Visual-Relation Conscious Image Generation from Structured-Text Supplementary Material

Duc Minh Vo[1] and Akihiro Sugimoto[2]

[1] Department of Informatics,
The Graduate University for Advanced Studies, SOKENDAI, Tokyo, Japan
[2] National Institute of Informatics, Tokyo, Japan
{vmduc, sugimoto}@nii.ac.jp

I. GAN architecture

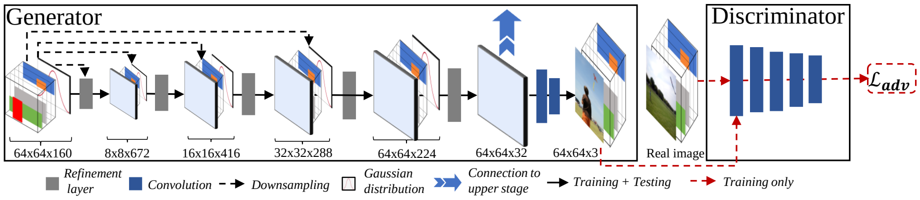


Fig. 1: Details of one GAN in the stacking-GANs. Illustrated is the first GAN which receives the visual-relation layout ($64 \times 64 \times 128$) and Gaussian distribution noise ($64 \times 64 \times 32$) as its inputs. The second (the third) GAN receives the upsampled visual-relation layout and the hidden features of previous GAN.

We employ stacking-GANs to progressively generate coarse-to-fine images. It consists of three GANs, each of which is conditioned on $\theta(t)$ (Fig. 1).

Generator. Our generator consists of five refinement layers [18] producing 512, 256, 128, 64, 32 outputs and two convolution layers outputting 32 and 3 channels.

We concatenate $\theta(t)$ and Gaussian noise (for the first generator) or the output of the last refinement layer of the previous generator (for the second and the third generators) to produce the input of $l \times l \times 160$ ($l = 64, 128, 256$). The size of the input is upsampled using the bilinear interpolation [21] to be consistent with that of the generator input. At each level of the refinement layers of each generator, the generator input is downsampled and concatenated with the output of the previous refinement layer (upsampled using the bilinear interpolation) to produce the input (except for the first refinement layer that receives the generator input only).

Discriminator. Following [6], we design our discriminator as the classification task rewarding high probability for real images and low one for generated images. Our discriminator consists of five convolution layers, outputting 64, 128, 256, 512, and 4 channels, respectively.

II. More examples

We show some more examples in this supplementary material. These examples show that our visual-relation layout preserves the scene structure more precisely than the layouts by the compared state-of-the-arts. We remark that we have confirmed that for all the results in the experiments, our method preserves the scene structure consistent with text descriptions. Figure 2 shows results obtained by our method, Johnson+ [1], Zhang+ [6], and Xu+ [9] on COCO-stuff [13]. Figure 3 shows results obtained by our method, Johnson+ [1], Zhang+ [6], and Xu+ [9] on GENOME [14].

Figures 4 and 5 show some examples obtained by the ablation models. We see that layouts by the completed model are consistent with those in ground-truth. Moreover, we see that the quality of generated images by the models w/o any subnet in our visual-relation layout module is degraded. This indicates that the well-structured layout is crucial in generating photo-realistic images.

Next, we show the relation-units obtained by the individual usage subnet on COCO-stuff [13] (Fig. 6) and GENOME [14] (Fig. 7). These examples illustrate effectiveness of the individual usage subnet in keeping entities' relations as much as possible.

Finally, we show the outputs obtained along with the stacking-GANs on COCO-stuff [13] (Fig. 8) and GENOME [14] (Fig. 9). These examples illustrate effectiveness of the stacking-GANs in our method. From 64×64 to 128×128 resolution, details of images are significantly improved. From 128×128 to 256×256 resolution, on the other hand, the generated images become sharper. This can be explained as follows. At the third GAN (256×256), the visual-relation layout is upsampled twice using bilinear interpolation [21] (from 64×64 to 256×256). This twice upsamplings may weaken the entity embeddings in the visual-relation layout. As a result, the layout is not strong enough for the third GAN to generate more details of entities (compared to the second GAN). Nevertheless, thanks to the output of the second GAN, the third GAN can retain the scene structure and generate sharper images. This observation leaves a room to improve the quality of generated images.

