Kestrel: 3D Multimodal LLM for Part-Aware Grounded Description

Supplementary Material

A. Mask3D Baseline

To extend the GLaMM model from 2D to 3D, we incorporate our multimodal large language model (MLLM), $\mathcal{F}_{\mathcal{VL}}$, pretrained on PointLLM Objaverse [64] and 3DCoMPaT-GRIN datasets, with the Mask3D segmentation model. Mask3D [53] comprises a voxel-based ResUNet encoder, $\mathcal{E}_{voxel}$, and a segmentation decoder, $\mathcal{D}$, both pretrained on the ScanNet200 [52] dataset. The Mask3D model is designed to process point clouds containing 8,192 RGB points with a voxel size of 0.01, employing intermediate voxel resolutions of 600, 1200, 2048, 4096, and 8192 within its ResUNet upsampling blocks.

During grounded caption generation for 3D shapes, $\mathcal{F}_{\mathcal{VL}}$ outputs special [SEG] tokens, from which we extract the corresponding hidden states, h_{seg}. These hidden states are projected into the query embedding space of $\mathcal{D}$ and serve as positional queries for the segmentation decoder. The decoder then leverages these queries to predict part segmentation masks, aligning the textual grounding queries with the spatial representation of the 3D point cloud.

This process effectively combines the language understanding capabilities of $\mathcal{F}_{\mathcal{VL}}$ with the spatial reasoning and segmentation strength of Mask3D, enabling robust 3D part-aware grounding. The process is described mathematically below:

$$\mathbf{f}_{voxel} = \mathcal{E}_{voxel}(\mathbf{x}_{pc}), \tag{8}$$

where $\mathbf{x}_{pc}$ is the input point cloud with 8,192 RGB points, and $\mathcal{E}_{voxel}$ represents the voxel-based ResUNet encoder that extracts voxel features $\mathbf{f}_{voxel}$.

$$\mathbf{h}_{seg} = \mathcal{F}_{\mathcal{VL}}(\mathbf{x}_{pc}, \mathbf{x}_{txt}), \tag{9}$$

where $\mathcal{F}_{\mathcal{VL}}$ is the MLLM that generates the grounded caption and produces hidden states $\mathbf{h}_{seg}$ corresponding to the [SEG] tokens.

$$\mathbf{q}_{pos} = \mathcal{P}(\mathbf{h}_{seg}), \tag{10}$$

where $\mathcal{P}$ is the projection layer that maps the hidden states $\mathbf{h}_{seg}$ into the query embedding space of the segmentation decoder $\mathcal{D}$.

$$\mathbf{m}_{seg} = \mathcal{D}(\mathbf{f}_{voxel}, \mathbf{q}_{pos}), \tag{11}$$

where $\mathcal{D}$ is the segmentation decoder, and $\mathbf{m}_{seg}$ represents the predicted part segmentation masks.

We set the max number of queries for Mask3D to 16

Point Enc.	Projector	Grounded Desc. 3D-CALC	Grounded Desc. mIoU	D.S. mIoU	R.S. mIoU
PointBert	Q-Former	43.80	72.24	67.40	64.9
PointBert	MLP	44.65	77.05	69.03	66.2
Uni3D	Q-Former	49.45	74.50	70.26	68.4
Uni3D (Final Choice)	MLP (Final Choice)	50.10	86.70	80.70	71.8

Table 6. **Ablation on different architecture designs**

# GD Samples	# DS Samples	GD mIoU	DS mIoU
80K	0	80.67	49.97
80K	40K	81.37	**79.86**
80K*	8K*	**86.07**	78.78

Table 7. **Dataset Ablation.**

B. Additional Ablation Studies

Model Architecture. Tab. 6 presents an ablation study on different architectural choices for the point encoder and projector. The results indicate that both the choice of encoder and projector significantly influence the model's performance across all evaluation metrics. Using Uni3D as the point encoder consistently improves results over PointBert, regardless of the projector type. Similarly, MLP outperforms Q-Former as the projector, showing higher mIoU scores across Grounded Descriptions (GD), Direct Segmentation (DS), and Reasoning Segmentation (RS). The final chosen configuration, Uni3D with an MLP projector, achieves the best overall performance. This suggests that using an MLP to project the original point features into the LLM space produces tokens that closely match those used during upsampling in the segmentation decoder, leading to better results.

