Spatially Compositional Diffusion

Ryan Lian[1*] Xingjian Bai[2*] Joy Hsu[1] Weiyu Liu[1] Jiayuan Mao[3] Jiajun Wu[1]
[1]Stanford University [2]University of Oxford [3]MIT

Abstract

Despite proficiency in generating realistic images, current text-to-image diffusion models often fail to render spatial relationships within complex scenes faithfully, especially in scenarios involving multiple objects and relations. This suggests that such models lack an inherent mechanism to understand and represent the compositional spatial relations as indicated by the input text. To address this shortfall, we introduce an innovative approach to enhance the relational compositionality of diffusion models. In particular, our model takes scene graphs that encode object descriptions and their relations as the input specification, and generates images with a two-stage generation pipeline. It first generates a spatial layout from the scene graph, and then generates images conditioned on the created layouts. In each stage, our method leverages composable diffusion models for each individual object and relation in the scene graph, integrating their outputs during denoising steps. Our framework shows improved spatial compositionality on the CLEVR dataset. Moreover, when trained on simple two-object scenes, our model can generalize to multi-object scenes with complex spatial relations. Leveraging compositionality, our model demonstrates potential for generating complicated scenes with high fidelity.

1. Introduction

In recent years, deep generative models, such as GANs and diffusion models [2, 12, 30], have made significant strides, demonstrating versatility in diverse applications. Applications based on large vision-language models like DALL-E and Stable Diffusion have enabled the creation of highly photorealistic images from textual descriptions [23, 26, 27, 29]. These generative models excel at creating images in various styles and scenes. However, they often fail to accurately capture simple yet specific spatial relationships in prompts, such as "an apple on a plate, a banana to the right." Prior works have shown that, when given input text with multiple objects, these models often resort to an imprecise bag-of-words approach [9, 31, 36].

*These authors contributed equally to this work. Correspondence to Ryan Lian: ryanlian@stanford.edu

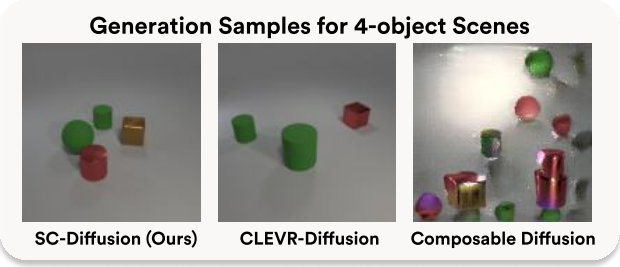

Figure 1. Comparison of generated samples using our method and prior works, with 4 objects specified in the prompt. SC-Diffusion successfully generates accurate object attributes and spatial relationships, while CLEVR-Diffusion shows missing objects and incorrect attributes, and Composable Diffusion struggles with noisiness. See Appendix D for the prompt.

Our goal is to address this failure mode by enhancing the models' ability to learn and compose spatial relationships. This spatial compositionality is a crucial desideratum for generative models, as it allows users to control the manipulation of image contents and supports generalization to an exponentially diverse range of scenes by recomposing learned visual features from a finite set of objects and relations. Our approach improves spatial compositionality by modularizing the diffusion process to reflect the structure of an object-centric scene description. Specifically, we represent scenes as scene graphs comprising objects and their interrelations. We present Spatially Compositional Diffusion (SC-Diffusion), a model that accepts scene graphs encoding object descriptions and their relations as input, and generates images using a two-stage generation pipeline. Stage I generates a layout, and Stage II produces the final image guided by this layout. By incorporating compositionality into the model's structure, we enable not only faithful image generation but also generalization across more complex scenes with more objects and relationships.

2. Related Works

Controllable text-to-image generation Recent text-to-image generation models, like Imagen [29] and DALL-E 2 [26], have shown impressive realism and stylistic flexibility, primarily using diffusion-based generative models [12]. Latent diffusion models (LDMs) improve upon these with enhanced generation fidelity and efficiency [27]. De-

spite their advancements, challenges remain in generating accurate representations from complex prompts, particularly in maintaining spatial and semantic relationships [9, 21, 31, 36]. Efforts to increase control have led to the development of techniques such as cross-attention layer manipulation [10], mask-guided edits [3], and zero-shot editing [7]. These methods help navigate the limitations of initial generations, especially for long and complex prompts.

Spatially conditioned image generation Spatial accuracy in image generation has been explored through structured visual inputs like segmentation maps and layouts [6, 24, 39]. Diffusion models have been extended to use structured inputs like bounding box layouts and edge maps for more grounded generation [17, 32, 37]. Additionally, the use of large language models (LLMs) in creating visual priors from text [8, 18, 38] and using scene graphs as inputs [16, 33] have shown promise. However, these methods still struggle with complex prompts with complete descriptions of scenes due to their holistic rather than compositional processing of inputs. RPG Diffusion [34] is a prime example. It can leverage vision language models to iteratively plan detailed layouts that enforce 2D spatial accuracy, but the complementary regional diffusion does not guarantee spatial accuracy with heavily overlapping layouts as no form of spatial reasoning is embedded in the diffusion process.

