LayoutTransformer:
Scene Layout Generation with Conceptual and Spatial Diversity

Cheng-Fu Yang[1*] Wan-Cyuan Fan[1*] Fu-En Yang[1,2] Yu-Chiang Frank Wang[1,2]
[1]Graduate Institute of Communication Engineering, National Taiwan University, Taiwan
[2]ASUS Intelligent Cloud Services, Taiwan
{b05901082, r09942092, f07942077, ycwang}@ntu.edu.tw

Abstract

When translating text inputs into layouts or images, existing works typically require explicit descriptions of each object in a scene, including their spatial information or the associated relationships. To better exploit the text input, so that implicit objects or relationships can be properly inferred during layout generation, we propose a LayoutTransformer Network (LT-Net) in this paper. Given a scene-graph input, our LT-Net uniquely encodes the semantic features for exploiting their co-occurrences and implicit relationships. This allows one to manipulate conceptually diverse yet plausible layout outputs. Moreover, the decoder of our LT-Net translates the encoded contextual features into bounding boxes with self-supervised relation consistency preserved. By fitting their distributions to Gaussian mixture models, spatially-diverse layouts can be additionally produced by LT-Net. We conduct extensive experiments on the datasets of MS-COCO and Visual Genome, and confirm the effectiveness and plausibility of our LT-Net over recent layout generation models. Codes will be released at LayoutTransformer.

1. Introduction

Text-to-image (T2I) generation takes the input text descriptions and converts them into realistic images, which would benefit a massive number of applications including computer-aided design, art generation and image editing emerging, it attracts the attention from researchers in computer vision and deep learning communities. In addition to the need to output high-quality images [19, 26, 27], the main challenge in T2I lies in the generation of plausible images, with semantic relationships preserved across different objects. Thus, how to bridge the gap between semantic and perceptual information requires the efforts from researchers in the fields of computer vision and machine learning.

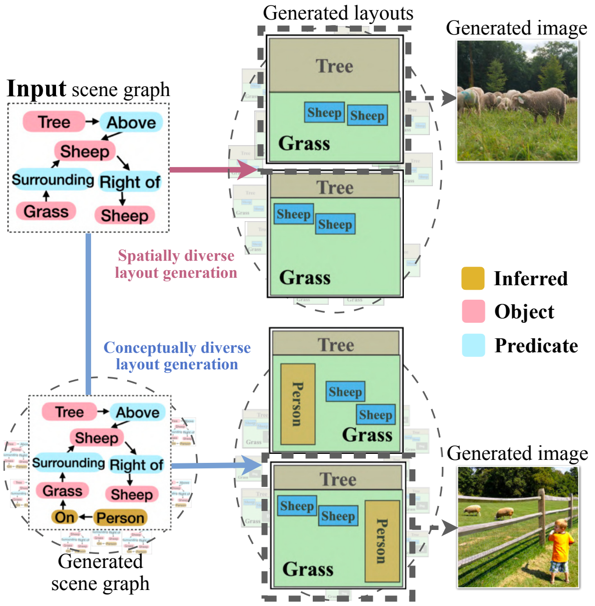

Figure 1. Layout generation with conceptual and spatial diversity. Given a scene graph input, we aim to produce diverse plausible layouts with the ability to exploit implicit objects/relations.

A common challenge in text-to-layout or image is that, objects and their relations described in text sentences may not be easily described, which would result less plausible outputs. Therefore, Johnson et al. [8] choose to convert textual input into scene graph (SG) as an intermediate text representation, which explicitly defines the objects and their relationships in a scene. Generally, the task of SG-to-image can be decomposed into the following two steps: SG-to-layout [6, 9, 12] and layout-to-image [13, 25, 21, 1] generation. The former focuses on modeling the geometric properties of objects while retaining their semantic characteristics. The latter targets at synthesizing realistic images conditioned on the given layout configuration. To address text-to-layout generation, [9] apply a variational au-

[*] Equal Contribution

	VAE based	GCN based				Transformer based
	LayoutVAE	Sg2Im	Grid2Im	NDN	CanonicalSg2Im	Ours
Layout Configuration	-	✓	✓	✓	✓	✓
Inferring spatial relation	-	-	✓	✓	✓	✓
Generating semantic-equivalent relation	-	-	-	-	✓	✓
Complex scene graph	-	-	-	-	✓	✓
Spatially-diverse Layout Generation	✓	-	-	✓	✓	✓
Conceptually-diverse Layout Generation	-	-	-	-	-	✓

Table 1. Comparisons of recent approaches on scene graph to layout generation.

toencoder (VAE) to model the layout distribution of objects. Alternatively, [12, 6] utilize Graph Convolution Network (GCN) [5] to extract the semantic information of the input SG and utilize VAE to regress the output layouts.

Although the use of SG allows possible inference of implicit relationships between objects in a scene, it cannot easily exploit and handle static objects or background, and thus the resulting layout might not be satisfactory. For example, the SG triplet of *Man-walks-Dog* implies the potential objects of *tree, grass, pavement*, since such static objects or those in the background are related to *walk* and *dog*. Such co-occurrences are conceptually realistic and can be possibly exploited during training. Moreover, given a properly derived SG, how to produce diverse layouts (i.e., those with different yet plausible compositions) and avoid possible mode collapse problems would be among the difficulties in text-to-layout generation.

To overcome the above challenges, we propose a novel LayoutTransformer Network (LT-Net) in this paper. Following [8], our LT-Net starts from SG as the input text description. In order to exploit the implicit objects or relations in a SG input, we present a masked language model (MLM) based on BERT [4]. Preserving the input graph characteristics, our MLM uniquely learns the contextual representation with object co-occurrence information exploited, allowing generation of *conceptually related yet diverse* outputs. As for layout synthesis, we advance the transformer decoder which sequentially produces the bounding boxes for each object/relation, whose distribution is modeled by Gaussian Mixture Models [20]. This decoder design allows *spatially diverse* layout components. Finally, a visual-textual co-attention module is deployed for layout refinement.