Dataset Amount. Tab. 7 Analyzes the impact of single-part data by varying the amount of Direct Segmentation (DS) data while keeping the Reasoning Segmentation (RS) data at zero as a control variable. Results show that adding a small amount of DS samples (8K) helps Kestrel learn part grounding. However, increasing DS to 40K raises training cost and slightly degrades performance on grounded descriptions. This suggests that a small amount of DS data offers a good trade-off for overall performance.

Training Data	Grounded Desc.	Direct Segmentation	Reasoning Segmentation
GD	80.67	49.97	48.32
GD+DS	86.07	72.57	52.03
GD+DS+RS	**86.70**	**80.70**	**71.8**

Table 8. **Further ablation on the dataset distribution.** GD: Grounded Dscription subset. DS: Direct Segmentation subset. RS: Reasoning Segmentation subset.

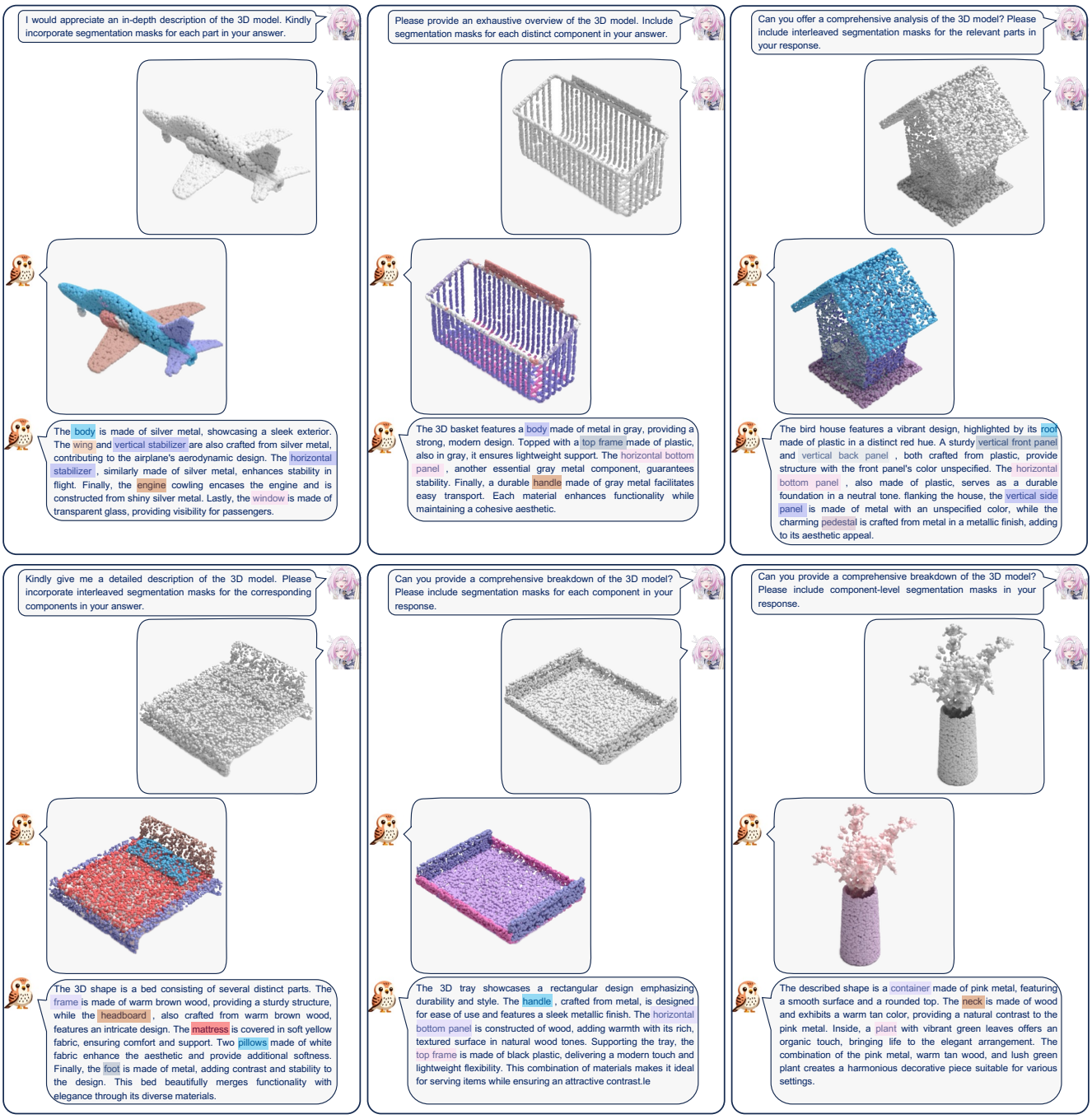

Figure 6. **Qualitative results of Kestrel on Part-Aware Point Grounded Description.**

