LayoutTransformer: Scene Layout Generation with Conceptual and Spatial Diversity (Supplementary Material)

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

1. Implementation Details

Our implementations are based on the Transformer [9] and Pytorch [7]. 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.:

$$\begin{aligned} p_{\theta_t}(b_t \mid c_t) &= p_{\theta_t}(x_t, y_t, w_t, h_t \mid c_t) \\ &= p_{\theta_t}(x_t, y_t \mid c_t) p_{\theta_t}(w_t, h_t \mid c_t, x_t, y_t). \end{aligned} \tag{1}$$

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.

Layout to Image Generation Since our work focuses on scene-graph-to-layout generation, we simply leverage the existing model of LostGAN [8] for layout-to-image synthesis.

*Equal Contribution

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

Table A. 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.

2. Experiments

2.1. Datasets

We perform our experiments on the COCO-Stuff dataset [1] and VG-MSDN dataset provided by [6]. Since the raw VG [4] 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 A.

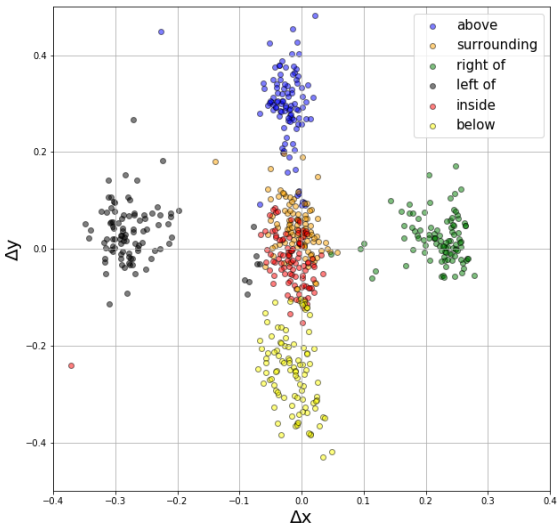

Figure A. Distribution visualization of relation priors generated by our LT-Net. Note that x and y axes represent the differences between the associated bounding boxes of subject and object pair in horizontal and vertical directions, respectively. Different colors denote each relation of interest. For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

2.2. Training details of the baselines

For Sg2Im and CanonicalSg2Im, the authors have made their implementation publicly available[1] [2]. To be more specific, we directly use the released model of Sg2Im; as for CanonicalSg2Im, we apply their model of scene-graph-to-layout synthesis, and disregard the one for layout-to-image generation. Since the authors of NDN [5] did not release their code, we follow Sects. 3 and 4.1 of their paper to implement their model.

2.3. Details of evaluation protocols

mIOU. As noted in Sec. 3.3, our LT-Net utilizes GMM to model the distributions of the bounding box coordinates for each entity (i.e., subject or object in the scene graph). With the learned GMM parameters, we predict the coordinates of each bounding box based on the associated distribution given the input scene graph. Based on VAE, NDN samples a vector from the latent space for each entity, which is fed into MLP to generate the coordinates of each bounding box. As for Sg2Im and CanonicalSg2Im, they do not predict bounding box distributions as the outputs. Instead, they feed the extracted scene graph features into MLP layers to produce layouts. When calculating mIOU, relation accuracy and FID, sampling diverse layout outputs is *not* required. We simply apply the mean of each distribution for predicting the bounding box coordinate outputs. As for NDN, it inputs the derived mean vector into the MLP for producing layouts. The above process follows the protocols of generative models for quantitative evaluation.

Diversity Score. Given the input scene graph, five layouts are sampled from the bounding box distributions derived by each model (including ours). The same LostGAN is applied to convert the layouts into images for calculating the diversity scores (ten image pairs produced by the five sampled layouts).

2.4. 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 A, which confirms that our model learns the mapping between semantic words and spatial relations. Take the distribution in green in Figure A) (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.

2.5. Qualitative Results

In this section, we present additional qualitative results following the same setting as that in Experiments 4.2. The presentation order follows that in Sect. 4.2. We compare our proposed LT-Net with recent state-of-the-art models, including Sg2Im [3], NDN [5] and CanonicalSg2Im [2].

Plausible layout generation. We conduct the experiment on both COCO and VG-MSDN datasets, and the results are shown in Figures B (COCO) and C (VG-MSDN), respectively. We demonstrate that our model is capable of handling complex objects and relations by incrementally adding more objects in images. From top to bottom we gradually increase the complexity of the input scene graph. Fig. B presents the synthesized results on the COCO dataset with 4 to 8 objects in an image. For the VG-MSDN dataset, Fig. C shows our generated images from 5 up to 10 objects in an image.

