SKE-Layout: Spatial Knowledge Enhanced Layout Generation with LLMs

Supplementary Material

Abstract In this supplementary material, we provide our dataset generation in Section A, implementation details in Section B, additional examples in Section C, and real-world experiments in Section D.

A. Dataset Generation

A.1. SK dataset

We create the SK dataset, a benchmark specifically designed to evaluate complex object rearrangement tasks. The dataset consists of task instructions paired with their corresponding layouts, where each instruction specifies a desired spatial arrangement of objects. These high-level instructions are generated by a large language model (LLM) and incorporate diverse semantic and geometric properties, focusing on both object attributes and spatial relationships. To ensure physical realism and plausibility, we procedurally generate stable, collision-free layouts using the PyBullet physics simulator and render them with high-quality visuals using Blender. The dataset includes more than 100 everyday household objects, as shown in Figure 1.

Figure 1. Objects used in SK dataset

To generate the task instructions, we collect rearrangement tasks from previous studies on object rearrangement as contextual examples and use them as prompts for the LLM, where we use the following LLM template.

> **Template:** *You are a desk object rearrangement project assistant. Your task is to generate table object arrangement tasks. Tasks to consider are:*
> *1. Relative position relationship between objects.*
> *2. The relative position of objects and the table.*
> *3. Creating shapes that represent a single number or single letter using objects.*
> *4. Forming simple geometric shapes (e.g., rectangle, circle).*
> *5. Forming semantic geometric shapes (e.g., star, smile).*
> *The list of items you can use and their serial numbers are: [obj info]. You only need to list tasks with object numbers without providing specific calculations or placement plans. Here are some examples: [examples].*

The generated tasks reflect a wide variety of scenarios and challenges, leveraging the reasoning capabilities of LLMs to create realistic and meaningful rearrangement instructions. This approach ensures the tasks are both contextually relevant and aligned with real-world complexity while maintaining consistency with the benchmarks established in prior research. After generating the task, we further utilized LLM to determine the corresponding object placement layout. We imported it into the Pybullet simulation environment to verify its feasibility, where we use the following LLM template.

> **Template:** *You are a desk object rearrangement project assistant. Your task is to determine whether the results of the robot placement meet the instructions' requirements. The given instruction: [task]. Please judge whether the task is successfully completed. Select one of the following options as the result output:*
> *1. Completed task*
> *2. Uncompleted task*
> *3. Manual judgment required.*

In summary, this comprehensive pipeline seamlessly integrates task generation, layout generation, and validation, providing a reliable and scalable framework for benchmarking object rearrangement tasks.

A.2. StructFormer Dateset

As mentioned in the previous experiment section, the StructFormer dataset is utilized in the StructFormer framework and is designed for the robotic object rearrangement task. Since it's not feasible to perform experiments directly in the point cloud environment, we extracted the necessary information and constructed a precise simulation in Pybullet for testing. This approach allowed us to accurately replicate the object arrangements and constraints from the original environment, ensuring that the model can execute precise object placements for tasks. The figure 2 shows a detailed comparison between the two environments.

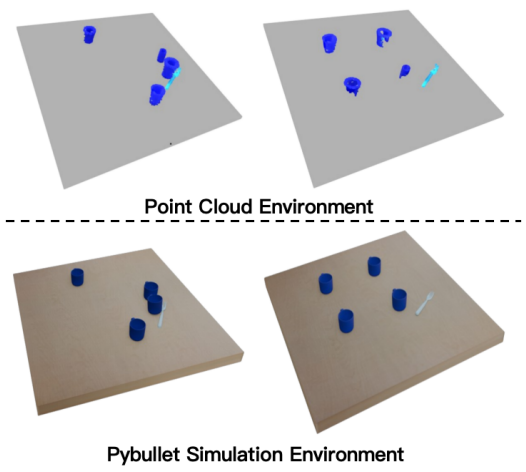

Figure 2. Point Cloud and Pybullet Simulation environment on Structformer dataset

