PointLLM: Empowering Large Language Models to Understand Point Clouds

Supplementary Material

Appendix

A D.4. C.II. 4

A Data Collection	I
B Benchmarks and Evaluation	1
B.1. Discussions about Close-Set Classification .	1
B.2. GPT Evaluation Prompts	2
B.3. Human Verification of GPT Evaluation	2
B.4. Human Scoring Criteria	2
C Implementation and Training	3
D Win Rate Comparison	3
E. Qualitative Results	3
F. Text-to-3D Generation	4

A. Data Collection

Instruction lists. The 30 pre-defined instructions used to prompt the model to briefly and elaborately describe the objects are shown in Tab. 2 and Tab. 3 respectively. These prompts are generated with the assistance of GPT-4 and are coupled with captions to form our description-type data.

Data generation with GPT-4. In Tab. 4 we show an example of using GPT-4 for data generation as well as the system prompt of GPT-4. The input is one human-written caption provided by Cap3D[14] and the outputs are one expanded detailed caption, three single-round conversations, and one multi-round conversation. The system prompt is used for all samples, which guides the model to analyze existing captions based on the general knowledge of 3D objects and generate detailed captions, diverse Q&As, and logically connected multi-round conversations.

Dataset distribution. The comprehensive statistics of our newly compiled point-text instruction following dataset are detailed in Tab. 1. The dataset encompasses approximately 730K samples, designed for effective training. Fig. 1 illustrates the length distributions of instructions and responses across various data types. In Fig. 2, we present word clouds (after removing generic words like "model", "object", etc.) and verb-noun pair distributions following [18] from our instruction following dataset, highlighting its extensive coverage of diverse topics such as color, shape, usage, material, and more.

Table 1. Statistics of our point-text instruction following data.

Statistics Number of all samples 731851 - brief-description type 661577 - detailed-description type 15055 - single-round type 40122 - multi-round type 15097 - multi-round responses 45287 Avg. len. of all instruction/responses (in words) 11/17- brief-description type 11/15- detailed-description type 9/82 10/15 - single-round type - multi-round type 10/21

B. Benchmarks and Evaluation

B.1. Discussions about Close-Set Classification

Initially, we consider formatting the close-set zero-shot classification task on ModelNet40[20] as a multiple-choice problem, including indexed candidate category names in the prompt, and prompting our model to select one of the 40 categories given the point cloud as input. However, since our model is not designed for multiple-choice problems but for real-world usage where it can generate any word or phrase as output, we cannot directly parse its response for evaluation. Therefore, we use ChatGPT as a post-processor to select one of the ModelNet40 categories based on the model's answer. In the meantime, we find that including category names in the prompt results in meaningless responses from InstructBLIP[2], which is the model we compare with, making meaningful comparisons challenging. Consequently, we opt for a more generalized prompt, without including the candidate lists in the prompt. This allows us to make balanced comparisons.

Including candidate lists in the prompt, we also tried to calculate the conditional probability of different options given the model's output following [19], but this method did not work well for our model. As our instruction-following training data lacks such scenarios where it's needed to choose from a fixed set of options, our model always produces very low probabilities on these options with biased results. For example, among the options "00" to "39", our model predicts very low probabilities and among these low probabilities, "00" and "39" are the highest most of the time, which leads to biased predictions. Therefore, we choose to use general prompts and utilize ChatGPT for postprocessing. This approach more accurately reflects realworld scenarios where the model is expected to provide natural, free-form responses to diverse, unstructured questions.

B.2. GPT Evaluation Prompts

Close-set zero-shot classification. In this task, we use ChatGPT to post-process the model output by selecting the most probable class index from the 40 ModelNet40 categories. The process is detailed in Tab. 5, where {candidiate_lists} refers to the ModelNet40 category list, and {model_output} refers to the model's response. Chat-GPT is required to directly output the category index, category name, and a short reason for the choice. If the description doesn't clearly refer to any one of the categories, ChatGPT must make an educated guess based on the information provided. If ChatGPT cannot infer, then "-1" is returned and a random index will be chosen as the model's classification prediction. We do not use a system prompt for ChatGPT but directly input the prompt.

