Supplementary Material of MMGDreamer

In this Supplementary Material, we report the following:

- Section : Additional Results.
- Section : Additional Qualitative.
- Section : Additional Experimental Details.
- Section : Limitations and Future Work.

Additional Results

Scene Graph Manipulation

We follow EchoScene (Zhai et al. 2024b) in manipulating scene graphs by either adding a node with relevant edges or altering the relationships between existing nodes, the results presented in Tab. 1. Our method, MMGDreamer (MM+R), consistently outperforms other approaches in most categories, particularly excelling in the ”left/right,” ”smaller/larger,” and ”taller/shorter” relationships, where it achieves the highest scores. For example, in the ’left/right’ category for relationship change mode, MMGDreamer (0.95) outperforms both Graph-to-3D (0.91) and CommonScenes (0.91), demonstrating its ability to maintain consistency between the generated scene’s spatial relationships and the input graph structure. Notably, in the Node addition mode, MMGDreamer achieves a perfect score 1.00 in the ”front/behind” category, indicating its superior capability in preserving spatial relationships during scene graph manipulations. Across all manipulation modes, MMGDreamer (MM+R) demonstrates a clear superiority in the symmetrical metric compared to other methods. This consistent performance underscores the advantage of incorporating visual information into the mixed-modality graph, which enables more precise geometry control and leads to the generation of scenes with objects that exhibit enhanced symmetry.

Inter-object Consistency

To evaluate inter-object consistency, we measure the Chamfer Distance (CD) values for object shapes that share the *same as* relationship within a scene. Low CD values indicate higher consistency in object shapes. As shown in Tab. 2, MMGDreamer (I+R) consistently achieves lower CD values across various objects and room types, demonstrating stronger geometry control and significantly higher inter-object consistency compared to CommonScenes and EchoScene. For example, in the Bedroom, MMGDreamer reduces the CD for the Nightstand to 1.33, which is 1.36 lower than CommonScenes and 0.35 lower than EchoScene. Even for the Lamp, where EchoScene performs poorly with a CD of 30.07, MMGDreamer (I+R) shows much better consistency with a CD of 2.29, representing an improvement of 27.78, demonstrating MMGDreamer’s ability to maintain object shape consistency even in more challenging object types. In the Living room, MMGDreamer achieves a CD of 0.16 for the Chair, outperforming EchoScene by 0.83. For the Table, MMGDreamer (I+R) yields a CD of 2.44, which is 8.31 lower than CommonScenes and 0.58 lower than EchoScene. In the Dining room, MMGDreamer maintains a low CD of 0.83 for the Table, a notable improvement of 8.21 over CommonScenes and 0.43 over EchoScene. Overall, MMGDreamer (I+R) demonstrates superior control over object geometry, as evidenced by consistently lower CD values across Bedroom, Living room, and Dining room scenes.

Additional Qualitative

Qualitative Results On Scene Generation

We present a qualitative comparison of scene generation results across different methods, as shown in Fig. 2. MMGDreamer consistently excels in maintaining object geometry and spatial relationships, resulting in more detailed and realistic scenes compared to other methods. For example, in the bedroom scene, MMGDreamer successfully maintains the precise geometric alignment between the Nightstand and Bed, as indicated by the green rectangles. In contrast, Graph-to-3D and CommonScenes exhibit issues with the geometry of the Nightstand and Bed, leading to unrealistic shapes. In particular, EchoScene generates a visibly distorted Sofa with incorrect placement, leading to significant inconsistencies in both shape and spatial location. In the dining room scene, MMGDreamer accurately captures the complex geometry of the Lamp and maintains the correct spatial relationship between the Chair and Table. Other methods, like CommonScenes and EchoScene, struggle to reproduce the Lamp’s intricate details, leading to visible distortions, and fail to maintain the correct positioning of the Chair and Table. This highlights MMGDreamer’s clear advantage in handling both complex shapes and spatial relationships. In the complex living room scene, MMGDreamer effectively demonstrates superior geometry controllability by accurately generating the sofa, chair, and lamp, maintaining both precise object shapes and a consistent spatial arrangement that closely aligns with the input graph. In contrast, other methods exhibit significant geometry issues, particularly with the chair and lamp.