B. Implementation Details

B.1. Task instructions

As is shown in Table 1, two distinct prompt templates are designed to guide layout planning tasks for LLMs, targeting image generation and object rearrangement, respectively. For image generation, the template outlines a structured approach to create object layouts within a constrained 64px-by-64px canvas. The instructions require the layout to follow a CSS-style format, detailing object attributes such as width, height, and absolute position (left and top). This ensures that all properties adhere to the spatial limitations while maintaining precision. Relevant knowledge is also provided to assist the model in generating accurate layouts based on given instructions. For object rearrangement, the prompt template focuses on reorganizing objects on a table according to specific instructions. It incorporates environmental awareness and three-dimensional spatial definitions, including coordinates (x, y, z) and rotation (yaw). The task requires calculating positions and orientations while considering object size to avoid collisions. Contextual knowledge is included to support further effective arrangement planning. These templates demonstrate a systematic approach to facilitating diverse layout-related tasks for LLMs.

B.2. Implementation and Training Details

We implement our retrieval model using the Sentence Transformer library and select *distilbert-base-nli-stsb-mean-tokens* as the backbone. For task-specific adaptations, we introduce an embedding head f, which consists of additional Transformer layers followed by an MLP head g, which consists of one hidden layer of size 768 and outputs vectors of size 128, providing compact and task-relevant embeddings for retrieval. The experiment is done on an NVIDIA RTX 4090 GPU, and the hyperparameter settings are listed in Table 2. For the LLM, we set the sampling temperature to 0.7 to generate diverse data and results. The number of examples related to spatial knowledge is fixed at

Table 1. Instructions for Layout Planning for LLMs.

Task	**Instruction for LLMs**
Image Generation	Instruction: You are an image layout generation project assistant. Your task is to generate the layout of the objects in the image according to specific instructions. The generated layout should follow the CSS style, where each line starts with the object description and is followed by its absolute position. Formally, each line should be like `object{width:?px; height:?px; left:?px; top:?px;}`. The image is 64px wide and 64px high. Therefore, all properties of the positions should not exceed 64px, including the addition of left and width and the addition of top and height. The given instruction: `{instruction}`. Here are some relative tasks and corresponding layouts you can refer to: `{knowledge}`
Object Rearrangement	Instruction: You are a desk object rearrangement project assistant. Your task is to reposition items on the table according to specific instructions. Table range: `{min-x, min-y, max-x, max-y, table-z}` The given instruction: `{instruction}`. Please calculate and provide the position and orientation of each item on the table. Here is the detailed information about the objects: `{objects info}`. Explanation of terms: `{x y z}`: These represent the coordinates in a three-dimensional space. `{yaw}`: Rotation. You need to focus on the size of the object and try to avoid bumping into each other. Here are some relative tasks and corresponding layouts you can refer to: `{knowledge}`