References

[1] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in CVPR, 2018.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in NIPS, 2014.

[6] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in ICCV, 2017.

[9] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in CVPR, 2018.

[13] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in CVPR, 2018.

[14] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," in IJCV, 2017.

[18] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in ICCV, 2017.

[21] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in NIPS, 2015.

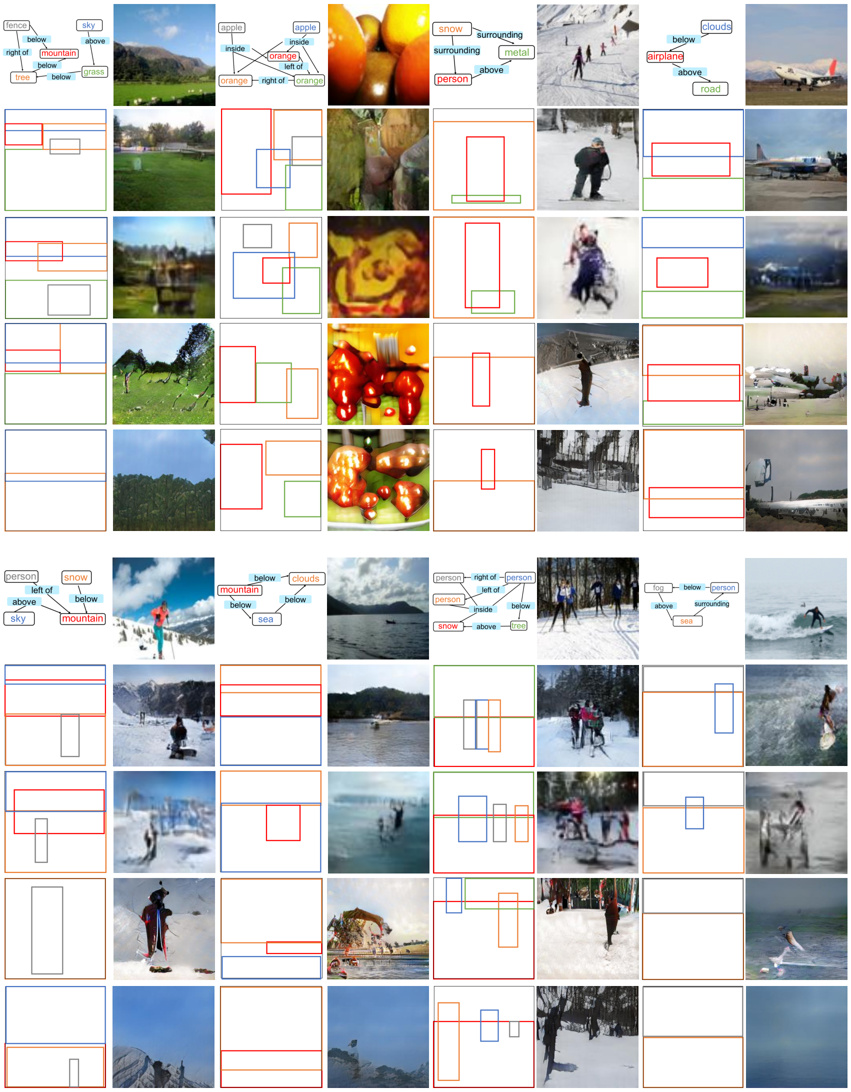


Fig. 2: Visual comparison on COCO-stuff [11]. For each example, we show the scene graph and reference image at the first row. From second to the last rows, we show the layouts and images generated by our method (256 × 256), Johnson+ [1] (64 × 64), Zhang+ [6] (256 × 256), and Xu+ [9] (256 × 256). The color of each entity bounding-box corresponds to that in the scene graph. Scene graphs and layouts are enlarged for best views.