Qualitative Results On Object Generation

We provide a qualitative analysis of the objects generated by MMGDreamer (I+R) in Fig. 3. The results demonstrate a high degree of consistency between the generated object shapes and the input images, showcasing the strong geometry controllability of our method. For example, in generating complex objects like the Chair and Lamp, MMGDreamer successfully produces highly consistent geometries. The Chair, with its intricate structure and unique shape, is accurately captured in the generated object, maintaining consistency with the input image in both shape and proportions. Similarly, the Lamp’s complex geometry and fine details are faithfully reproduced, showcasing our model’s high precision in capturing and generating intricate shapes.

Table 1: **Scene graph consistency** (higher is better). **MM** represents nodes using mixed-modality representations. **R** denotes the relationships of nodes. Top to bottom: Relationship change mode, Node addition mode, and Generation only.

Method	Shape Representation	Mode	left/ right	front/ behind	smaller/ larger	taller/ shorter	close by	symmetrical
Graph-to-3D (Dhamo et al. 2021)	DeepSDF (Park et al. 2019)	Change	0.91	0.92	0.86	0.89	0.69	0.46
CommonScenes (Zhai et al. 2024c)	rel2shape		0.91	0.92	0.86	0.91	0.69	0.59
EchoScene (Zhai et al. 2024b)	echo2shape		0.94	0.96	0.92	0.93	0.74	0.50
MMGDreamer (MM+R)	echo2shape		0.95	0.96	0.93	0.93	0.71	0.53
Graph-to-3D (Dhamo et al. 2021)	DeepSDF (Park et al. 2019)	Addition	0.94	0.95	0.91	0.93	0.63	0.47
CommonScenes (Zhai et al. 2024c)	rel2shape		0.95	0.95	0.91	0.95	0.70	0.61
EchoScene (Zhai et al. 2024b)	echo2shape		0.98	0.99	0.96	0.96	0.76	0.49
MMGDreamer (MM+R)	echo2shape		0.98	1.00	0.97	0.97	0.80	0.61
Graph-to-3D (Dhamo et al. 2021)	DeepSDF (Park et al. 2019)	None	0.98	0.99	0.97	0.95	0.74	0.57
CommonScenes (Zhai et al. 2024c)	rel2shape		0.98	1.00	0.97	0.95	0.77	0.60
EchoScene (Zhai et al. 2024b)	echo2shape		0.98	0.99	0.96	0.96	0.74	0.55
MMGDreamer (MM+R)	echo2shape		0.98	0.99	0.97	0.96	0.76	0.62

Table 2: **Inter-object consistency.** The object shapes corresponding to the *same as* relationship within a scene demonstrate a high degree of consistency, as reflected by the low CD values (scaled by ×0.001). **I** represents nodes using image representations. **R** denotes the relationships of nodes.

Method	Bedroom			Living room			Dining room		
	Wardrobe	Nightstand	Lamp	Chair	Table	Lamp	Chair	Table	Sofa
CommonScenes (Zhai et al. 2024c)	0.61	2.69	-	6.64	11.75	-	1.96	9.04	-
EchoScene (Zhai et al. 2024b)	0.14	1.68	30.07	0.99	3.02	10.06	1.75	1.26	3.47
MMGDreamer (I+R)	0.11	1.33	2.29	0.16	2.44	0.18	0.23	0.83	0.18

Compared to methods specifically designed for object generation, such as One-2-3-45++ (Liu et al. 2024), which require large amounts of training data, MMGDreamer (I+R) achieves impressive object geometry generation results with only a small amount of training data. This demonstrates the robustness and geometric controllability of our approach, even under data-limited conditions, while still generating high-quality object shapes.

