Appendix

Contents

A. Supplementary Material Introduction

In this supplementary material, we extend the discussions presented in the main conference paper. Appendix B provides a more in-depth exploration of related work, focusing on defining the scope of large language models family and examining the developments in point-text multimodal approaches. Appendix C supplements more details about the data annotation engine Pyramid-XL and the diffusion architecture. Moving to Appendix D, we expand on the superiority of our benchmark. Initially, we introduce examples from our ObjaverseXL-LVIS QA 1K dataset, which includes concise QAs for evaluation and long QAs for instructive tuning. Then, we show more 3D generation failure cases where GPT4Point can figure it out while 2D VLM can not underscore the necessity and relevance of our 3D point-text benchmark. Finally, in Appendix E, we give more qualitative results of Point-text inference tasks, including caption and QA tasks and Controllable point diffusion.

B. Additional Related Work

In this section, we provide detailed insights into related work. Appendix B.1 classifies key concepts of large language models, including LLMs, MLLMs, and VLMs. Appendix B.2 presents the evolution of point-text multimodal models through an illustrative flowchart.

B.1. The Family of LLMs and MLLMs

Although the concepts related to large language models are already familiar, we still wish to detail these concepts here. We briefly introduce some families of LLMs and MLLMs. First are the LLMs based on the Transformer architecture, such as ChatGPT [16] and GPT-4 [17]. Currently, there

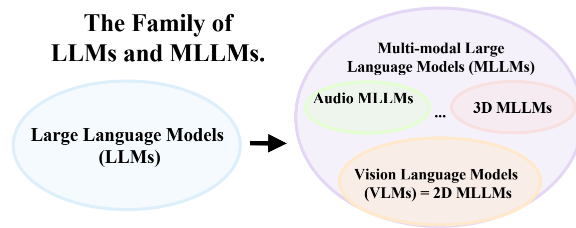


Figure S1. **The Family of LLMs and MLLMs.**

are several open-source, deployable models [4, 20, 23, 25]. They exhibit strong comprehension and reasoning abilities after extensive pre-training on a vast corpus. Multimodal Large Models (MLLMs) aim to enable LLMs to understand information in other modalities. The fundamental approach involves retrieving text features with other modality features. Among them, image-text multimodal large models, also known as 2D MLLMs or Visual Language Models (VLMs), stand out due to the abundant image-text pairs and strong image backbones provided by computer vision [7, 12, 13]. Beyond images, there are other modalities, such as Audio MLLMs [9] that combine with the audio modality and Video MLLMs with the video modality [2]. In the 3D domain, some existing work, like 3D-LLM [8], utilizes 2D image features combined with depth projections to generate 3D features. We propose a unified text understanding and generation model based on point clouds and develop a real 3D MLLM.

B.2. The development of Point-text Multimodal

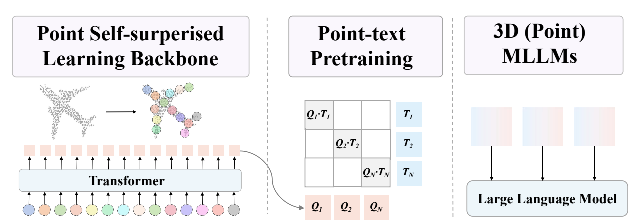


Figure S2. **The development of Point-text Multimodal.**

This section delves into the evolution of point-text multimodal models for single objects.

- **Backbone Development:** Like texts and images, point clouds undergo self-supervised training for a strong backbone. PointBert exemplifies this, dividing point clouds into patches and reconstructing masked patches via a Transformer-based backbone [24].
- **Text Modality Alignment:** Inspired by CLIP [18], this phase aligns point patches with text features, enhancing the backbone's processing of textual information.
- **3D MLLMs Integration:** Following the alignment, point features are integrated into LLMs, similar to approaches in VLMs, enabling LLMs to understand point data.

C. Additional Method

Here, we provide additional information on our method. We first give more details about the data text annotation engine Pyramid-XL in Appendix C.1. And then, in Appendix C.2 about the model architecture, we give the details about the point diffusion branch.

C.1. Pyramid-XL: Data Annotation Engine

First, we introduce the approach to acquire point clouds from Objaverse-XL [5]. Then, we introduce the cost and prompts of our data annotation engine Pyramid-XL. Finally, we give more qualitative results that finetune the Point-E [15] by our Pyramid-XL level 3 dense captions.