Spatially-diverse layout generation. In Fig. D, we show example layout generation results given the same scene graph input. It is worth noting that the original implementation of Sg2Im [3] did not present diverse layouts, and the diversity of the image is produced by the layout-to-image model [8]. We also note that, though NDN [5] and

[1] https://github.com/google/sg2im
[2] https://github.com/roeiherz/CanonicalSg2Im

Model	COCO		VG-MSDN	
	mIOU ↑	R@0.5 ↑	mIOU ↑	R@0.5 ↑
Sg2Im	0.29	35.69	0.168	10.92
NDN	0.33	28.03	-	-
WSGC	0.42	38.2	0.174	10.94
Ours	**0.49**	**38.76**	**0.183**	**12.03**

Table B. **Quantitative evaluation.** The bold numbers represent the best scores. Recall that NDN is not applicable on VG-MSDN since it requires complete graph annotation as inputs. Note that **WSGC** stands for CanonicalSg2Im.

CanonicalSg2Im [2] generate diverse layout outputs, their result did not semantically match the input scene graph. As for our LT-Net, spatial diversity can be produced, while the semantic plausibility is properly preserved.

Conceptually-diverse scene graph generation. To infer novel objects from an input scene graph, we construct a new triplet of subject-relation-object. Either subject or object shares the same Obj ID as those already presented in the input scene graph, while the MASK token is assigned to the remaining two entities (e.g., Fig. 5b in the main paper). To predict the masked entities, we assign the sentence ID to this newly added triplet (following the consecutive numbers from the last known triplet), feed the inputs defined in Line 324 into Predictor $\mathcal{P}$, and infer the associated embedding outputs.

Finally, we demonstrate the capability of our LT-Net to manipulate implicit objects and relationships in the scene in Figures E and F, verifying the ability of our model in producing conceptually-diverse yet plausible layouts given an input scene graph. Take the first row in Fig. E for example, it can be seen that our model predicts *sky* as an additional object in the input scene graph. Such a prediction is reasonable since *sky* is a common object in the outdoor scene. More results can be found in Fig. E and Fig. F.

2.6. Quantitative comparisons

In this section, we present additional quantitative results following the same setting as that in Experiments 4.2. The presentation order follows that in Sect. 4.2. We compare our proposed LT-Net with recent state-of-the-art models, including Sg2Im [3], NDN [5] and CanonicalSg2Im [2].

Layout generation. In Table B, we report mIOU and recall score at 0.5 (IOU threshold) and compares our LT-Net with Sg2Im [3], NDN [5] and CanonicalSg2Im [2]. We observe that our LT-Net achieved the best mIOU and recall score among all methods both on COCO and VG-MSDN dataset. This verifies the design of our LT-Net in encoding contextual features while enforcing layout recovery with relation consistency.

Image generation. In Table C, we provide additional quantitative evaluation on the generated image by taking Layout2Im [10] as another layout-to-image model for comparisons. From this table, we observe that our method achieved comparable or improved image quality, which is consistent with the results presented in Table 2 of the main paper.

Model	#Para	COCO			VG-MSDN		
		FID ↓	DS ↑	FPS	FID ↓	DS ↑	FPS
Sg2Im	0.97M	**78.46**	18.77	180	**128.03**	14.61	156
NDN	44M	106.24	**34.01**	50	-	-	-
WSGC	5.5M	113.70	25.59	59	135.63	<u>20.15</u>	19
Ours	6.2M	<u>86.53</u>	<u>27.68</u>	46	<u>133.89</u>	**33.69**	27

Table C. **Quantitative evaluation using Layout2Im.** The bold numbers represent the best scores, and the underline ones are the second highest. Recall that Sg2Im requires ground truth image for training, and NDN is not applicable on VG-MSDN since it requires complete graph annotation as inputs. Note that **WSGC** stands for CanonicalSg2Im.