Qualitative results on relation-free scene generation

We demonstrate the generated results of MMGDreamer (I) and MMGDreamer (T) when provided with mixed-modality graphs that lack any explicit object relationships, as shown in Fig. 4. Despite the absence of predefined relationships, our method successfully generates coherent and realistic layouts. This highlights the effectiveness of the Relation Predictor within MMGDreamer, which can infer the spatial relationships between objects, leading to well-organized scene layouts. For example, in the Bedroom scene generated by MMGDreamer (I), the bed, nightstands, and lamp are not only arranged logically but also exhibit a high degree of fidelity. The objects' geometries in the generated scene closely match the corresponding input images, showcasing MMGDreamer (I)'s ability to maintain geometric consistency and high detail throughout the scene generation process. Similarly, MMGDreamer (T) successfully arranges the objects in the Living Room scene, where the sofa, tables, and chairs are organized into a cohesive layout that reflects real-world spatial arrangements, again without any predefined relationships. These results demonstrate the robustness of MMGDreamer's Relation Predictor, which predicts object relationships and generates reasonable layouts under relation-free conditions, ensuring consistent and visually harmonious scene generation.

Additional Experimental Details

Baselines

Graph-to-3D. Graph-to-3D (Dhamo et al. 2021)is an approach that directly generates 3D shapes from a scene graph in an end-to-end manner. Unlike previous methods that rely on retrieving object meshes from synthetic data, Graph-to-3D leverages GCN within a variational autoencoder framework to generate both object shapes and scene layouts. This model allows for flexible scene synthesis and modification, using the scene graph as an interface for semantic control, providing a more robust and direct method for 3D scene generation. We utilize the DeepSDF (Park et al. 2019) variant of Graph-to-3D for SDF-based shape generation, training twelve category-specific models (excluding ”floor”) for 1500 epochs using SG-FRONT. The latent codes for each object are optimized and stored, then used to train Graph-to-3D. During inference, the model directly generates 3D shapes and full scenes using the predicted latent codes.

CommonScenes. CommonScenes (Zhai et al. 2024c) is a fully generative model that effectively converts scene graphs into controllable 3D scenes that are semantically realistic and conform to commonsense. Unlike previous methods that rely on database retrieval or pre-trained embeddings, CommonScenes uses a dual-branch pipeline to predict scene layouts and generate object shapes while capturing global and local relationships in the scene graph. We follow the training procedure outlined in (Zhai et al. 2024c) and train the network end-to-end on the SG-FRONT dataset using the AdamW optimizer with an initial learning rate of 1×10^{-4} for 1000 epoch.

EchoScenes. EchoScene (Zhai et al. 2024b) is a generative model designed to create 3D indoor scenes from scene graphs by utilizing a dual-branch diffusion model. It handles the complexities of scene graphs, such as varying node counts and diverse edge combinations, by introducing an information echo scheme. This allows for collaborative information exchange between nodes, ensuring that the generated scenes are both globally coherent and controllable. Adhering to the training protocol from (Zhai et al. 2024b), we trained EchoScene on the SG-FRONT dataset for 2050 epochs.

Text-to-shape Series. This series includes two generative baselines. One is built upon CommonScenes, called CommonLayout+SDFusion, and the other builds upon Echoscene, referred to as EchoLayout+SDFusion. Both methods first generate bounding boxes and then use the text-to-shape method SDFusion (Cheng et al. 2023) to further generate shapes within each bounding box, based on the textual information from the graph nodes.

Implementation Details

Hardware and Software. We demonstrate the hardware and software specifications of our experimental setup, including CPU, GPU, and system configuration, as shown in Tab. 3. In addition, we utilize Blender 4.1 with the CYCLES engine to render high-quality images for our qualitative comparison experiments. In our Blender setup, we configure the Noise Threshold to 0.001, set the maximum samples to 300, and use the RGBA color mode with a color depth of 16. We also ensure that these parameters remain consistent across all qualitative comparison experiments to maintain uniformity in the rendering process.