We now highlight the contributions as follows:

- We propose a novel framework of LayoutTransformer Network (LT-Net), which takes scene-graph inputs for layout generation, with the ability to produce *conceptually* and *spatially* diverse outputs.

- Our LT-Net jointly encodes object/relation, pair and sentence-wise information from SG inputs, which not only learns to recover the semantic information also exploits implicit objects and relations for conceptual diversity guarantees.

- Our decoder utilizes Gaussian mixture models to describe spatial outputs for each object/relation, modeling the desirable distributions.

- Visual-textual co-attention is performed to refine the layout output, which jointly observes the conditions of derived semantic and spatial representations.

2. Related works

2.1. Text-to-image synthesis

Generating realistic images from text descriptions benefits a wide range of computer vision applications. [19] propose an end-to-end trainable network generating image conditioned on sentence description. [26] use a two-stage GAN to progressively generate images with higher resolution. Following [24], they design a cross-modality attention module with an eye to align the content of the generated image and the conditioned text. [7] decompose the generating process into multiple stages. They first predict the objects and their layout in the scene, then construct the segmentation masks conditioned on the predicted layout and image. Recently, [14] present a novel object-level attention mechanism to generate semantically meaningful images. Nevertheless, existing methods might not be able to describe and manipulate the image content in terms of the composition and particular attributes.

2.2. From scene graphs to layouts or images

For the task of layout generation, [9] propose LayoutVAE which takes a set of object labels as inputs and predicts both the number of instances of each category, as well as the corresponding scene layout. Since the input of LayoutVAE might not sufficiently describe the scene of interest, the use of scene graphs (SG) as an intermediate text representation [8] is generally preferable. For example, [12] propose neural design networks (NDN) by integrating graph convolution network (GCN) and conditional VAE to generate design layouts from the given user-specified constraints. However, it is only designed to model a limited number of classes and relationships. [1, 3, 25, 17, 22] also struggle

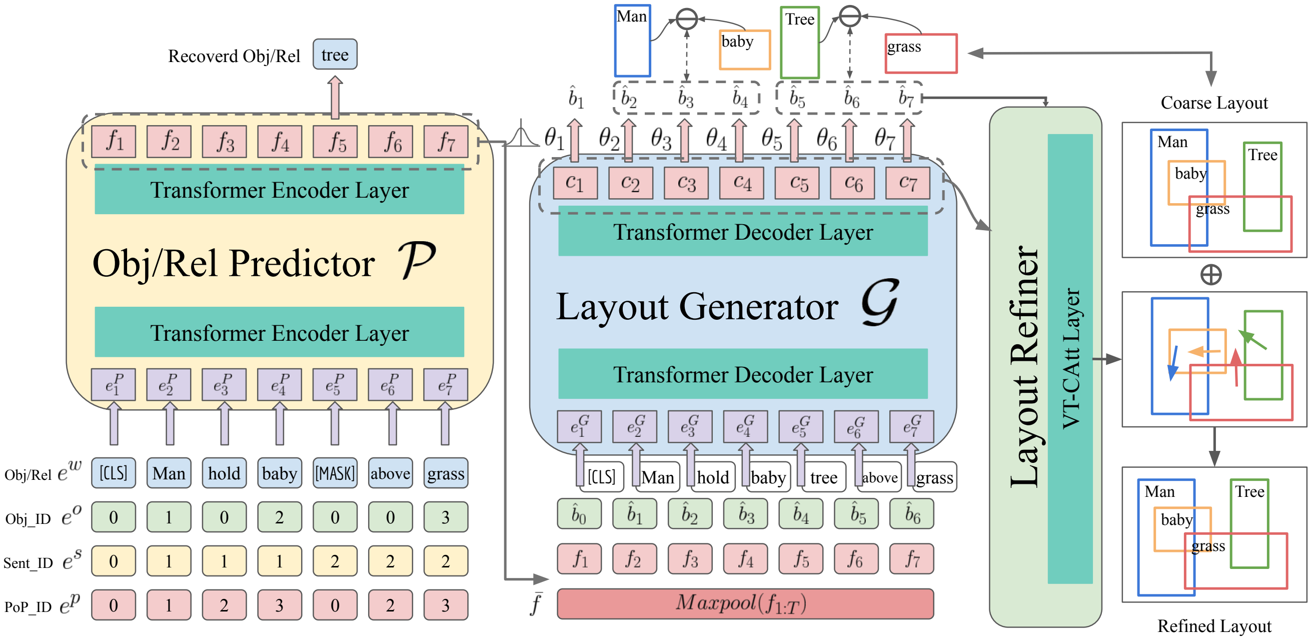
Recoverd Obj/Rel
tree
f_1 f_2 f_3 f_4 f_5 f_6 f_7
Transformer Encoder Layer
Obj/Rel Predictor $\mathcal{P}$
Transformer Encoder Layer
e^P_1 e^P_2 e^P_3 e^P_4 e^P_5 e^P_6 e^P_7
Obj/Rel e^w
[CLS] Man hold baby [MASK] above grass
Obj_ID e^o
0 1 0 2 0 0 3
Sent_ID e^s
0 1 1 1 2 2 2
PoP_ID e^p
0 1 2 3 0 2 3
$\bar{f}$
Man
baby
Tree
grass
$\hat{b}_1$ $\hat{b}_2$ $\hat{b}_3$ $\hat{b}_4$ $\hat{b}_5$ $\hat{b}_6$ $\hat{b}_7$
θ_1 θ_2 θ_3 θ_4 θ_5 θ_6 θ_7
c_1 c_2 c_3 c_4 c_5 c_6 c_7
Transformer Decoder Layer
Layout Generator $\mathcal{G}$
Transformer Decoder Layer
e^G_1 e^G_2 e^G_3 e^G_4 e^G_5 e^G_6 e^G_7
[CLS] Man hold baby tree above grass
$\hat{b}_0$ $\hat{b}_1$ $\hat{b}_2$ $\hat{b}_3$ $\hat{b}_4$ $\hat{b}_5$ $\hat{b}_6$
f_1 f_2 f_3 f_4 f_5 f_6 f_7
$Maxpool(f_{1:T})$
Layout Refiner
VT-CAtt Layer
Coarse Layout
Man
baby
Tree
grass
$\oplus$
Man
baby
Tree
grass
Refined Layout

