

# Supplementary Material

## SceneGenie: Scene Graph Guided Diffusion Models for Image Synthesis

### 1. Architecture Details

#### 1.1. Denoising UNet

The denoising UNet of latent diffusion model [5] takes latent code of image compressed by KL-regularized autoencoder as input. Table 1 shows our denoising UNet model hyperparameters.

Table 1: Model hyperparameters

<i>z</i> -shape	(32,32,4)
Diffusion steps	1000
Noise Schedule	linear
Channels	320
Channel Multiplier	1,2,4,4
Number of Heads	8
Conditioning	CA
Cross-Attention Resolutions	32,16,8
Transformers Depth	1
Context dimension	1280

#### 1.2. SG2SEG

**B.BoxNet Architecture** The B.Box Net predicts bounding box for each object embeddings from the GCN. Specifically, it predicts the 2D coordinates of the bottom-left vertex ( $x_l, y_l$ ) and upper-right vertex ( $x_r, y_r$ ) of the bounding box. The B.Box Net contains two layers of MLP with ReLU activation function. The architecture is shown in Table 2.

Table 2: B.Box Net Architecture

Block	Operation	Output Shape
-	Object embedding	[128]
MLP	Linear	[512]
	ReLU	[512]
	Linear	[4]

**MaskNet Architecture** The MaskNet for semantic segmentation prediction in the SG2SEG network consists of a set of convolutional blocks that upsample input embeddings

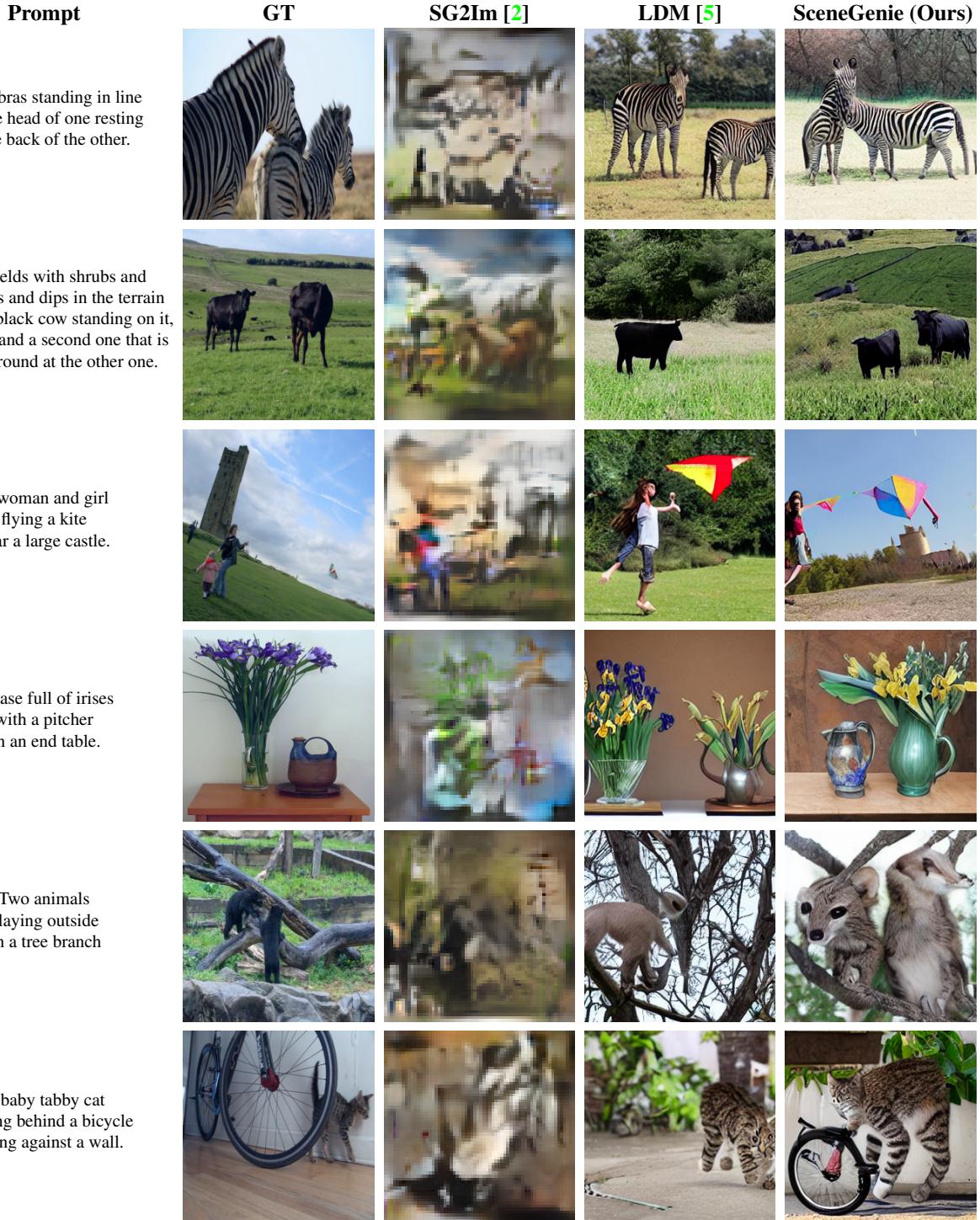
to the desired resolution in a cascaded manner. Each convolutional block contains an Upsample layer with ratio 2, a BatchNorm layer, a Convolution layer with kernel size 3 and padding 1, and a ReLU activation function. We stack the convolutional blocks to generate  $64 \times 64$  segmentation maps. The architecture is shown in Table 3.