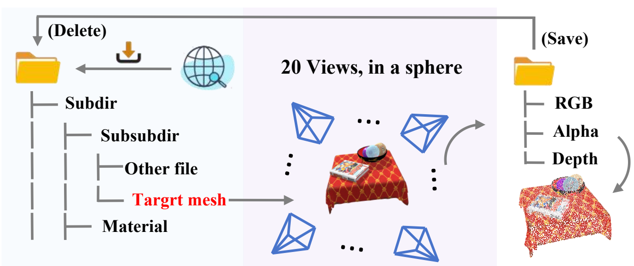


Figure S3. **Acquire Data Pipeline from Objaverse-XL [5].**

Acquire data from Objaverse-XL. Here, we detail our processing approach for the Objaverse-XL dataset [5]. It has 10M objects and is the extension of Objaverse-1.0 [6], which only has 800K 3D objects. Objaverse-XL offers only unprocessed downloads for its 3D objects, most of which originate from sources like GitHub. Downloading these mesh files necessitates completing the project, as materials and related components are often stored in separate directories. Downloading the raw dataset in this format is impractical due to excessive memory requirements, with an average project consuming about 1GB of space. Therefore, we render object images and clear the cache upon completion to manage space. We render 20 random views of each object, capturing the RGB, alpha values, and depth, which are then used to generate point clouds. In addition to the 780K objects from Objaverse-1.0, we rendered an additional 220K from Objaverse-XL, totaling 1M objects.

Dataset	Num Obj	Data Type	Cost/K (GPU + GPT)
Level 1	1M	Single-View Caption	\$0.47 + \$0
Level 2 (GPT-4)	660K	Multi-View Caption	\$4.17 + \$4.18*
Level 2 (ChatGPT)	660K	Multi-View Caption	\$4.17 + \$0.14*
Level 3	70K	QA, Detailed Caption	\$1.64 + \$0

Table S1. **Comparing Costs across Different Dataset Levels.** Costs are calculated based on generating annotation for 1K objects. * is directly from Cap3D [14]. As levels increase, the cost rises, indicating larger scales for lower-level datasets.

The cost of the Pyramid-XL. We now focus on the cost analysis of our data annotation engine, detailed in Tab. S1. The primary costs, detailed under the '1K Cost' column, include GPU resources on the left and GPT API usage on the right. We use the same GPU settings as Cap3D [14], employing A40s on an identical cloud platform. Given GPUs' parallel processing, costs are equal for single or multiple units. For simplicity, we calculate usage time by assuming a single GPU. For Level 1, we use BLIP-2 [12] to generate one short caption for one object. It needs 0.074 hours and costs $0.074h \times \$1.28/h = \$0.095$. For Level 2, the cost is the same as the Cap3D [14]. The GPU resource fees include BLIP-2 [12] and CLIP [18]. BLIP-2 generates eight views for each object, each with five captions, so the fee is $\$0.095 \times 8 \times 5 = \$3.76$. Moreover, the CLIP uses 0.3h and costs $0.3h \times \$1.28/h = \$0.38$. All GPU resource fee is $\$3.76 + \$0.38 = \$4.17$. For the GPT API fee, it costs $\$0.03/1k$ tokens and needs 139.3 tokens for each object, and the total cost is $\$139.3/1000k \times \$0.03/1k \times 1000 = \$4.18$. For Level 3, We use the open-source Visual Language Model (VLM) Qwen-VL [1] for processing the final captions. It needs 1.28h for the CLIP filter and Qwen-VL generation captions, so the cost is $1.28h \times \$1.28/h = \$1.64$.

We can observe that Level 2 captions account for most of the costs, primarily due to GPT usage fees. Our findings show that using GPT-4 for text-based multi-view caption synthesis does not substantially outperform ChatGPT. Furthermore, we can eliminate API call expenses by utilizing open-source Large Language Models (LLMs). The other significant cost is the GPU resources, as it uses BLIP-2 to generate five captions for each view, which can lead to redundancy in information. We can reduce the number of captions and even the number of views for each view.

The prompts of the Pyramid-XL. We present the prompt part of the Pyramid-XL data text annotation engine, as illustrated in Fig. S6 and Fig. S7. We primarily illustrate how to construct GPT-based Level 2 captions, ChatCaptioner-based Level 3 short QA pairs, and MLLM-based Level 3 instruction captions and long QA pairs.

For Level 2 captions, we use Level 1 captions of rendered images from 6 views. Through carefully designed prompts, we integrate captions from the six captions to obtain a comprehensive and relatively accurate caption with fewer than 30 words. In our paper, we use GPT-4 to get the comprehensive caption, but we find that ChatGPT can be replaced by GPT-4 to generate Level 2 captions and reduce the cost.

For Level 3 short QA, we follow the approach outlined in ChatCaptioner [26]. We use ChatGPT or other LLMs (we choose Vicuna-7B [4]) as the questioner and BLIP-2 [12] as the answerer. By providing appropriate instructions and context (Level 2 caption) to the LLM and BLIP-2, we observe that LLM generates diverse questions that include color, type, material, purpose, and more. Also, BLIP-

2 tends to output concise answers without restricting the number of words. These form the basis for our Objaverse-XL short QA dataset.