The number of parameters and inference time. For the practical concern, we also provide the comparison on the number of parameters and the inference time. As for the number of parameters, we use the "count_params" tool [3] in PyTorch to calculate the amount of parameter in the model. For the inference time, we report frame per second (FPS) rate in the inference time. Table C lists the results, confirming satisfactory efficiency of our model.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018. 1

[2] 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. 2, 3

[3] 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. 2, 3

[4] 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. 1

[5] 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. 2, 3

[6] 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 Interna-*

[3] https://gist.github.com/zackenton/12a86b6e0ff274b39608e40f4a412f2b

tional Conference on Computer Vision, pages 1261–1270, 2017. 1

[7] 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*, pages 8026–8037, 2019. 1

[8] 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, 2, 5, 6, 7

[9] 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. 1

[10] 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*, pages 8584–8593, 2019. 3

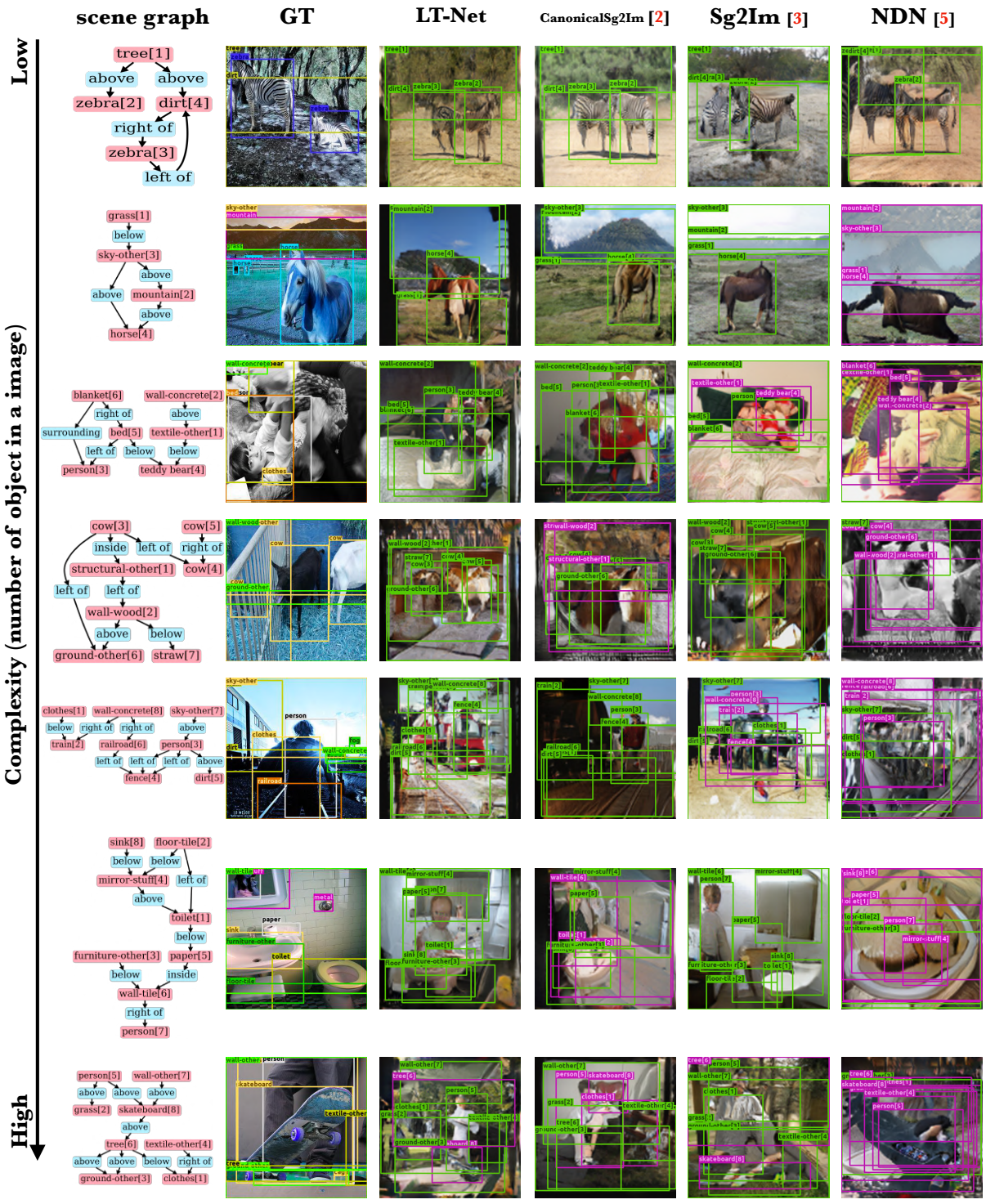

Figure B. **Qualitative comparisons of layout generation on COCO-Stuff.** For each row, we show the scene graph input, ground truth layout, and those produced by different approaches. Note that the synthesized images are converted from the corresponding layouts by [8]. Also, those bounding boxes in **green** denote the layout components matching the given description, while those in **red** do *not*.