Open-vocabulary classification. In this task, we use GPT-4 as an evaluator to classify whether the model's response and the human caption are referring to the same object type. The process is outlined in Tab. 6, where {ground_truth} and {model_output} refer to the human caption and the model's response. We do not require the model's response to match exactly with the human caption, as long as it conveys the same object type. We also directly input the prompt for GPT-4 instead of using a system prompt.

Object captioning. In this task, we utilize GPT-4 as an evaluator to assess model-generated captions against human-generated captions (ground truth) of 3D models. GPT-4 is tasked with identifying aspects mentioned in the human caption and calculating the percentage of these aspects that are either correctly mentioned or partially matched in the model's caption on a scale of 0 to 100, with each aspect contributing equally to the score. The evaluation process is detailed in Tab. 7, where {ground_truth} refers to the human caption, and {model_output} refers to the model's response.

B.3. Human Verification of GPT Evaluation

To verify the effectiveness of using GPT models for evaluation, the first author manually checks the evaluation results of ChatGPT and GPT-4.

In the close-set classification task on ModelNet40, the author finds the following:

- 1. ChatGPT consistently outputs in the desired format, selecting the category or "-1" and providing a reason.
- 2. When the model output clearly refers to or hints at a category with salient information regarding one of the candidate categories, ChatGPT can accurately identify the corresponding category based on the model's output, showing a high degree of consistency with human-

selected options. False negatives or false positives are rare in these cases.

3. If the model output is ambiguous, ChatGPT's selection appears random, aligning with our expectations for handling such cases in classification tasks, because when the model encounters uncertainty or lacks confidence in its identification, random guessing is permissible.

For open-vocabulary classification and object captioning tasks on Objaverse, the author finds that ChatGPT underperforms in identifying the same object concept, acting as a strict judge, and producing more false negatives in classification. It often considers two words or phrases not to refer to the same object type, even when they do. In contrast, GPT-4 demonstrates accurate recognition. After reviewing 50 samples of classification results, the first author has 100% consistency with GPT-4's evaluations. As a result, we opt to use GPT-4 for the open-vocabulary and object captioning tasks on Objaverse. Examples of GPT evaluation can be found in Tab. 5, Tab. 6, and Tab. 7.

B.4. Human Scoring Criteria

Human evaluators were employed to assess captions in the object captioning benchmark. Outputs for the same object from various models were grouped and randomly shuffled, and evaluators independently scored these captions while manually inspecting objects in the Objaverse using the official explorer at https://objaverse.allenai.org/explore.

Scoring criteria. The evaluation process involved assigning correctness scores and hallucination scores following these guidelines:

1. Correctness score.

- Each distinct correct attribute in a model output (*e.g.*, category, color, shape, usage, material) was awarded one point. For example, a black tire correctly identified as a tire and being black would receive two points.
- Partial correctness was graded on a scale of 0 to 1, depending on the degree of accuracy. For instance, if a model output described "a cartoon figure" but the object was specifically a cartoon horse, it would be awarded 0.5 points.

2. Hallucination score.

- Hallucination points were assigned for each incorrect detail in the model output, mirroring the correctness scoring mechanism. For instance, if the model incorrectly described two yellow tires instead of four black ones, it would incur two hallucination points, one for color and one for number.
- Repetitive inaccuracies based on one attribute were not subject to multiple penalties. As an example, erroneously mentioning a black tire when no tire existed would lead to only one hallucination point.
- Penalties were also applied for content that was irrelevant to the object description.

3. General considerations.

- Generic terms like '3D model' or 'image' were disregarded, as were references to black backgrounds or environmental colors, and viewpoints.
- Elements that were indeterminable as either correct or incorrect were not considered in the scoring.
- Within each group of evaluations, a range of scores should be established to differentiate between high and low-quality captions. After the initial scoring, a final review adjusted the scores to ensure a clear distinction between better and worse captions.

Precision score. We also calculate the precision score as the proportion of correct information within the model-generated content as follows:

$$\mathbf{Precision} = \frac{\mathbf{C. Score}}{\mathbf{C. Score} + \mathbf{H. Score}} \times 100\% \qquad (1)$$

where **C. Score** refers to the correctness score and **H. Score** refers to the hallucination score. The precision score in the main paper is reported after summing all **C. Score** and **H. Score** of all samples for robust evaluation.