Figure 2. Architecture of LayoutTransformer Network (LT-Net), with modules of Object/Relation Predictor $\mathcal{P}$, Layout Generator $\mathcal{G}$, and Layout Refiner (Visual-Textual Co-Attention). Note that predictor $\mathcal{P}$ encodes the input scene graph in terms of different semantic attributes into contextual features f. Generator $\mathcal{G}$ interprets such contextual features into layout-aware representation c, which predicts the bounding box information b with distributions matching the learned Gaussian distribution models θ. Finally, the co-attention module jointly observes the generated bounding boxes and the contextual representations for refining the final layout. See Section 3 for detailed discussions.

as the complexity of the SG increases. Recently, [6] learns canonical graph representations from the data, resulting in performance improvement on layout generation with complex scenes. Nevertheless, existing layout generation approaches generally are not designed to observe or model implicit objects and relationships from the given SG. Thus, conceptually diverse layouts cannot be produced.

As for layout-to-image generation, [13, 29] utilize GAN to generate layouts from the component attributes, while [1, 25, 21, 17] are designed to handle configurable inputs. Nevertheless, the above methods cannot infer and synthesize implicit objects or background information. In Table 1, we compare the characteristics of recent layout/image generation approaches and highlight the novelty of our LT-Net.

3. Methodology

3.1. Notations and Algorithmic Overview

For the sake of completeness, we first define the notations to be used in this paper. To convert SG into the input format required by the transformer network [23], we convert each *subject-relation-object* triplet in the SG into a sequential data in a randomly order $S = \{s_1, s_2, ...s_T\}$ (T denotes the number of words) and separate triplets with a special token of "SEP". Given such inputs, our goal is to produce plausible yet diverse layouts in terms of bounding boxes $B = \{b_1, b_2, ..., b_T\}$. To achieve this, we propose a *LayoutTransformer Network (LT-Net)*, as shown in Fig. 2. In LT-Net, we have a unique object/relation predictor $\mathcal{P}$ for modeling the semantic information from the observed sequential input $s_{1:T}$, which extracts the contextualized representations $f_{1:T}$ describing the associated semantic information, with $\bar{f}$ max-pooled over $f_{1:T}$) describing that of the input scene. On the other hand, the module $\mathcal{G}$ in our LT-Net serves as a decoder/generator, containing a layout feature extractor $\mathcal{F}$, followed by a GMM components parameterized by θ. This is to model and produce diverse layouts $B = \{b_1, b_2, ...b_T\}$ from the learned contextualized representations of each object. Finally, by jointly observing the sequentially-produced layout and the contextual representation of the entire scene, we introduce a Visual-Textual Co-Attention (VT-CAtt) module for layout refinement with *spatial* and *semantic* information guarantees.

It is worth repeating that, our LT-Net not only generates spatially-diverse layouts given the SG input, it also learns *co-occurrences* among objects and relationships, so that conceptually diverse outputs exploiting implicit object/relation information can be achieved.

3.2. Learning Object/Relation-aware Embedding

Given a SG input S, our Object/Relation Predictor $\mathcal{P}$ derives contextualized representations $f_{1:T} = \{f_1, f_2, ..., f_T\}$

for each object/relation, with the goal of describing its semantic and spatial information, exhibiting the ability in inferring the implicit objects or relations in the given SG.

Instead of applying standard recurrent models for encoding S, we have $\mathcal{P}$ embed $s_{1:T}$ into a relation-aware and object-discriminative embedding $e^P_{1:T}$ by decomposing $s_{1:T}$ into different types of features: word embedding $e^w_{1:T}$, object ID embedding $e^o_{1:T}$, sentence ID embedding $e^s_{1:T}$, and part-of-pair (PoP) ID embedding $e^p_{1:T}$. Following [4], the *word embedding* e^w_t describes the features of the tth object/relation. The *object ID embedding* e^o_t, as depicted in Fig. 2, is expressed order numbers which distinguish between different instances of the same object category (i.e., with the same e^w_t). The *sentence ID embedding* e^s_t is to identify different triplets in the input SG. In order to specify the *semantic role* (i.e., subject, relation, or object) s_t in each sentence of the input, we uniquely utilize the *Part-of-Pair (PoP) ID* e^p_t for each s_t in a sentence.