For Level 3 dense captions, we use the Level 2 caption as context, feed the rendering image that best matches the context into MLLM, and input suitable instructions. Due to high-quality conversational performance and cost-effectiveness, we choose the Qwen-VL [1] model to generate. The construction method for Level 3 instruction (long) QA pairs is similar to the above steps, with the critical difference in the instruction variation.

The effectiveness of Pyramid-XL Level 3 caption. We use dense captions from Level 3 of Pyramid-XL to finetune Point-E and compare the results with those of Cap3D, as shown in Fig. S11. Ours significantly outperforms Cap3D's captions, demonstrating the precision of our captions.

C.2. Point Diffusion Architecture

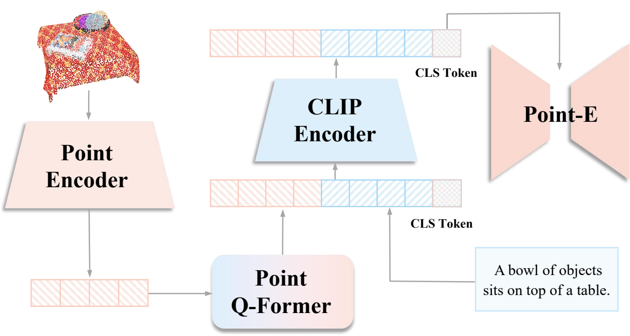


Figure S4. **Point Diffusion with GPT4Point.**

There are some explorations into controllable text-to-3D work [10, 21, 22]. However, we are attempting to combine understanding and controllable 3D generation. Here, we offer an in-depth look at the Diffusion branch's structure in Stage 2, illustrated in Fig. S4. Initially, the point cloud undergoes processing via the Point Encoder (Backbone) and Point Q-Former, yielding Q-Former Tokens. For text, instead of Point Q-Former's text tokenizer, we utilize Point-E's CLIP tokenizer. The resulting text tokens are then concatenated with the Q-Former Tokens. Subsequently, the CLS token from the Text Token is fed into Point E. The concatenation method in GPT4Point differs notably from BLIP-Diffusion [11]. In BLIP-Diffusion, Q-Former Tokens are inserted between the CLS and input tokens. In contrast, GPT4Point appends Q-Former Tokens directly to the text token sequence, allowing the CLS token to integrate geometric and color information, which is crucial for guiding the 3D generation.

D. Additional Benchmark

In this section, we mainly introduce some additional content about the benchmark. In Appendix D.1, we give more examples of the ObjaverseXL QA dataset. Note that the short QA dataset is evaluated based on the accuracy metric. Then, in Appendix D.2, we show more qualitative results about Generation Failure Cases, which can not be recognized by 2D VLMs through a single view but are judged by our GPT4Point.

D.1. Objaverse-XL QA Dataset

Short QA Dataset In this section, we mainly introduce some additional content about the benchmark. In Appendix D.1, we give more examples of the ObjaverseXL QA dataset. Note that the short QA dataset is evaluated based on the accuracy metric. Then, in Appendix D.2, we show more qualitative results about Generation Failure Cases, which can not be recognized by 2D VLMs through a single view but are judged by our GPT4Point.

Long (Instruction) QA Dataset The long (Instruction) QA dataset is used to fine-tune-finetuning the model to enhance the model's conversational capabilities significantly. We impose length constraints on prompts, requiring approximately 50 words for answers to dense caption questions and at least ten words for other questions. As illustrated in Fig. S8, we constructed a Long (Instruction) QA dataset for 70K objects, comprising 344,996 QA pairs. Among these, 69K data are used for finetuning, while the remaining 1K are reserved for testing. This aims to encourage LLMs to generate long and more comprehensive results.

D.2. Anomalous Objects: Generation Failure Cases

In this section, we will demonstrate more qualitative results to show the failure case, which can not be recognized by 2D VLMs through a single view but can be judged by our GPT4Point. This section mainly shows the failure cases produced by the state-of-the-art text-to-3D generation methods like Dream-Gaussian [19] and Fantasia3d [3]. Due to technical constraints, these models will likely generate 3D objects with multi-heads or multi-bodies. If provided with render images from only a single perspective, 2D VLMs [1, 4], and even humans, in most cases, may make incorrect judgments, as illustrated in the upper part of Fig. S9. This hinders the assessment of 3D object generation. However, our GPT4Point provides a better solution to this issue. More examples are showcased in Fig. S9.