C. Implementation and Training

Implementation details. We use ULIP-2[21] to pre-train our point cloud encoder (Point-BERT[22]). ULIP-2 is a method for aligning the latent space of the point cloud encoder to that of CLIP[16] through contrastive learning, endowing the encoder with a strong zero-shot capability for 3D object recognition. As the original implementation of ULIP-2 only supports point clouds with spatial coordinates (xyz), we re-train Point-BERT from scratch with color information (xyzrgb), following the same procedure outlined in the ULIP-2 paper. For training Point-BERT, we employ ViT-L/14 trained on DataComp-1B[6] for 12.8B steps and batch size 90k (denoted as 'ViT-L/14datacomp_xl_s13b_b90k') from OpenCLIP[10]. We use point clouds from Cap3D[14], which contains 660K objects. We filter out 3000 objects from this dataset and reserve them for future testing. These 3000 objects are not used during any stage of the entire model training and the 200 objects utilized for our benchmarks are part of these 3000 unseen objects to prevent information leakage.

Training details. All training are conducted on $8 \times 80G$ A100 GPUs with BF16 data type, leveraging flash-attention [3], the AdamW [13] optimizer, and a cosine learning rate scheduler. For the feature alignment stage, we train our model for 3 epochs with a batch size of 128 and a learning rate of 2e-3. For the instruction tuning stage, we train our model for 3 epochs with a batch size of 32 and a learning rate of 2e-5. For efficiency, the 7B model completes the feature alignment and instruction tuning stages in approximately 13.3 and 2.5 hours, respectively, while the 13B model takes around 22.3 and 4.3 hours for the same stages.

D. Win Rate Comparison

In Fig. 3, we present the win rate comparisons of the 7B and 13B model variants across different scores. Note that due to truncation errors in plotting figures, some cumulative rates (win, tie, lose) may not sum to exactly 100%.

The win rate analysis reveals that both the 7B and 13B models not only outperform baselines but also surpass human annotators in terms of correctness scores by a significant margin. This superior performance in correctly identifying object attributes underscores the models' advanced understanding and processing capabilities of 3D objects.

Regarding hallucination, our models exhibit a marked improvement over the baselines, reflecting their better ability to avoid generating incorrect or fabricated details about the objects. This is further corroborated by the precision scores, where our models demonstrate a higher ratio of correct information in their outputs compared to other models. Compared with human annotators, our models show comparable performance in about 50% of samples in terms of hallucination and precision.

However, it must be acknowledged that there is still room for improvement in reducing hallucination rates to match the levels achieved by human annotators. Striving towards the precision demonstrated by human evaluations remains a target for future enhancements.

E. Qualitative Results

In this section, we provide the qualitative results from different datasets of the 13B models for comparison. All samples used were unseen by our models during training.

Results on ModelNet40. Tab. 8 illustrates the classification results from different models on the ModelNet40 dataset. These examples highlight the inherent limitations of image-based models, which depend on suitable views for accurate object identification. Notable challenges include the failure to recognize the guitar in Sample 3, the monitor in Sample 6, and the ambiguity in depth perception leading to misclassifications, such as confusing a chair with a bed in Sample 2, and a bathtub with a bowl in Sample 5. In contrast, PointLLM bypasses these challenges by using point clouds, which provide direct access to object geometry without concerns over ambiguous depth, occlusion, or viewpoint.

Moreover, there are evident hallucination issues with other models. For instance, Point-Bind LLM[7] erroneously describes a person lying on the couch in Sample 1 and someone holding a wine bottle in Sample 4. In comparison, our 13B model consistently provides accurate and realistic classifications. Its superior performance, devoid of such hallucinatory inaccuracies, underscores the model's advanced comprehension of 3D structures and its effectiveness in handling diverse object types.

Results on Objaverse. Due to the limited capability of

Point-Bind LLM in producing meaningful outcomes, it has been excluded from our comparative analysis on Objaverse. It is noteworthy that InstructBLIP also occasionally yields nonsensical results as in Sample 2 of Tab. 9. As depicted by the results, PointLLM consistently generates captions that are both more accurate and detailed compared to other baselines and human annotators. For example, in Sample 2, PointLLM accurately describes the golden brown eyes of an insect, a detail overlooked by human annotators who provide only a generic description, and completely missed by other models failing to identify the object type correctly. Similarly, in Samples 3 and 4, PointLLM offers elaborate descriptions encompassing shape and color without errors, in stark contrast to the simplistic captions from human annotators and erroneous information from other baselines.