Composable diffusion models Addressing the limitations of spatial and semantic complexity, several works have explored composable diffusion models. Recent work involves parametrizing diffusion models as energy-based models for simple compositions [4, 5, 19, 20]. Approaches such as Collaborative Diffusion [13] and MultiDiffusion [1] suggest using an orchestrator module or treating composition as an optimization problem. These models offer higher-level semantic juxtapositions but still face challenges in composing specific objects and relations in detailed scenes.

3. Spatially Compositional Diffusion

We study the text-to-image generation task and simplify this setting by considering every scene as primarily composed of a set of objects and a set of relations between the objects. The input to our method is a *scene graph* [14] (See Figure 2), and our model outputs an image $\mathbf{x}$. This eliminates some ambiguity from natural language while preserving flexibility, and, in practice, we can conceivably get the scene graphs from a language-to-code model. More concretely, given a set of all objects $\mathcal{C}_o$ (e.g. "small metal red ball") and relations $\mathcal{C}_r$ (e.g. "on"), a scene graph is the tuple $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{o_i \in \mathcal{C}_o\}_{i=1}^n$ is a set of objects, and $\mathcal{E} = \{e_{ij} = (o_i, r_{ij}, o_j), r_{ij} \in \mathcal{C}_r\}_{1 \le i,j \le n}$ is a set of relationships between objects. It is the set of directed edges from o_i to o_j, analogous to (subject, predicate, object). Note that objects can be general subsets of a scene.

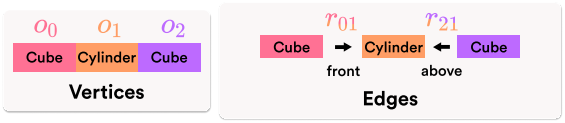

Figure 2. An example of a scene graph input to SC-Diffusion.

Since current diffusion models (See appendix A) have demonstrated strong performance in generating objects, our main challenge under this formulation is modeling spatial relations. Our insight is to improve spatial accuracy by enforcing local object relationships with the new relation denoiser. Due to the complexity of generating images with accurate spatial relations, we take inspiration from recent works and propose a two-stage generation process. The first stage is generating a bounding box layout that satisfies the relations in the scene graph for the final image. This allows us to focus on modeling the higher-level spatial relations (e.g., "left/right", "above/below") without considering pixel-level details. Once a spatially correct layout is generated, our second stage is to generate the image grounded by the layout — the goal for this stage is to model the lower level, more semantically complex, and spatially ambiguous relations between overlapping objects (e.g., "front/behind", "holding"). Figure 3 shows the overall architecture SC-Diffusion, and Appendix B contains more details.

Stage I: Compositional layout generation Given a scene graph $S = (O, R)$, we aim to generate a layout $\mathbf{b} = (\mathbf{b}^{(1)}, \cdots, \mathbf{b}^{(n)})$ where for each object $o_i \in \mathcal{V}$ we generate an axis-aligned bounding box $\mathbf{b}^{(i)} \in \mathbb{R}^4$ parameterized as $[\text{center } x, \text{center } y, \text{width}, \text{height}]$.

Given all the objects and relations specified in a scene graph, we aim to learn a generative distribution $p(\mathbf{b}^{(i)}|\mathcal{V}, \mathcal{E})$ of the bounding boxes. We parameterize this distribution by composing an object-conditioned diffusion model with a relation-conditioned diffusion model. Concretely, the object-conditioned diffusion model is a denoiser ϵ_θ^O that takes in the current diffusion timestep t, an object attribute o_i, a noisy bounding box $\mathbf{b}_t^{(i)}$ for o_i, and predicts noise over $\mathbf{b}_t^{(i)}$. The relation-conditioned diffusion model is a separate denoiser ϵ_θ^R that takes the current diffusion timestep t, a directed edge e_{ij}, and the noisy bounding boxes of the objects o_i and o_j in e_{ij} concatenated together as $[\mathbf{b}_t^{(i)} : \mathbf{b}_t^{(j)}]$, and predicts noise over both bounding boxes. For simpler notation, let the output of the relation-conditioned denoiser $\epsilon_\theta^R([\mathbf{b}_t^{(i)} : \mathbf{b}_t^{(j)}]|e_{ij}, t)$ be denoted as $[\epsilon_\theta^R(\mathbf{b}_t^{(i)}|e_{ij}, t) : \epsilon_\theta^R(\mathbf{b}_t^{(j)}|e_{ij}, t)]$. Under the energy-based interpretation of diffusion models [4, 5, 20], we can then predict noise over $\mathbf{b}_t^{(i)}$ by combining the denoisers with

$$\epsilon_\theta(\mathbf{b}_t^{(i)}|\mathcal{V}, \mathcal{E}, t) = \epsilon_\theta^O(\mathbf{b}_t^{(i)}|o_i, t) + \sum_{e_{ij} \in \mathcal{E}} \epsilon_\theta^R(\mathbf{b}_t^{(i)}|e_{ij}, t).$$