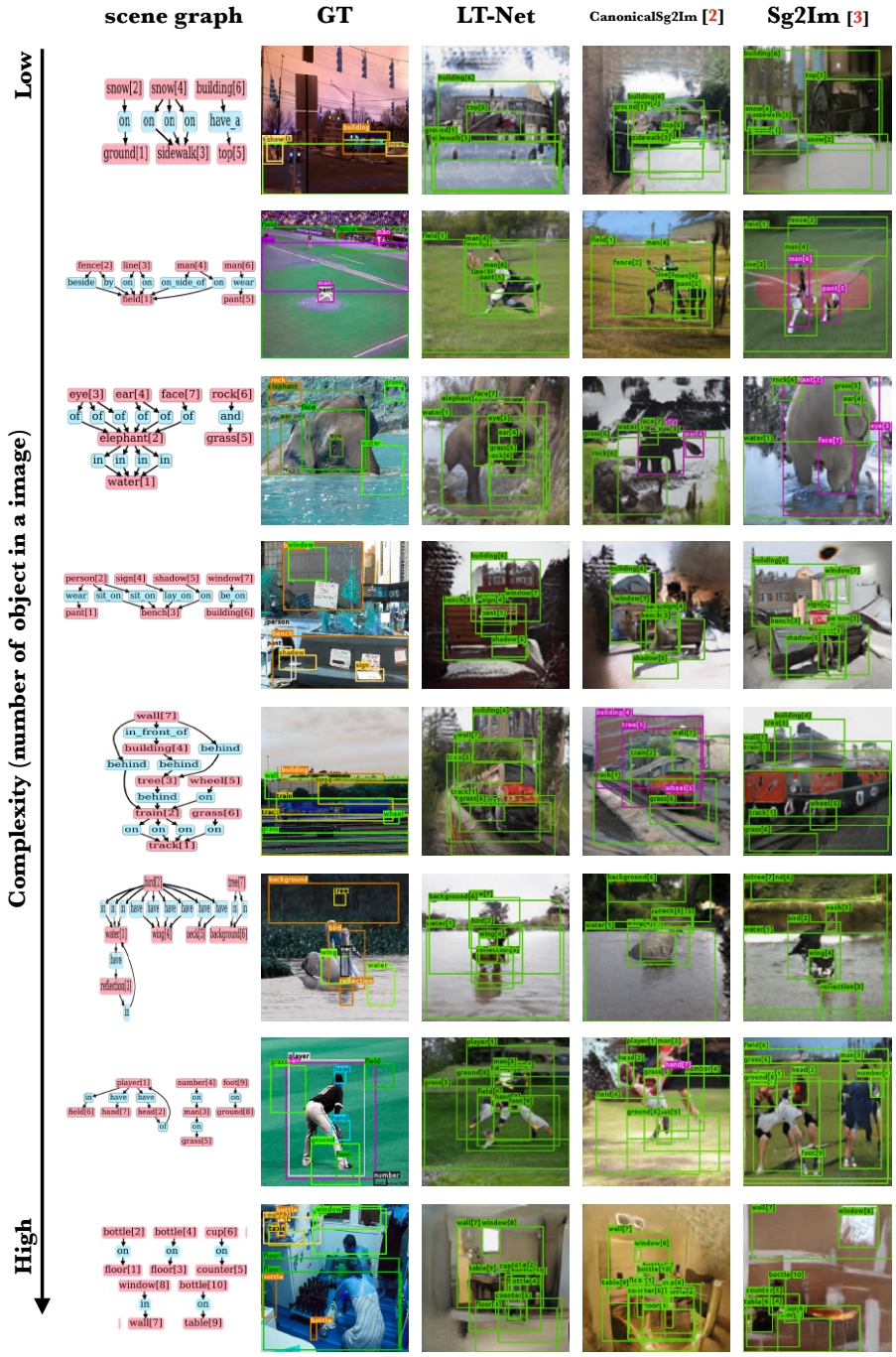

Figure C. **Qualitative comparison on VG-MSDN.** For each row we show the scene graph input, ground truth layout, synthesized layout, and image converted from the layout by LostGAN [8]. Also, those bounding boxes in **green** denote the layout components matching the given description, while those in **red** do *not*.