We concatenate the above semantic features to form the embedding $e^P_{1:T} = [e^w_{1:T} \oplus e^s_{1:T} \oplus e^p_{1:T} \oplus e^o_{1:T}]$, which serves as the input of our relation predictor $\mathcal{P}$ for learning the contextualized feature vectors $f_{1:T}$. To allow $\mathcal{P}$ for capturing conceptually diverse embedding and exploiting the co-occurrence among objects and relationships, we follow BERT [4] and adapt the masked language model to train the predictor $\mathcal{P}$. Specifically, we randomly choose 45% of the input triplets, with uniform probability for each *subject, relation, object* in the selected triplet to be masked. Moreover, if the t-th word is chosen, it would be replaced by the token "MASK" for 80% of the time, while it remains unchanged for 20% of the time. This strategy allows our LT-Net to preserve the semantic properties of the unmasked word. Finally, we have the output contextual features f_t to predict $\hat{s_t}$ via a single linear layer.

We note that, our predictor $\mathcal{P}$ not only outputs the masked words, it also recovers their object and PoP IDs. Thus, the objective function $\mathcal{L}_{pred}$ for training $\mathcal{P}$ observes the cross-entropy losses for the matching word, object ID, and PoP ID between the masked input s_t and the reconstructed $\hat{s_t}$, which is calculated by:

$$\mathcal{L}_{pred} = CrossEntropy(s_t, \hat{s_t}). \tag{1}$$

With the contextual feature embedding $f_{1:T}$ output by $\mathcal{P}$, we further perform max-pooling over $f_{1:T}$ into $\bar{f}$, which would serve as the contextual representation of the entire input SG (for later layout generation and refinement purposes.

3.3. Stochastic Layout Generation

3.3.1 Generative model based on Gaussian Mixture Models

The layout generator $\mathcal{G}$ aims to produce layout bounding boxes in a scene. Extended from a transformer-based decoder, $\mathcal{G}$ jointly and sequentially interprets semantic and spatial information, and translates them into diverse bounding box outputs $B = \{b_1, b_2, ..., b_T\}$. For each subject and object in $f_{1:T}$, its output is defined as the location and size of the associated bounding box, i.e., $b_t = (x_t, y_t, w_t, h_t)$, $b_t \in \{b^{sub}, b^{obj}\}$. As for the relation words, we output the box disparity between its associated subject and object pair instead, i.e., $b_t = (\Delta x_t, \Delta y_t)$, $b_t \in \{b^{rel}\}$. We now discuss how our generator $\mathcal{G}$ performs this process.

For each word, our $\mathcal{G}$ jointly takes the observed contextual features f_t, $\bar{f}$ and previously predicted layout b_{t-1} as the input e^G_t. By translating such inputs into the corresponding layout-aware contextual representation c_t, the associated bounding box b_t can be predicted accordingly. It is worth pointing out that, in the aforementioned input e^G_t, we do not directly apply the spatial coordinates and sizes to represent the bounding box feature b_{t-1}. Instead, we project such values into the same dimensional feature as those of the contextual features f_t and $\bar{f}$.

With derived c_t, it would be transformed int bounding box information b_t. However, in order to introduce the generative ability to our model, we do not directly map c_t into b_t. Inspired by [7, 14], we choose to model the distribution of each c_t by a *Gaussian Mixture Models (GMM)*. Then, each bounding box will be sampled from its corresponding posterior distribution $p_{\theta_t}(b_t \mid c_t)$, which is described by K multivariate normal distributions with i indicating the i-th distribution. Each distribution is parameterized by $\theta_{t,i}$ and a magnitude factor π_i. Mathematically, we have

$$\begin{gathered} p_{\theta_t}(b_t \mid c_t) = \sum_{i=1}^{K} \pi_i \mathcal{N}(b_t; \theta_{t,i}), \\ \theta_{t,i} = (\mu^x_{t,i}, \mu^y_{t,i}, \sigma^x_{t,i}, \sigma^y_{t,i}, \rho^{xy}_{t,i}), \sum_{i=1}^{K} \pi_i = 1, \end{gathered} \tag{2}$$

where $\mathcal{N}(b_t; \theta_{t,i})$ denotes the multivariate normal distribution. Note that for θ_t, we have μ^x and μ^y denote the means, σ^x and σ^y as the standard deviations, and ρ^{xy} as the correlation coefficient, which describe the multivariate normal distributions for the associated word (and bounding box).

To realize the above objective, we consider the bounding box reconstruction loss $\mathcal{L}_{box}$, which maximizes the log-likelihood of the generated GMM to fit that observed from the training data. More precisely, we have $\mathcal{L}_{box}$ calculated using the generated GMM parameters θ_t and the location of the ground-truth bounding box $\hat{b_t} = (\hat{x_t}, \hat{y_t}, \hat{w_t}, \hat{h_t})$:

$$\mathcal{L}_{box} = -\frac{1}{K} \log(\sum_{i=1}^{K} \pi_i \mathcal{N}(\hat{x_t}, \hat{y_t}, \hat{w_t}, \hat{h_t}; \theta_{t,i})). \tag{3}$$

We observe that, however, the above optimization task tends to suffer from over-fitting problems (e.g., degeneration to a Dirac delta function). Thus, as regularization, we additionally fit the GMM distributions to a multivariate normal

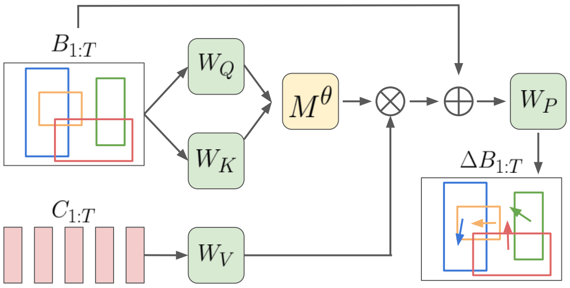

Figure 3. Architecture of our Visual-Textual Co-Attention (VT-CAtt) module. B denotes the coarse layout synthesized by $\mathcal{G}$, and C represents the contextual vectors produced by $\mathcal{F}$ in LT-Net. M^θ indicates the refined attention weights, while W_Q, W_K, W_V, and W_P are to be learned for performing co-attention. Note that ΔB is the output describing the residual for each bounding box.

distribution Q (with the same mean same as θ_t and unit variance) by calculating $\mathcal{L}_{KL} = \sum_{i=1}^{K} D_{KL}(P_i \| Q_i)$.

