LAYOUTTRANSFORMER: RELATION-AWARE SCENE LAYOUT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In the areas of machine learning and computer vision, text-to-image synthesis aims at producing image outputs given the input text. In particular, the task of layout generation requires one to describe the spatial information for each object component, with the ability to model their relationships. In this paper, we present a LayoutTransformer Network (LT-Net), which is a generative model for text-conditioned layout generation. By extracting semantics-aware yet object discriminative contextual features from the input, we utilize Gaussian mixture models to describe the layouts for each object with relation consistency enforced. Finally, a co-attention mechanism across textual and visual features is deployed to produce the final output. In our experiments, we conduct extensive experiments on both MS-COCO and Visual Genome (VG) datasets, and confirm the effectiveness and superiority of our LT-Net over recent text-to-image and layout generation models.

1 INTRODUCTION

Text-to-image generation aims at synthesizing realistic images that semantically match the text descriptions. With 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. While remarkable progresses have been made by deep learning models in synthesizing high-quality images Xu et al. (2018); Zhang et al. (2018), generating plausible layout compositions with relationships preserved across different objects remains a challenging task, which requires one to bridge the gap between semantic and perceptual spaces .

To address text-to-layout generation. Jyothi et al. (2019) introduce a variational autoencoder (VAE) to model the latent distributions of spatial relationships between objects. Alternatively, Hong et al. (2018); Li et al. (2019b) use a LSTM (Hochreiter & Schmidhuber, 1997) to encode textual conditions, and then design an auto-regressive decoder for box coordinates generation. Li et al. (2019b) propose a similar seq2seq (Sutskever et al., 2014) model with attention mechanism (Bahdanau et al., 2014) to better capture the correspondence between boxes and words. Recently, Lee et al. (2019) introduce Graph Convolution Network (GCN) (Duvenaud et al., 2015) to iteratively generate bounding boxes for a set of components and user-specified constraints.

However, existing layout generation methods generally share a common limitation, which only models the spatial/relation information explicitly defined by textual inputs (e.g., *left, right, below, etc.*). In other words, language predicates such as *ride, wear, etc.* might not be sufficiently modeled. For example, the description '*A man wears a jacket.*' implies the layout of the man should be overlapped with the one of the jacket and for the layout corresponding to the description '*A man rides a horse.*', in no way would the box of the man lies under the box of the horse. Moreover, it is also desirable to have the layout both plausible and semantically consistent with the textual input, when particular objects or relation predicates are modified in the textual input.

To overcome the above challenges, we propose a novel LayoutTransformer Network (LT-Net) in this paper. Our LT-Net models the relations across multiple objects with their stochasticity properly observed. As illustrated in Fig. 1, our model converts textual inputs into semantics-aware and object discriminative representations, which extract and preserve both explicit and implicit relationships between objects. Together with the Gaussian Mixture Models (GMM) (Reynolds, 2009), the above contextual features allow our LT-Net to predict the diverse layout components in the output. Finally, a visual-textual co-attention module is deployed, which produces plausible layout outputs.

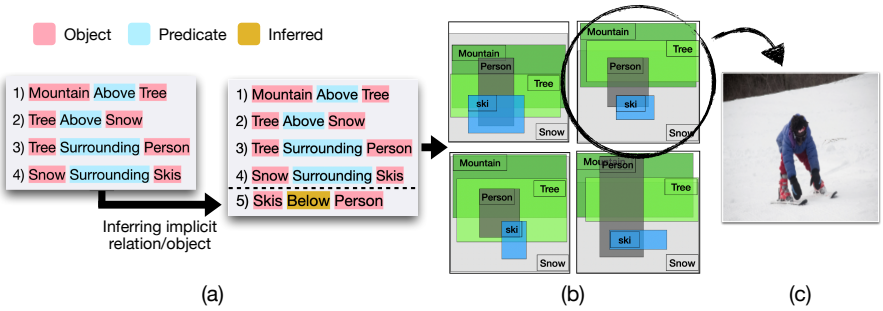

Figure 1: Illustration of our LayoutTransformer Network (LT-Net) for layout generation. (a) Inferring implicit relation across objects from the textual inputs. (b) Producing layouts from learned contextual features with diversity observed. (c) Image generation from the predicted layout.

We now highlight the contributions as follows:

- We propose a novel framework of LayoutTransformer Network (LT-Net) for layout generation, which models implicit object/relation from textual inputs for producing semantics-consistent yet diverse layout outputs.
- Our LT-Net encodes textual inputs into semantics-aware and object-discriminative representations, followed by layout generator which utilizes Gaussian mixture models for predicting the output layout with relation guarantees.
- A co-attention mechanism is introduced in our LT-Net. With the observed GMM-based layout distribution, it takes the bounding boxes of each object and the summarized contextual feature for producing the refined and plausible layout output.

2 Related Works

2.1 Text-to-image synthesis

Generating realistic images from text descriptions benefits a wide range of computer vision applications. Reed et al. (2016) propose an end-to-end trainable network generating image conditioned on sentence description. Zhang et al. (2017) use a two-stage GAN to progressively generate images with higher resolution. Following Xu et al. (2018), they design a cross-modality attention module with an eye to align the content of the generated image and the conditioned text. Hong et al. (2018) 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, Li et al. (2019b) present a novel object-level attention mechanism to generate semantically meaningful images. Nevertheless, most existing text-to-image methods only focus on nouns in the textual descriptions for synthesizing image outputs.

2.2 Layout generation

Layouts can be viewed as intermediate representations in text-conditioned image generation tasks (Hong et al., 2018; Tan et al., 2019; Li et al., 2019b). Instead of directly mapping from text to image domains, layout generation typically produces the outputs conditioned on the given inputs, followed by transforming such outputs into realistic images. Recently, generative models are applied to *graphic design* layout generation, which aims to design the document layout (e.g., Magazine (Tabata et al., 2019)). For example, Li et al. (2019a); Zheng et al. (2019) utilize GAN to generate layouts from the given attributes of components. Despite promising results, however, such GAN-based methods cannot explicitly exploit relationships between components. On the other hand, Lee et al. (2019) 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.