Figure D. **More example results of spatially-diverse layout generation on COCO-Stuff.** For each row, we show the scene graph input, ground truth layout, and two spatially-diverse layouts synthesized by different methods. Note that the produced layouts are further converted into images by [8] for visualization purposes. Note that Sg2Im does not exhibit sufficient spatial diversity, while CanonicalSg2Im and NDN might not produce results semantically matching the input. Also, those bounding boxes in **green** denote the layout components matching the given description, while those in **red** do *not*.

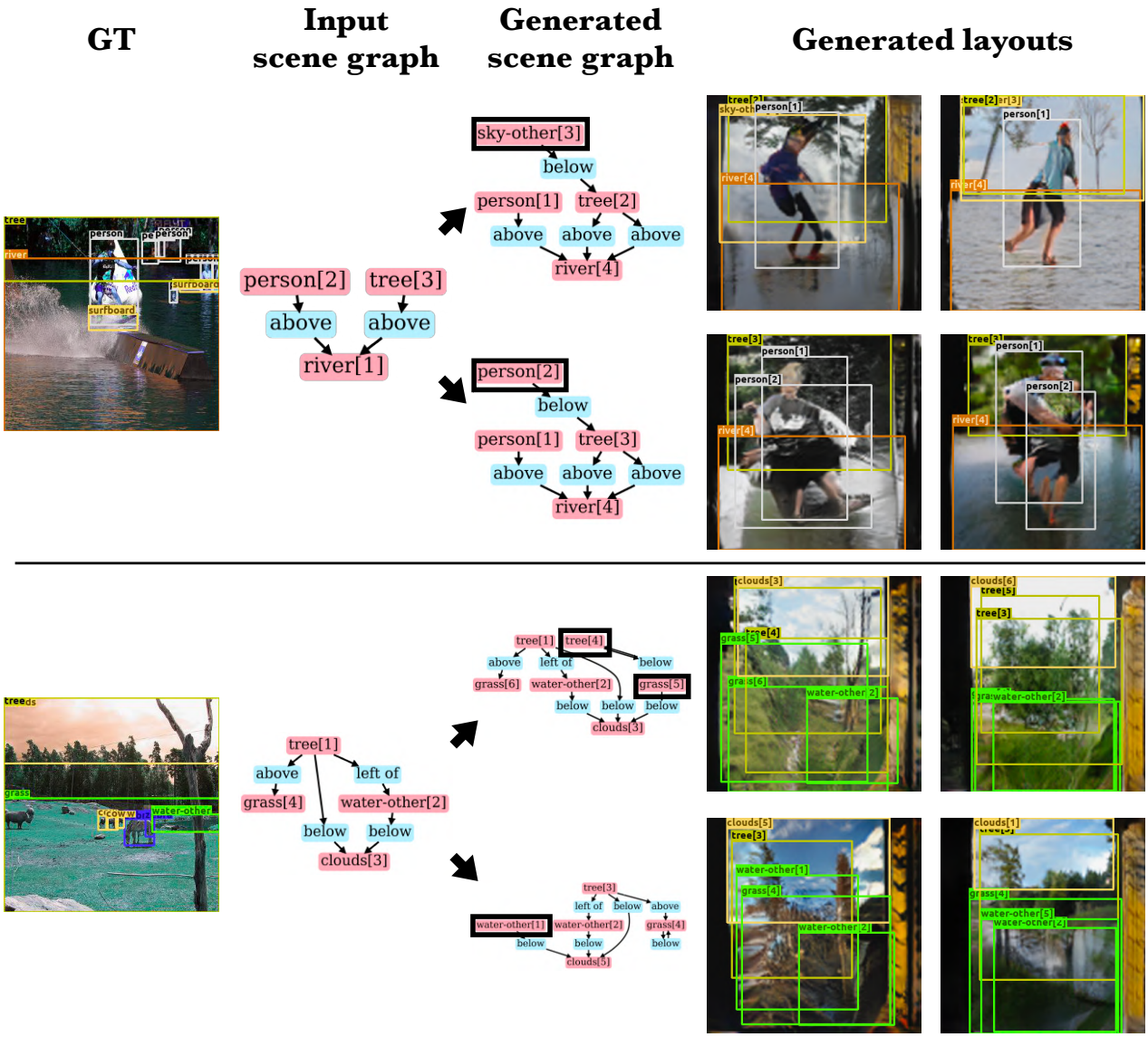

Figure E. **More example results of conceptually-diverse layout generation.** Given an input scene graph, our LT-Net generates conceptually diverse scene graphs inferring plausible objects and relationships, allowing generation of conceptually diverse layouts.