Dataset Details. Our experiments are conducted on SG-FRONT, a dataset that enhances the 3D-FRONT synthetic dataset by incorporating comprehensive scene graph annotations. These annotations are organized into three key categories: spatial/proximity, support, and style relationships. Spatial relationships dictate object positions (e.g., left/right), size comparisons (e.g., bigger/smaller), and height comparisons (e.g., taller/shorter). Support relationships capture structural dependencies such as proximity and relative placement (e.g., close by, above, standing on). Style relationships reflect similarities in material, shape, and category. SG-FRONT contains around 45k samples, covering three types of indoor scenes: bedrooms, dining rooms, and living rooms, with annotations for 15 different relationship types. We follow the training and testing procedures outlined in EchoScene (Zhai et al. 2024b) to assess all methods on SG-FRONT. The dataset consists of 4041 bedrooms, 900 dining rooms, and 813 living rooms. For training, we use 3879 bedrooms, 614 dining rooms, and 544 living rooms, while the remaining scenes are reserved for testing.

Table 3: **Hardware and software** specifications for experimental setup.

System & Hardware Overview	
CPU	Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
GPU	8 ×NVIDIA A100 Tensor Core GPU
Memory	10T DRAM
Operating System	Ubuntu 22.04.4 LTS
CUDA Version	11.3
NVIDIA Driver	530.30.02
ML Framework	Python 3.8.18 Pytorch 1.11.0
GPU Specifications	
CUDA Cores	6912
Memory Capacity	80GB
Memory Bandwidth	1935GB/s

ChatGPT Prompt. The prompt for Mixed-Modality Graph Generation using GPT-4V is shown in Fig. 5. The design of this prompt focuses on enabling GPT-4V to effectively interpret and generate structured scene graphs from both text descriptions and image inputs. By leveraging GPT-4V's multimodal capability, the prompt enables seamless integration of diverse inputs, ensuring that all relationships between objects are captured accurately and consistently within the generated scene graph.

Batch Size Definition. During the training of the dual-branch diffusion model, we follow the approach used in EchoScene (Zhai et al. 2024b), where each branch operates with its batch size to accommodate distinct training objectives. For the layout branch, we define a scene batch B_l, which includes all bounding boxes in the scenes during each training step. For the shape branch, we define a maximum batch size B_s^* and select scenes where the total number of objects B_s closely approaches but does not exceed this limit. This allows efficient use of batch capacity, though the batch size B_s fluctuates slightly due to varying object counts. Both B_l and B_s^* are set to 128 during training. Additionally, when training the Visual Enhancement Module and the Relation Predictor module separately, the batch size is also set to 128.

Training Procedure. The training process is divided into two distinct stages. In the first stage, we train the Visual Enhancement Module and the Relation Predictor separately. In the second stage, we train the dual-branch diffusion model.

The Visual Enhancement Module is trained on over 4,000 3D objects extracted from the SG-FRONT training set, where each object's image information is sourced from the 3D-FRONT dataset. For each object, we use CLIP ViT-B/32 to extract textual and visual features, forming corresponding textual-visual pairs. The textual features are then quantized by selecting the 4 closest entries from a codebook $\mathcal{C} \in \mathbb{R}^{64\times512}$, where the codebook consists of 64 entries,

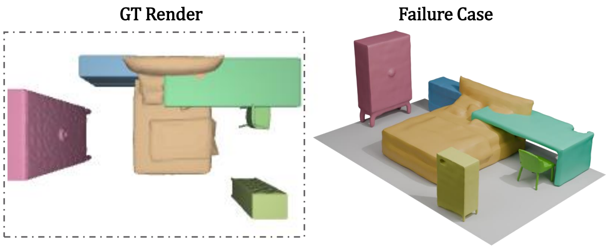

Figure 1: **Failure case.** The dashed box on the left is a top-down view rendered using the ground truth, while the result on the right is generated scene by MMGDreamer.

each with a dimension of 512. The Visual Enhancement Module is trained for 1,000 epochs with a batch size of 128, which runs for 500 steps per epoch, employing the loss function $\mathcal{L}_r$. The AdamW optimizer is used with a learning rate of 1×10^{-4} and a weight decay of 0.02. Additionally, an exponential moving average (EMA) with a decay factor of 0.9999 is applied to stabilize the training process. This training strategy ensures consistent and robust learning of the module across the dataset.