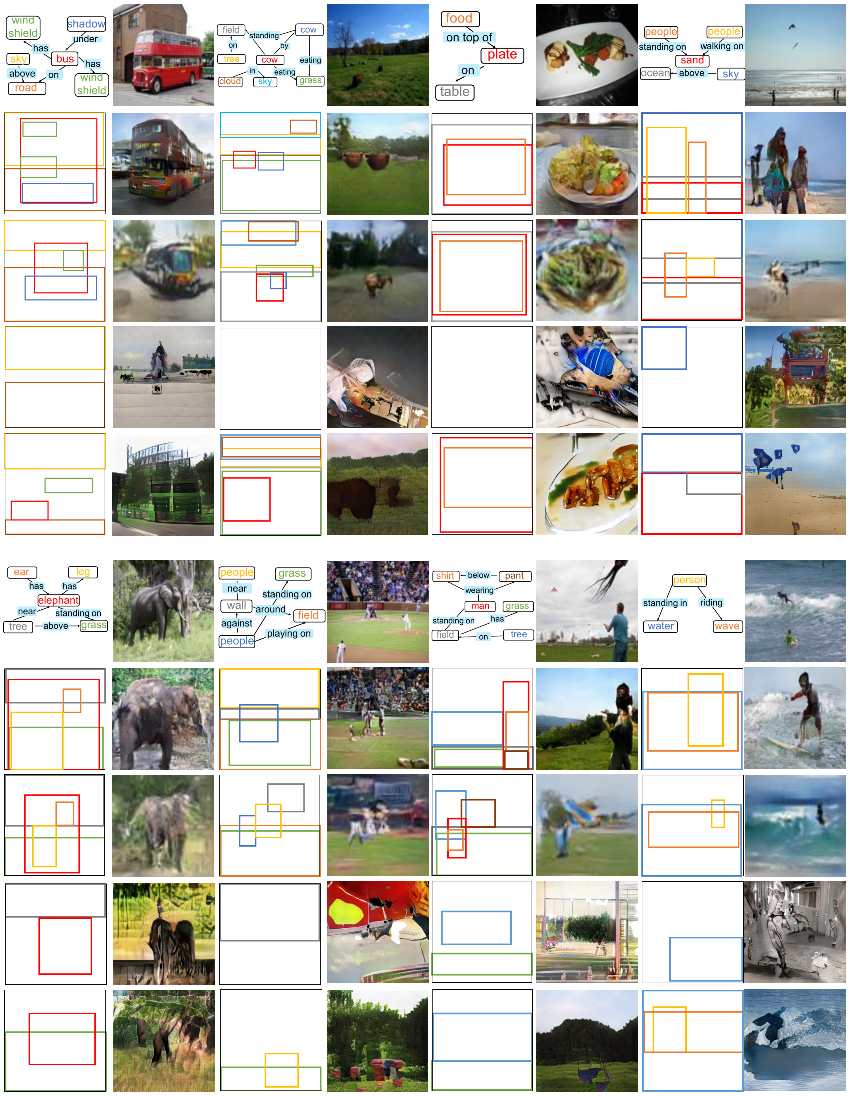
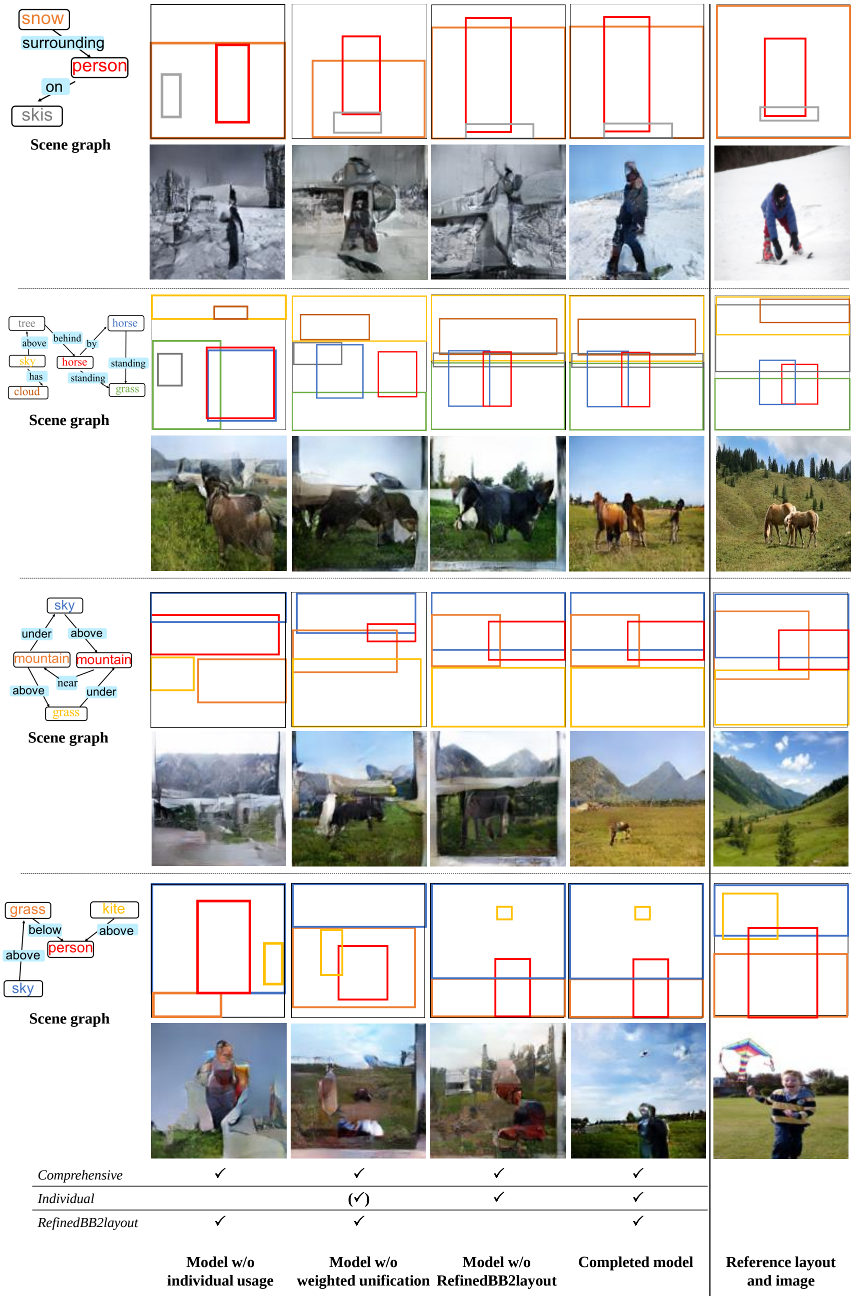