E. Additional Experiments

In this supplementary section, we elaborate on the experimental details. Initially, we present a table in Appendix E.1, enumerating all hyperparameters. Subsequently, we furnish additional qualitative outcomes. Specifically, Appendix E.2 illustrates textual reference tasks, such as 3D object point captioning and QA, while Appendix E.3 showcases our findings regarding point diffusion.

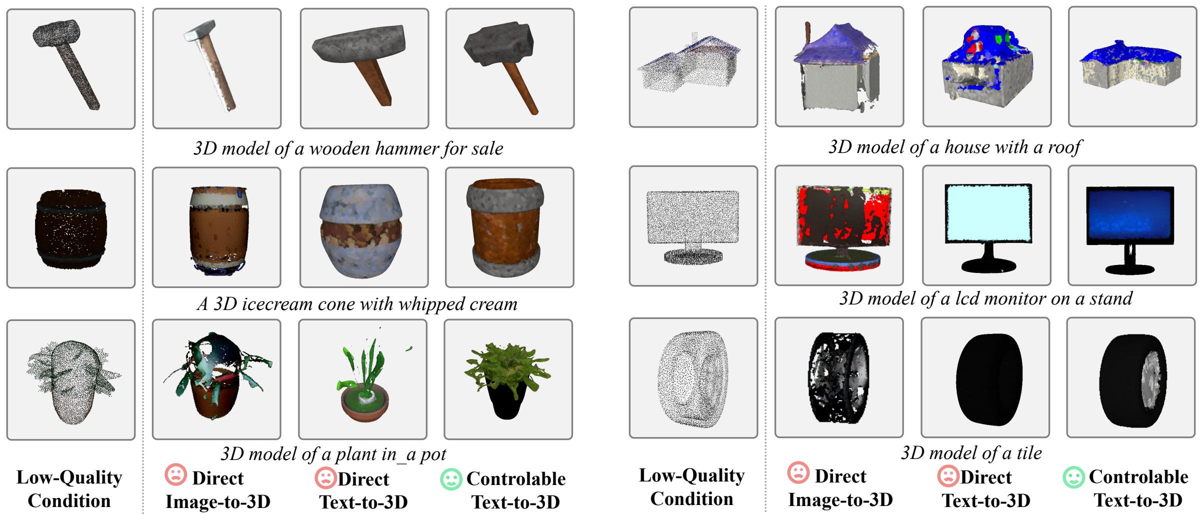


Figure S5. **Point Diffusion Results: our controllable text-to-3D.** Given a low-quality point cloud prior, it can generate outcomes superior to direct text-to-3D and image-to-3D methods and more closely align with the low-quality priors, demonstrating controllability.

Hyperparameters	Value/Type
batchsize	32
training epochs	10
optimizer	AdamW
init lr	1e-4
min lr	1e-5
warmup lr	1e-6
weight decay	0.05
lr schedule	cosine annealing
warmup type	linear
warmup iters	5000
Point size	8192
Q-Former queries	32

Table S2. **Training settings and hyperparameters for Stage1.**

E.1. Training Details

We detail the hyperparameters of GPT4Point, mainly mirroring those used in BLIP-2 [12] during the training stage. These parameters are maintained for Stage1: Point-text alignment, and the LLM branch in Stage2. Tab. S2 lists them. The parameters for the LLM branch in Stage 2 are almost identical to those of Stage 1, except for the warmup iterations, which changed from 5K to 2K. For BLIP-2, finetuning is performed on a smaller dataset and subtasks after pretraining on multiple datasets. Additionally, different image backbones were used in the pretraining and finetuning phases. However, in our GPT4Point, we only use the training stage in the BLIP-2, and all tasks are evaluated by zero-shot. We need to make the learning rate very small for the diffusion branch because we only train the fully connected layers here. The init, min, and warmup learning rates are 1e-7, 0, and 1e-8, and we only train one epoch.

E.2. Point-text Captions and QA Demos

In this section, we delve deeper into the qualitative assessment of GPT4Point's performance. Specifically, we explore additional examples showcasing its proficiency in understanding point clouds and engaging in meaningful conversations with users. Referencing Fig. S10, we present specific instances where GPT4Point accurately generates text descriptions corresponding to point cloud inputs, demonstrating its robustness and effectiveness in point-text tasks.

E.3. Point Diffusion Results

In Fig. S5, additional qualitative outcomes of the point diffusion process by GPT4Point are depicted. Notably, GPT4Point adeptly directs the text-to-3D conversion, yielding outputs characterized by enhanced fidelity in both color representation and geometric form.