The Relation Predictor uses training data generated by first masking 50% of the text, image, and relationship information in the Full-Modality Graph, and then encoding the masked data into triplet representations. The Relation Predictor model consists of a 10-layer GCN with a hidden dimension of 256, followed by two fully connected MLP layers with dimensions of 256 and 128, respectively. The model is trained using the loss function $\mathcal{L}_e$, focusing on predicting the masked relationships. Training proceeds for 1,000 epochs with a batch size of 128, using CrossEntropyLoss. AdamW is employed as the optimizer, with an initial learning rate of 5×10^{-3}.

The dual-branch diffusion model's training data is generated by applying a random masking ratio to the text and image of the Full-Modality Graph, which is subsequently encoded into LMMG. The model is trained using the loss function $\mathcal{L}_o$ for 2050 epochs, with a batch size of $B_s^* = 128$ for the shape branch and $B_l = 128$ for the layout branch. We utilize the AdamW optimizer, setting the initial learning rate to 1×10^{-4}.

Limitations and Future Work

Our method demonstrates strong potential in generating complex 3D indoor scenes, yet it occasionally encounters failure cases, as shown in Fig. 1. These errors primarily stem from the limitations of the 3D-FRONT dataset, where noisy data often leads to interpenetrating objects in the generated scenes. While we implement post-processing techniques to minimize this noise, a small amount of erroneous data, such as overlapping furniture instances, remains in the dataset. This issue is reflected during inference, with some generated scenes showing minor collisions between objects. Nevertheless, these errors are infrequent, and our method consistently outperforms others in maintaining coherence between shape and layout despite the dataset's limitations.

While our method successfully integrates visual information, we have intentionally focused on generating objects with accurate geometric shapes and coherent scene layouts, deliberately excluding texture and material details for simplicity and control. Incorporating textures and material properties would add a new layer of complexity to the method, as modeling complex 3D shapes with detailed textures is a challenging task. Nevertheless, we recognize that including texture and material information presents an exciting opportunity for future work. By enhancing the method to better leverage visual data, we plan to generate scenes with richer texture details and achieve a higher degree of control over both geometry and texture, which will significantly improve the realism and versatility of our generated scenes.