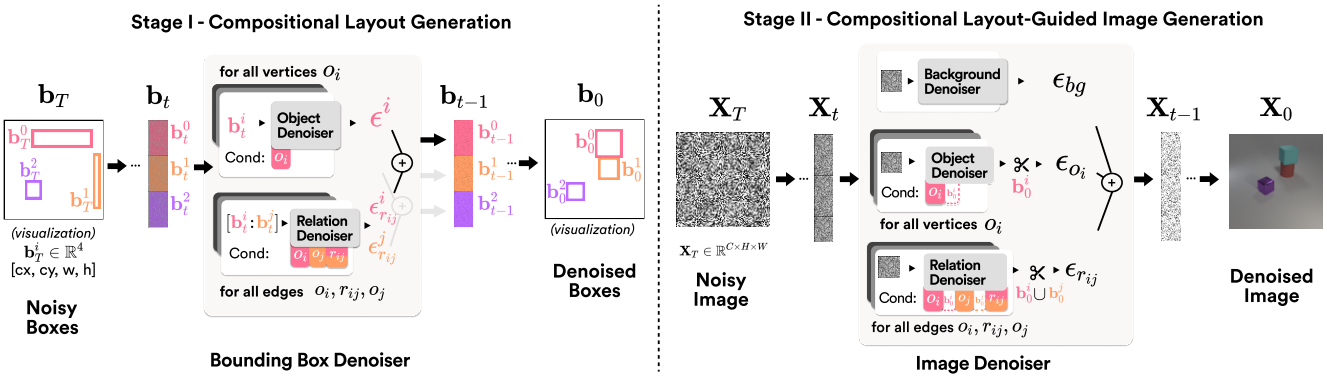

Figure 3. A detailed illustration of the two-stage process and SC-Diffusion architecture. The first stage focuses on generating spatial layouts from detailed scene graphs, utilizing separate diffusion models for individual objects and their interrelations. These layouts then serve as inputs for the second stage, where images are generated conditioned on the spatially accurate layouts.

During training, we follow the standard diffusion protocol by uniformly sampling a timestep t and randomly sampling a Gaussian noise ϵ, then optimizing the following loss:

$$\mathcal{L}_{mse} = ||\epsilon - \epsilon_\theta(\sqrt{1-\beta_t}\mathbf{b} + \sqrt{\beta_t}\epsilon \mid \mathcal{V}, \mathcal{E}, t)||_2^2,$$

where β_t is the diffusion noise schedule [12] and $\mathbf{b}$ is the ground truth layout.

Stage II: Compositional layout-guided image generation Our goal is to modularly generate an image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ conditioned on the generated layout $\mathbf{b} = (\mathbf{b}^{(1)}, \cdots, \mathbf{b}^{(n)})$ and following the object and relation attributes. We assume that the layout is accurate here and focus on enforcing the relational accuracy inside overlapping bounding boxes.

Formally, we parameterize the generative distribution $p(\mathbf{x}|\mathcal{V}, \mathcal{E}, \mathbf{b})$ with the composition of three diffusion models: object-conditioned diffusion, relation-conditioned diffusion, and unconditional diffusion. Given a timestep t, an object attribute o_i, a noisy image $\mathbf{x}_t$, and a bounding box $\mathbf{b}^{(i)}$, the object denoiser ϵ_ϕ^O predicts noise over $\mathbf{x}_t$. To ensure that the object denoiser only denoises within the relevant region in the layout, we construct a binary mask $\mathbf{m}^{(i)} \in \mathbb{R}^{H\times W\times 3}$ from the bounding box and apply the mask to the denoiser output. For brevity, let us denote

$$\eta_{o_i}^O = \epsilon_\phi^O(\mathbf{x}_t|o_i, \mathbf{b}^{(i)}, t) \otimes \mathbf{m}^{(i)},$$

where $\otimes$ is the Hadamard product. Similarly, the clipped relation denoiser can be written as the following:

$$\eta_{e_{ij}}^R = \epsilon_\phi^R(\mathbf{x}_t|e_{ij}, \mathbf{b}^{(i)}, \mathbf{b}^{(j)}, t) \otimes (\mathbf{m}^{(i)} \cup \mathbf{m}^{(j)}).$$

The unconditional background denoiser ϵ_ϕ^B simply predicts the noise over $\mathbf{x}_t$ given $\mathbf{x}_t$ and t for the purpose of denoising the background components that are not affected by the object and relation denoiser, ensuring a cohesive result. Composing the three diffusions together by adding the predicted noises, we arrive at the combined denoiser ϵ_ϕ:

$$\epsilon_\phi(\mathbf{x}_t|V, E, \mathbf{b}, t) = \epsilon_\phi^B(\mathbf{x}_t|t) + \sum_{o_i\in\mathcal{V}} \eta_{o_i}^O + \sum_{e_{ij}\in\mathcal{E}} \eta_{e_{ij}}^R$$

