Supplementary Materials
FreeScene: Mixed Graph Diffusion for 3D Scene Synthesis from Free Prompts

Tongyuan Bai[1] Wangyuanfan Bai[1] Dong Chen[1] Tieru Wu[1,3] Manyi Li[2] Rui Ma[1,3*]
[1]School of Artificial Intelligence, Jilin University
[2]School of Software, Shandong University
[3]Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, MOE, China

A. Implementation Details

A.1. Data Processing

Edge Matrix Construction. In this paper, we define ten types of spatial relationships and a 'None' type, similar to InstructScene, including: 'Left of', 'Right of', 'In front of', 'Behind', 'Closely left of', 'Closely right of', 'Closely in front of', 'Closely behind', 'Above', 'Below' and 'None'. Additionally, based on the dataset, we defined different category list tailored to different room types as shown in Figure 1. Upon receiving the triplet list output from the Graph Designer, we construct a specialized symmetric matrix where each pair of symmetric positions stores opposite spatial relationships, such as 'Left of' and 'Right of'.

Bedroom: [0:'armchair', 1:'bookshelf', 2:'cabinet', 3:'ceiling_lamp', 4:'chair', 5:'children_cabinet', 6:'coffee_table', 7:'desk', 8:'double_bed', 9:'dressing_chair', 10:'dressing_table', 11:'kids_bed', 12:'nightstand', 13:'pendant_lamp', 14:'shelf', 15:'single_bed', 16:'sofa', 17:'stool', 18:'table', 19:'tv_stand', 20:'wardrobe']

Living&Dining Room: [0:'armchair', 1:'bookshelf', 2:'cabinet', 3:'ceiling_lamp', 4:'chaise_longue_sofa', 5:'chinese_chair', 6:'coffee_table', 7:'console_table', 8:'corner_side_table', 9:'desk', 10:'dining_chair', 11:'dining_table', 12:'l_shaped_sofa', 13:'lazy_sofa', 14:'lounge_chair', 15:'loveseat_sofa', 16:'multi_seat_sofa', 17:'pendant_lamp', 18:'round_end_table', 19:'shelf', 20:'stool', 21: 'tv_stand', 22:'wardrobe', 23:'wine_cabinet']

Figure 1. Category lists for different room types.

Text Prompt Generation. We followed the methodology of InstructScene to generate text prompts for each scene, which include spatial relations extracted from the 3DFront dataset and object captions obtained using BLIP. Finally, 1∼2 of these relationships are randomly selected and refined using ChatGPT.

A.2. Model Details

Training Strategies. We employ a network containing five Transformer blocks with 8-head 512-dimensional attention and a dropout rate of 0.1. In order to improve the robustness of the training process, we employ a probabilistic approach to selectively fix certain variables. Specifically, we assign a 0.2 probability to nullify the text input (for classifier-free guidance), a 0.2 probability to preserve the input graph (steering the model to predict noise that aligns with the graph), and a 0.1 probability to generate only the edges while keeping the other features fixed. The remaining 0.5 probability is dedicated to denoising all the variables. The models are trained using the AdamW optimizer for 500,000 iterations with a batch size of 128, a learning rate of 1e-4, and a weight decay of 0.02. All experiments are conducted on a single NVIDIA A40 48GB GPU.