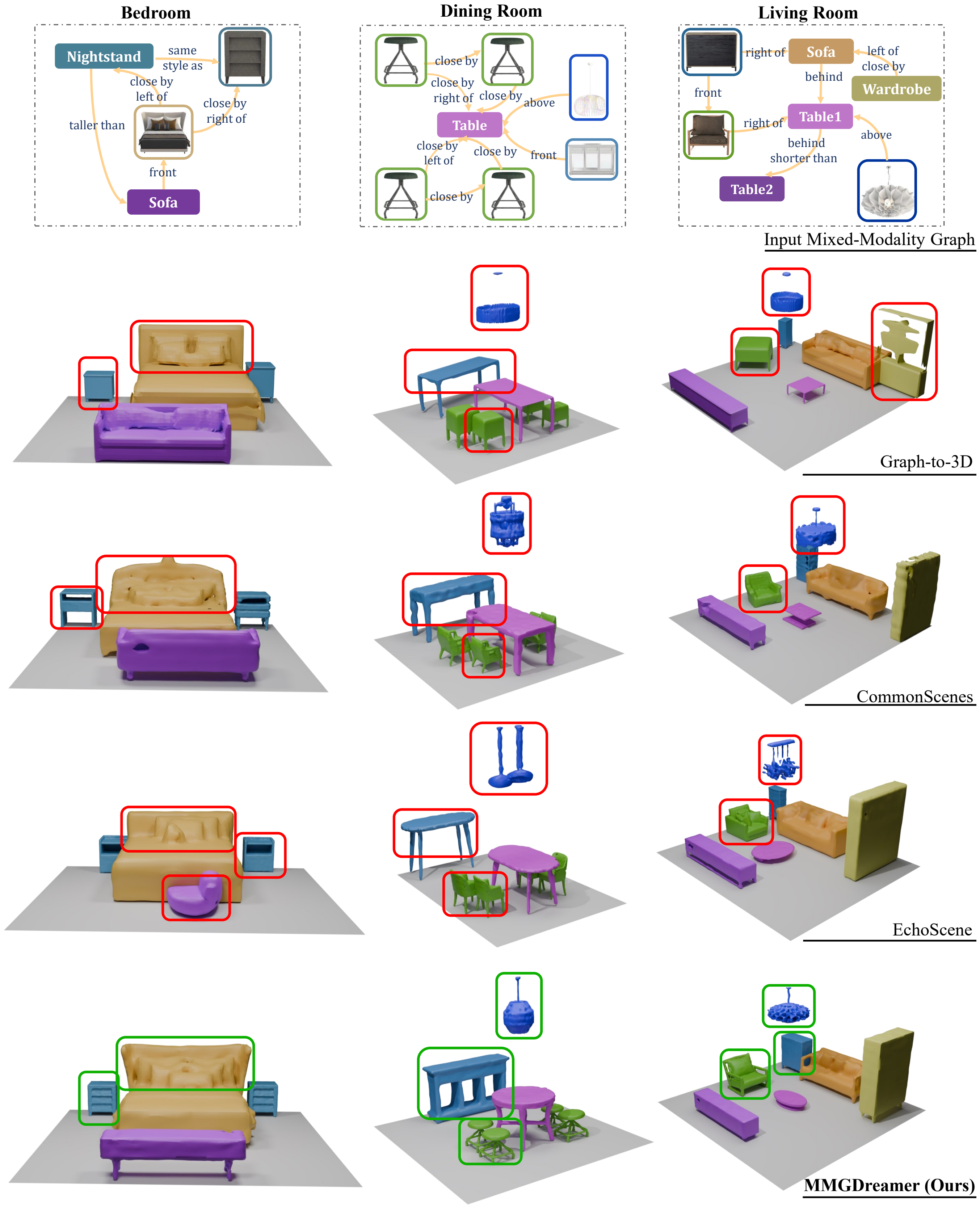

Figure 2: **More qualitative comparison on scene generation.** The first row shows the input mixed-modality graph, which visualizes only the most critical edges in the scene. Red rectangles denote areas of inconsistency in the generated scenes, while green rectangles signify regions of consistent generation.