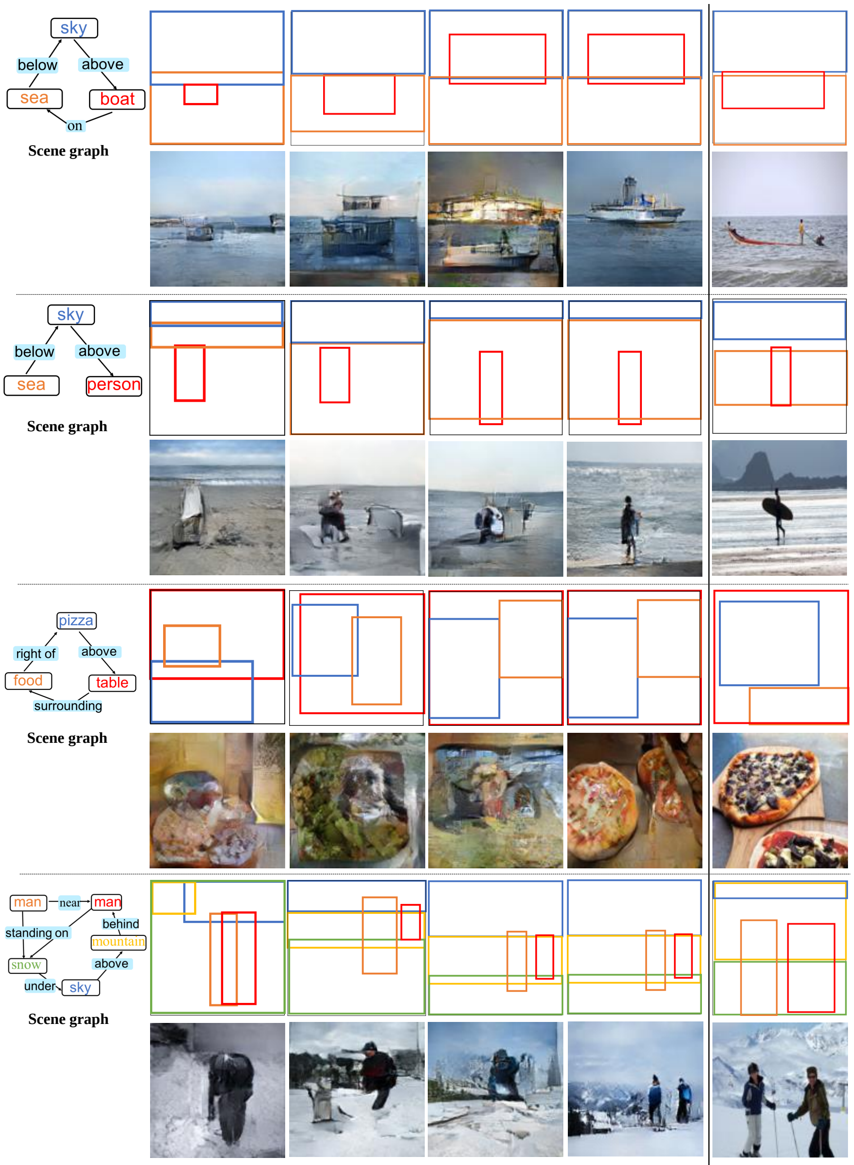