3.3.2 Self-supervised relation consistency

In addition to the above objectives associated with bounding box generation, we observe and enforce a novel Relation Consistency Loss $\mathcal{L}_{rel}$ as a self-supervision guidance. That is, we additionally observe the consistency between the box disparity of the relation word b^{rel} and that of the corresponding subject-object pair $\Delta b = ((b_x^{sub}, b_y^{sub}) - (b_x^{obj}, b_y^{obj}))$. Thus, this loss is calculated by the Mean Square Error (MSE) between box disparity $\mathit{\Delta} b$ and b^{rel}:

$$\mathcal{L}_{rel} = \frac{1}{N} \sum (\mathit{\Delta} b - b^{rel})^2, \tag{4}$$

where N is the number of the relation pairs in S.

With the above objectives, our layout generator $\mathcal{G}$ can be trained by minimizing the following loss terms:

$$\mathcal{L}_{gen} = \mathcal{L}_{box} + \lambda_{KL}\mathcal{L}_{KL} + \mathcal{L}_{rel}, \tag{5}$$

where λ_{KL} indicates the regularization weight and is fix as 0.1 in this work for simplicity.

3.4. Visual-Textual Co-Attention for Layout Refinement

Recall that, as depicted in Fig. 2, the coarse layout output $B_{1:T} = \{b_1, b_2, ...b_T\}$ is generated for each individual bounding box in a sequential fashion. Thus, it would be desirable to refine such outputs into the final layout, so that the complete set of bounding boxes and their associated semantic information would be jointly exploited. To achieve this, we present an Visual-Textual Co-Attention (VT-CAtt) mechanism, which predicts the residual $\Delta B_{1:T}$ for updating each bounding box, leading to the final output $B'_{1:T}$.

The module of VT-CAtt is depicted in Fig. 3. We see that such co-attention is based on the inputs of the coarse layout $B_{1:T}$ (as the visual feature) and the layout-aware contextual representation $C_{1:T} = \{c1, c2, ...c_T\}$ (as the semantic feature). For the depicted self-attention mechanism, the former is utilized as both *query* and *key*), while the latter is taken as *value*). More specifically, we perform matrix multiplication to the projection of *query* $W_Q(B)$ and *key* $W_K(B)$ to obtain the attention matrix M. We note that, since we generate the coarse bounding box B from the GMM distribution, the sampled bounding boxes with low probabilities are implied less likely to be the desirable spatial outputs. Therefore, we take the sampled GMM probability value as the confidence value ϵ for the associated bounding box. Thus, the resulting GMM-aware co-attention weight M^θ is calculated as:

$$M_{i,j}^\theta = \frac{\epsilon_j \cdot \exp(M_{i,j})}{\sum_{j=1}^{T} \epsilon_j \cdot \exp(M_{i,j})}, \tag{6}$$

where $M_{i,j}^\theta$ denotes the contribution of the j^{th} object to the i^{th} object in the layout, and ϵ_j is derived from calculating the probability density of the coarse bounding box b_j (i.e., $p_{\theta_j}(b_j)$). We feed the course B and the feature vectors produced by VT-CAtt to a single linear layer to predict ΔB. Similar to [18], we calculate the following refinement loss $\mathcal{L}_{ref}$ for updating this module:

$$\begin{aligned}\mathcal{L}_{ref} = \sum_{t=1}^{T} \lambda_{xy}[(x'_t - \hat{x}_t)^2 + (y'_t - \hat{y}_t)^2] \\ + \lambda_{wh}[(\sqrt{w'_t} - \sqrt{\hat{w}_t})^2 + (\sqrt{h'_t} - \sqrt{\hat{h}_t})^2],\end{aligned} \tag{7}$$

where $b'_t = (x'_t, y'_t, w'_t, h'_t)$ denotes each refined bounding box, and $\hat{b}_t$ represents the ground-truth one. With the objectives defined in equations (1), (5) and (7), our LT-Net can be trained accordingly.

4. Experiments

4.1. Datasets

COCO-stuff. We perform our experiments on the COCO-Stuff dataset [2], which augments a subset of the COCO dataset [16] with additional stuff categories. Thus, a total of 80 *thing* categories (car, dog, etc.) and 91 *stuff* categories (sky, snow, etc.) are available, with 118K/5K annotated images for training/validation. For the relationship annotations, we refer to Sg2Im [8], which utilizes coordinates of the objects in images to construct synthetic scene relationship. Following the definitions of the geometric relationships in Sg2Im, a total of six relationships are considered: *left of, right of, above, below, inside, and surrounding*.

Visual Genome. The original Visual Genome dataset [10] contains a large number of noisy annotation.