Table 3: MaskNet Architecture

Block	Operation	Output Shape
Conv Block 1	Object embedding	[128]
	Reshape	[128, 1, 1]
	Upsample	[128, 2, 2]
	BatchNorm	[128, 2, 2]
	Conv 3x3	[128, 2, 2]
Conv Block 2	ReLU	[128, 2, 2]
	Upsample	[128, 4, 4]
	BatchNorm	[128, 4, 4]
	Conv 3x3	[128, 4, 4]
	ReLU	[128, 4, 4]
Conv Block 6	...	...
	Upsample	[128, 64, 64]
	BatchNorm	[128, 64, 64]
	Conv 3x3	[128, 64, 64]
	ReLU	[128, 64, 64]
Output	Conv 1x1	[1, 64, 64]
	Sigmoid	[1, 64, 64]

## 2. Additional Qualitative Results on COCO

Figure 1 shows additional qualitative results of our method, compared against SG2Im [2] and LDM [5] on COCO [1] dataset.



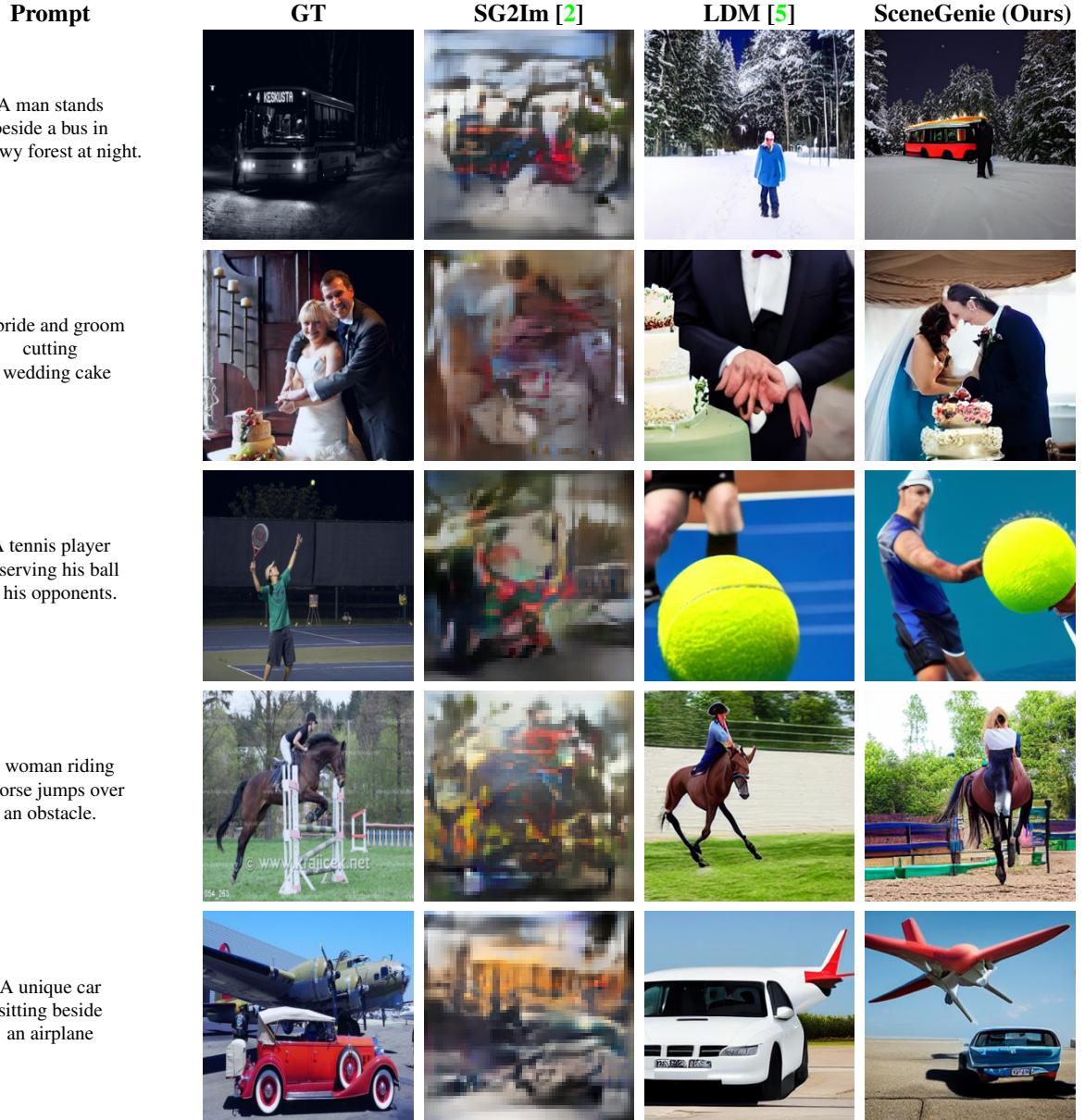


Figure 1: **Additional qualitative results on the comparison of SceneGenie against related work on the COCO stuff [1] test set.** As it can be seen, the images generated by SceneGenie represent the given prompt more accurately compared to previous work. SceneGenie, in addition to high quality image generation, correctly generates the number of given instances in the image and represents the scene more accurately overall.