Similar to Stage I, during training, we randomly sample a timestep t and Gaussian noise ϵ, then optimize with the loss

$$\mathcal{L}_{mse} = ||\epsilon - \epsilon_\phi(\sqrt{1-\beta_t}\mathbf{x} + \sqrt{\beta_t}\epsilon \mid \mathcal{V}, \mathcal{E}, \mathbf{b}, t)||_2^2, \quad (1)$$

where β_t is the diffusion noise schedule and $\mathbf{x}$ is the ground truth image.

4. Experiments

Relational CLEVR We present comparisons of our model's performance compared to that of prior methods on a variation of CLEVR [15]. In addition to the regular relations of *left/right* and *front/behind*, this variation also includes *above/below*. This introduces more complex spatial relationships by highlighting the 3D nature of the dataset with more overlaps and occlusions. Following prior work [20], we generate 40,000 examples with two objects as the train set. For evaluation, we generate 100 examples for 2, 3, 4, and 8 object scenes as the test set.[1] This set-up is designed to test the model's ability to generalize to scenes with more objects, so we train SC-Diffusion on the scenes with 2 objects only. Some baselines require calibration to prompts of different lengths and are trained on a mixture of 1, 2, and 3 object scenes. Nonetheless, our method still demonstrates better generalization. To assess image fidelity, we utilize the Frechet Inception Distance (FID) metric [11]. To evaluate spatial accuracy, we use the Scene Relation Score (RA) [32] and introduce an object accuracy metric (OA) (details in Appendix C).

[1] Due to the cost of the OpenAI API required for LayoutGPT [8]

Table 1. Overview of quantitative results. Our model consistently outperforms all baselines in object accuracy (OA), relational accuracy (RA), and Fréchet Inception Distance (FID) metrics, demonstrating substantial improvements as the number of objects in a scene increases. These results underscore our model's superior generalization ability in scenes with high spatial complexity.

Method	2 Objects			3 Objects			4 Objects			8 Objects		
	OA (%)	RA (%)	FID ↓	OA (%)	RA (%)	FID ↓	OA (%)	RA (%)	FID ↓	OA (%)	RA (%)	FID ↓
LayoutGPT + GLIGEN	19.0	9.0	157.2	16.3	9.2	157.3	14.0	7.9	161.8	14.7	3.3	222.0
Composable Diffusion	74.4	51.5	87.2	75.3	46.5	88.6	53.7	45.7	161.4	22.8	14.0	262.0
CLEVR Diffusion	69.5	73.4	50.9	35.7	22.1	50.4	24.0	9.5	46.9	14.2	4.0	60.4
SC-Diffusion (Ours)	**100.0**	**99.6**	**43.7**	**98.6**	**98.5**	**43.9**	**93.8**	**93.0**	**43.8**	**42.6**	**27.3**	**54.8**

Our method outperforms baselines in all metrics, as seen in Table 1. For image quality, SC-Diffusion has the lowest FID score for all scene complexities. Unsurprisingly, models not trained explicitly on CLEVR (e.g., LayoutGPT + GLIGEN) suffer the most in this metric. Composable Diffusion's FID score monotonically increases with more objects, whereas CLEVR diffusion remains consistent until the 8-object scenes.

In terms of relational accuracy, SC-Diffusion is the only method that successfully generalizes to scenes with more objects. While the RA and OA for all the baselines plummet to below 55% in 4 object scenes, our method maintains above 93%. Composable Diffusion starts off with the second highest accuracy, but as the number of objects increases, the OA steeply declines, and as a result, the RA also drops. CLEVR Diffusion performs decently in 2 and 3 object scenes, but as the scene complexity increases to out-of-training distribution, the OA drops significantly.

Qualitative examples (Appendix E) explain the differences in generalization. For Composable Diffusion, the images become overwhelmingly noisy with more object scenes. This explains the drop in OA, and thus RA, as it struggles to generate even the objects. It shows the importance of grounding the composition in objects and relations instead of only composing energy densities in a general manner (e.g., set operations) [20]. CLEVR Diffusion results are valid with 2,3 object scenes because they are in distribution, but the struggle to generate more than 4 objects in scenes highlights its struggle to generalize.

Ablation We investigate the contribution of Stage I and Stage II in this section. Performance is evaluated using the Relational Accuracy metric on the generated bounding boxes or images. For Stage I, we assess the importance of relational diffusion by comparing our model against object-only diffusion and LayoutGPT. For Stage II, we compare our model against object-only diffusion, LayoutGPT, and GLIGEN [17]. We present the Relational Accuracy of each setting across scenes with varying numbers of object results in Table 2. The experimental results suggest that, in both stages, incorporating relational attributes significantly improves the model's ability to maintain spatial relations, especially as the scene complexity increases.