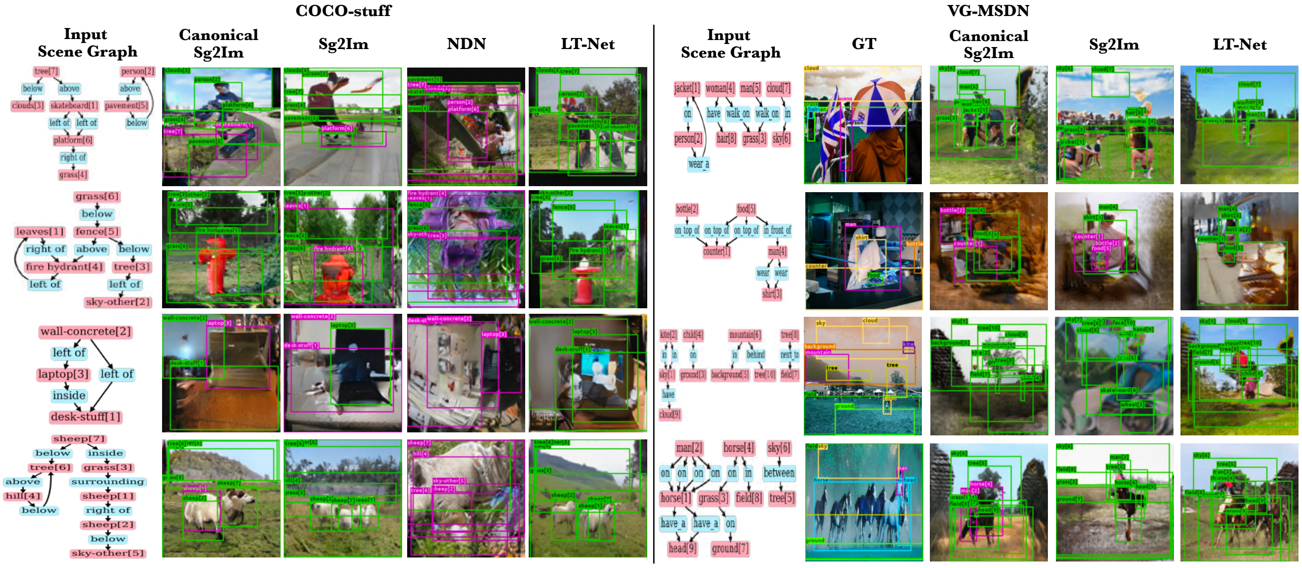

Figure 4. **Qualitative evaluation on COCO-Stuff and VG-MSDN.** Each row shows the scene-graph input, ground truth layout and those generated by different approaches. For visualization purposes, we apply the pretrained LostGAN [21] to convert the output layout into images. Note that bounding boxes in **green** denote the layout components matching the given description, while those in **red** do *not*.

Thus, we consider the VG-MSDN dataset [15] instead, which comprises more than 46K training images and a testing set with 10K images. Note that Visual Genome deals with a more challenging setting since 50 types of relation are regarded, including both semantic and spatial relation words such as verbs and prepositions. Please refer to our supplementary material for the details about this dataset.

4.2. Qualitative results

Plausible layout generation. We compare our proposed LT-Net with recent state-of-the-art models, including Sg2Im [8], NDN [12] and CanonicalSg2Im [6], with results shown in Fig. 4. From this figure, we observe that the outputs of CanonicalSg2Im, Sg2Im, and NDN did not necessarily match the relations between the objects (i.e., bounding boxes shown in red), while our LT-Net was able to generate consistent layout components with the given textual descriptions (i.e., bounding boxes shown in green), especially on the more challenging dataset of VG-MSDN.

Spatially-diverse layout generation. In Fig. 5(a), we show example layout generation results given the same scene graph input. We see that spatial diversity can be produced by our model, while semantic plausibility is preserved. It is worth repeating that, this is due to our modeling of GMM distributions at the decoder layer, which turns LT-Net into a generative model for output stochasticity. Additional qualitative results and comparisons with previous layout generation works on spatial diversity can be found in our supplementary materials. More importantly, we later quantitatively compare to recent methods on this and present the results (in terms of diversity scores) in Table 2.

Conceptually-diverse scene graph generation. A key novelty of our LT-Net is to manipulate implicit objects and relations in the input SG. As shown in Fig. 5(b), our model successfully added additional objects like *person* or *grass* which is not explicitly presented in the input. More importantly, with such inferred objects/relations, the final layouts still exhibit sufficient plausibility. Please refer to our supplementary materials for more qualitative results.

4.3. Quantitative results

4.3.1 Evaluation metrics

For quantitative evaluation, we consider the following different four metrics:

(1) **Mean intersection over union (mIOU).** The mIOU score measures how well the generated layout fits the ground truth bounding box information.

(2) **Relation accuracy.** The relation accuracy only considers the relation with explicit spatial meaning (i.e., *left of, right of, above and below*). Since the Visual Genome dataset contains the relation other than these spatial description such as *wear, hold*, we only calculate the relation accuracy on the MSCOCO dataset. We randomly select 1000 images and calculate the relation accuracy for each pair of objects by measuring the x, y distances between the boxes.

Assuming that better layouts imply images produced with improved quality, we convert the output layouts to images using the pretrained layout-to-image model LostGAN [21], and take the following metrics for comparisons:

(3) **Fréchet inception distance (FID)** to evaluate the qual-

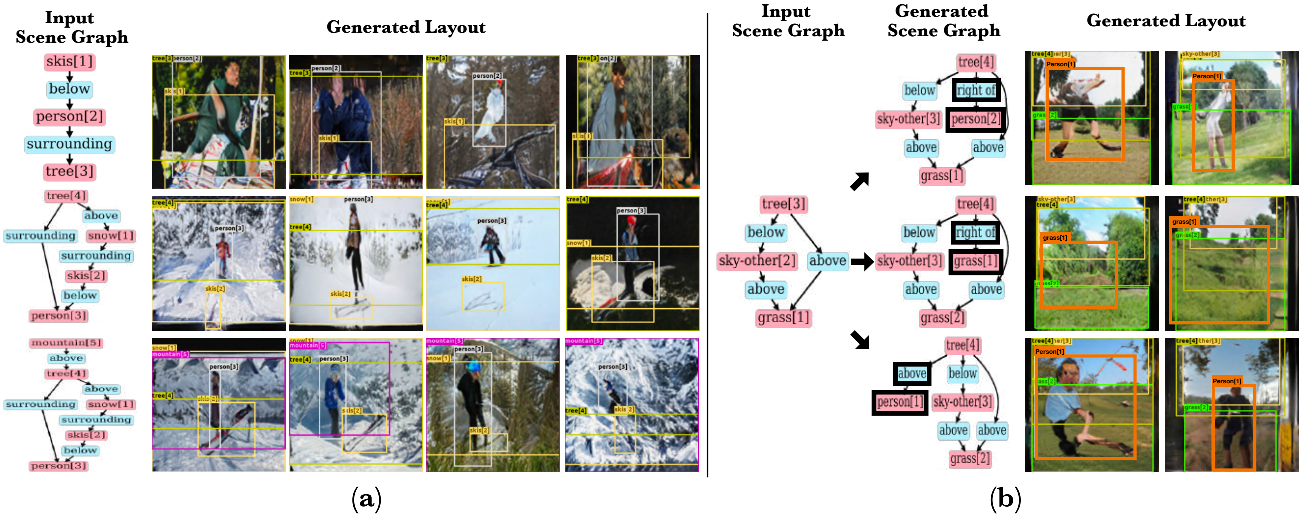

Figure 5. **Example layout outputs with (a) spatial and (b) conceptual diversity.** Note that each row in (a) shows layouts conditioned on the same input, while objects and relations inferred from the input are additionally recovered in (b) (in **orange** bounding boxes).

Model	Rel	Img	Layout			Image			
			mIOU (↑) COCO	mIOU (↑) VG-MSDN	Rel. Acc. (↑) COCO	FID (↓) COCO	FID (↓) VG-MSDN	D.S. (↑) COCO	D.S. (↑) VG-MSDN
Sg2Im	✓	✓	0.29 ±0.06	0.168 ±0.043	49.12 ±0.29	**69.2** ±0.4	**92.1** ±1.7	0.50 ±0.01	0.41 ±0.02
NDN	✓		0.33 ±0.04	-	48.89 ±0.67	90.9 ±0.9	-	**0.55** ±0.02	-
CanonicalSg2Im	✓		0.42 ±0.01	0.174 ±0.025	50.12 ±0.21	97.9 ±1.0	101.6 ±0.7	0.49 ±0.02	0.42 ±0.01
Ours	✓		**0.49** ±0.03	**0.183** ±0.036	**51.36** ±0.45	76.8 ±0.3	99.8 ±1.7	0.53 ±0.03	**0.45** ±0.001

Table 2. **Quantitative evaluation.** Note that **Rel** denotes the ability in exploiting relations between objects during training, and **Img** indicates the additional requirement of ground-truth images for training. Note that NDN is not evaluated on VG-MSDN since they require complete graph annotation as inputs. The bold numbers represent the best scores, and the underline ones are the second highest.

ity of the generated images based on our predicted layouts via measuring the distance between the generated distribution and the real image input.

(4) **Diversity score. (D.S.)** We calculate the diversity score using the LPIPS metric proposed by [28]. The LPIPS metric measures the diversity of the generated images via computing the distances between the generated image pairs at AlexNet [11] feature space.

4.3.2 Quantitative comparisons

Layout generation. Table 2 compares our LT-Net with Sg2Im [8], NDN [12] and CanonicalSg2Im [6]. We see that our LT-Net achieved the best mIOU and relation accuracy scores among all methods. This verifies the design of our LT-Net in encoding contextual features while enforcing layout recovery with relation consistency.

Image generation. In Table 2, we additionally demonstrate the effectiveness of our model for the downstream task of image generation. From this table, we see that Our LT-Net reported comparable or improved FID and diversity scores, confirming the plausibility and diversity of the generated images. It is worth pointing out that, while the best FID scores were reported by Sg2Im [8], it requires ground-truth images during training, while other methods (including ours) do not have such a requirement.

We note that, for the above experiments, we randomly split the testing set into 5 groups and report the associated mean and standard deviation, which follows the settings applied in [8]. To sum up, our model quantitatively performed favorably against state-of-the-art methods, supporting the use of our model for producing plausible and satisfactory layout or image outputs.

4.4. Ablation studies

4.4.1 Learning contextual embedding from scene graph inputs

We first conduct ablation studies verifying the design of the LT-Net encoder (i.e., object/relation predictor). As listed in Table 3, the baseline model (in the first row) contains only word embedding and segment embedding as inputs, which

Obj ID	PoP ID	Masked Acc	Obj ID Acc	PoP ID Acc
		74.45 ±0.52	92.27 ±0.17	83.16 ±0.05
	✓	85.68 ±0.21	91.87 ±0.19	99.89 ±0.01
✓		75.73 ±0.08	**96.26** ±0.13	83.19 ±0.06
✓	✓	**87.12** ±0.17	96.21 ±0.17	**99.99** ±0.01

Table 3. **Ablation studies on input embedding** using the accuracy score on masked word, object ID and POP ID. We show that object ID and POP ID embeddings utilized in LT-Net particularly resulted in improve masked accuracy.

Model	C. Box mIOU (↑)	R. Box mIOU (↑)	FID (↓)	D.S. (↑)
Baseline	0.43 ±0.01	-	85.6 ±0.3	**0.54** ±0.03
+ $\mathcal{L}_{rel}$	0.45 ±0.02	-	81.8 ±0.9	0.53 ±0.06
+ VT-CAtt	0.46 ±0.01	0.49 ±0.02	**75.3** ±0.8	0.53 ±0.02
+ ϵ	**0.46** ±0.04	**0.49** ±0.03	76.8 ±0.3	0.53 ±0.03

Table 4. **Ablation studies of LT-Net on COCO-Stuff.** Note that $\mathcal{L}_{rel}$ and ϵ denote the Relation Consistency Loss and confidence score weight, respectively. For each added component, we train the LT-Net for 50 epochs and report the results on the testing set.)

are the default inputs of BERT [4]. From rows 2 to 4 in Table 3, we further compare the performances of the encoder with different combination of embeddings. From the results listed in this table, we see that our design of input embedding achieved the best accuracy across different categories, and thus would be preferable in exploiting co-occurrences between objects and relationships.