Content	Figure
Appendix C.1: 2 types of short annotation prompts	Fig. S6
Appendix C.1: 2 types of long annotation prompts	Fig. S7
Appendix C.1: Level 3 caption finetune Point-E	Fig. S11
Appendix D.1: ObjaverseXL-LVIS QA Data	Fig. S8
Appendix D.2: Generation Failure Cases	Fig. S9
Appendix E.2: Point-text Captions and QA	Fig. S10
Appendix E.3: Point Diffusion Results	Fig. S5

Table S3. **Chapter-Experiment Result Image Correspondence.**

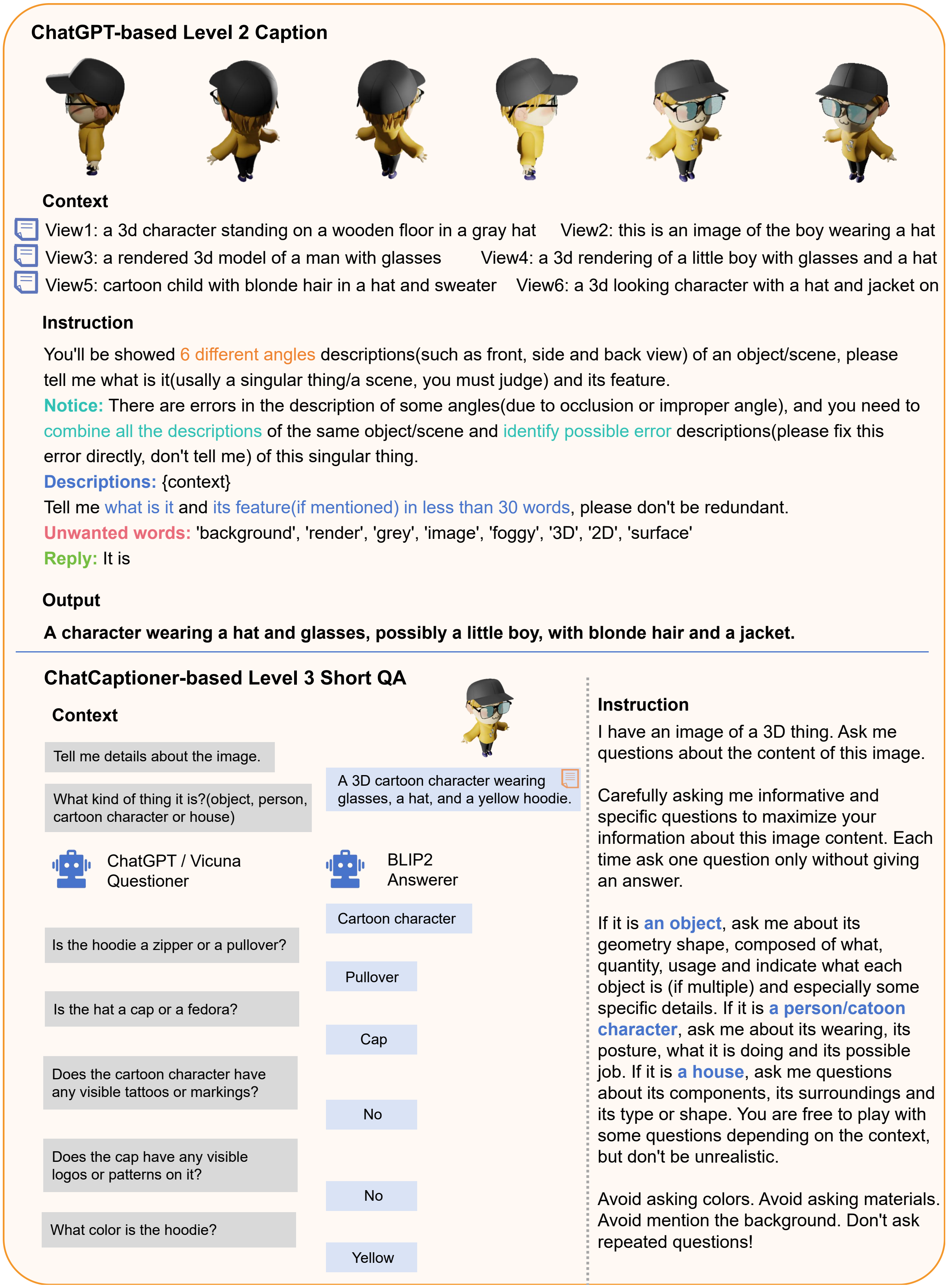
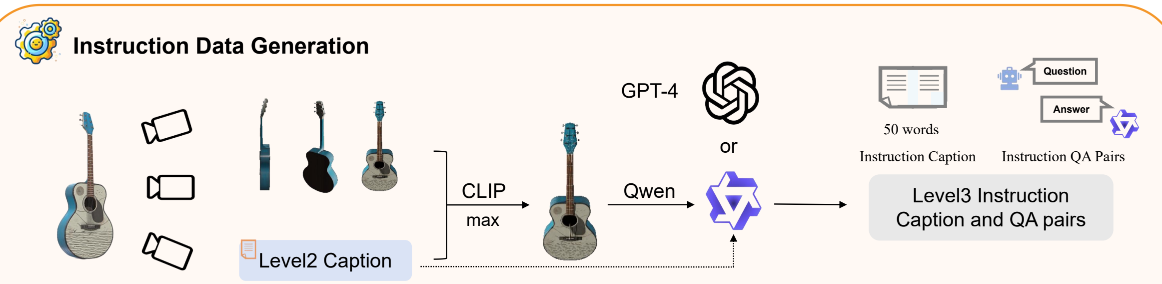