Dialogues. Fig. 4 showcases dialogues between PointLLM and a human user, which reveal PointLLM's capacity to understand point clouds' shapes, appearances, functionalities, and more. Notably, our PointLLM is unaffected by occlusion, capable of discerning the car's internal two-seat structure and identifying a logo on the back of a shoe, tasks challenging for image inputs. Furthermore, our model engages with human instructions using common sense and avoids biases, as seen in its refusal to declare a 'best' shoe brand. Collectively, these samples validate PointLLM's proficiency in understanding point clouds and responding to human instructions both accurately and effectively.

F. Text-to-3D Generation

The burgeoning interest in text-to-X generation tasks [1, 11, 15, 17] has led to significant advancements. Notably, [1] demonstrated that text-to-image generation models benefit greatly from training on highly descriptive, generated captions. Leveraging PointLLM's capability to generate detailed and accurate captions for 3D models, we explore its potential in enhancing text-to-3D generation models.

We employed PointLLM-13B to generate captions for Objaverse[4] objects with LVIS[8] labels. The prompt "Describe this 3D model in detail and accurately." was used for caption generation. We trained the text-to-3D generation model from [5] using captions generated by our PointLLM and the Cap3D[14] respectively for comparison.

Qualitative comparisons of the generation results, as illustrated in Fig. 5, reveal that the model trained with PointLLM-generated captions generates objects more closely aligned with text prompts and exhibits more precise detailing. This highlights the advantages of utilizing detailed and accurate captions from our model for text-to-3D generation tasks, pointing towards improved fidelity and coherence in generated 3D objects.



Figure 1. Length distributions of instructions and responses of different types of point-text instruction following data.







(e) Single round-instruction.



(g) Multi round-instruction.



(b) Brief description-response.



(d) Detailed description-response.



(f) Single round-response.



Figure 2. Word distributions of instructions and responses of different types of point-text instruction following data.

Table 2. The instruction list for brief descriptions. An instruction from the list is randomly selected and coupled with a human-written caption from Cap3D[14] to form a brief-description instruction following sample.

- Summarize the 3D point cloud object briefly.
- What kind of object is depicted by this point cloud?
- Provide a short explanation of this 3D structure.
- What does this collection of points represent?
- Offer a succinct summary of this 3D object.
- Can you give a brief overview of this point cloud?
- Characterize the object this point cloud is illustrating.
- Share a brief interpretation of this 3D point cloud.
- Provide an outline of this 3D shape's characteristics.
- What object is this point cloud rendering?
- Deliver a quick description of the object represented here.
- How would you describe the 3D form shown in this point cloud?
- What is the nature of the object this point cloud is representing?
- Present a compact account of this 3D object's key features.
- What can you infer about the object from this point cloud?
- Offer a clear and concise description of this point cloud object.
- How would you summarize this 3D data set?
- Give a brief explanation of the object that this cloud of points forms.
- What kind of structure does this 3D point cloud depict?
- Could you delineate the form indicated by this point cloud?
- Express in brief, what this point cloud is representing.
- Give a quick overview of the object represented by this 3D cloud.
- Convey a summary of the 3D structure represented in this point cloud.
- What kind of object is illustrated by this collection of points?
- Describe the object that this point cloud forms.
- How would you interpret this 3D point cloud?
- Can you briefly outline the shape represented by these points?
- Give a concise interpretation of the 3D data presented here.
- Explain the object this point cloud depicts succinctly.
- Offer a summary of the 3D object illustrated by this cloud.

Table 3. The instruction list for detailed descriptions. An instruction from the list is randomly selected and coupled with a GPT-4 generated caption to form a detailed-description instruction following sample.