Table 2. This table displays an ablation study assessing the performance of our model's two stages against alternatives (Object-only diffusion, LayoutGPT, GLIGEN) across different scene complexities (2 Obj, 3 Obj, 4 Obj, and 8 Obj). The results, measured in accuracy percentages, underscore the superior efficacy of our method's components. Substituting either stage with alternative methods leads to significant performance declines.

Stage I	Stage II	2 Obj	3 Obj	4 Obj	8 Obj
Ours	Ours	99.6%	98.5%	93.0%	27.3%
LayoutGPT	Ours	95.2%	42.5%	21.3%	14.5%
GT	Obj Only	**99.9%**	91.3%	42.9%	16.2%
GT	GLIGEN	8.0%	10.1%	7.6%	5.6%
GT	Ours	99.8%	**98.8%**	**97.2%**	**73.3%**

5. Discussion and Conclusion

We propose a framework designed to improve the spatial compositionality of diffusion models for text-to-image generation. Through a scene-graph-based input encoding and a two-stage generation process that uses bounding boxes as an intermediate representation, our approach highlights the potential of injecting modularity into generative models to improve relational compositionality. Therefore, it enables a faithful and controllable generation of complex scenes.

One of the inherent limitations of our approach stems from the usage of multiple denoisers, which leads to a sharp increase in the resources needed for inference as the scene complexity increases. However, recent advancements in the distillation [22, 35] and sampling optimization of diffusion models present viable pathways for mitigating these issues.

Our work serves as a proof of concept, illustrating the feasibility and benefits of embedding compositional reasoning within the image generation process. One direction for future work is to extend our framework to more complete real-world images. Given the abilities of large pretrained generative models such as Stable Diffusion and Diffusion Transformers to generate faithful objects and relations in the real world, a potential method is to finetune these models as object and relational denoisers. By using a similar compositional pipeline, one could improve the generative of real-world complex scenes.

References

[1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation, 2023. 2

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019. 1

[3] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance, 2022. 2

[4] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation and inference with energy based models, 2020. 2

[5] Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc, 2023. 2

[6] Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A. Efros. Blobgan: Spatially disentangled scene representations, 2022. 2

[7] Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation, 2023. 2

[8] Weixi Feng, Wanrong Zhu, Tsu jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models, 2023. 2, 3, 12

[9] Tejas Gokhale, Hamid Palangi, Besmira Nushi, Vibhav Vineet, Eric Horvitz, Ece Kamar, Chitta Baral, and Yezhou Yang. Benchmarking spatial relationships in text-to-image generation, 2023. 1, 2

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022. 2

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017. 3

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1, 3, 7

[13] Ziqi Huang, Kelvin C. K. Chan, Yuming Jiang, and Ziwei Liu. Collaborative diffusion for multi-modal face generation and editing, 2023. 2

[14] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[15] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society, 2017. 3

[16] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs, 2018. 2

[17] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation, 2023. 2, 4, 12

[18] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models, 2023. 2

[19] Nan Liu, Shuang Li, Yilun Du, Joshua B. Tenenbaum, and Antonio Torralba. Learning to compose visual relations, 2021. 2

[20] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models, 2023. 2, 3, 4, 11

[21] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally?, 2023. 2

[22] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models, 2023. 4

[23] OpenAI. Dall·e 3, 2023. 1

[24] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization, 2019. 2

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 7

[26] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 7

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 7

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 1, 7

[30] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. 1

[31] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality, 2022. 1, 2

[32] Binbin Yang, Yi Luo, Ziliang Chen, Guangrun Wang, Xiaodan Liang, and Liang Lin. Law-diffusion: Complex scene generation by diffusion with layouts, 2023. 2, 3, 7

[33] Ling Yang, Zhilin Huang, Yang Song, Shenda Hong, Guohao Li, Wentao Zhang, Bin Cui, Bernard Ghanem, and Ming-Hsuan Yang. Diffusion-based scene graph to image generation with masked contrastive pre-training, 2022. 2

[34] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms, 2024. 2

[35] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation, 2023. 4

[36] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023. 1, 2

[37] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2

[38] Tianjun Zhang, Yi Zhang, Vibhav Vineet, Neel Joshi, and Xin Wang. Controllable text-to-image generation with gpt-4, 2023. 2

[39] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout, 2019. 2

Supplementary Material for Spatially Compositional Diffusion

A. Background on Diffusion Models

In the realm of generative modeling, diffusion probabilistic models have achieved remarkable performance recently. These models iteratively introduce noise into an initial sample X_0 drawn from a predefined data distribution $q(X_0)$. This process is controlled by a predefined variance schedule $\{\beta_t\}_{t=0}^T$, which dictates the incremental transformations as $q(X_t|X_{t-1}) = \mathcal{N}(X_t; \sqrt{1-\beta_t}X_{t-1}, \beta_t I)$. The noise added at each step is Gaussian and always implemented by a well-known reparameterization trick:

$$X_t \approx q(X_t|X_0); \quad X_t = \sqrt{\alpha_t}X_0 + \epsilon\sqrt{1-\alpha_t}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, and $\epsilon \sim \mathcal{N}(0, I)$.