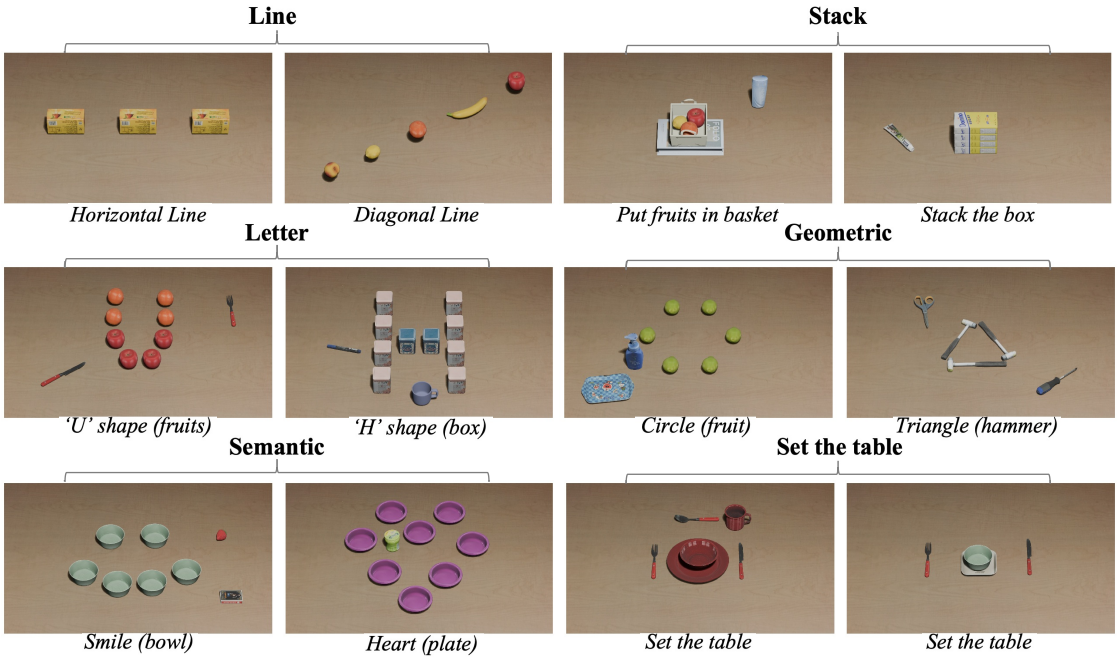

Figure 3. Examples of the Object Rearrangement task

Table 2. hyperparameters

Parameter	Value
Optimizer	Adam
Learning rate	10^{-5}
β	(0.9, 0.999)
Learning rate schedule	None
Epochs	100
Batch size	400
λ_{image}	0.5
λ_{object}	0.5

8 to ensure consistent input for LLM.

B.3. Details about Baselines

Robotic Object Rearrangement Task

- **StructFormer**: StructFormer is a structure-based Transformer model specifically designed for generating complex object placement layouts. By leveraging its ability to capture spatial relationships among objects, StructFormer excels in arranging objects in a coherent and logical manner, taking into account the dependencies between them. Its transformer architecture allows it to efficiently process object-to-object interactions, ensuring that the generated layouts adhere to predefined spatial constraints and logical rules. StructFormer is particularly suited for tasks that require a fine-grained understanding of spatial dependencies, making it a robust baseline for robotic rearrangement.
- **LLM-GROP**: LLM-GROP (Large Language Model with Generate-and-Operate Planning) is an advanced model that combines the capabilities of large language models with generate-and-operate planning techniques. This model uses language models to generate initial proposals for object layouts based on the input instructions and then applies planning algorithms to refine and execute these layouts. By integrating linguistic reasoning with operational precision, LLM-GROP achieves high accuracy in placing objects according to both spatial and contextual requirements. Its two-step approach ensures that the final arrangement is not only logical but also feasible in a real-world setting.

Image Generation Task

- **Stable Diffusion**: Stable Diffusion is a state-of-the-art Text-to-Image generation model that employs a diffusion-based approach to create high-resolution, photorealistic images from textual prompts. This model excels in capturing intricate details and producing visually coherent outputs, even for complex scenes with multiple objects and nuanced relationships. By iteratively refining noisy images toward the desired output, Stable Diffusion ensures high-quality results, making it a powerful tool for diverse image generation tasks.
- **Attend-and-Excite**: Attend-and-Excite is an end-to-end Text-to-Image generation model that leverages attention mechanisms to emphasize critical details in the input textual descriptions. By focusing on the most salient parts