Figure S6. **Prompts of Level 2 Caption and Level 3 Short Q&A in Pyramid-XL.** We present the construction process of GPT-based Level 2 Caption and ChatCaptioner-based Level 3 Short Q&A, along with the prompts utilized, consisting of context and instruction.



MLLM-based Level 3 Instruction Caption

Context

Blue and white acoustic guitar 3D model, royalty-free.

Instruction

Analyze the 3D object model from the image and the given caption: "{context}"

Write a new `detailed` caption by describing it in more words*(no less than 50 words)*, **identifying its type, appearance**, and any other inferred information such as **functionalities, usages, daily-life knowledge**, etc. Exclude any uncertain details. Use **rich language descriptions**.

Output

The 3D object model is a **blue and white acoustic guitar**. It has a classic design with a **wooden body and a rosewood fretboard**. The guitar is painted in a distressed style, giving it a vintage look. This guitar is a high-quality, royalty-free model that can be used for various purposes, **such as in music videos, advertising, or as a decorative item**.

MLLM-based Level 3 Instruction QA pairs

Context

Blue and white acoustic guitar 3D model, royalty-free.

Instruction

Analyze the 3D object model from the image and the given caption: "{context}"
Generate a set of 1-round Q&As(both question and answer pairs) **regarding diverse aspects of the object based on the image and provided captions**. Question begin with 'Q:'. Answer begin with 'A:'. Don't mention size.

Continue to generate 1 single-round Q&As(both question and answer pairs) regarding diverse aspects of the object based on the image. Don't mention size. Different aspects from the previous questions.

Continue to generate 1 single-round Q&As(both question and answer pairs) regarding diverse aspects of the object based on the image. Don't mention size. Different aspects from the previous questions. The answer should be more than 10 words, but should not be too redundant with the question.
……

Output

What is the **purpose** of the acoustic guitar?

The acoustic guitar is a musical instrument that is used to create music by plucking the strings. It is commonly used in a variety of musical genres, including folk, rock, and classical music.

What is the **color** of the guitar?

What is the color of the guitar?

What is the **material** of the guitar?

The guitar is made of wood.

Figure S7. **Prompts of MLLM-based Level 3 Instruction Caption and QA pairs in Pyramid-XL.** The top part details the process of constructing the dataset, while below are the specific instructions provided to the MLLM (Qwen-VL[xx]) and the model output.

Figure S8. **Objaverse-XL QA dataset.** The upper section of the dataset features short Q&A samples, and the lower part includes long Q&A samples covering color, shape, type, material, and purpose. The short Q&A dataset evaluates performance, while the long Q&A is for finetuning, promoting more detailed language generation, and promoting understanding and conversational capabilities.