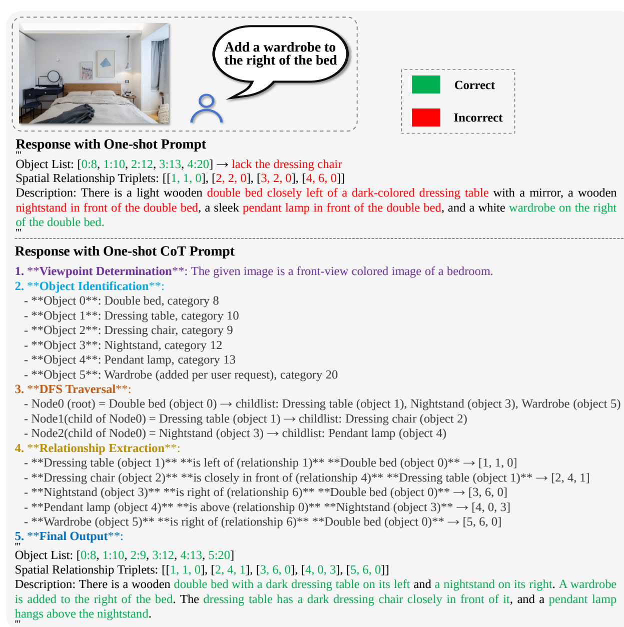

Figure 2. A comparison of the results between one-shot CoT prompts and one-shot prompts on a specific example.

A.3. Prompt Templates

In Figure 3, we provide the one-shot Chain-of-Thought (CoT) prompt utilized by the Graph Designer, illustrated

*Corresponding author

Rank\Method	FreeScene	MG-DiT	InstructScene
Top1$_{\%}$	57.10	24.52	18.39
Top2$_{\%}$	87.74	63.23	49.03

Table 1. The user study involving 31 participants to evaluate 10 sets of scenes generated by three methods and focused on two key aspects: text-scene consistency and scene plausibility. For each result, we provided a top-down view and a randomly selected perspective. Participants were asked to rank the outputs from the three methods based on the given criteria, from best to worst. We subsequently analyzed and calculated the probability of each method being ranked as Top1 and Top2 across all evaluations.

with the example of a bedroom. For room types such as dining room and living room, the prompt template can be effortlessly adapted by substituting the category list with the corresponding range.

B. Additional Results

B.1. Evaluation on One-Shot CoT Prompts

In Figure 2, we also present a comparison of the graph extraction results on an image-text pair using one-shot CoT prompts and one-shot prompts. It is clearly demonstrated that one-shot CoT prompts significantly enhance the performance of the Graph Designer through stepwise guidance, particularly in the accuracy of spatial relationships, making the graph extraction process more reliable.

B.2. More Qualitative Results

In this section, we present additional qualitative results for text-to-scene, graph-to-scene, completion, and stylization tasks, which are showcased in Figures 4, 5, 6, and 7, respectively. All qualitative results are rendered using Blender's Python API.

B.3. User Study

To further evaluate these methods, we conducted a user study comparing InstructScene, MG-DiT, and full version of FreeScene. As shown in Table 1, FreeScene demonstrates clear advantages over other methods

You are an experienced 3D room designer. Now you are given a reference picture or a piece of user text or both, and you need to extract the main objects and their spatial relationships in the picture and user text. Note that all available object class indices and names should in the following list: [0:'armchair', 1:'bookshelf', 2:'cabinet', 3:'ceiling_lamp', 4:'chair', 5:'children_cabinet',6:'coffee_table', 7:'desk', 8:'double_bed', 9:'dressing_chair', 10:'dressing_table', 11:'kids_bed', 12:'nightstand',13:'pendant_lamp', 14:'shelf', 15:'single_bed', 16:'sofa', 17:'stool', 18:'table', 19:'tv_stand', 20:'wardrobe'], and all the spatial relationship class indices and names should in the following list: [0:'above', 1:'left of', 2:'in front of',3:'closely left of', 4:'closely in front of', 5:'below', 6:'right of', 7:'behind', 8:'closely right of', 9:'closely behind'].