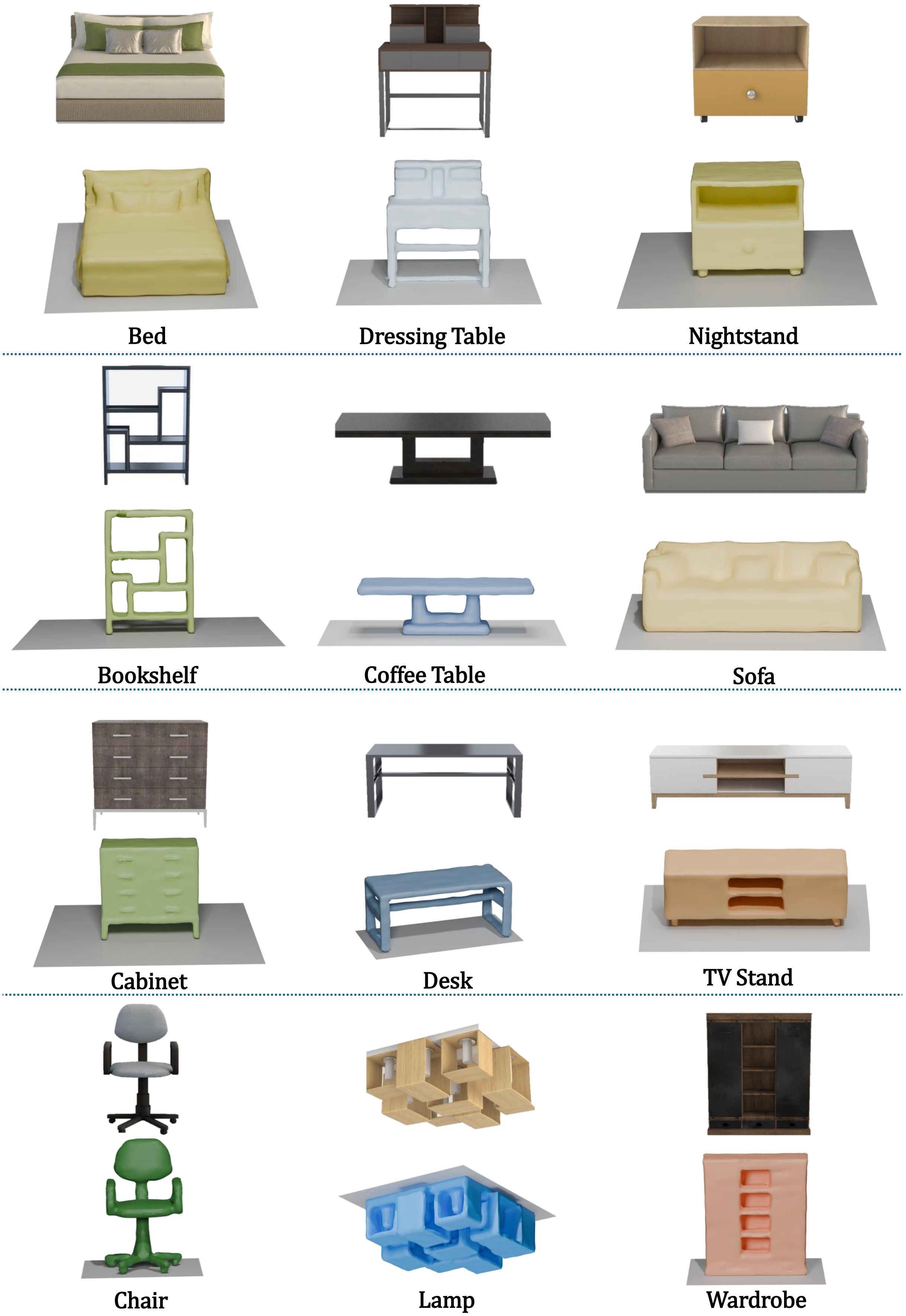

Figure 3: **Qualitative results on object generation.** The figure is divided into three sections by dashed lines. In each section, the top row shows the input images of various furniture items, the middle row displays the corresponding generated objects in the scenes, and the bottom row provides the object categories.