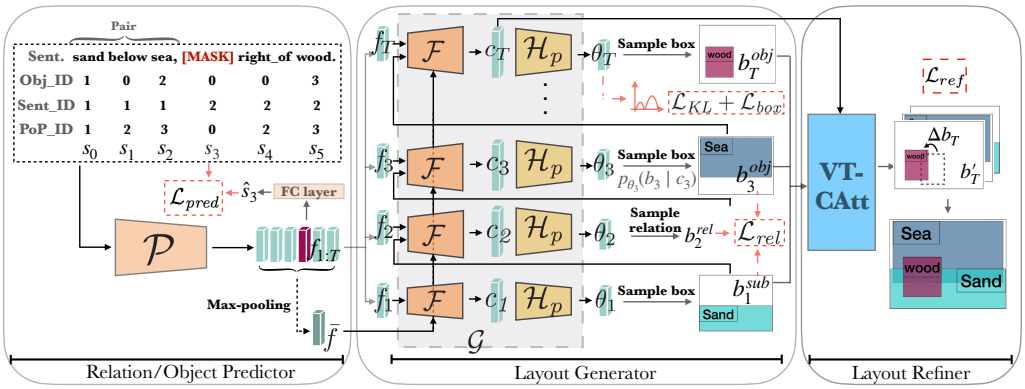

Figure 2: LayoutTransformer Network (LT-Net) with Relation/Object Predictor $\mathcal{P}$, Layout Generator $\mathcal{G}$, and Layout Refiner (Visual-Textual Co-Attention) Modules. Note that $f_{1:T}$ denote the contextualized features of each object/relation, and $\bar{f}$ is that of the entire input. $\mathcal{G}$ contains a layout feature extractor $\mathcal{F}$ to incorporate contextual and bounding box features, followed by a prediction head $\mathcal{H}_p$ for modeling the associated feature distribution θ. We have b_T and b'_T indicate coarse and refined layout outputs, respectively.

Recently, Jyothi et al. (2019) propose LayoutVAE to tackle *scene* layout generation, i.e., producing layouts of natural scenes. LayoutVAE directly applies VAE for generating stochastic scene layouts from a given label set. Thus, unlike our proposed work, LayoutVAE is not able to synthesize diverse layouts using textual inputs, and thus lacks the ability in manipulating such outputs.

3 Methodology

3.1 Notations and Algorithmic Overview

For the sake of completeness, we first define the notations to be used in this paper. Our model takes textual data $S = \{s_1, s_2, ...s_T\}$ as the input (T denotes the number of words). More specifically, each sentence in the textual input (i.e., s_{i-1}, s_i and s_{i+1}) consists of three words representing the *subject*, *relation*, and *object* describing the interaction between the associated objects in the scene. Given such inputs, the goal of our model is to synthesize 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 relation/object predictor $\mathcal{P}$ for modeling the semantic information from the observed sequential input $s_{1:T}$. More precisely, $\mathcal{P}$ extracts the contextualized representations $f_{1:T}$ describing the semantics of each object in the scene. With the semantic representation of the input sentence $f_{1:T}$ and its summarized $\bar{f}$ (via max pooling over $f_{1:T}$) obtained, the second module $\mathcal{G}$ of our LT-Net serves a generator, which contains a layout feature extractor $\mathcal{F}$, followed by a prediction head $\mathcal{H}_p$ to produce the layouts $B = \{b_1, b_2, ...b_T\}$ from the learned contextualized representations of each object with sufficient stochasticity. Finally, we additionally introduce Visual-Textual Co-Attention(VT-CAtt) for layout refinement, which performs cross-modal attention to optimize the predicted image layout by jointly taking both the *spatial* and *semantic* information into consideration. It is worth noting that, our LT-Net not only generates diverse layouts conditioned on the given textual input. As discussed later, it further exhibits the ability of inferring the *implicit relations* among the objects, which explains why satisfactory layouts can be expected by our LT-Net.

3.2 Learning Relation-Aware and Object-Discriminative Embedding

Given a sequential text input $s_{1:T}$, our Relation/Object Predictor $\mathcal{P}$ in LT-Net aims at deriving contextualized representations $f_{1:T} = \{f_1, f_2, ..., f_T\}$ for each object/relation, describing its semantic and spacial information. Moreover, as one of our goals of our LT-Net, this introduced predictor exhibits abilities in inferring *implicit* relationships between objects.

Instead of applying standard recurrent models for taking textual input directly, this predictor $\mathcal{P}$ embeds $s_{1:T}$ into a relation-aware and object-discriminative Embedding $e^i_{1:T}$ 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 (Devlin et al., 2018), 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 indicates the number of *subject-relation-object* pairs in the input. In order to specify the *semantic role* (i.e., subject, relation, or object) s_t in each sentence of the textual input, we uniquely utilize the *Part-of-Pair (PoP) ID* e^p_t for each s_t in a sentence.

With the above four types of features extracted from the textual input, we concatenate them to form the embedding $e^i_{1:T} = [e^w_{1:T} \oplus e^s_{1:T} \oplus e^p_{1:T} \oplus e^o_{1:T}]$, which serves as the input of our relation predictor $\mathcal{P}$ for learning the contextualized feature vectors $f_{1:T}$. We pretrain our Relation/Object Predictor $\mathcal{P}$ based on BERT (Devlin et al., 2018), and have the output contextual features $f_{1:T}$ to predict $\hat{s_t}$ via a single linear layer. We note that, in addition to predicting the masked word as BERT does, our model also recovers both the masked PoP ID and object ID. Thus, to train this relation/object predictor, we have the objective function $\mathcal{L}_{pred}$ calcualte the cross-entropy losses from matching word, object ID, and PoP ID between the input s_t and the predicted $\hat{s}_t$, respectively.

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

3.3 Stochastic Layout Generation

With contextualized representations $f_{1:T}$ and $\bar{f}$ derived, our Layout Generator $\mathcal{G}$ aims to produce the layout, while the probability distribution of the position/size of the resulting bounding boxes simultaneously observed. Our $\mathcal{G}$ comprises two components: a Layout Feature Extractor $\mathcal{F}$ to extract the spatial information from contextual inputs, and a prediction head $\mathcal{H}_p$ to model the distributions of the position/size of the bounding boxes. We now detail these two components.

3.3.1 Modeling layout distribution via Gaussian Mixture Model

For each word s_t in the textual input (either subject or object), we define the produced bounding box in terms of its location and size, i.e., $b_t = (x_t, y_t, w_t, h_t)$, $b_t \in \{b^{sub}, b^{obj}\}$. If s_t denotes a relation, we output the box disparity between its associated subject and object pair, i.e., $b_t = (\Delta x_t, \Delta y_t)$, $b_t \in \{b^{rel}\}$. In order to introduce the generative ability to our model, we follow Hong et al. (2018); Li et al. (2019b) and the above layout distribution with *Gaussian Mixture Models (GMM)*. Each bounding box is 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

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

where $\mathcal{N}(b_t; \theta_{t,i})$ denotes the multivariate normal distribution. Note that for θ_t, μ^x and μ^y are the means, σ^x and σ^y are the standard deviations, and ρ^{xy} is the correlation coefficient, describing the associated multivariate normal distributions.