Fig. 3: Visual comparison on GENOME [14]. For each example, we show the scene graph and reference image at the first row. From second to the last rows, we show the layouts and images generated by our method (256 × 256), Johnson+ [1] (64 × 64), Zhang+ [6] (256 × 256), and Xu+ [9] (256 × 256). The color of each entity bounding-box corresponds to that in the scene graph. Scene graphs and layouts are enlarged for best views.



	Model w/o individual usage	Model w/o weighted unification	Model w/o RefinedBB2layout	Completed model	Reference layout and image
Comprehensive	✓	✓	✓	✓	
Individual		(✓)	✓	✓	
RefinedBB2layout	✓	✓		✓	

Fig. 4: Example of layouts and generated images by the ablation models. For each model in each sample, the 1st row shows the layout, the 2nd row shows the generated image. All images are at 256 × 256 resolution.



	Model w/o individual usage	Model w/o weighted unification	Model w/o RefinedBB2layout	Completed model	Reference layout and image
Comprehensive	✓	✓	✓	✓	
Individual		(✓)	✓	✓	
RefinedBB2layout	✓	✓		✓	

Fig. 5: Example of layouts and generated images by the ablation models. For each model in each sample, the 1st row shows the layout, the 2nd row shows the generated image. All images are at 256 × 256 resolution.