Dataset Distribution. Tab. 8 explores how different training data subsets influence the model's overall performance. Starting with only the grounded description subset yields moderate performance in each task. Adding the direct segmentation subset leads to a noticeable boost, particularly for its task. Finally, incorporating the reasoning segmentation subset achieves the best results, confirming that diverse training data covering grounded descriptions, direct, and reasoning-based instructions is essential for robust part-aware vision-language understanding. Together, these findings underscore the effectiveness of both progressive query refinement and comprehensive dataset coverage in enhancing language understanding and segmentation accuracy.

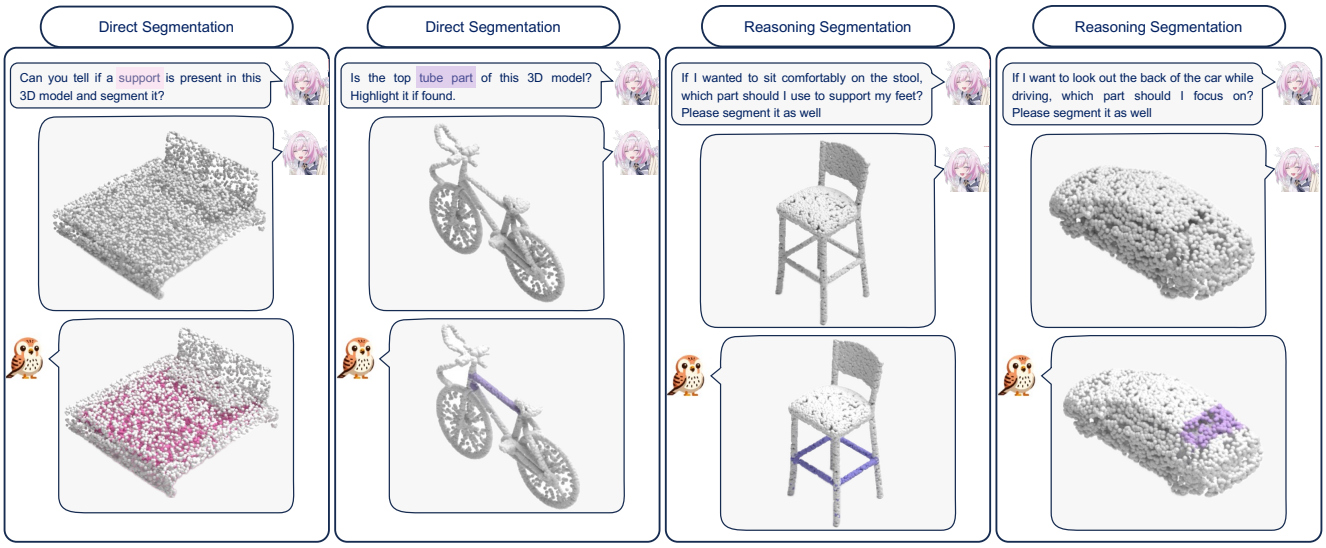

Figure 7. **Qualitative results of Kestrel on Single-Part Grounding.**

C. PartNet-Mobility

Tab. 9 presents a comparison of our method against prior baselines on a sample of categories from the PartNet-Mobility dataset, along with the overall accuracy. While PartSTAD achieves the highest overall score, it relies on training a separate model for each object category. In contrast, our method uses a single unified model across all categories. Notably, our few-shot setting outperforms or matches prior methods on several categories, including Lamp and Chair, and achieves strong overall performance. The results demonstrate the effectiveness of our approach in generalizing to diverse part segmentation tasks with limited supervision.

D. Qualitative Examples

Due to space constraints in the main paper, we present additional qualitative experiments in this section. As shown in Fig. 6, Kestrel demonstrates its ability to provide comprehensive explanations of 3D objects, offering detailed, part-level descriptions for a given 3D object. For each part-level phrase in the generated description, Kestrel predicts its corresponding position within the 3D space, represented by segmentation masks. Fig. 7 showcases the single-part segmentation grounding results. Kestrel demonstrates its ability to interpret part-aware instructions, understand user intent, and predict the corresponding position based on the given text input.