Inspired by the Variational AutoEncoder (VAE) (Kingma & Welling, 2013), we have the loss function as the sum of following two terms for learning $\mathcal{F}$ and $\mathcal{H}_p$: the bounding box reconstruction loss $\mathcal{L}_{box}$ and the Kullback-Leibler Divergence Loss, $\mathcal{L}_{KL}$. Note that the aforementioned reconstruction loss maximizes the log-likelihood of the generated GMM to fit that observed from the training data, which is calculated using the generated GMM parameters θ_t and the location of the ground-truth bounding box $\hat{b_t} = (\hat{x_t}, \hat{y_t}, \hat{w_t}, \hat{h_t})$:

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

Empirically, the maximum likelihood framework might suffer from over-fitting problems (i.e., degeneration to a Dirac delta function). To alleviate this, the Kullback-Leibler (KL) divergence with

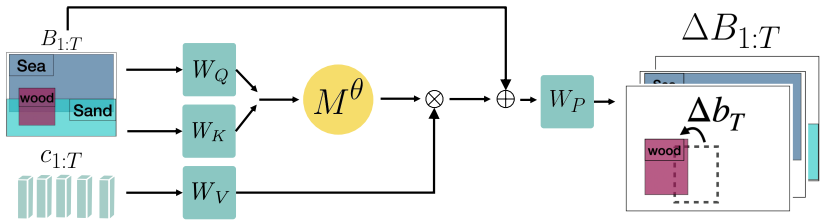

Figure 3: Architecture of ouf Visual-Textual Co-Attention (VT-CAtt) module. B denotes the coarse layout synthesized by $\mathcal{G}$, and C represents the contextual vectors produced by $\mathcal{F}$ in LT-Net. M^θ indicates the refined attention weights, while W_Q, W_K, W_V, and W_P are to be learned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

respect to multivariate normal distributions serves as a regularization term. With such GMM distributions observed, our model allows one to generate diverse yet plausible layouts. We calculate the KL Loss term, $\mathcal{L}_{KL}$, by measuring the distance between the generated distribution P controlled by θ_t and a multivariate normal distribution Q with its mean same as θ_t and unit variance.

$$\mathcal{L}_{KL} = \sum_{i=1}^{K} D_{KL}(P_i \| Q_i) = \sum_{i=1}^{K} D_{KL}(\mathcal{N}(\mu_{t,i}^x, \mu_{t,i}^y, \sigma_{t,i}^x, \sigma_{t,i}^y, \rho_{t,i}^{xy}) \| \mathcal{N}(\mu_{t,i}^x, \mu_{t,i}^y, 1, 1, 0)). \tag{4}$$

3.3.2 Observing self-supervised relation consistency

In addition to the above bounding box recovery and KL divergence losses, we introduce a novel Relation Consistency Loss $\mathcal{L}_{rel}$ to our layout generator. Served as a self-supervised objective, we enforce this 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 Δb and b^{rel}:

$$\mathcal{L}_{rel} = \frac{1}{N} \sum (\Delta b - b^{rel})^2, \tag{5}$$

where N is the number of the relation pairs in $S = \{s_1, s_2, ...s_T\}$.

With the above objectives, we train our layout generator $\mathcal{G}$ by calculating the following loss term:

$$\mathcal{L}_{gen} = \mathcal{L}_{box} + \mathcal{L}_{rel} + \lambda_{KL}\mathcal{L}_{KL}, \tag{6}$$

where λ_{KL} represents the magnitude of the regularization (we fix $\lambda_{KL} = 0.1$ in this work).

3.4 Visual-Textual Co-Attention for Layout Refinement

Since the coarse layout $B_{1:T}$ is generated for each individual bounding box in an incremental fashion, the produced layout might not be optimal. Thus, we present an Visual-Textual Co-Attention (VT-CAtt) mechanism for refining $B_{1:T}$ into the final output. By leveraging both spatial and semantic information, our VT-CAtt outputs the residual $\Delta B_{1:T}$ updating each bounding box, leading to more accurate and realistic layouts $B'_{1:T}$. It is worth noting that, when jointly taking the above contextual and visual features as the refinement inputs, we particularly leverage the visual information of the coarse bounding boxes into the contextual features. Realized by our unique co-attention process, this allows the contextual features to exhibit spatial awareness.

Our VT-CAtt module is depicted in Fig 3, which takes the coarse layout $B_{1:T}$ as the visual features (as both *query* and *key*) and the contextual vectors $C_{1:T}$ as the semantic feature (as *value*). We perform matrix multiplication to the projection of *query* $W_Q(B)$ and *key* $W_K(B)$ to obtain the attention matrix M. Moreover, since we generate the coarse bounding box B from the GMM distribution, the sampled bounding boxes with low probabilities would imply less likely spatial outputs. Therefore, we derive the box confidence ϵ from the sampled GMM probability value to penalize the each bounding box. And, the resulting GMM-aware attention weights M^θ is calculated as:

$$M_{i,j}^\theta = \frac{\epsilon_j \cdot \exp(M_{i,j})}{\sum_{j=1}^{T} \epsilon_j \cdot \exp(M_{i,j})}, \tag{7}$$

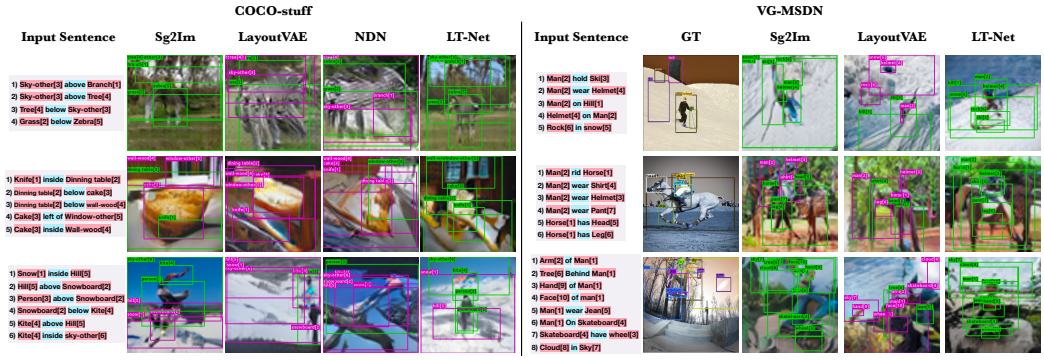

Figure 4: **Qualitative evaluation on COCO-Stuff and VG-MSDN.** Each row shows the textual input, ground truth layout and those generated by different approaches. For visualization purposes, we apply the pretrained layout2im (Zhao et al., 2019) to convert the output layout into images. Note that bounding boxes in **red** indicate layout components **not** matching the input relationships.