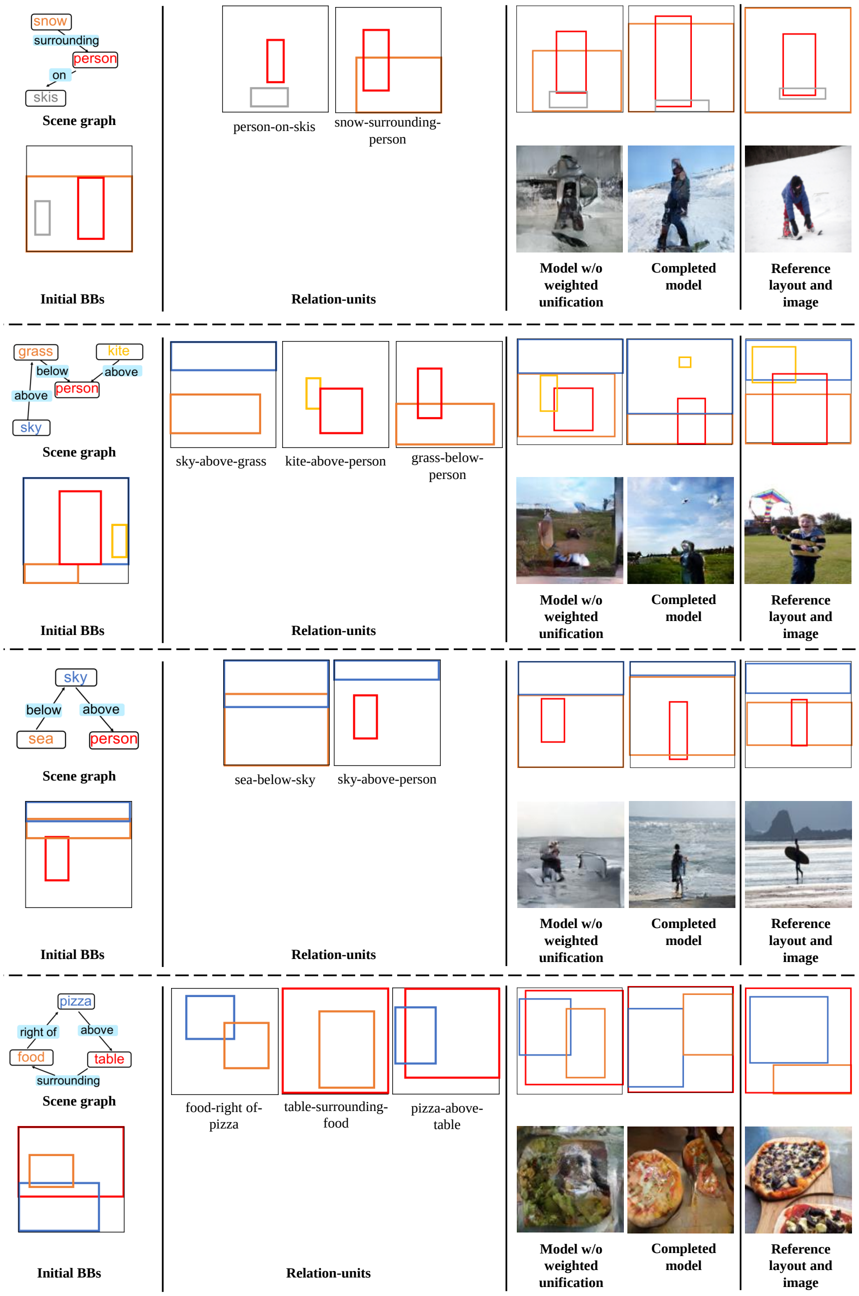


Fig. 6: Example of relation-units in the individual usage subnet on COCO-stuff [11]; layouts and generated images by model w/o weighted unification and completed model.