4.4.2 Design of LT-Net

Table 4 lists the performances and compares contributions of the deployed modules in our LT-Net. The baseline model in Table 4 only includes the Obj/Rel predictor $\mathcal{P}$ and layout generator $\mathcal{G}$, with $\mathcal{P}$ pretrained ass the masked language model, while $\mathcal{G}$ is trained using only $\mathcal{L}_{box}$ and $\mathcal{L}_{KL}$. To confirm our introduction and enforcement of relation consistency during training, we apply this objective to the baseline model, and report the results in the second row of Table 4. With the added Visual-Textual Co-Attention (VT-CAtt) module, the layout outputs are further refined, with results listed in the third row of Table 4. We note that, we further consider the confidence score ϵ) for each object/relation during the co-attention refinement process. By comparing the performances listed in Table 4, we see that the full version of our LT-Net achieved the best performance in terms of both layout and image generation. Thus, the design of our LT-Net can be successfully verified.

Decoder	C. Box mIOU (↑)	FID (↓)	D.S. (↑)
Linear	0.40 ±0.03	225.5 ±1.2	0.42 ±0.01
Gaussian	0.43 ±0.03	114.9 ±8.4	0.52 ±0.01
GMM	**0.43** ±0.01	**85.6** ±0.3	**0.54** ±0.03

Table 5. **Ablation studies of Gaussian Mixture Model.** Note that D.S. denote the diversity score. For each row, we train the LT-Net for 50 epochs and report the results on the testing set.)

4.4.3 Fitting Gaussian Mixture Models

Finally, we conduct ablation experiments to verify and support the use of GMM distribution matching in our layout predictor $\mathcal{G}$, and the results are shown in Table 5. In this table, we consider two different modules for distribution matching: **linear module** directly uses the linear layer to predict coordinates of the bounding boxes in the layout with the regression loss. **Gaussian module** is applied to replace each GMM (i.e., a Gaussian distribution for each bounding box) for training LT-Net with the same objective functions. Comparing these two methods, we see that our GMM decoder not only achieved the best mIOU score (in fitting the ground-truth bounding boxes), it further exhibits the ability of our LT-Net in producing diverse yet plausible images translated from the sampled layouts (i.e., with improved FID and diversity scores).

5. Conclusion

We proposed a generative model of LayoutTransformer Network (LT-Net) for text-conditioned layout generation. The unique design of the encoder in LT-Net not only encodes objects and relations explicitly presented in the input scene graph, it also exploits the implicit ones and allows manipulation of conceptually diverse yet plausible outputs. The decoder of LT-Net translates the encoded contextual features into layouts n terms of bounding boxes. With the enforcement of GMM-like distributions at the decoder layer, together with the self-supervised relation consistency, spatially diverse layouts can be further produced. A post-processing module of visual-textual co-attention jointly observes the contextual information from the scene graph and sequentially produced bounding boxes, allowing the output to be further refined with semantic and visual plausibility guarantees. We conducted experiments on COCO and VG-MSDN datasets, which qualitatively and quantitatively demonstrated the effectiveness of our model over state-of-the-art layout generation methods.

Acknowledgement This work is supported in part by the Ministry of Science and Technology of Taiwan under grant MOST 110-2634-F-002-036.

References

[1] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4561–4569, 2019. 1, 2, 3

[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. 5

[3] Zhiwei Deng, Jiacheng Chen, Yifang Fu, and Greg Mori. Probabilistic neural programmed networks for scene generation. In *Advances in Neural Information Processing Systems*, pages 4028–4038, 2018. 2

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 4, 8

[5] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015. 2

[6] 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*, 2020. 1, 2, 3, 6, 7

[7] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7986–7994, 2018. 2, 4

[8] 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. 1, 2, 5, 6, 7

[9] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9895–9904, 2019. 1, 2

[10] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 5

[11] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014. 7

[12] Hsin-Ying Lee, Weilong Yang, Lu Jiang, Madison Le, Irfan Essa, Haifeng Gong, and Ming-Hsuan Yang. Neural design network: Graphic layout generation with constraints. *arXiv preprint arXiv:1912.09421*, 2019. 1, 2, 6, 7

[13] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv preprint arXiv:1901.06767*, 2019. 1, 3

[14] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12174–12182, 2019. 2, 4

[15] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1261–1270, 2017. 6

[16] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[17] Gaurav Mittal, Shubham Agrawal, Anuva Agarwal, Sushant Mehta, and Tanya Marwah. Interactive image generation using scene graphs. *arXiv preprint arXiv:1905.03743*, 2019. 2, 3

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 5

[19] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 1, 2

[20] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741, 2009. 2

[21] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10531–10540, 2019. 1, 3, 6

[22] Subarna Tripathi, Anahita Bhiwandiwalla, Alexei Bastidas, and Hanlin Tang. Heuristics for image generation from scene graphs. 2019. 2

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[24] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018. 2

[25] LI Yikang, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. Pastegan: A semi-parametric method to generate image from scene graph. In *Advances in Neural Information Processing Systems*, pages 3948–3958, 2019. 1, 2, 3

[26] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017. 1, 2

[27] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018. 1

[28] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[29] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 3