where $M^{\theta}_{i,j}$ denotes the contribution of the j^{th} object to the i^{th} object, 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 the residual ΔB. Following (Redmon et al., 2016), the loss $\mathcal{L}_{ref}$ for this refinement module is defined below:

$$\mathcal{L}_{ref} = \sum_{t=1}^{T} \lambda_{xy}[(x'_t - \hat{x_t})^2 + (y'_t - \hat{y_t})^2] + \lambda_{wh}[(\sqrt{w'_t} - \sqrt{\hat{w_t}})^2 + (\sqrt{h'_t} - \sqrt{\hat{h_t}})^2], \quad (8)$$

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, 6 and 8, our LT-Net can be trained accordingly.

4 Experiments

4.1 Datasets

COCO-stuff. We perform our experiments on the COCO-Stuff dataset (Caesar et al., 2018), which augments a subset of the COCO dataset (Lin et al., 2014) 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 (Johnson et al., 2018), 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 Visual Genome dataset (Krishna et al., 2017) comprises more than 108K images annotated with scene graphs. In our experiments, we specifically consider the VG-MSDN dataset (Li et al., 2017). Please refer to Appendix A.2.1 for the details about this dataset such as the total number of training data and testing data.

4.2 Qualitative Results

Plausible layout generation. We compare our proposed LT-Net with recent state-of-the-art models, including sg2im (Johnson et al., 2018), LayoutVAE (Jyothi et al., 2019), and NDN (Lee et al., 2019). In Fig. 4, we observe that the outputs of Sg2Im, LayoutVAE, and NDN did not necessarily match the relations between the objects, while our LT-Net was able to generate consistent layout components with the given textual descriptions, especially on the more challenging dataset of VG-MSDN.

Multi-modal layout generation. Comparing with other state-of-the-art models such as Sg2Im, NDN, and LayoutVAE, our LT-Net is a generative model, and thus is capable of generating diverse yet plausible layouts given the same textual input. From top to bottom rows in Fig. 5(a), we see that we were able to produce diverse layout outputs given simple to more complex textual inputs.

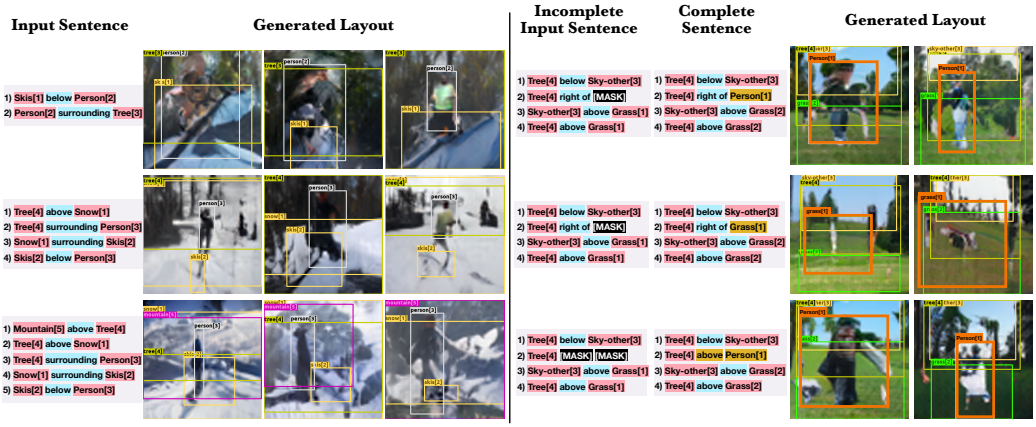

Figure 5: **Examples of (a) diverse layout outputs and (b) layout with implicit relation/object inferred.** Note that all three outputs in (a) are conditioned on the same textual input, while objects and relations from the incomplete textual inputs are recovered in (b).

Inferring implicit objects and relations. Fig. 5(b) demonstrates the ability of our LT-Net in inferring the implicit objects or relations across existing objects given the textual input. Take the first row in Fig. 5(b) for example: given four input sentences of (1) Tree below Sky-other, (2) Tree right of [MASK], (3) Sky-other above Grass, and (4) Tree above Grass, our LT-Net was able to infer the masked word of "Person", which is not explicitly presented in the textual input. And, with such inferred objects/relations, the final layout would still exhibit plausibility.

From the above qualitative evaluation and comparisons, we see that our LT-Net is able to generate sufficiently plausible layouts, which contain inferred objects/relations with output diversity preserved. More qualitative results are available in Appendix A.2.

4.3 Quantitative Results

4.3.1 Evaluation metrics

For quantitative evaluation, we consider the following five different metrics: (1) **Mean intersection over union (mIOU).** The mIOU score measures how well the generated layout fits the ground truth data. (2) **Relation accuracy.** The relation accuracy only considers the relation with explicit spatial meaning (i.e., *left of, right of, above and below*). We randomly select 1000 images and calculate the relation accuracy for each pair of objects by measuring the x, y distance between the boxes. (3) **R-precision.** is a common evaluation metric for ranking retrieval results. Following Li et al. (2019b), we calculate the percentage of successful retrievals as the R-precision score. (4) **Fréchet inception distance (FID)** to evaluate the quality of the generated images based on our predicted layouts via measuring the distance between the generated distribution and the real image input.

4.3.2 Quantitative comparisons

Table 1 compares our LT-Net with Sg2IM, LayoutVAE, and NDN. Note that, for each experiment, we randomly sample 3000 images for 5 times and report the associated mean and standard deviation. As can be seen in Table 1, our LT-Net achieved improved mIOU scores than others by significant margins, which supports our use of GMM for fitting layout distributions. Moreover, our LT-Net reported satisfactory FID scores, which indicate that high quality synthesized images can be produced by our predicted layouts. Moreover, we apply *R-Precision* and *Relation Accuracy* to estimate the consistency between the input textual descriptions and the generated layouts. To sum up, our model performed favorably against state-of-the-art methods in terms of mIOU, R-precision, and Relation Accuracy, and reported satisfactory FID scores. The above quantitative results support the use of our model for producing plausible and satisfactory layouts.