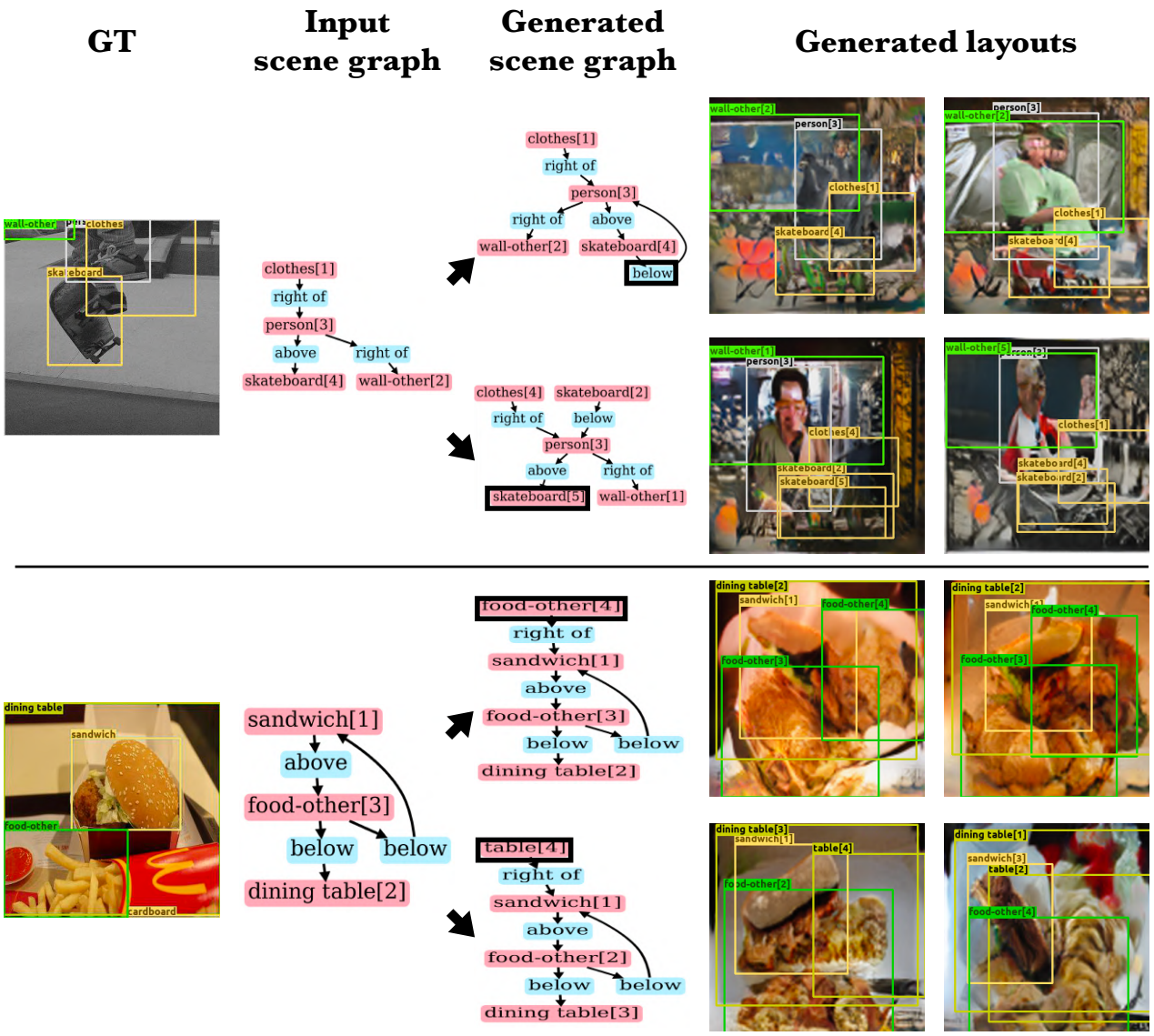

Figure F. **More example results of conceptually-diverse layout generation.** Given an input scene graph, our LT-Net generates conceptually diverse scene graphs inferring plausible objects and relationships, allowing generation of conceptually diverse layouts.