- Can you tell me more about this?
- What does this represent?
- Can you describe this in more detail?
- I'm interested in this, can you explain?
- What is this object made of?
- Could you provide more info about this?
- What exactly am I looking at here?
- What is this?
- Could you describe the detailed structure of this?
- This looks interesting, can you expand on it?
- Can you explain more about this form?
- What can you tell me about the shape of this object?
- Could you delve deeper into this?
- I want to know more about this, can you help?
- Can you walk me through the details of this object?
- Can you provide a comprehensive account of this object?
- Offer a detailed interpretation of this point cloud.
- Please elucidate on the characteristics of this form.
- Could you provide an in-depth description of this structure?
- What does this cloud represent in its entirety?
- Elaborate on the details of this point cloud, please.
- Kindly furnish me with more information about this object.
- Please expand on the intricate structure of this form.
- Provide a meticulous explanation of what these points represent.
- I request a detailed breakdown of this structure.
- Give a thorough rundown of this point cloud.
- Can you offer a complete analysis of this object?
- I would like a comprehensive explanation of this form.
- Please detail the specific features of this point cloud.
- Could you elaborate extensively on what this represents?

Table 4. An example of data generation with GPT-4. The input consists of a human-written caption provided by Cap3D[14], guided by a system prompt to analyze the existing caption based on the general knowledge of 3D objects. The outputs include an expanded detailed caption, three single-round conversations, and one multi-round conversation.

Uid	4887da0aab51406dab3c5cb69ec82404	
System Prompt	 Analyze the 3D object model from the given caption: 1. Write a new detailed caption by describing it in 50-100 words, identifying its type, appearance, and any other inferred information such as functionalities, usages, daily-life knowledge, etc. Exclude any uncertain details. 2. Generate 3 single-round Q&As regarding diverse aspects of the object based on the provided captions and your new captions. 3. Construct 1 set of 3-round Q&As, ensuring logical relevance between the questions and answers. Questions should be different from those in "2". 	
	Format your response as: ```json { "caption": "description", "single_conversation": [{"Q": "Q", "A": "A"} x 3], "multi_conversation": [{"Q1": "Q", "A1": "A", "Q2": "Q", "A2": "A", "Q3": "Q", "A3": "A"} x 1]	
Input	A female doll with long back cream hair and with black high boots.	
Output	<pre>{ "caption": "The 3D object model represents a detailed doll figure of a female character with long, cream- colored hair cascading down her back. It is dressed in fashionable attire with black high boots being one of its highlighted features. The doll, resembling those used in play or as collectible items, exhibits a high degree of intricate detailing, showcasing its fashionable outfit and engaging features, making it appealing for users, especially ones intrigued by doll collection or imaginative play.", "single_conversation": [{"Q": "What color is the doll's hair?", "A": "The doll's hair is cream-colored."}, {"Q": "What is specific about the doll's footwear?", "A": "The specific footwear feature of the doll is its black high boots."}, {"Q": "What are possible usages of this doll?", "A": "Possible usages for this doll might include being part of a toy collection, used for imaginative play, or as a detailed model for hobbyists or artists."}, </pre>	
	"multi_conversation": [{"Q1": "What makes this doll distinguishable?", "A1": "The doll is distinguishable by its long cream- colored hair cascading down its back and its outfit featuring black high boots.", "Q2": "Could this doll be part of a collection or series?", "A2": "Yes, based on the intricate detailing and appearance, it's possible that the doll could be part of a collection or series.", "Q3": "Who would be the likely audience or users of this doll?", "A3": "The likely users of this doll would be individuals who enjoy doll collecting, children or adults for imaginative play, or even hobb yists and artists who may use it as a model."}]	

Table 5. The prompt and examples of ChatGPT in close-set zero-shot classification. ChatGPT post-processes the model output and selects the most probable class index from the available 40 categories, even if the description is vague or ambiguous. The blue placeholders {candidate_lists} and {model_output} refer to the ModelNet40 category list and the model's response, respectively.