Table 1: **Quantitative evaluation.** Note that **Rel** denotes the exploitation of relation information during training, and **Img** indicates the requirement of real images (instead of layouts) for training.

Model	Rel	Img	**mIOU** (↑) COCO	**mIOU** (↑) VG-MSDN	**FID** (↓) COCO	**FID** (↓) VG-MSDN	**R-Pre.** (↑) COCO	**Rel. Acc.** (↑) COCO
Sg2Im	✓	✓	0.29 ±0.06	0.168 ±0.063	**48.8** ±0.3	**90.5** ±1.7	0.26 ±0.01	49.12 ±0.29
LayoutVAE			0.19 ± 0.02	0.041 ± 0.028	60.7 ±0.4	113.9 ±7.7	0.23 ±0.03	-
NDN	✓		0.33 ±0.04	-	79.5 ±1.0	-	0.25 ±0.02	48.89 ±0.67
Ours	✓		**0.49** ±0.03	**0.183** ±0.036	55.7 ±0.9	**90.5** ±1.7	**0.35** ±0.02	**51.36** ±0.45

Table 2: **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 test set.)

Model	**C. Box mIOU** (↑)	**R. Box mIOU** (↑)	**FID** (↓)	**R-pre.** (↑)
Baseline	43.12 ±0.04	-	80.89 ±6.53	0.30 ±0.02
+ $\mathcal{L}_{rel}$	45.33 ±0.03	-	60.65 ±0.54	0.33 ±0.02
+ VT-CAtt	46.45 ±0.05	49.57 ±0.13	56.09 ±0.19	0.34 ±0.02
+ ϵ	**46.42** ±0.06	**49.72** ±0.03	**55.74** ±0.91	**0.35** ±0.02

4.4 Ablation Studies

Finally, we perform ablation studies on our model design. More precisely, we demonstrate the effectiveness of our model by incrementally adding each component to the baseline model, which only contains Object/Relation Predictor $\mathcal{P}$ and Layout Generator $\mathcal{G}$ without relation consistency $\mathcal{L}_{rel}$. Additionally, we assess the each component of our relation-aware and object-discriminative embeddings as described in Sect. 3.2, and the supporting results can be seen in Appendix A.2.3.

Relation consistency. 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 2. We see that not only the mIOU score was raised by 0.3, both FID and R-precision scores were also improved. These results indicate that the self-supervised relation consistency would be beneficial to our model design.

Visual-Textual Co-Attention(VT-CAtt). With the deployment of VT-CAtt module for refinement, the third row of Table 2 confirm that the mIOU, FID, and R-precision scores all made remarkable improvements. Thus, the joint exploitation of contextual representations and visual layout features would be a critical component in our LT-Net.

Confidence score (ϵ) re-weighting. As described in Sect. 3.4, we additionally take the confidence scores from the GMM probability outputs to weight the bounding box features during attention refinement. As can be seen from the last row in Table 2, this would finally boost the performances and thus would be desirable in our co-attention process.

5 Conclusion

In this paper, we proposed a generative model of LayoutTransformer Network (LT-Net) for text-conditioned layout generation. By deriving semantics-aware and object discriminative contextual features from the textual descriptions. our LT-Net is able to produce layout components which not only reveal implicit objects/relations, sufficient output diversity can be guaranteed via fitting Gaussian mixture models. Finally, a visual-textual co-attention mechanism exploits cross-modal features for refining the final layout, which exhibit semantics consistency and plausibility. We conducted extensive experiments on COCO and VG-MSDN datasets, which qualitatively and quantitatively demonstrated the effectiveness of our model over state-of-the-art methods.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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*, pp. 1209–1218, 2018.

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.

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*, pp. 2224–2232, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

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*, pp. 7986–7994, 2018.

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*, pp. 1219–1228, 2018.

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*, pp. 9895–9904, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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.

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.

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

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*, pp. 12174–12182, 2019b.

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*, pp. 1261–1270, 2017.

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*, pp. 740–755. Springer, 2014.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pp. 8026–8037, 2019.

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*, pp. 779–788, 2016.

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.

Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741, 2009.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Sou Tabata, Hiroki Yoshihara, Haruka Maeda, and Kei Yokoyama. Automatic layout generation for graphical design magazines. In *ACM SIGGRAPH 2019 Posters*, pp. 1–2. 2019.

Fuwen Tan, Song Feng, and Vicente Ordonez. Text2scene: Generating compositional scenes from textual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6710–6719, 2019.

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*, pp. 5998–6008, 2017.

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*, pp. 1316–1324, 2018.

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*, pp. 5907–5915, 2017.

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.

Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8584–8593, 2019.

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.

A Appendix

A.1 Implementation Details

Our implementations are based on the Transformer (Vaswani et al., 2017) and Pytorch (Paszke et al., 2019). All the models are trained with one GeForce 1080 Ti GPU, with a batch size of 64. Learning rates will be detailed in the later paragraph.

Relation/Object Predictor. Our Relation/Object Predictor $\mathcal{P}$ is a 4-layer Transformer Encoder, with 4 attention heads, hidden size of 256, and we use a dropout probability of 0.1 on all layers. The Encoder is followed by three linear layers to predict the masked word, PoP ID, and object ID, respectively. We pretrain our Relation/Object Predictor $\mathcal{P}$ for 50 epochs, using Adam optimizer with learning rate of 4e-4, $\beta 1$ = 0.9, $\beta 2$ = 0.999, L2 weight decay of 0.01, learning rate warmup over the first 10 epochs, and linear decay of the learning rate.