The user text is: { user text }, the user image is: { base64_image }

In the subsequent explanations, every [object_name] refers to the category name of a specified object within the given range, and every [class_index] represents the class index corresponding to the object's category name. Every [color and texture] refers to the color and texture of the object. Based on the object's color and texture in the colored image, you can incorporate these adjectives into the final description, and please adjust them according to the user's prompt. If the input image is a sketch or a diagram without any color, please apply the colors and textures specified by the user's text. If no colors and textures are provided by the user text, do not include descriptive adjectives in the description. Your answer should encompass three parts:

1. The first part is the main objects list consists of all the object indices and class indices.
e.g. [0:8, 1:12, 2:7], that means the object zero is a object with class index 8, and so on.
2. The second part is the spatial relationship triplets list e.g. [[0, 6, 1], [2, 6, 0]]. Note that for every triplet such as [0, 6, 1], each of its elements sequentially represents the object index, predicate(relationship index), and the subject index. This means the object zero is right of the object one (relation 6). Be sure not to interchange the object and subject order; otherwise, the inverse relationship should be used.
3. The third part is the simple description of the room. e.g. "there is a [color and texture] [obj_name] on the left of a [color and texture] [obj_name], and a [color and texture] [obj_name] on the right of the [color and texture] [obj_name]".

You can derive your answer following these steps.

1. Assuming the given image is either top-view or front-view, please determine the viewpoint of the input image to ascertain its orientation, and please recgonize if the picture is a colored image, sketch or a diagram.
2. Identify all the objects in the image and the user text that match the specified object list, categorizing them by type. You can list every object you find as following, note that you have to take the object in user text into account as well:
eg : object 0: [obj_name], catogory 12; object 1: [obj_name], category 12; object 2: [obj_name], category 8;... Please ensure that objects outside of the specified object list are not extracted, even if they are present in the picture.
3. Define that an object node is a main object surrounded by its several child objects, such as a bed surrounded by nightstands and wordrobe, or a table surrounded by chairs. You should start from the most essential object as root node to identify its child objects, then recursively start from some of the child objects which have their own child objects, and so on, until all objects in the last step have been traversed. You can use the DFS (Depth-First Search) algorithm to traverse the object hierarchy. This implies that any subsequent node n > 0 must serve as child objects to a preceding node. such as :
- Node0(root) = The [obj_name] (object 0) → child_list: [obj_name] (object 1), [obj_name] (object 2), ...
- Node1(child of Node0) = The [obj_name] (object 1) → child_list: [obj_name] (object 3), ...
4. Extract the relationships between the root node objects and their surrounding child items, as well as the relationships between the subsequent node objects and their own child objects, representing these relationships using the previously mentioned triples. You can list every triplet you find as following.
eg: object 0 nightstand is left of (relationship 1) the object 1 [obj_name], triple: [0, 1, 1]; ...
5. Finally, organize the final output according to the specified format. Here is an reasonable example:

[BEGIN]
1. **Viewpoint Calibration**: The given image is a front-view colored image of a bedroom.
2. **Object Extraction**:
   - **Object 0**: [obj_name], category [class_index]
   - **Object 1**: [obj_name], category [class_index]
   - **Object 2**: [obj_name], category [class_index]
   - **Object 3**: [obj_name], category [class_index]
   ...
3. **DFS Traversal**:
   - Node0(root) = [obj_name] (object 0) → childlist: [obj_name] (object 1), [obj_name] (object 2) ...
   - Node1(child of Node0) = [obj_name] (object 1) → childlist: [obj_name] (object 3) ...
   ...
4. **Relationship Extraction**:
   - **[obj_name] (object 1)** **is left of (relationship 1)** **[obj_name] (object 0)** → [1, 1, 0]
   - **[obj_name] (object 2)** **is in front of (relationship 2)** **[obj_name] (object 0)** → [2, 2, 0]
   - **[obj_name] (object 3)** **is in front of (relationship 2)** **[obj_name] (object 1)** → [3, 2, 1]
   ...
5. **Final Output:**:
'''
object list: [0:class_index, 1:class_index, 2:class_index, 3:class_index]
spatial relationship triplets: [[1, 1, 0], [2, 2, 0], [3, 2, 1]]
description: there is a [color and texture] [obj_name] on the left of a [color and texture] [obj_name], a [color and texture] [obj_name] in front of the [obj_name], and a [color and texture] [obj_name] in front of the [obj_name].
'''
[END]