The reverse diffusion process is adopted to generate images. Inverting the diffusion process involves learning the conditional distribution $q(X_{t-1}|X_t)$ through a neural network $p_\theta(X_{t-1}|X_t) \approx q(X_{t-1}|X_t)$. This allows for an approximate sampling from $q(X_0)$ by initiating the process with a noise sample $X_T \sim q(X_T)$, followed by iterative sampling from $q(X_{t-1}|X_t)$. When the sampling step size is adequately small, the conditional distribution $q(X_{t-1}|X_t)$ can be closely approximated by an isotropic Gaussian with a fixed small covariance. Consequently, one only needs to predict the mean of $q(X_{t-1}|X_t)$.

B. Implementation Details

Since our object and relation categories were small, we chose to first perform label encoding on the scene graph, but one could also use any pre-trained encoders like CLIP [25] for an open vocabulary set. This resulted in each object attribute $o_i \in \mathbb{R}^4$ being represented as feature vectors $[\text{material}, \text{size}, \text{shape}, \text{color}]$, and each directed edge $(o_i, r_{ij}, o_j) \in \mathbb{R}^9$ as $[o_i : o_j : \text{relation}]$.

Stage I Both the object and relation denoisers share the same underlying MLP-based architecture. The timestep is encoded through a Sinusoidal Positional Embedding and then fed into an MLP to get a time embedding. Then, the core component is an MLP with four fully connected layers. The condition is concatenated with the input before feeding into the first fully connected layer. In the subsequent fully connected layers, the temporal conditioning is integrated via two additional fully connected layers that modulate the activations with a learnable affine transformation dependent on the time embedding. The final fully-connected layer maps from the hidden dimension back to the output dimension. In the combined model, we initialize the same denoiser with different conditioning dimensions. In the forward pass, we keep the noisy bounding boxes in the layout in the batch dimension and apply the object and relation denoiser in parallel. Since the data distribution we are modeling is very low dimensional, the MLP-based architecture is sufficient.

Stage II In the image generation pipeline, all denoisers share the underlying architecture of a U-Net [28] modified for conditional input, similar to the denoisers seen in prior text-to-image diffusion models [12, 27, 29].

Since we wanted to condition not just on the object and relation encoding but also the corresponding bounding box coordinates for the objects, we concatenated them together as conditions: for object denoiser, the condition becomes $[o_i : \mathbf{b}^{(i)}] \in \mathbb{R}^8$; for relation denoiser, the condition is $[o_i : \mathbf{b}^{(i)} : o_j : \mathbf{b}^{(j)} : r_{ij}] \in \mathbb{R}^{17}$.

We continue using Sinusoidal Positional Encoding followed by an MLP to get time embedding from timesteps. To inject the condition into the U-Net, we simply feed the condition into another MLP and combine it with the temporal conditioning through the sum.

In the combined model, we simply initiate three U-Nets for three denoisers with varying conditional dimensions. In the forward pass, we again apply the object denoiser and relational denoiser in parallel, applying the corresponding mask to their outputs, and sum the noises together along with the background denoiser outputs.

C. Evaluation Metrics

FID score uses a pre-trained inception model to extract features for ground truth and generated datasets and measures their distributional similarity. To evaluate the generation quality on an object level, we introduce an object accuracy metric (OA). This involves pretraining an object classifier on the training dataset to identify each object's attributes, applying this classifier to the generated images, and calculating the mean accuracy of attribute alignment with specified requirements. To evaluate the plausibility of the object relations within the scene, we adapt the Scene Relation Score (SRS) metric proposed by [32]. We train a relational predictor on the ground truth dataset, which, given the features of two objects, predicts the relation between them. We apply this classifier to all relations in the ground truth scene graph to report the mean relational accuracy. This metric evaluates the given generative model's ability to recover the specified object relations.

D. Prompt

We include the prompt used below in text form.