Layout Generator. Our Layout Generator $\mathcal{G}$ comprises two modules: a Layout Feature Extractor $\mathcal{F}$ and a prediction head $\mathcal{H}_p$. The Layout Feature Extractor $\mathcal{F}$ is a single-layer Transformer Decoder with 4 attention heads, hidden size of 256, and dropout ratio of 0.1. The Layout Feature Extractor $\mathcal{F}$ first extracts the feature of the synthesized layout $B_{1:t-1}$, denoted as e_t^b, and concatenates e_t^b with contextualized feature vectors f_t and $\bar{f}$ to form the context vector $c_t = [f_t \oplus \bar{f} \oplus e_t^b]$ for the prediction head $\mathcal{H}_p$. We implement our prediction head $\mathcal{H}_p$ by decomposing the quadravariate distribution into two bivariate distributions, i.e.:

$$p_{\theta_t}(b_t \mid c_t) = p_{\theta_t}(x_t, y_t, w_t, h_t \mid c_t) = p_{\theta_t}(x_t, y_t \mid c_t) p_{\theta_t}(w_t, h_t \mid c_t, x_t, y_t). \tag{9}$$

Table 3: Descriptions of the COCO-stuff and VG-MSDN datasets. Note that, **#Img** and **#Rel** represent the total number of images and that of relation pairs in the dataset, respectively. In the last two columns, **Obj** and **Pred** denote the numbers of unique object classes and predicates, respectively.

Dataset	**Training Set**		**Testing Set**		#Obj	#Pred
	#Img	#Rel	#Img	#Rel		
COCO-Stuff	∼106K	∼800K	5,000	∼36K	155	6
VG-MSDN	46,164	∼507K	10,000	∼111K	150	50

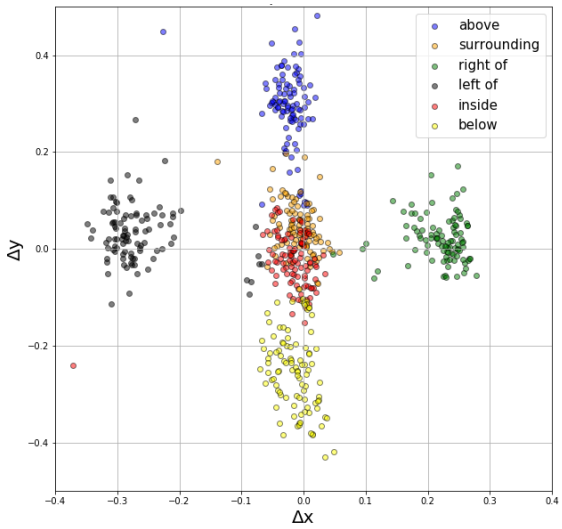

Figure 6: Distribution visualization of relation priors generated by our LT-Net. Note that x and y axes represent the differences between the associated bounding boxes of subject and object pair in horizontal and vertical directions, respectively. Different colors denote each relation of interest. For example, the circles in **green** describe subject-object pairs with the relation word "right of".

In practice, we use two linear layer to model the parameters of the bivariate normal distribution of (x_t, y_t) and (w_t, h_t), respectively.

Visual-Textual Co-Attention. Our Visual-Textual Co-Attention (VT-CAtt) is a 4-layer Transformer, with 4 attention heads, hidden size of 256, and we use a dropout probability of 0.1, followed by a bounding box prediction head W_P which is a single linear layer predicting the offset of the coarse bounding boxes .

After pretraining the Relation Predictor $\mathcal{P}$, we train our LT-Net in an End-to-End fashion with different learning rates for each module. The Relation/Object Predictor $\mathcal{P}$ is fine-tuned with learning rate of 1e-5 and linear decay of the learning rate. We jointly optimize our Layout Generator $\mathcal{G}$ and Visual-Textual Co-Attention (VT-CAtt) using Adam optimizer with learning rate of 1e-4, $\beta 1$ = 0.9, $\beta 2$ = 0.999, L2 weight decay of 0.01, learning rate warmup over the first 5 epochs, and linear decay of the learning rate.

A.2 Additional Experiments

A.2.1 Datasets

We perform our experiments on the COCO-Stuff dataset (Caesar et al., 2018) and VG-MSDN dataset provided by Li et al. (2017). Since the raw VG (Krishna et al., 2017) dataset may contain a large number of noisy data, we use a cleansed-version VG-MSDN dataset. The statistics of these datasets are provided in Table 3.

Table 4: Ablation studies on our input embedding in terms of the prediction accuracy for the masked word, object ID and PoP ID. We show that the uses of both object and PoP ID embeddings are desirable for exploiting the relation/object (as our LT-Net does).

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

A.2.2 Distribution of relation priors on COCO-stuff.

To demonstrate the ability to infer the spatial information implied by the relation constraints, we visualize the spatial prior of some predefined words: *surrounding, inside, left of, right of, above and below*. To achieve this, we randomly select 100 samples of corresponding relation words and plot the mean of the distribution induced by these relation words. To be more specific, we plot the means (μ_x and μ_y) of the distribution induced by these relation words, which represent the box disparity between the associated subject and object pairs. The result can be found in Figure 6, which confirms that our model learns the mapping between semantic words and spatial relations. Take the distribution in green in Figure 6) (i.e., the relation word "right of") for example, it can be seen that the green circles are on the right hand side of the y axis, indicating that the x coordinate values of the subject boxes were observed to be generally larger than those of the object boxes, matching the relation of "right of". Note that both μ_x and μ_y are normalized by the width and height of each image.

A.2.3 Ablation Studies

Input embedding analysis. The input embedding of the baseline model (as first row in Table 4) contains only word embedding and segment embedding which are default input according to BERT model (Devlin et al., 2018). In rows 2 to 4 in Table 4, we show the performance of the predictor with different combination of embeddings. From the results shown in this table, we see that these input embedding benefit the task of learning contextualized representation for layout generation. Take PoP ID for example, it significantly improved the masked accuracy by 10% comparing to the baseline method. Also, Object ID slightly improved the masked accuracy while enabling our model to discriminate distinct objects in the output scene.

A.2.4 Qualitative Results

In this section, we present additional qualitative results following the same setting as that in Experiments 4.2. We conduct the experiment of **plausible layout generation** on both COCO and VG-MSDN datasets, and the results are shown in Figures 7, 8 (COCO), 9, and 10 (VG-MSDN). We demonstrate that our model is capable of handling complex objects and relations by incrementally adding more objects in images. For Fig. 7, it presents the results on the COCO dataset with less than 5 objects in one image and Fig. 8 with more than 5 objects. For the VG-MSDN dataset, Fig. 9 shows images with 6 objects or fewer, and Fig. 10 with 6 objects or more. The qualitative results of **multi-modal layout generation** and **inferring implicit objects and relations** on COCO are shown in Figures 11 and 12 respectively.