Prompt	Given the following free-form description of a 3D object, please determine the most probable class index from the fol- lowing 40 available categories, even if the description doesn't clearly refer to any one of them. Make your best-educated guess based on the information provided. If the description already contains a valid index, then the index should be selected. If it contains more than one valid index, then randomly select one index (specify your reason). If there is no valid index and it cannot be inferred from the information, return "-1#NA#Cannot infer". Categories: 0: airplane 1: bathtub {candidate_lists} 38: wardrobe 39: xbox
	Examples: Input: This is a 3D object model of a cartoon white truck. Output: 7#car#Closest match to "car"in categories
	Input: A green leaf in a flower pot. Output: 26#plant#The primary subject "leaf" directly indicates a plant.
	Input: It's difficult to determine the exact type of this object due to insufficient details. But it seems to be like a piece of furniture. Output: 33#table#Randomly select one kind of furniture from the list.
	Input: I cannot determine the specific type of the object without additional information or context. Output: -1#NA#Cannot infer.
	Now analyze the following: Input: {model_output} Output:
Example 1	Input: This 3D model represents a war plane, typically used for combat and military purposes. It is characterized by a large, dominating hue of military gray. The detailing on the model suggests it's a fighter jet, hinting at its usage in air combats. Its posture seems ready for flight, showcasing a typical attribute of war planes - preparedness for battle. This could be used in game design or other virtual reality applications that require detailed, lifelike renderings of objects. Output: 0#airplane#The description explicitly mentions "war plane" and "fighter jet".
Example 2	Input: The 3D model object is a tall, cylindrical-shaped clay container with a flat, smooth top and a slightly wider base. The cylindrical form and the structure suggest that it's possibly used for storing liquids or semi-liquids due to its wide top and flat base allowing easy pouring and stability. Its material and structure indicate it could be a common household item, like a vase or a jug. Output: 37#vase#The description includes key characteristics of a vase (tall, cylindrical shape, wide top, flat base).
Example 3	Input: This is a 3D model of a sleek, black rectangular unit with its sides elegantly curved. On one of its sides, there is a distinctive circle inscribed, adding a unique character to the model. The object's form and design suggest that it can be a piece of technology, like a gadget or a home appliance, with the circle possibly indicating a button or a sensor. Its usage, however, cannot be ascertained without specific context. Output: 22#monitor#The description suggests a sleek, black rectangular unit which can be a monitor. The inscribed circle may indicate a power button or a sensor on the monitor.

Table 6. **The prompt and examples of GPT-4 in open-vocabulary classification.** GPT-4 needs to analyze two sentences to determine if they refer to the same general object or concept, focusing on the type of object, not attributes such as color, size, or shape. The placeholders {ground_truth} and {model_output} refer to the human caption and the model's response, respectively.

Prompt	Analyze two sentences and determine if they're referring to the same general object or concept, focusing on the type of object, not attributes such as color, size, or shape. Respond with "T" if they refer to the same thing and "F" if not. Also, provide a brief rationale (no more than 20 words) for your judgment. Example: Input: 1. Spiral staircase that goes from a ground floor. 2. This is a 3D model of wooden stairs in light brown Output: T#Both refer to a staircase.		
	Now, analyze the following: Input: 1. {ground_truth} 2. {model_output} Output:		
Example 1	Input: 1. A black and brown colored gun. 2. The 3D object is a representation of a futuristic, high-tech gun crafted from a glossy black material. Distinctive features include its metallic handrail, giving an impression of a robust mechanized design. The gun, possibly used in a sci-fi or futuristic setting, denotes advanced technology and might include functionalities such as voice recognition, aiming systems, or biometric triggers. Output: T#Both refer to a gun.		
Example 2	Input: 1. A yellow and white fish with black stripes and fins. 2. This is a 3D model of a vibrant, polka-dotted toy fish that is predominantly orange on the body, shifting to white on the belly. The toy has dark brown spots that enhance its appearance, potentially mimicking the natural patterns found on real-life fish. It's an ideal object for educational purposes, helping to introduce children to marine life, as well as serving as a playful item in a playroom or nursery. Output: T#Both refer to a fish.		
Example 3	Input: 1. A white cartoon scorpion with eight legs. 2. This is a 3D object model representing a cartoon version of a rare type of spider. The entire model is rendered in white, which highlights its unique and exaggerated characteristics such as multiple legs and a funnel-like body. Its cartoonish appeal makes it more appealing to a younger audience, and it could possibly be used in animations or educational materials to teach children about spiders in a less intimidating way. Output: F#One is a scorpion and the other is a spider.		

Table 7. The prompt and examples of GPT-4 in object captioning. GPT-4 evaluates the model's response by identifying aspects mentioned in the human caption and calculating the percentage of aspects that are correctly or partially matched in the model's caption. The placeholders {ground_truth} and {model_output} refer to the human caption and the model's response, respectively.