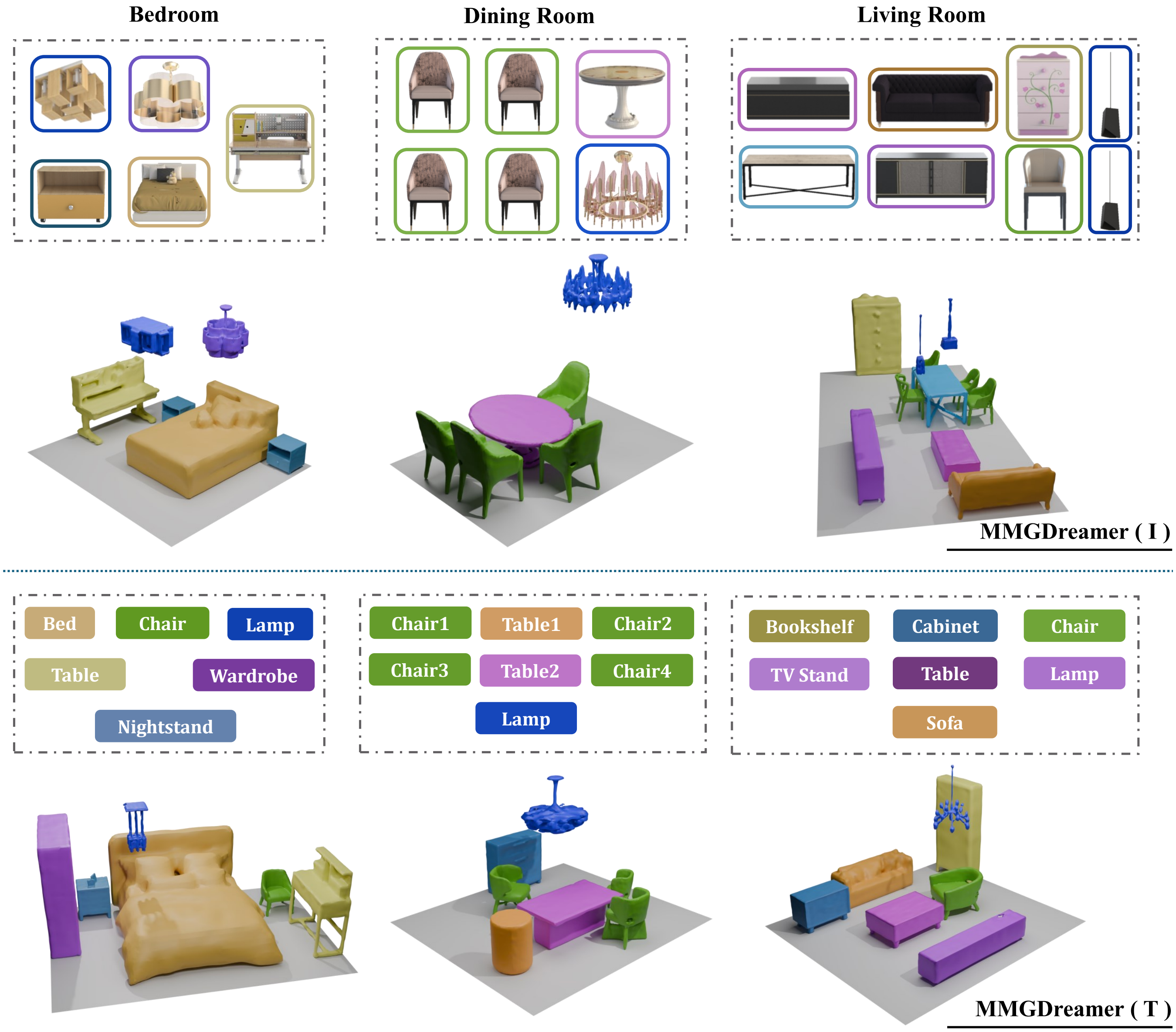

Figure 4: **Qualitative Results On Relation-Free Scene Generation.** The figure is divided into two sections by dashed lines. In each section, the dashed boxes represent the input mixed-modality graphs, where nodes are depicted either as text or images, without any explicit relationships. Below each input graph, the corresponding generated indoor scenes are displayed.

Mixed-Modality Graph Generation Prompt: Assume you are an interior designer, and I will provide you with a multimodal scene design request that may include textual descriptions or images of furniture.
Please create a graph based on my input and list all nodes along with the relationships between them (in the format A -> <relationship> -> B). Here are the constraints:
1. For furniture described in text, the node name should be the corresponding English word.
2. For furniture presented in images, you will first need to identify the type of furniture depicted in each image (Only need to identify its type without describing its attributes).
3. Additionally, number the images sequentially as Image1, Image2, etc., according to the order they were provided, and use the format "number (English word)" as the node name. When providing design requirements, focus solely on outlining the nodes and their relationships, without including any introductory or concluding remarks.

Please note that only these twelve relationships are allowed: left of, right of, front, behind, close by, above, standing on, bigger than, smaller than, taller than, shorter than, symmetrical to, same style as, same super category as, same material as. When the input relationship description is not one of these twelve expressions, you need to replace it with a synonym from this list.

ONE-SHOT EXAMPLE

Here is an example of the output. Please make sure to output in this format:

Nodes:
- Image1 (Nightstand)
- Image2 (Bed)
- Wardrobe
- Pendant Lamp
- Nightstand

Relationships:
- Image1 (Nightstand) -> close by -> Image2 (Bed)
- Image1 (Nightstand) -> right of -> Pendant Lamp
- Wardrobe -> close by -> Image1 (Nightstand)
- Image1 (Nightstand) -> smaller than -> Image2 (Bed)
- Pendant Lamp -> behind -> Nightstand
- Image2 (Bed) -> front-> Wardrobe
- Pendant Lamp -> smaller than -> Image1 (Nightstand)

Here is my design requirement:

Figure 5: **Prompt template** for Mixed-Modality Graph Generation with GPT-4V.