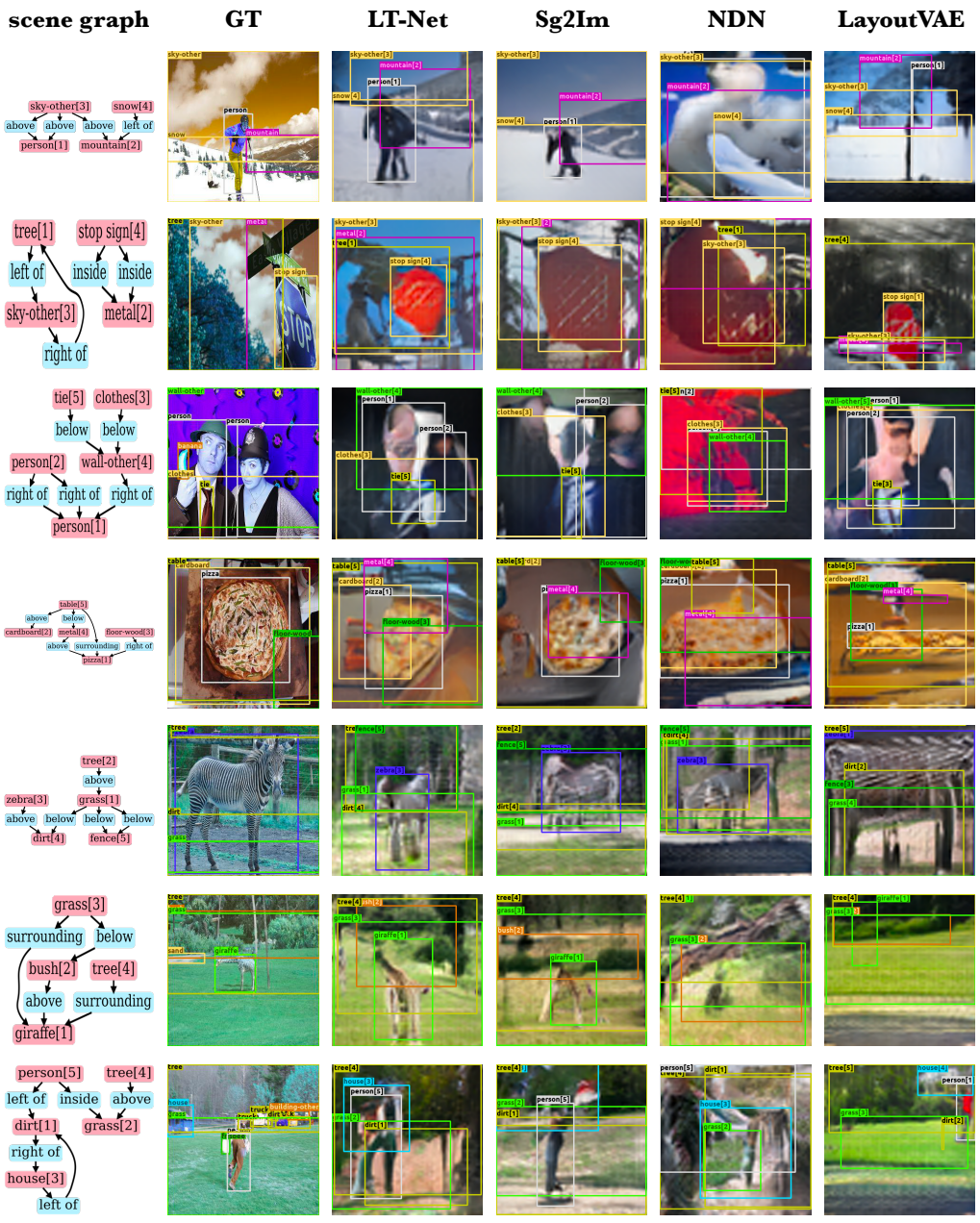

Figure 7: **Qualitative comparison on COCO-Stuff (less than 5 objects).** For each row we show the textual input, ground truth layout, synthesized layout, and image converted from the layout by layout2im (Zhao et al., 2019). Note that, for simplicity, we use the scene graph to represent the textual input.

Figure 8: **Qualitative comparison on COCO-Stuff (more than 5 objects).** For each row we show the textual input, ground truth layout, synthesized layout, and image converted from the layout by layout2im (Zhao et al., 2019). Note that, for simplicity, we use the scene graph to represent the textual input.