E. Data Collection

E.1. GPT4 collection

In order to generate a detailed caption for each shape in 3DCoMPaT, we leverage the metadata of each shape to create part-material pair assignment text in the form of: "*part_name* made of *material_name*, $\cdots$". This metadata text is used in the GPT-4o prompt along with 8 views of the shape to caption it accurately in terms of the part, material, and color. a sample of the prompt can be seen in Tab. 10. For the reasoning segmentation task, we use the metadata to prompt GPT4o to create an indirect prompt and response each part. A sample of the prompt can be found in Tab. 11

E.2. Prompts

A list of 30 predefined instructions is utilized to prompt the model to generate descriptions and ground segmentation masks, as detailed in Table 12. Additionally, another set of 15 predefined prompts, illustrated in Table 13, is used to evaluate direct segmentation task. To complement these instructions, a collection of 11 template responses, as shown in Table 14, guides the expected output format for the direct segmentation task. Each instruction and template was created with the assistance of GPT-4o to ensure diversity and relevance in the dataset.

The dataset creation process is divided into two distinct parts:

1. **Grounded Description Dataset**: This dataset is designed for multi-part grounding tasks, where each caption describes multiple parts of a shape along with their segmentation masks. To generate captions, we randomly assign an instruction from the 30 predefined instruction list to each sample and use GPT-4o to produce a descrip-

Model	Bottle	Chair	Display	Door	Knife	Lamp	Storage Furniture	Table	Camera	Cart	Dispenser	Kettle	Kitchen Pot	Oven	Suitcase	Toaster	Overall
PointNeXt	68.4	91.8	89.4	43.8	58.7	64.9	68.5	52.1	33.2	36.3	26.0	45.1	57.0	37.8	13.5	8.3	50.2
PartSLIP	83.4	85.3	84.8	40.8	65.2	63.9	66.0	53.6	58.3	88.1	73.7	77.0	69.6	73.5	70.4	60.0	59.4
PARIS3D	84.0	81.0	70.1	68.4	47.2	61.2	39.4	45.1	29.3	71.7	40.1	59.3	78.8	59.1	61.6	24.9	57.6
PartSTAD	83.6	85.1	82.3	61.4	63.8	68.3	59.5	47.7	64.3	85.0	73.7	84.2	73.5	71.8	68.2	58.6	**65.0**
Ours (zero-shot)	67.6	58.0	57.9	56.0	56.5	58.3	32.4	40.1	35.1	59.1	43.6	67.5	72.4	48.4	75.6	41.2	47.7
Ours (few-shot)	80.0	83.3	85.2	79.1	67.3	84.0	54.7	54.7	40.4	61.9	63.9	79.3	76.3	67.9	75.1	31.4	*63.9*

Table 9. **Sample Category Results on PartNet-Mobility** Performance on selected categories and overall accuracy. All models are evaluated on the full set; only a subset is shown here.