Prompt	Evaluate a model-generated caption against a human-generated caption (ground truth) for a 3D model. Identify the aspects mentioned in the human caption and calculate the percentage of these aspects correctly mentioned or partially matched in the model caption. Score from 0 to 100, where each aspect contributes equally to the score. Consider similar concepts for a partial score.
	Provide your score (0-100) and a short justification (less than 15 words) in the format of "score#reason"
	Example:
	Model: This is a 3D model of a small, cartoon-like robot. It has a spherical body and is covered in a layer of white dust. Output: 50#mention white; skeleton and robot have similar appearence.
	Now score the following: Human: {ground_truth} Model: {model_output}
 Example 1	Output: Human: A white presentation of a planted trace plantation ragion
Example 1	Model: The model is a representation of a table saw with a simple, four-legged design. Output: 0#No aspects from the human caption are mentioned in the model's.
Example 2	Human: Private jet Model: This is a 3D model of a cartoon-style airplane. Output: 50#mentioned airplane, but missed private jet aspect.
Example 3	Human: A cartoon look like a tree in 3d Model: The 3D model is an animated depiction of a tree with gray bark, characterized by its lifelike details and textured surface that mimic the realistic grain of bark. Output: 100#mentioned tree, 3D and artistic/cartoonish aspect.



Figure 3. Win rate comparisons of 7B and 13B models across various scores. Our models surpass baselines and human annotators in correctness scores, while exhibiting lower hallucination rates and superior precision compared to various baselines.

Table 8. **Qualitative results on ModelNet40.** The first image in each sample serves as the input for image-based models, with additional point cloud views provided for reference. PointLLM consistently and accurately identifies object types, whereas other models struggle with correct identification or produce hallucinated, incorrect content.

Samples 1, 2		A A A A A A A A A A A A A A A A A A A
Ground Truth	Sofa	Bed
Prompt InstructBLIP[2]	What is this? sofa	This is an object of person cutting meat into small pieces with a large knife
LLaVA[12] Point-Bind LLM[7]	The image is a black and white drawing of a couch. This is a drawing of a person laying on a couch, with a dog nearby	The image is a black and white depiction of a chair. This is an object of art.
PointLLM	This 3D object is a comfortable, black, three- seater sofa.	This is a model of a neatly made bed, primarily composed of a solid black mattress.
Samples 3, 4		
Ground Truth	Guitar	Bottle
Prompt InstructBLIP[2] LLaVA[12] Point-Bind LLM[7] PointLLM	What is this? telescope The image is a close-up view of a black ink pen sit- ting on a white surface. This is a bird flying in the sky. This 3D model represents a black electric guitar equipped with a distinctive headstock.	This is an object of wine bottle This image features a black and white depiction of a bottle. This is an object of a person holding a bottle of wine. This is a 3D model of a bottle, possibly intended for liquids.
Samples 5, 6		
Ground Truth	Bathtub	Monitor
Prompt InstructBLIP[2] LLaVA[12]	What is this? table The image features a black and white, close-up view of a grater, which is an object used for grating cheese	This is an object of the foot The image features a small table with a black base and a white or light-colored top.
Point-Bind LLM[7] PointLLM	This is a bowl or a vase, which is oval-shaped and has a smooth surface. The 3D object model represents a realistic, dark grey hathtub.	This is an object of art, a sculpture, or a decorative piece. This is a 3D model of a desktop computer monitor.

Table 9. **Qualitative results on Objaverse.** The first image in each sample serves as the input for image-based models, with additional point cloud views provided for reference. PointLLM can generates more accurate and detailed captions for 3D objects than other baseline models and human annotators.