Your response should be formatted exactly according to the template provided between the [BEGIN] and [END] tags. You can use the provided template as a guide, and fill in the blanks in each step by step. Please make sure to include all the necessary information and follow the format strictly. Note that if the picture/diagram is a top view, in the 2D image/diagram, 'above' should correspond to 'behind' or 'closely behind', and 'below' should correspond to 'in front of' or 'closely in front of', and There is no relationship between 'above' and 'below'. And in the 'diagram', the determination of whether A is in front of, behind, to the left, or to the right of B is based on which edge of B's rectangle A is closest to.

Figure 3. One-shot CoT prompt template for bedroom.

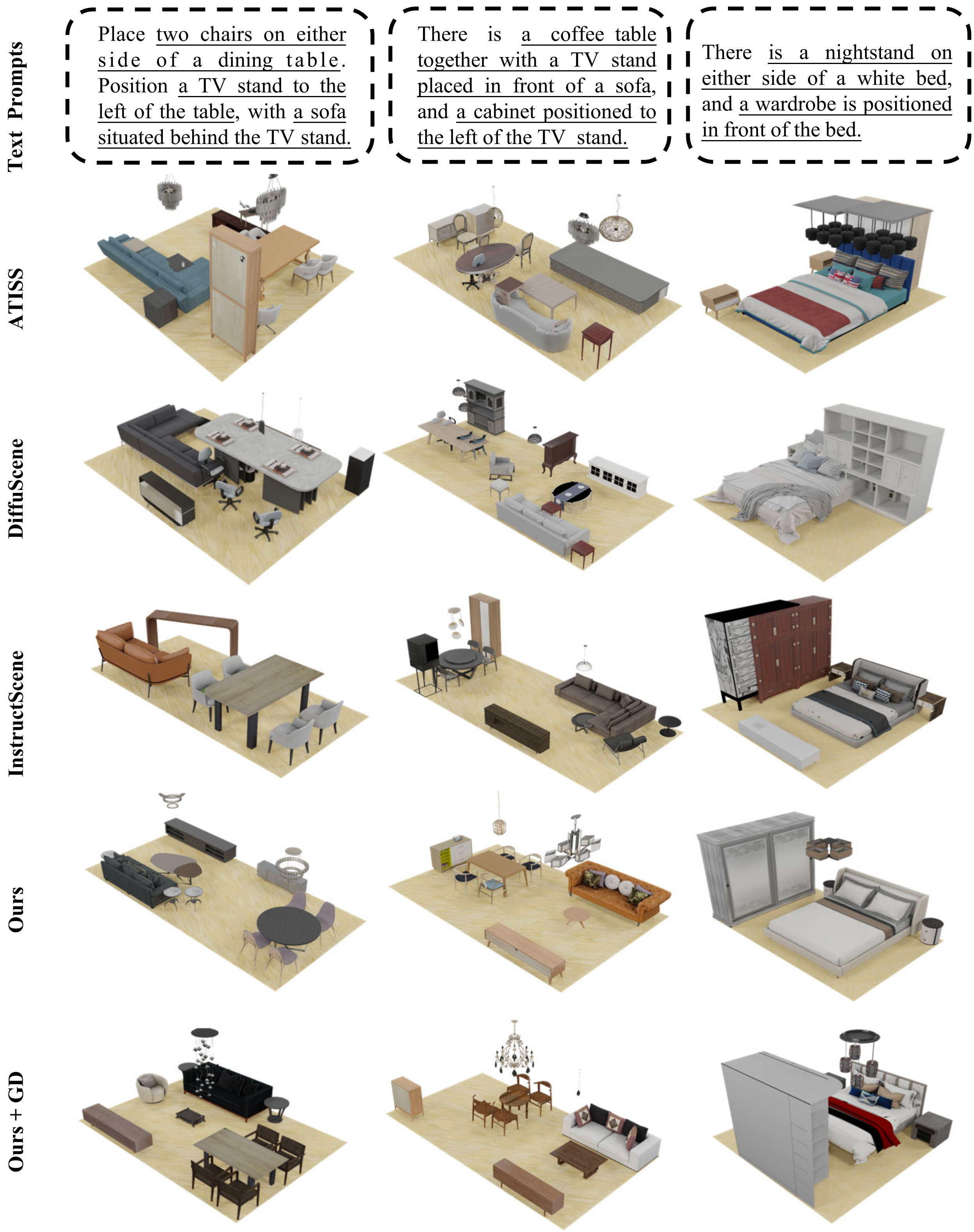

Figure 4. Qualitative comparisons on text-to-scene generation.

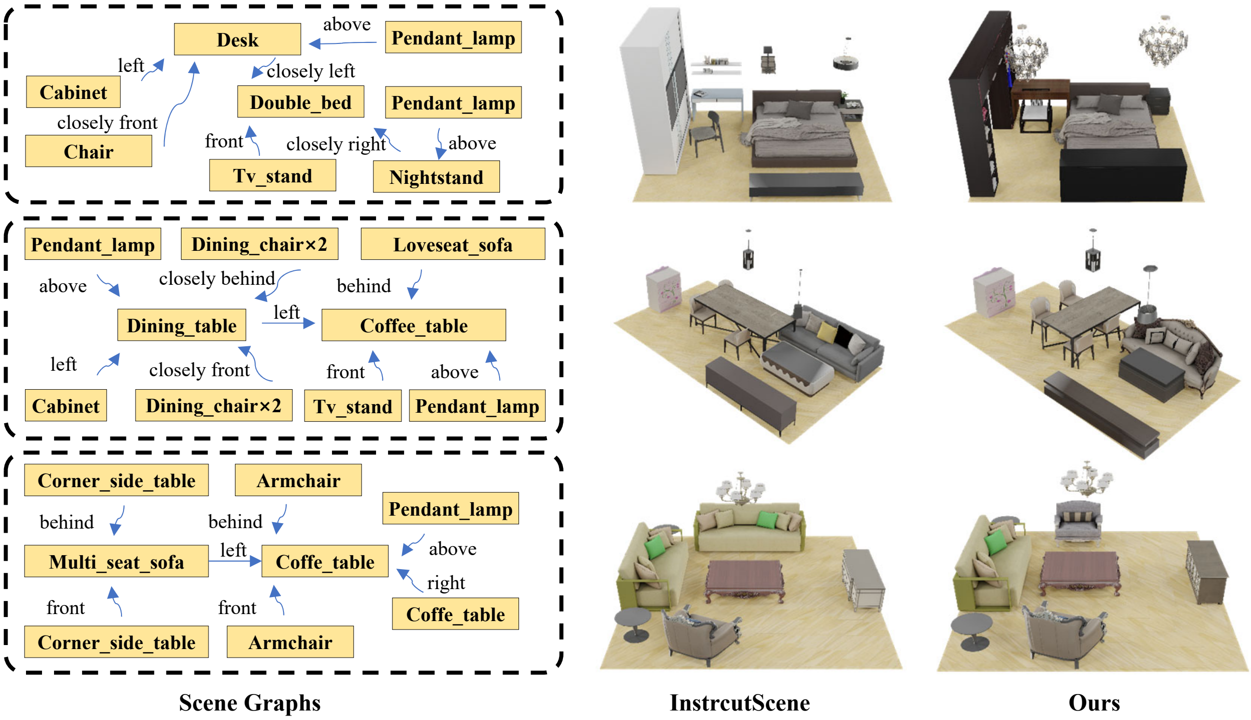

Figure 5. Qualitative comparisons on graph-to-scene generation.