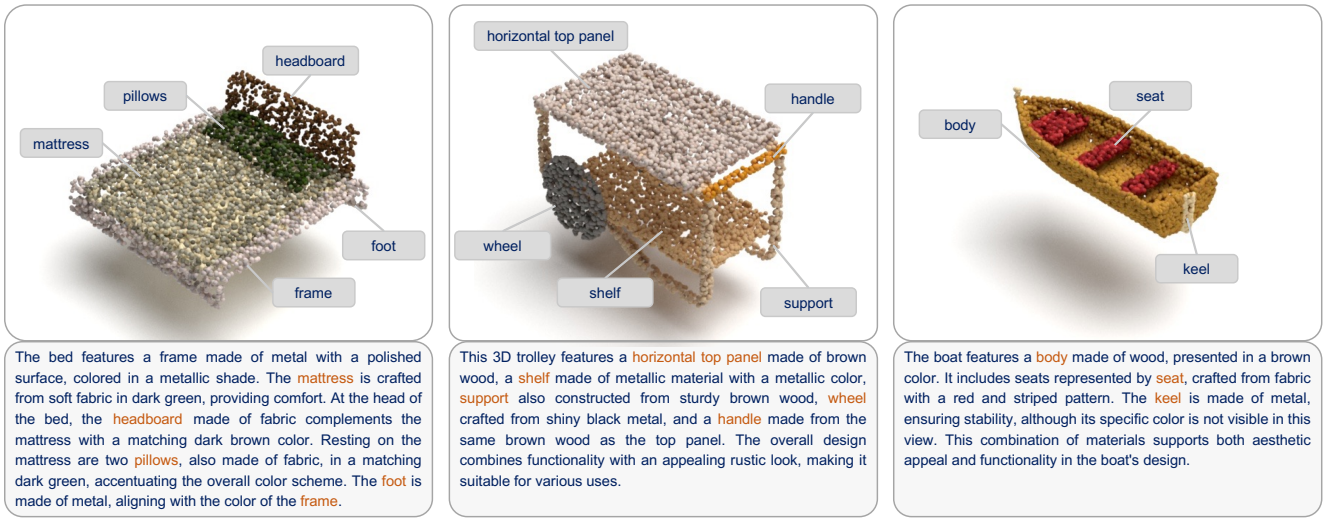

Figure 8. **Visualizations of collected 3DCoMPaT-GrIn .**

tive and coherent output.

2. **Reasoning Segmentation Dataset**: For this dataset, a single part from the shape is targeted for segmentation. A random instruction is selected from the single-part instruction set, and it is paired with a corresponding response template from the predefined list. This ensures consistency in both input and expected output formats.

By leveraging these structured prompts and templates, we ensure that our dataset provides comprehensive coverage of both multi-part and single-part grounding tasks, effectively addressing the challenges of part-aware segmentation and language grounding in 3D models.

E.3. Human Evaluation

To evaluate our validation set of grounded descriptions, we conducted a human evaluation on the 6,770 grounded description samples. Annotators were provided with the interface shown in Fig. 9 and were tasked with evaluating the following aspects:

1. Whether the caption includes all the ground truth parts and their corresponding materials.
2. Whether the part color described in the caption matches the true color visible in the rendered views.
3. If the caption includes extra part names, we ask the annotator to mention them separated by a comma in order to correct the sample afterward.

Based on the annotators' responses, we identify samples with inaccuracies and reran the GPT-4o pipeline on these samples. This iterative process ensured the creation of a fully accurate validation set, establishing a reliable benchmark for evaluating grounded descriptions.

E.4. Examples

Using the proposed dataset collection pipeline, we collect a total of 88,836 training samples and 6,770 validation samples for part-aware point-grounded description. Additionally, 677 validation samples are collected for single-part grounding. Fig 8 shows examples of the colored point clouds alongside their corresponding grounded descriptions. As shown, the collected data effectively captures the various components of 3D objects, accurately representing each part-level component and its position.