Samples 1, 2		State State
Uid	0ea33b6617174530b97d6b7a92c275fb	de8ec2a724f14fc4b54624512f80f13e
Prompt Human InstructBLIP[2] LLaVA[12]	What is this? A cartoon green and red like a fruit an appleavatar 3d model This is a computer-generated image (CGI) or a 3D model of an apple with a leaf on it.	This is an object of A black insect 26 This is an image of a large, close-up, and dark- colored insect, possibly a big cockroach, with long antennae.
3D-LLM[9] PointLLM	A 3D model of a red apple. This is a 3D model of a unique apple, distinctively adorned with a single, vibrant green leaf at the top.	A small, black spider with a long tail. This 3D model depicts a realistic, jet-black insect with a pair of striking, golden brown eyes.
Samples 3, 4		
Uid	059a7936ed89419ba9eae3153753ae86	82413f74efb84bc3b872a7a8f26cca81
Prompt	Caption this 3D model in detail.	Caption this 3D model in detail.
InstructBLIP[2]	There is a cute blue and white pony in a still, looking	There is a fish hanging on a string in front of a black
LLaVA[12]	cheerful and cheerful. It's standing next to a larger blue-colored object, potentially a barrier or struc- ture, which provides the backdrop for this scene. The overall image is designed to create a fun, whimsical setting for the pony. The image features a 3D model of a cute, colorful cartoon unicorn with a blue and white body and a hint of pink. The unicorn has a magical aura and is set against a black background. Its long, flow- ing mane and tail add to its whimsical charm. The unicorn appears to be in motion, possibly flying or prancing, as it captures the viewer's imagination.	background. The fish is swimming in the middle of the image and is a little bit up from the bottom of the screen. The fish has a red spots on the left and right sides, and is positioned slightly towards the top of the screen. The image features a 3D model of a fish floating in a dark, empty space. The fish is positioned towards the left side of the image and appears to be swimming upwards. The fish is large and occupies a significant portion of the image, giving the impression of a life- like, computer-generated representation. The dark background and lack of other objects in the scene emphasize the fish's presence and create a sense of depth and dimension.
3D-LLM[9]	A 3D model of a purple and pink pony with long hair, standing on its hind legs.	A small fish with a long tail is visible in the image. It has no distinguishing features or markings, and its color appears to be brown.
PointLLM	A 3D model of a vibrant blue cartoon pony, char- acterized by its large white eyes, two wings, and a distinctive horn. The creature's overall design exudes a playful and friendly aura, making it ap- propriate for children's animations or toy pro- duction. The noticeably large white eyes and the unique wing and horn features endow the pony with an element of fantasy, while its blue body color gives it an overall appealing look.	The 3D model is of a long fish, illustrating a strik- ing color combination of red, yellow, and grey. It's characterized by a slender tail. The yellow color is mostly seen on the body of the fish, fading into a subtler, grayish hue towards the fins and tail. The model depicts a typical long fish species, often found in the ocean, and can be used in a variety of digital applications, like animation, games, or virtual reality.



Figure 4. **Dialogues between PointLLM and a human user.** The dialogues show PointLLM's ability to understand point clouds' shapes, appearances, functionalities, etc. PointLLM's resilience to occlusion enables it to perceive detailed features such as the interior structure of a car or a logo on a shoe, which are challenging for image inputs. Additionally, PointLLM demonstrates abilities to respond to human instructions with common sense, avoiding biases.



(a) PointLLM.

(b) Cap3D.

Text prompt: "a chimpanzee stirring a bubbling purple potion in a cauldron."



(c) PointLLM.

(d) **Cap3D.**





(e) PointLLM.

(f) Cap3D.

Text prompt: "a confused beagle sitting at a desk working on homework."

Figure 5. Text-to-3D generation results of models trained with different captions. The Model trained with PointLLM-generated captions generates objects more closely aligned with text prompts and exhibits more precise detailing.

References

- [1] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. https://cdn.openai.com/ papers/dall-e-3.pdf, 2023. 4
- [2] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards generalpurpose vision-language models with instruction tuning. *arXiv:2305.06500*, 2023. 1, 14, 15
- [3] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022. 3
- [4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In CVPR, 2023. 4
- [5] Lihe Ding, Shaocong Dong, Zhanpeng Huang, Zibin Wang, Yiyuan Zhang, Kaixiong Gong, Dan Xu, and Tianfan Xue. Text-to-3d generation with bidirectional diffusion using both 2d and 3d priors. arXiv:2312.04963, 2023. 4
- [6] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. arXiv:2304.14108, 2023. 3
- [7] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, et al. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023. 3, 14
- [8] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 4
- [9] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *NeurIPS*, 2023. 15
- [10] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 3
- [11] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 4
- [12] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. arXiv:2304.08485, 2023. 14, 15
- [13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. **3**
- [14] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. arXiv:2306.07279, 2023. 1, 3, 4, 7, 9

- [15] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv:2212.08751*, 2022. 4
- [16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. 2021. 3
- [17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In CVPR, 2022. 4
- [18] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. arXiv:2212.10560, 2022. 1
- [19] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2021. 1
- [20] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 1
- [21] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. arXiv:2305.08275, 2023. 3
- [22] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*, 2022. 3