> "a large green rubber sphere to the left of a small brown metal cube, AND a large green rubber sphere in front of a small brown metal cube, AND a large green rubber sphere to the left of a small red metal cylinder, AND a large green rubber sphere behind a small red metal cylinder, AND a large green rubber sphere to the left of a small green rubber cylinder, AND a large green rubber sphere in front of a small green rubber cylinder, AND a small brown metal cube to the right of a large green rubber sphere, AND a small brown metal cube behind a large green rubber sphere, AND a small brown metal cube to the right of a small red metal cylinder, AND a small brown metal cube behind a small red metal cylinder, AND a small brown metal cube to the right of a small green rubber cylinder, AND a small brown metal cube in front of a small green rubber cylinder, AND a small red metal cylinder to the right of a large green rubber sphere, AND a small red metal cylinder in front of a large green rubber sphere, AND a small red metal cylinder to the left of a small brown metal cube, AND a small red metal cylinder in front of a small brown metal cube, AND a small red metal cylinder to the left of a small green rubber cylinder, AND a small red metal cylinder in front of a small green rubber cylinder, AND a small green rubber cylinder to the right of a large green rubber sphere, AND a small green rubber cylinder behind a large green rubber sphere, AND a small green rubber cylinder to the left of a small brown metal cube, AND a small green rubber cylinder behind a small brown metal cube, AND a small green rubber cylinder to the right of a small red metal cylinder, AND a small green rubber cylinder behind a small red metal cylinder"

E. Generation Samples

In Figure 4, 5, 6 and 7, we present additional qualitative examples of Relational CLEVR.

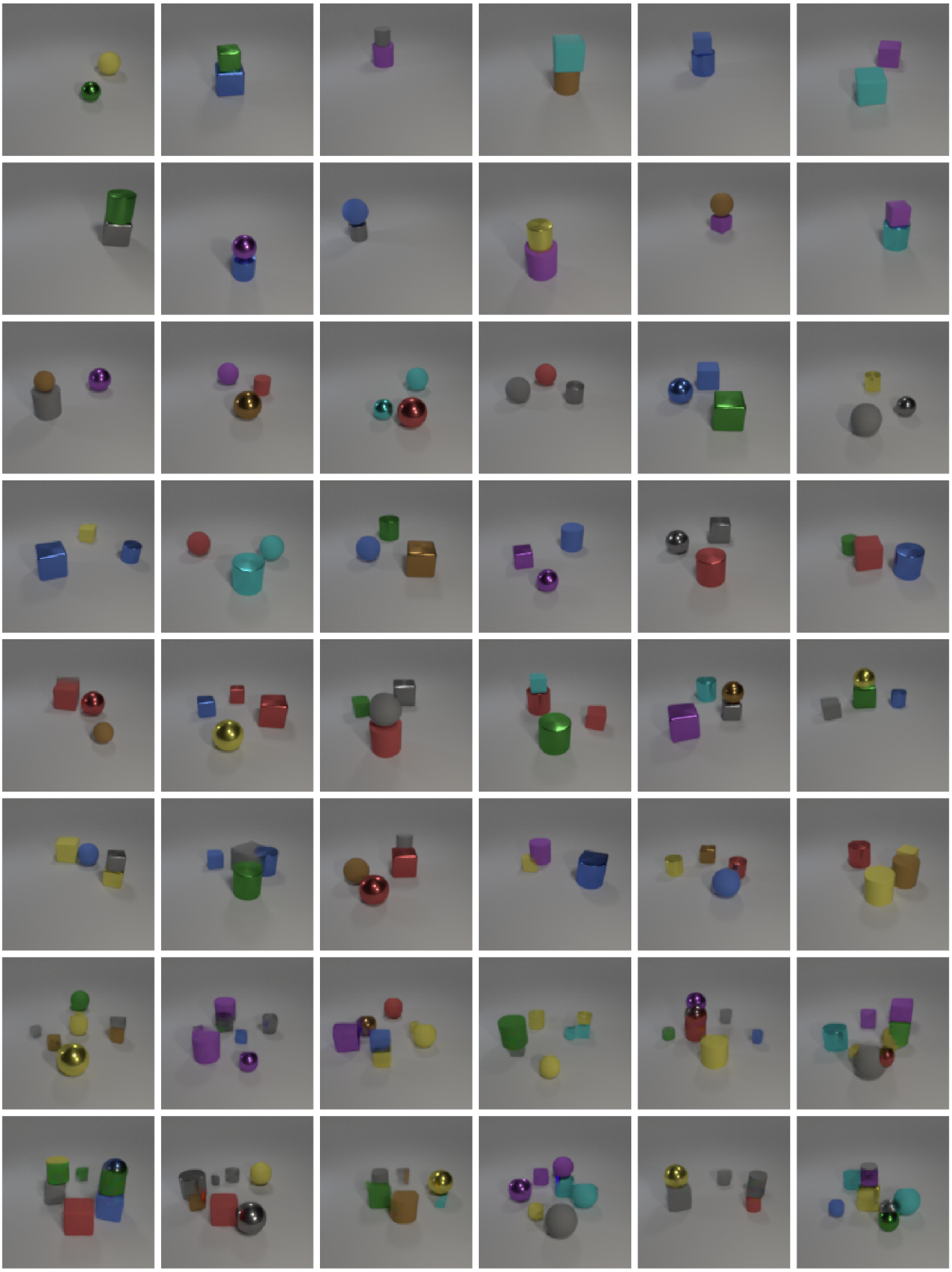

Figure 4. **Additional qualitative examples from SC-Diffusion.** Examples are sampled using the DDPM sampler. The first two rows are samples of two-object scenes; the next two are three-object scenes, then four-object and eight-object, respectively.