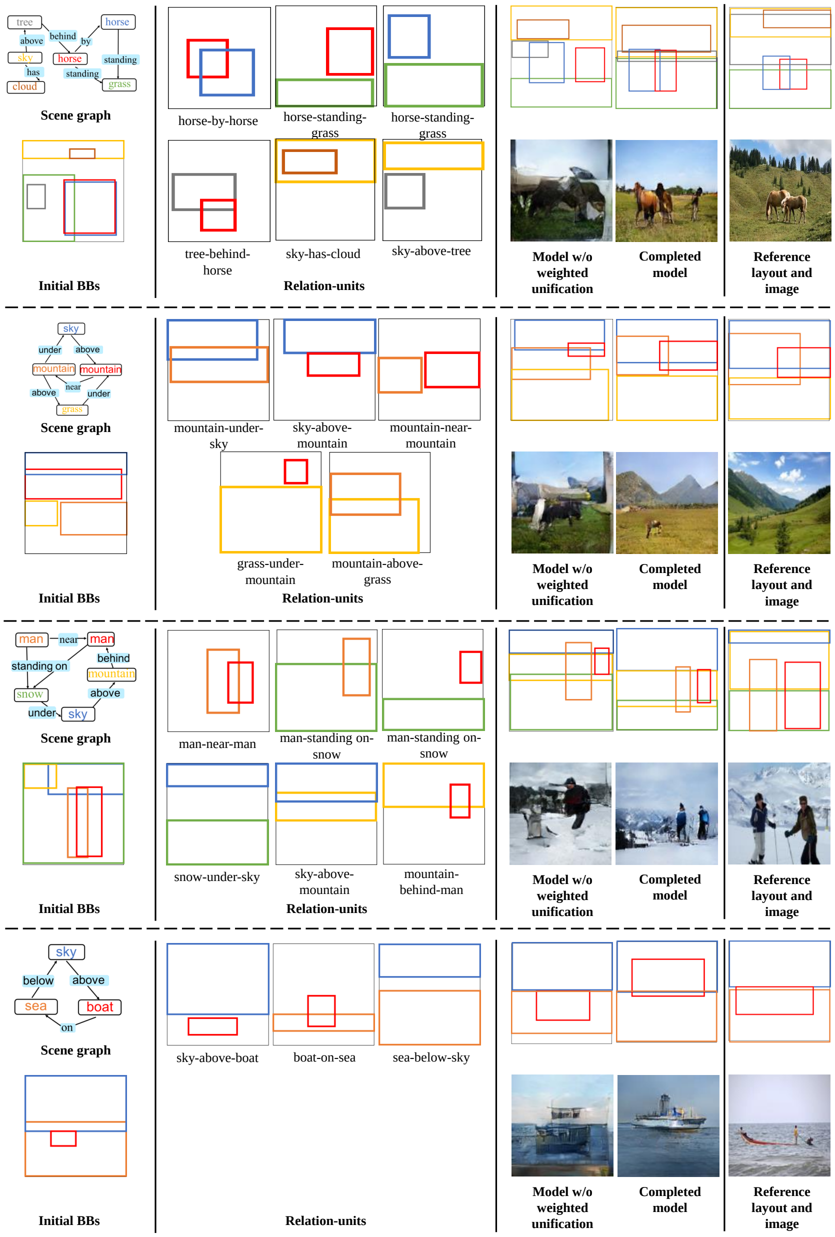


Fig. 7: Example of relation-units in the individual usage subnet on GENOME [14]; layouts and generated images by model w/o weighted unification and completed model.