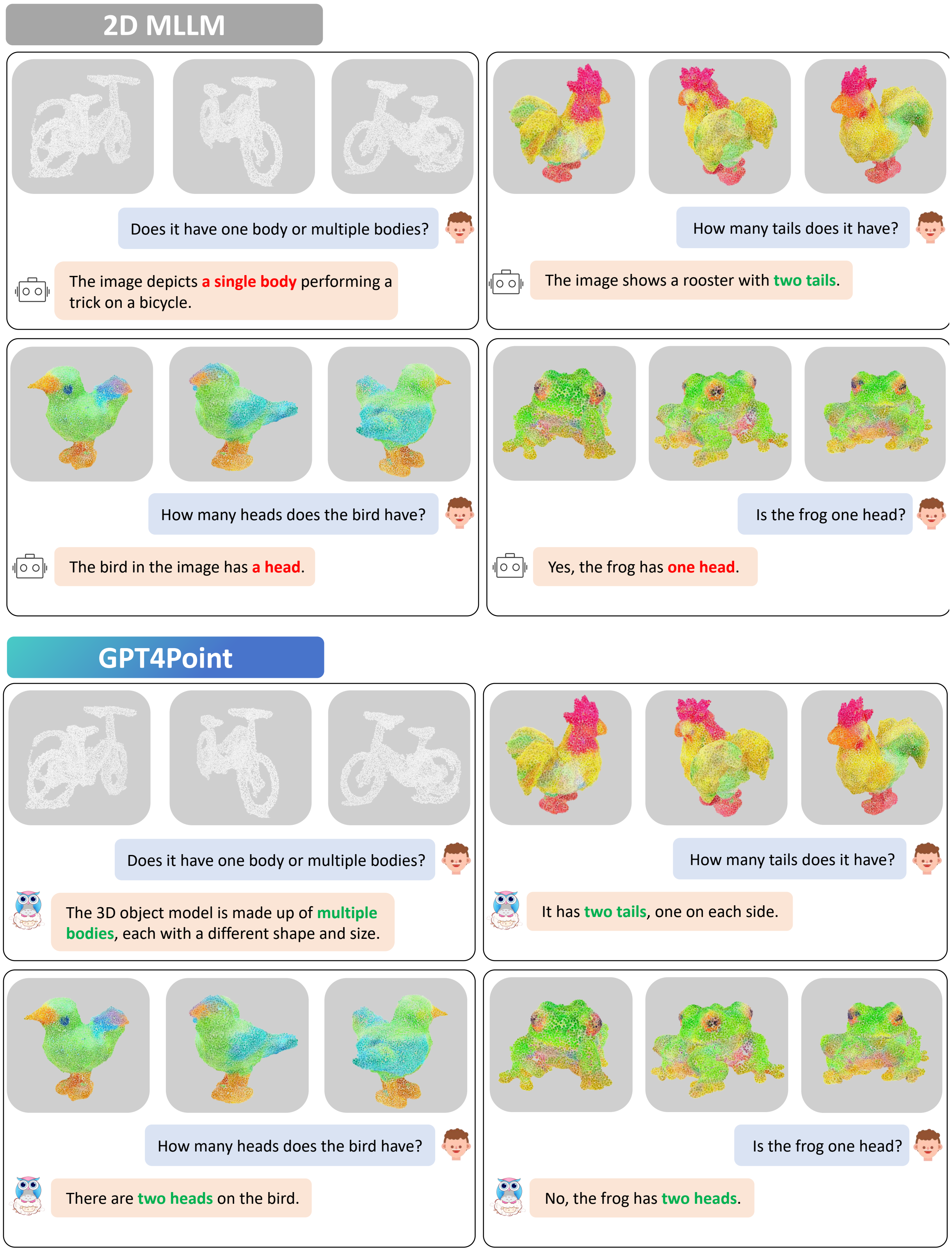


Figure S9. **Anomalous Objects: Generation Failure Cases.** The upper and lower parts, respectively, depict the performance of 2D MLLM and GPT4Point in identifying abnormally generated objects with multi-body and multi-head structures. GPT4Point effectively makes accurate judgments, whereas 2D MLLM fails to correctly identify most cases due to the lack of information from single-view images.

Figure S10. **Point-text Captions and QA Demos.** We use the finetuned GPT4Point with OPT6.7B model to generate results on the test set, demonstrating that our model performs well on dense captioning tasks and long (instruction) question answering. The results show our model's capability to comprehend object color and geometry information.