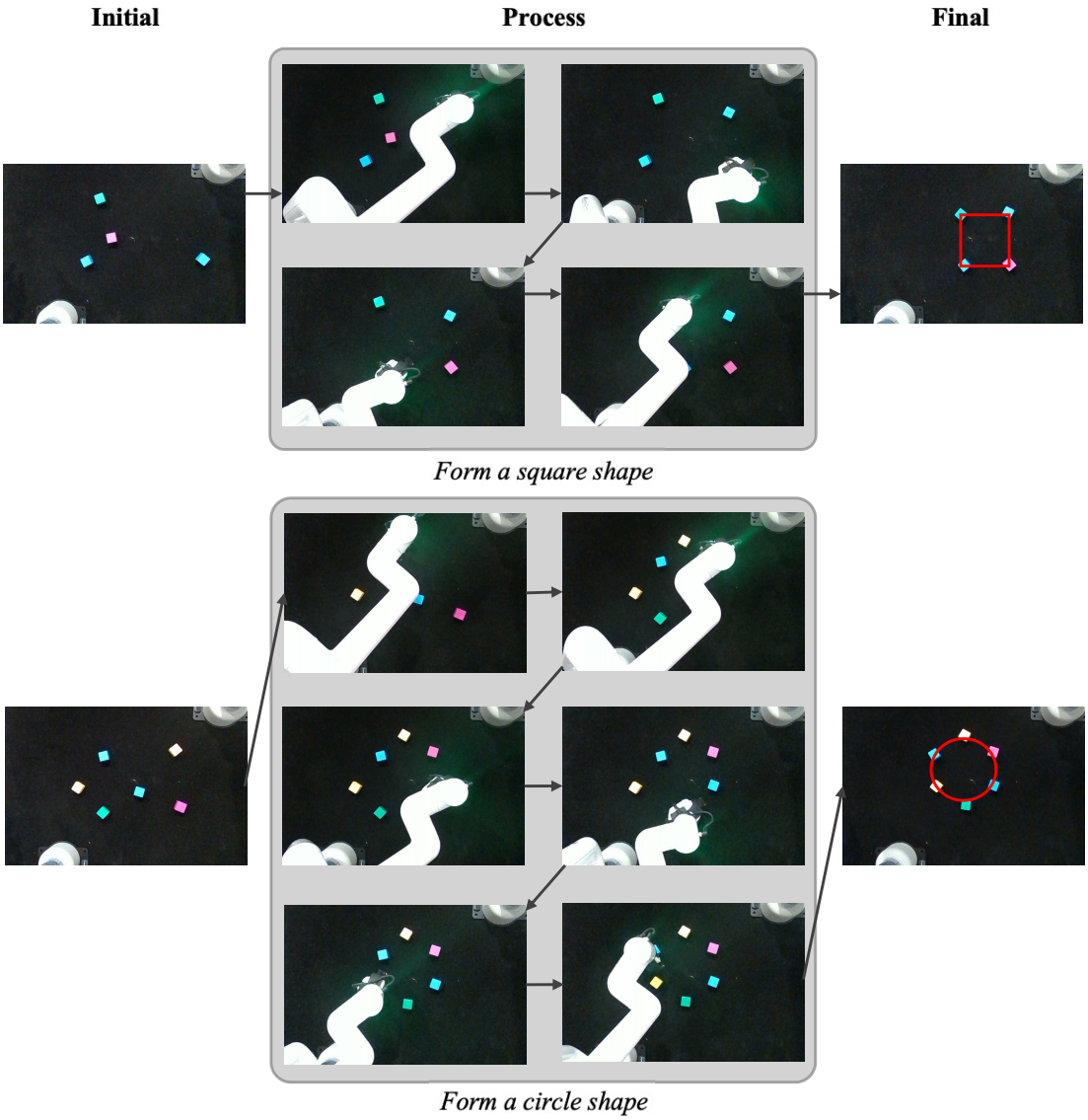

Figure 4. Real-world Experiments

of the prompt, this model generates realistic and contextually relevant images. Its attention-driven approach ensures that the generated visuals align closely with the key elements of the textual input, making it particularly effective for scenarios where precision and detail are paramount.

- **LayoutTransformer**: LayoutTransformer is a Text-to-Layout generation model that utilizes a transformer architecture to produce structured layouts from textual descriptions. This model focuses on creating spatially accurate and logically organized layouts that serve as a blueprint for subsequent image generation. By generating intermediate layouts, LayoutTransformer bridges the gap between textual input and visual output, providing a structured representation that can guide downstream image generation models.
- **LayoutGPT**: LayoutGPT is a Text-to-Layout generation model that harnesses the power of large language models to create layouts directly from textual prompts. By combining the linguistic understanding of LLMs with layout generation capabilities, LayoutGPT can produce spatially organized and contextually relevant layouts. Its flexibility and adaptability make it an ideal choice for generating layouts that accurately reflect complex textual descriptions, serving as a robust baseline for Text-to-Layout tasks.

C. Additional Examples

We provide additional visual examples to highlight the performance of SKE-Layout in various contexts. Figure 3 showcases some examples of the object rearrangement task.

D. Real-world Experiments

We used a RealSense D435i as a fixed camera and deployed our system on an Elephant Pro630 robot to evaluate real-world object manipulation. The environment perception algorithm integrates DETR and VLM, as mentioned in the main text, while high-level execution instructions are generated by the LLM. DMPL is used for motion planning of the robotic arm. As shown in the figure 4, the final object placements are semantically logical, successfully completing the object rearrangement tasks (e.g., square, circle). These results effectively demonstrate the capability of our approach to perform real-world tasks using a robotic platform.