### 3. Additional Qualitative Results on VG

Figure 2 shows additional qualitative results of our method, compared against SG2Im [2] and LDM [4] on VG [3] dataset.

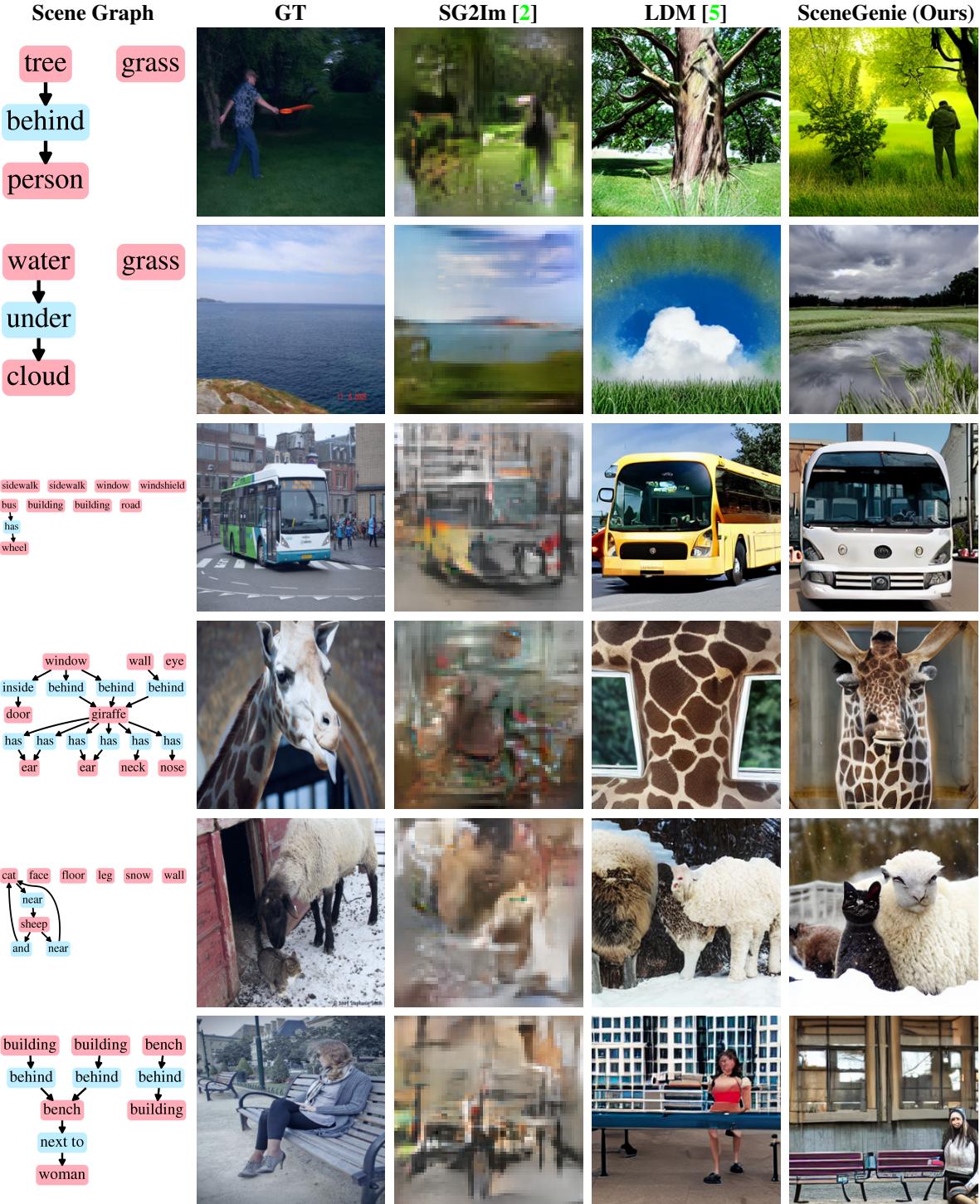




Figure 2: Additional qualitative results on the comparison of SceneGenie against related work on the VG [3] test set.

## References

- [1] 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. [2](#), [3](#)
- [2] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. [2](#), [3](#), [4](#), [5](#)
- [3] 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. *IJCV*, 123(1):32–73, 2017. [4](#), [5](#)
- [4] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. [4](#)
- [5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. [1](#), [2](#), [3](#), [4](#), [5](#)