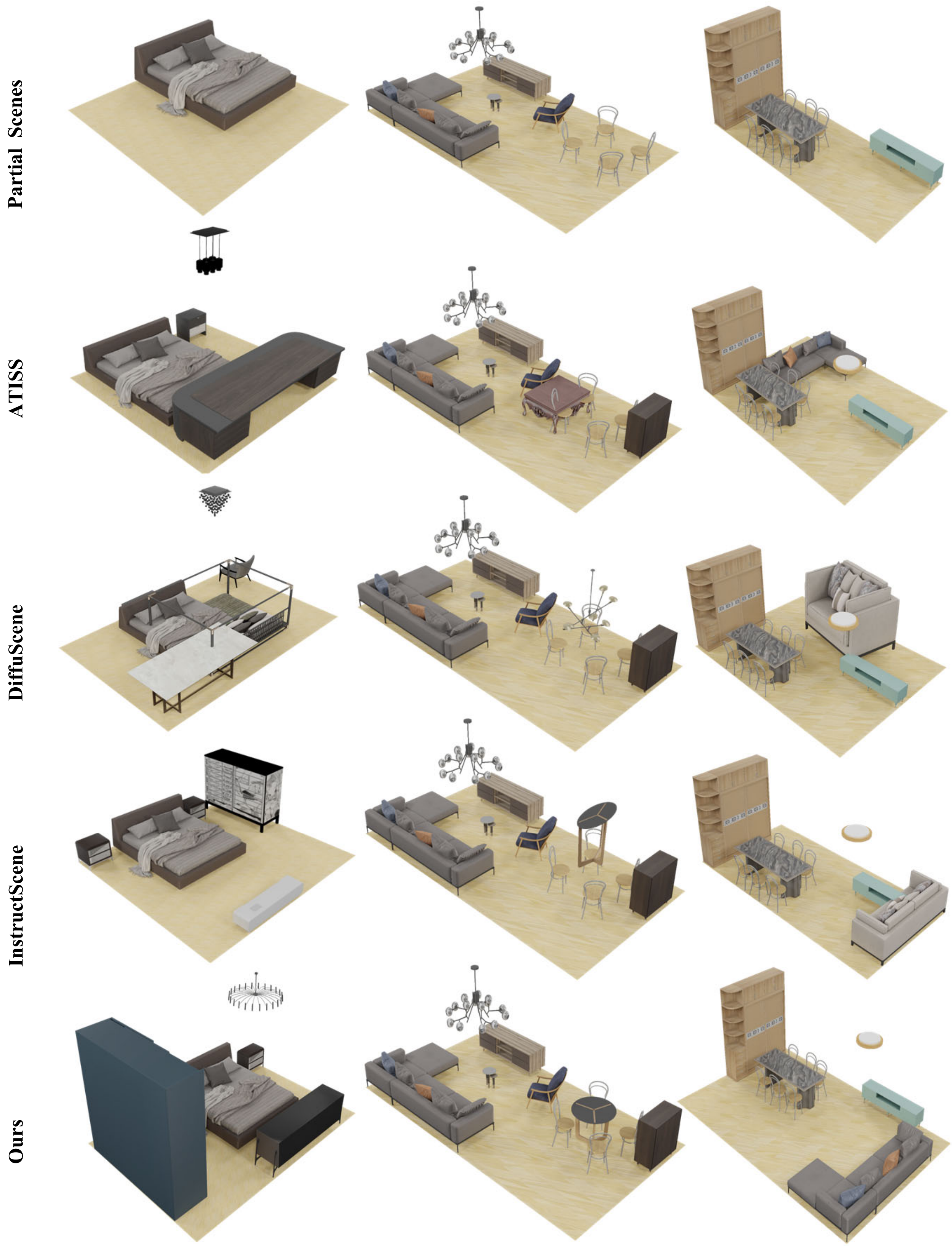

Figure 6. Qualitative comparisons on completion. Note that, in this task, the main goal is to complete the scene by considering the given objects in the input partial scene and the floor size is automatically adapted to the boundary of the updated scene.