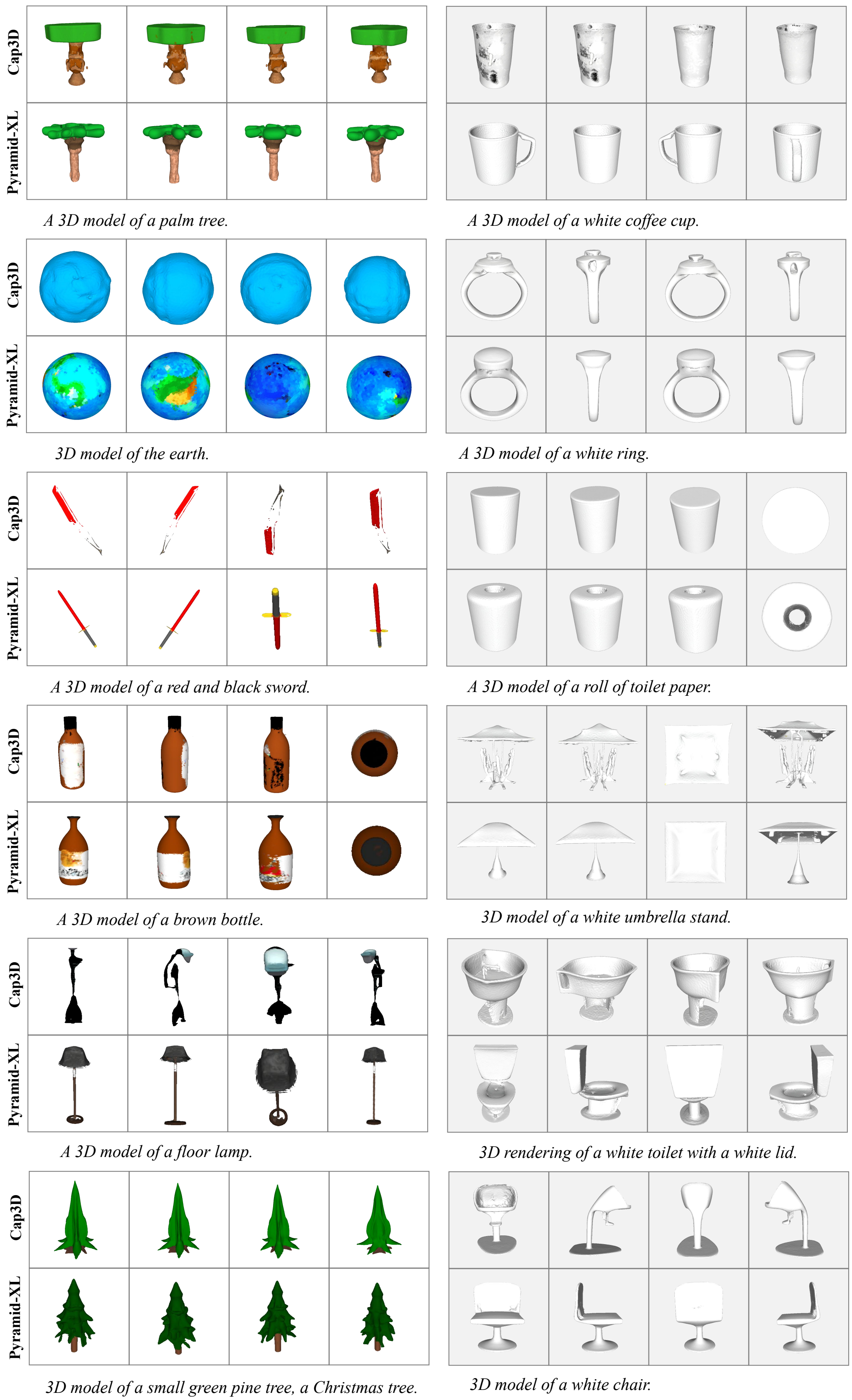


Figure S11. **Pyramid-XL Level 3 Point-E Finetune Results.** We found that the results of finetuning with dense captions from our Pyramid-XL significantly outperform those finetuned with Cap3D captions, demonstrating the greater accuracy of the captions we generated.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv:2308.12966*, 2023. 2, 3

[2] Guo Chen, Yin-Dong Zheng, Jiahao Wang, Jilan Xu, Yifei Huang, Junting Pan, Yi Wang, Yali Wang, Yu Qiao, Tong Lu, et al. Videollm: Modeling video sequence with large language models. *arXiv:2305.13292*, 2023. 1

[3] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *ICCV*, 2023. 3

[4] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 1, 2, 3

[5] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *NeurIPS*, 2023. 2

[6] 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. 2

[7] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *ICML*, 2020. 1

[8] 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. In *NeurIPS*, 2023. 1

[9] Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, et al. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *AAAI*, 2024. 1

[10] Yukun Huang, Jianan Wang, Yukai Shi, Xianbiao Qi, Zheng-Jun Zha, and Lei Zhang. Dreamtime: An improved optimization strategy for text-to-3d content creation. In *ICLR*, 2024. 3

[11] Dongxu Li, Junnan Li, and Steven CH Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *NeurIPS*, 2023. 3

[12] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 1, 2, 4

[13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1

[14] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. In *NeurIPS*, 2023. 2

[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. 2

[16] OpenAI. Chatgpt. https://openai.com/blog/chatgpt, 2022. 1

[17] OpenAI. GPT-4 technical report, 2023. 1

[18] 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. In *ICML*, 2021. 1, 2

[19] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024. 3

[20] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023. 1

[21] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. *arXiv:2304.12439*, 2023. 3

[22] Haowei Wang, Jiji Tang, Jiayi Ji, Xiaoshuai Sun, Rongsheng Zhang, Yiwei Ma, Minda Zhao, Lincheng Li, Zeng Zhao, Tangjie Lv, et al. Beyond first impressions: Integrating joint multi-modal cues for comprehensive 3d representation. In *ACMMM*, 2023. 3

[23] 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

[24] 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. 1

[25] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv:2205.01068*, 2022. 1

[26] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *TMLR*, 2024. 2