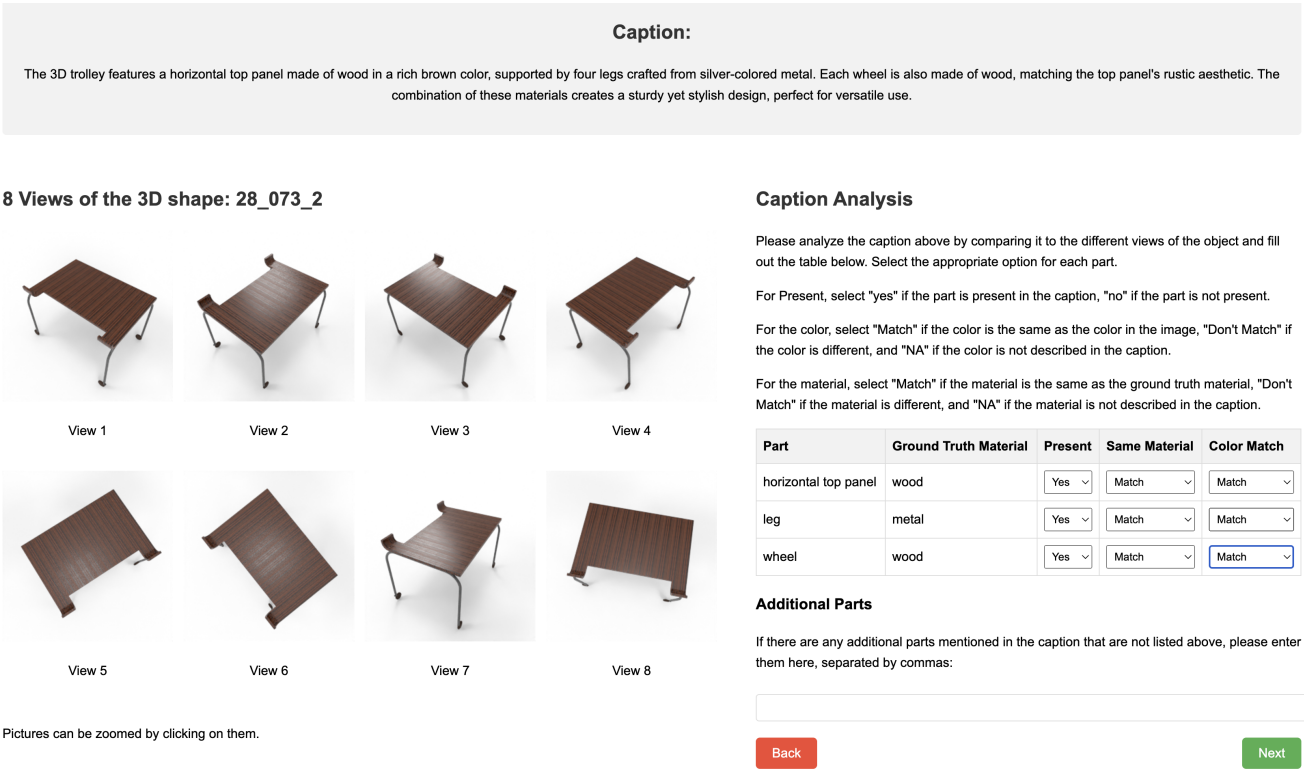

Figure 9. **Caption Validation Website.** The annotators are asked to compare the shape's caption with the ground truth parts and material and check that the part color described in the caption matches its color in the rendered images.

Given the following different views of the same 3D bench, caption the 3D shape by giving a description of the shape and its parts also describe the parts and their materials and add the exact color of each part from the provided part material assignment list: "seat is made of leather, seat_frame is made of plastic, stretcher is made of plastic, leg is made of metal". Make the caption short but comprehensive and descriptive. Your output must be without styling or line breaks and under 500 characters. You must mention the color of each part explicitly using their exact names from the list. Do not add any extra part names that are not on the list. Avoid describing the background or adding any unnecessary text or mentioning the words images, views, or objects. Do not use any words that shows you are not sure about the color. if the part name has _ in it, Do not replace it with a space. If a part is not visible, do not mention that it is not found or not visible, instead mention the material description of the part. The caption should be coherent and descriptive. The caption sentence is:

Table 10. **Sample GPT4o Prompt** An example of the prompt given to GPT4o

I will be giving you a 3D object category along with part name and its material. I want you to generate a question prompt that inquires about the part's funtionality and usage and then provide the answer seperate by a new line. For example, if the object is a teapot, the part is the handle and the material is wood, the question should be Ïf I wanted to hold the teapot, what part should I hold?̈. You have to use the word 'part' in the question and mention the object name. Do not say the part name in the question and do not ask about its functionality since you mention the functionality in the question. The question must be descriptive of the part and uniquely identify the part in the object. For the answer, you must use the same part name and the functionality of the part. The object category is: car, the part name is: door, the material is: metal. The question prompt is:

Table 11. **Sample GPT4o reasoning segmentation prompt** An example of the prompt used to create questions and answers