Figure 5. **Additional qualitative examples from CLEVR-Diffusion.** Examples are sampled using the DDPM sampler. The first two rows are samples of two-object scenes; the next two are three-object scenes, then four-object and eight-object, respectively.

Figure 6. **Additional qualitative examples from Composable Diffusion[20].** Examples are sampled using the DDPM sampler. The first two rows are samples of two-object scenes; the next two are three-object scenes; then four-object and eight-object, respectively.

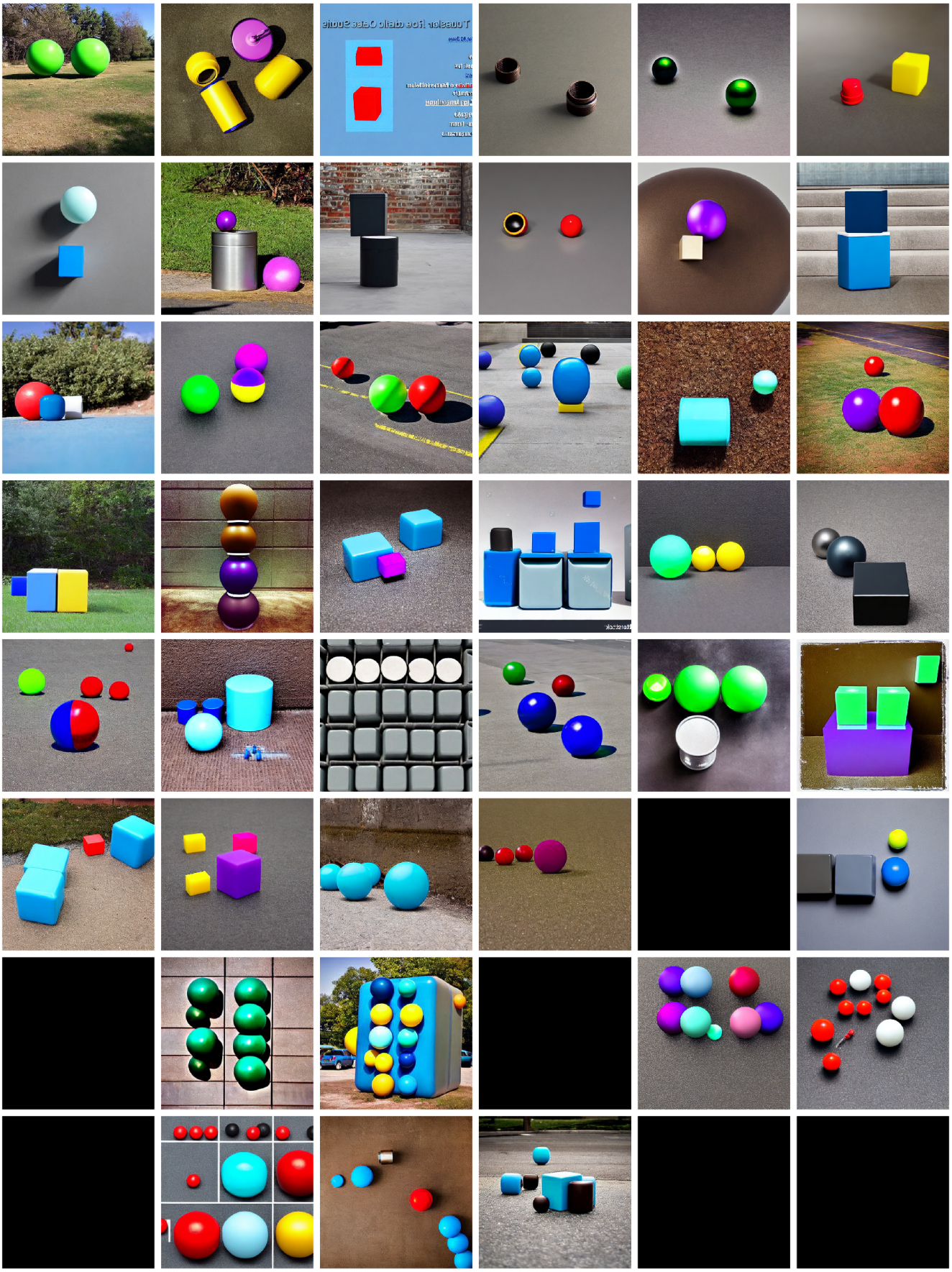

Figure 7. **Additional qualitative examples from LayoutGPT + GLIGEN [8, 17].** Examples are sampled using the DDPM sampler. The first two rows are samples of two-object scenes; the next two are three-object scenes; then four-object and eight-object, respectively. The black images indicate LayoutGPT's layout generation failure.