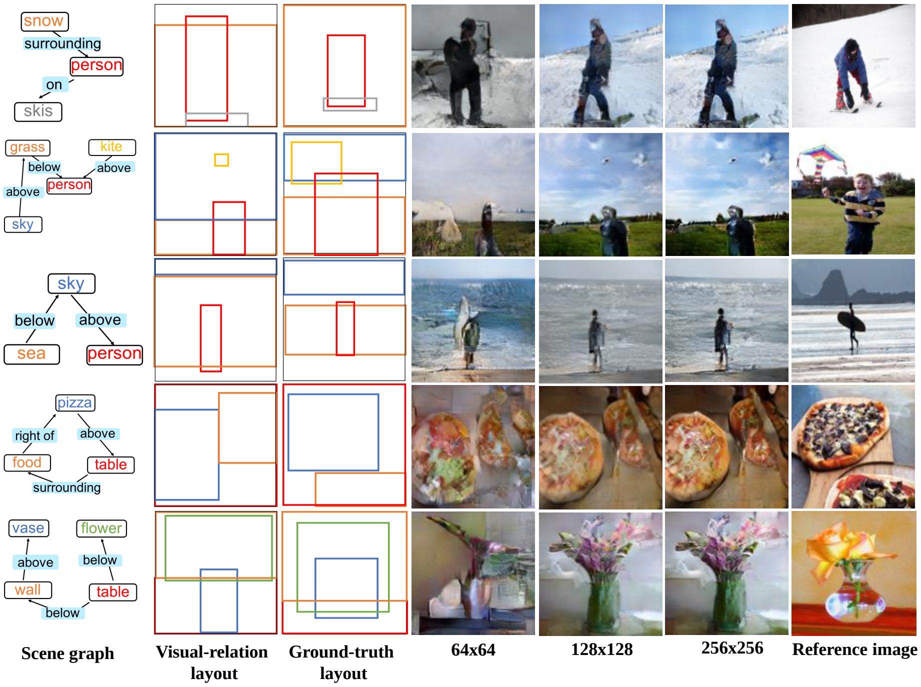
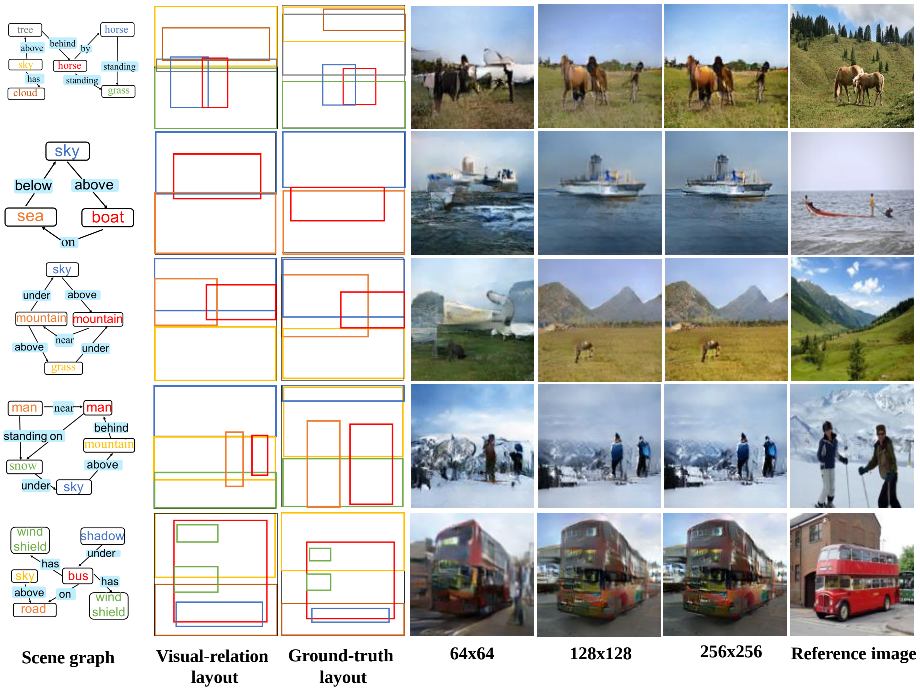


Fig. 8: Examples of outputs along with the stacking-GANs on COCO-stuff [11]. From left to right, scene graph, visual-relation layout, the outputs at 64 × 64, 128 × 128, 256 × 256 resolutions, and the reference image.



Scene graph

Visual-relation layout

Ground-truth layout

64x64

128x128

256x256

Reference image

Fig. 9: Examples of outputs along with the stacking-GANs on GENOME [14]. From left to right, scene graph, visual-relation layout, the outputs at 64 × 64, 128 × 128, 256 × 256 resolutions, and the reference image.