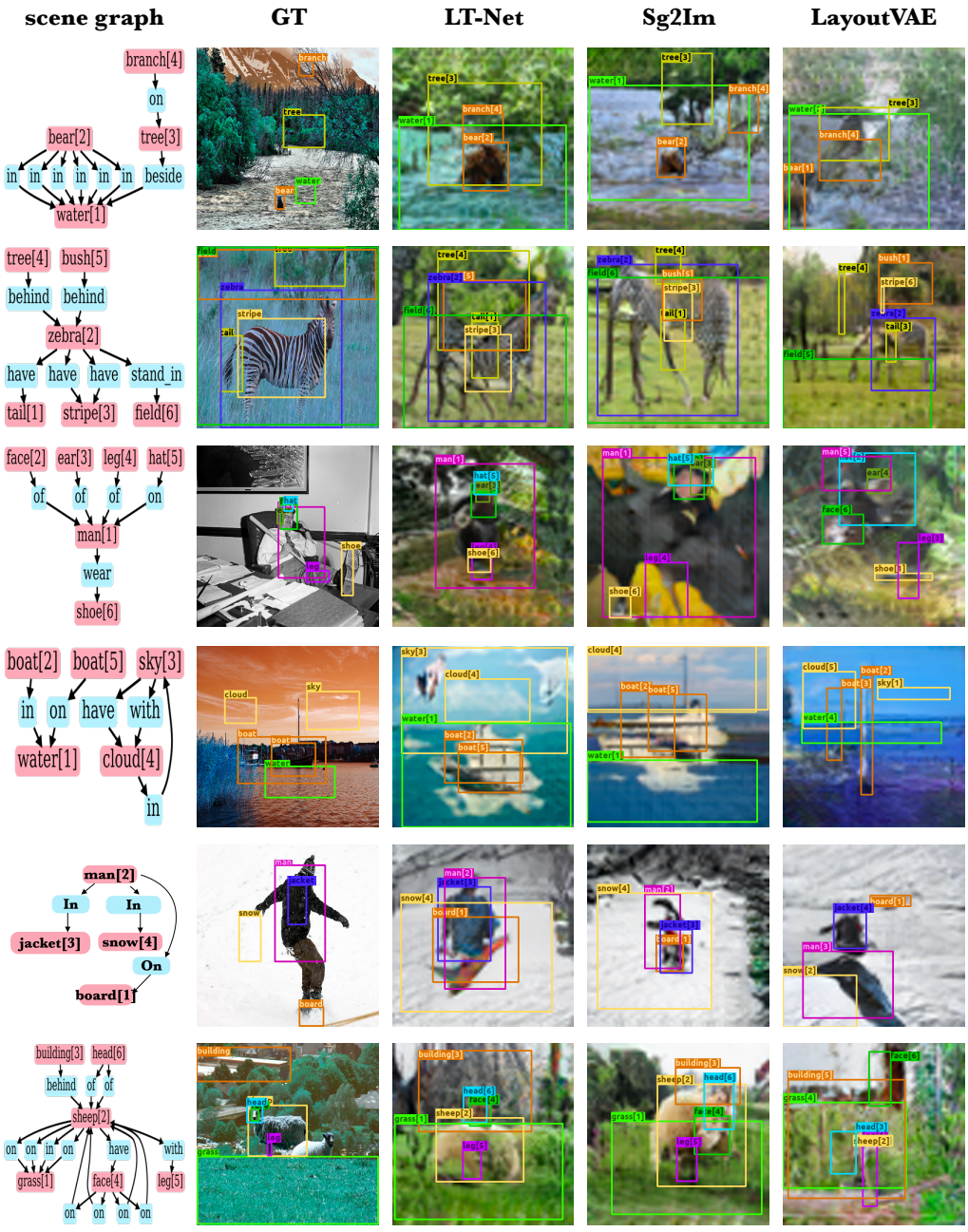

Figure 9: **Qualitative comparison on VG-MSDN (less than 6 objects).** For each row we show the textual input, ground truth layout, synthesized layout, and image converted from the layout by layout2im (Zhao et al., 2019). Note that, for simplicity, we use the scene graph to represent the textual input.

Figure 10: **Qualitative comparison on VG-MSDN (more than 6 objects).** For each row we show the textual input, ground truth layout, synthesized layout, and image converted from the layout by layout2im (Zhao et al., 2019). Note that, for simplicity, we use the scene graph to represent the textual input.

Figure 11: **More example results of multi-model layout generation on COCO-Stuff.** For simplicity, we take the scene graph to represent the textual input in each row, followed by three sampled layout outputs produced by our LT-Net.

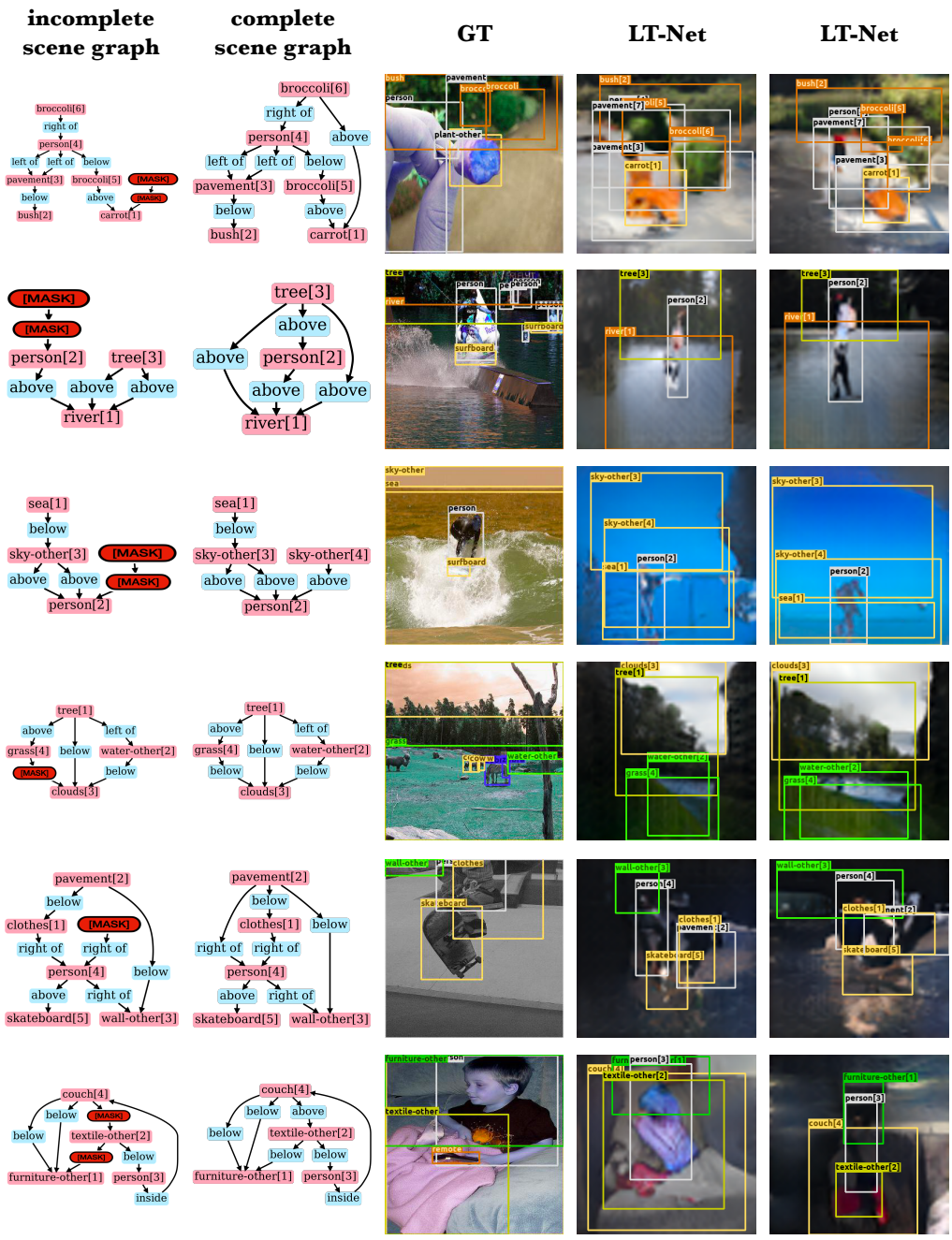

Figure 12: **More example results of inferring implicit objects and relations on COCO-Stuff.** For simplicity, we take an incomplete scene graph to represent the textual input with missing words/relation. Note that we use the word **[MASK]** in the scene graphs to represent the object/relation which is missing in the textual input. For each row, we depict two sampled layout outputs produced by our LT-Net (shown in the last two columns).