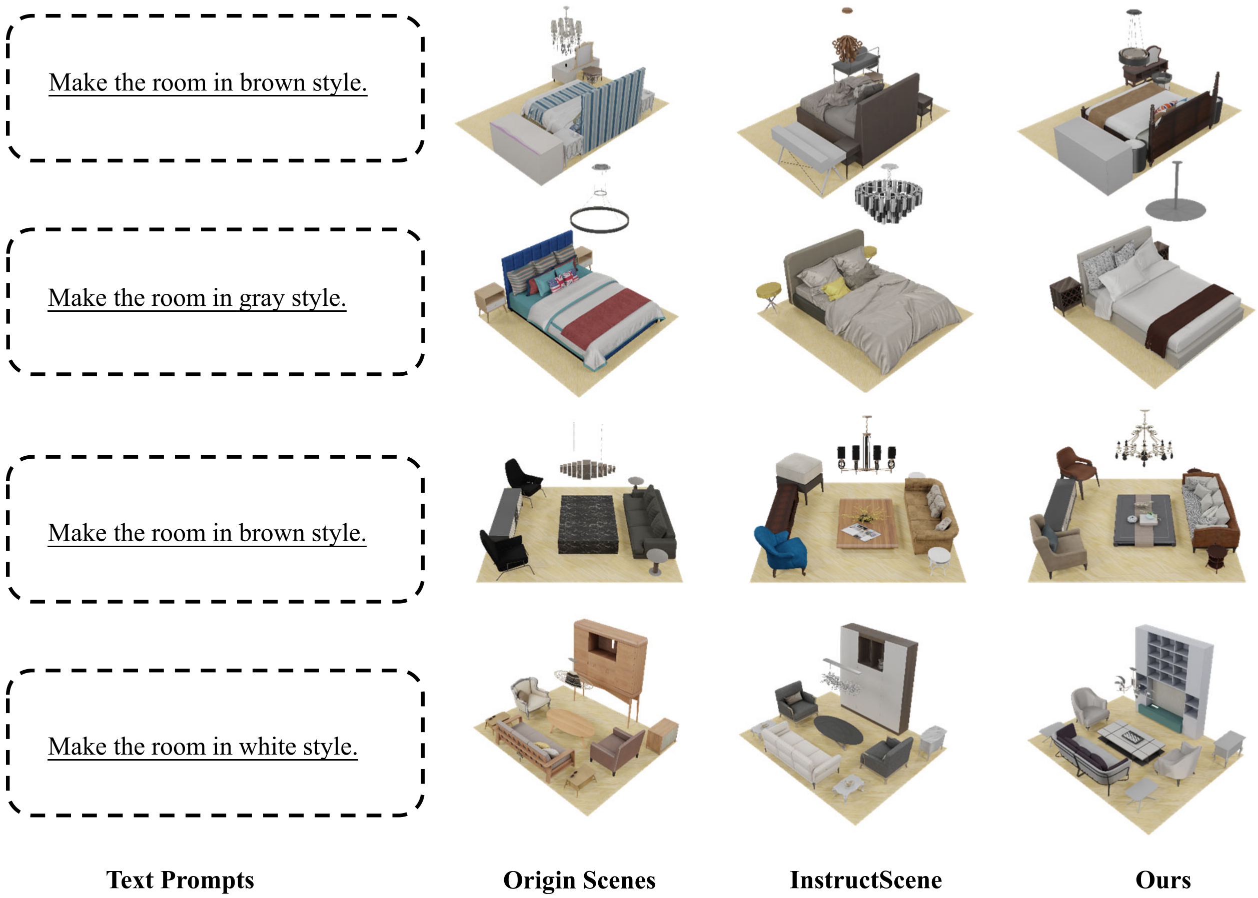

Figure 7. Qualitative comparisons on stylization.