- Can you provide a comprehensive breakdown of the 3D model? Please include segmentation masks for each component in your response.
- I would appreciate a thorough explanation of the 3D model. Kindly incorporate segmentation masks for the relevant parts in your answer.
- Please offer a meticulous analysis of the 3D model. Include interleaved segmentation masks for the corresponding sections in your reply.
- Could you give me an in-depth description of the 3D model? Please provide part-specific segmentation masks within your response.
- I would like a detailed overview of the 3D model. Please include segmentation masks for each distinct element in your answer.
- Kindly provide an extensive description of the 3D model. Please incorporate component-level segmentation masks in your explanation.
- Can you offer a comprehensive analysis of the 3D model? Please include interleaved segmentation masks for the relevant parts in your response.
- I request a thorough breakdown of the 3D model. Please provide segmentation masks for each part in your answer.
- Please give me a detailed explanation of the 3D model. Include part-specific segmentation masks in your reply.
- Could you provide an exhaustive description of the 3D model? Please include segmentation masks for each component in your response.
- I would appreciate a meticulous analysis of the 3D model. Kindly incorporate interleaved segmentation masks for the corresponding sections in your answer.
- Can you give me a comprehensive overview of the 3D model? Please provide segmentation masks for each distinct element in your explanation.
- Please offer an in-depth breakdown of the 3D model. Include component-level segmentation masks within your reply.
- I request a detailed description of the 3D model. Please incorporate part-specific segmentation masks in your response.
- Kindly provide a thorough explanation of the 3D model. Please include segmentation masks for the relevant parts in your answer.
- Could you give me an extensive analysis of the 3D model? Please provide interleaved segmentation masks for the corresponding components in your response.
- I would like a comprehensive breakdown of the 3D model. Please include segmentation masks for each section in your reply.
- Can you offer a meticulous description of the 3D model? Kindly incorporate part-level segmentation masks in your explanation.
- Please provide an exhaustive overview of the 3D model. Include segmentation masks for each distinct component in your answer.
- I request a detailed analysis of the 3D model. Please provide part-specific segmentation masks within your response.
- Could you give me a thorough explanation of the 3D model? Please include interleaved segmentation masks for the relevant sections in your reply.
- I would appreciate an in-depth description of the 3D model. Kindly incorporate segmentation masks for each part in your answer.
- Can you provide a comprehensive breakdown of the 3D model? Please include component-level segmentation masks in your response.
- Please offer an extensive analysis of the 3D model. Include segmentation masks for each distinct element in your explanation.
- I request a meticulous overview of the 3D model. Please provide part-specific segmentation masks in your reply.
- Kindly give me a detailed description of the 3D model. Please incorporate interleaved segmentation masks for the corresponding components in your answer.
- Could you offer a thorough breakdown of the 3D model? Please include segmentation masks for each section in your response.
- I would like an exhaustive explanation of the 3D model. Kindly provide part-level segmentation masks within your reply.
- Can you give me a comprehensive analysis of the 3D model? Please include segmentation masks for the relevant parts in your answer.
- Please provide a meticulous description of the 3D model. Include component-specific segmentation masks in your response.

Table 12. **Instruction list for grounding description task.** Each instruction is paired with a GPT-generated caption to guide the generation of part-specific segmentation masks for the 3D model.

- Does this 3D *shape_name* have a *part_name*? If yes, where is it located?
- Is there a *part_name* in this 3D *shape_name*? Please segment it if it exists.
- Can you tell if a *part_name* is present in this 3D *shape_name* and segment it?
- Is a *part_name* included in this 3D *shape_name*? Please highlight its location.
- Does this 3D *shape_name* contain a *part_name*? If so, please isolate it.
- Please check if there is a *part_name* in the 3D *shape_name*, and segment it if present.
- Is the *part_name* part of this 3D *shape_name*? Highlight it if found.
- Is there a part described as *part_name* within this 3D *shape_name*? Segment it if present.
- Can you confirm if the *part_name* exists in this 3D *shape_name* and segment it?
- Does this 3D *shape_name* have a *part_name*? Show me where it is, if applicable.
- Can you identify if the *part_name* is present in this 3D *shape_name* and segment it?
- Please check if the area corresponding to the *part_name* is part of the 3D *shape_name*.
- Can you verify the existence of a *part_name* and segment it within the 3D *shape_name*?
- Is there a *part_name* in this *shape_name*? Segment it if found.
- Check if the *part_name* is present in the 3D *shape_name*, and segment it if applicable.

Table 13. **Instruction list for direct segmentation task.** Each instruction is paired with a template answer. The *part_name* and *shape_name* words are replaced with the part and model names for each sample respectively.

- Yes, there is a *part_name* [SEG] part in the 3D *shape_name*.
- Yes, a *part_name* [SEG] part is present in the 3D *shape_name*.
- Confirmed, the 3D *shape_name* contains a *part_name* [SEG] part.
- The *part_name* [SEG] part is found in this 3D *shape_name*.
- Indeed, there is a *part_name* [SEG] part included in the 3D *shape_name*.
- A *part_name* [SEG] part is present within the 3D *shape_name*.
- Yes, the 3D *shape_name* includes a *part_name* [SEG] part.
- There is a *part_name* [SEG] part in this 3D *shape_name*.
- You can find a *part_name* [SEG] part in the 3D *shape_name*.
- The 3D *shape_name* contains a *part_name* [SEG] part.
- Yes, the 3D *shape_name* features a *part_name* [SEG] part

Table 14. **Template list for direct segmentation task.** The *part_name* and *shape_name* words are replaced with the part and model names for each sample respectively.