expansion (SGE) for both datasets. In Tab. 2, our methods and [20, 12] are compared on layout generation tasks (G2L). Specifically, on scene graph object generation, we improve by about 1.00 on rAVG and 3-5% on Hit@1. On the other hand, we improve roughly 2.0% on mIoU of novel objects on bounding box prediction.

5.3. Image Outpainting

Although we do not make any modification to the layout to image (L2I) module in SGT-outpaint [12], we did observe that the qualitative of the semantic outpainted images can be greatly improved with more accurate SGE and G2L.

To quantify the performance of L2I, we measure the FID [27] of the generated images and the ground truth images. The FID shows the perceptual distance between the two sets. Hence the lower the FID score is, the more natural the images are.

VG-MSDN / FID	
Outpaint-SRN [10]	66.89
AttSpade [22]	68.91
Boundless [16]	77.86
SGT-outpaint [12]	60.99
Ours	**58.71**

Table 3. **Results on L2I.**

VG-MSDN / Hit@1 Acc.	
SGT [12]	39.8
+Laplacian PE	43.0
+Multiscale Atten.	42.1
+both	**44.2**

Table 4. **Ablation studies.**

As shown in Tab. 3, our method with better scene graph prediction leads to about 2.0 performance gain on FID. In Fig. 3, we observe that our approach extends the boundaries of the existing objects more accurately than SGT [12], such as the rear and feet of the horse and the chassis of the bus.

5.4. Ablation Study

To demonstrate the efficacy of each of our proposed modules, we removed different modules from the SGT for scene graph object prediction. As shown Tab. 4, by only adding Laplaican positional encoder to the baseline [12], the model improves by 3.2% on accuracy with richer structural information. Next, incorporating multiscale graph attention improves the accuracy by 2.3%. Our best performance is achieved by having both modules together arriving at a performance gain of 4.4%.

6. CONCLUSION

We address two underlying problems in SGT [12], (a) the positional encoding was not designed for graph-structural signals and (b) the attention propagates information locally between neighboring nodes making it challenging to learn large graphs. We propose Laplacian positional encoders and multiscale graph attention tackling the two problems respectively. In our experiments, we show that the proposed positional encoding and multi-scale attention are both effective and robust with quantitative and qualitative comparisons.

7. ACKNOWLEDGEMENT

This work is supported in part by the Ministry of Science and Technology of Taiwan under grant MOST 110-2634-F-002-036 and in part by Qualcomm Technologies, Inc. through a Taiwan University Research Collaboration Project.

References

[1] Zongxin Yang, Jian Dong, Ping Liu, Yi Yang, and Shuicheng Yan, "Very long natural scenery image prediction by outpainting," in *ICCV*, 2019. 1, 2

[2] Zhenqiang Ying and Alan C. Bovik, "180-degree outpainting from a single image," *CoRR*, 2020. 1

[3] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *WACV*, 2022. 1

[4] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 1

[5] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa, "Globally and locally consistent image completion," in *ACM Trans. Graph.*, 2017. 1

[6] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *ECCV*, 2018. 1

[7] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang, "Free-form image inpainting with gated convolution," in *ICCV*, 2019. 1

[8] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *ICCV Workshops*, 2019. 1

[9] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo, "Foreground-aware image inpainting," in *CVPR*, 2019. 1

[10] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia, "Wide-context semantic image extrapolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 4

[11] Mark Sabini and Gili Rusak, "Painting outside the box: Image outpainting with gans," *CoRR*, 2018. 1, 2

[12] Chiao-An Yang, Cheng-Yo Tan, Wan-Cyuan Fan, Cheng-Fu Yang, Meng-Lin Wu, and Yu-Chiang Frank Wang, "Scene graph expansion for semantics-guided image outpainting," in *Proc. CVPR*, 2022. 1, 2, 3, 4

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017. 1, 2

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial networks," *Communications of the ACM*, 2020. 2

[15] Chia-Ni Lu, Ya-Chu Chang, and Wei-Chen Chiu, "Bridging the visual gap: Wide-range image blending," in *CVPR*, 2021. 2

[16] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T Freeman, "Boundless: Generative adversarial networks for image extension," in *ICCV*, 2019. 2, 3, 4

[17] Cheng-Yo Tan, Chiao-An Yang, Shang-Fu Chen, Meng-Lin Wu, and Yu-Chiang Frank Wang, "Robust image outpainting with learnable image margins," in *ICIP*, 2021. 2

[18] Bholeshwar Khurana, Soumya Ranjan Dash, Abhishek Bhatia, Aniruddha Mahapatra, Hrituraj Singh, and Kuldeep Kulkarni, "Semie: Semantically-aware image extrapolation," in *ICCV*, 2021. 2

[19] Cheng-Fu Yang, Wan-Cyuan Fan, Fu-En Yang, and Yu-Chiang Frank Wang, "Layouttransformer: Scene layout generation with conceptual and spatial diversity," in *CVPR*, 2021. 3, 4

[20] Vijay Prakash Dwivedi and Xavier Bresson, "A generalization of transformer networks to graphs," *arXiv preprint arXiv:2012.09699*, 2020. 3, 4

[21] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, 2003. 3

[22] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson, "Learning canonical representations for scene graph to image generation," in *ECCV*, 2020. 3, 4

[23] 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," in *IJCV*, 2017. 4

[24] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang, "Scene graph generation from objects, phrases and region captions," in *ICCV*, 2017. 4

[25] 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 *ECCV*, 2014. 4

[26] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari, "Coco-stuff: Thing and stuff classes in context," in *CVPR*, 2018. 4

[27] 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," in *NIPS*, 2017. 4