UniVerse3D: Emerging Properties of Unified Multimodal Models in 3D Understanding and Generation

Supplementary Material

1. More Implementation Details

1.1. Template Setting

As shown in Figure. 1, in the generation task, we specify the input conditions (text, image, or both) for the 3D-VLM through a fixed template. In contrast, in the replacement task, we adopt the prompt template illustrated in Figure. 2, with the aim of enhancing the 3D-VLM's ability to perceive the edited region.

1.2. Data Curation Details

200K Part-Centric Dataset. To construct our dataset, we collect 3D models from multiple sources, including Objaverse [5], Objaverse-XL [5], ShapeNet [3], and PartNet [11]. To ensure good generalization capability, we randomly sample 400,000 3D assets from these datasets while maintaining category balance. Following the procedure of P^3-SAM [10], we reverse-engineer part information from these assets and recursively merge components until all parts within an asset contribute more than 1% of the total surface area. After merging, we discard objects with too few (fewer than 5) or too many (more than 20) parts. After filtering, we obtain a final set of 200,000 objects, with each object containing an average of 12 part components.

We subsequently perform rendering to generate holistic renderings of each asset, along with individual renderings of its constituent components. All resulting images are submitted to the Vision Language Model (VLM) [1] for annotation. The annotation content includes: (1) coarse-grained and fine-grained descriptions of each component, (2) question-answering (QA) pairs constructed based on the components, and (3) component-level editing instructions. The editing instructions from the third category can be utilized for the subsequent construction of editing data. For instance, if a component is a piece of clothing, a corresponding editing instruction might be "replace the clothes with a down jacket." After obtaining the annotations from the VLM, we use this data to construct the 3D-SFT corpus for training the understanding module.

400K Editing Data. Leveraging the 200K Part-Centric Dataset, we construct a large-scale 3D editing dataset. This process employs our enhanced framework, Nano3D*, to translate 2D image edits into consistent 3D model modifications.

The workflow for each part-annotated asset proceeds as follows. First, we generate a target 2D edit. An image editing model [7], guided by a VLM-generated instruction and a holistic rendering of the asset, produces an edited image. This edited image, along with the original render, the source mesh, and the part's bounding box (bbox), is then input into the Nano3D* framework. The framework processes these inputs to generate the final, edited 3D model.

The Nano3D* framework incorporates two key improvements over the original Nano3D [16], significantly enhancing both its stability and its editing precision:

1) Direct Mask Generation from Bounding Box: The original Nano3D relies on rule-based comparisons between pre- and post-edit voxels to compute a mask, a process that can be unstable. We replace this by directly using the bounding box of the input part as the editing mask, ensuring a more robust initialization.

2) Mask Merging Strategy for Fine Details: Using a raw bounding box as a mask can be imprecise, as it may exclude small, connected details of a part (e.g., fingers being omitted from a mask intended for a jacket). This imprecision can lead to disconnected geometry or artifacts during the final mesh merging stage. To address this, we introduce a mask merging strategy. This strategy identifies all connected geometric components that intersect with the initial mask. It then merges any component whose area falls below a predefined threshold into the final mask. This ensures that fine-grained features are preserved, preventing fragmentation and improving the geometric integrity of the final edited model.

Reasoning-100K. Our data construction pipeline is built upon a predefined category matrix, formed orthogonally by content form (e.g., single object) and knowledge domain (e.g., historical civilization). By systematically traversing all category combinations, we use each combination as an instruction input to a large language model. The model first conceptualizes the category pair into an abstract idea and then, leveraging its extensive world knowledge, performs reasoning and concretization to expand this abstract concept into a detailed, richly specified description. The final output of this process is a structured pair: the abstract prompt is used to train subsequent 'concept-to-3D' tasks, while the corresponding detailed prompt drives the text-to-image and image-to-3D pipelines to generate high-quality 3D assets that serve as training targets.

Image-Text-Interleaved-300K. Conversely, for constructing our image-text interleaved dataset, we employ a

System Prompt for Gen. Task

<|im_start|>system
You are a helpful and creative assistant for generating 3D models. Your task is to analyze the user's input, which may include text, an image, or both. Your goal is to provide a comprehensive, detailed, and imaginative description for creating a 3D asset. <|im_end|>
<|im_start|>user
<|Generation Prompt|><|im_end|>
<|im_start|>assistant

Figure 1. System prompt for 3D-generation task, where <| Generation Prompt |> denotes the user-provided input, which may consist of text, images, or a combination of both.

System Prompt for Replace. Task

<|im_start|>system
You are a helpful and creative assistant for generating 3D models. Your task is to analyze the user's input, which may include text, an image, or both. Your goal is to provide a comprehensive, detailed, and imaginative description for creating a 3D asset. <|im_end|>
<|im_start|>user
Please provide the bounding box for the part corresponding to this description: **<|Description of the replaced object|>**, replace it with **<|Description of the replacing object|>**<|im_end|>
<|im_start|>assistant

Figure 2. System prompt for the Replace task, where <| Description of the replaced object |> refers to the description of the object being replaced, and <| Description of the replacing object |> refers to the description of the object that replaces it.

reverse data pipeline. This process begins with a 3D asset and aims to reverse-engineer its likely generation path. Specifically, we feed multi-view renderings of a randomly selected 3D asset into a Vision-Language Model (VLM). The VLM must then perform two tasks. First, it generates the textual component of the input condition. Second, it produces an image editing instruction. This instruction subsequently guides the automated transformation of the original asset rendering into the final image input. For example, given a 3D asset of SpongeBob, the VLM might generate the text prompt, "Generate the 3D asset for SpongeBob shown in the image." The corresponding image editing instruction might be, "Add a background scene to the rendering of SpongeBob." Finally, we create a data pair by matching the edited image and the generated text (the inputs) with the original 3D asset (the output). Through this inverse construction method, we systematically generate well-structured and logically consistent training triplets, formatted as (Image Input, Text Input) -> 3D Asset Output, for use in multi-modal generation tasks.

2. More Comparison

2.1. 3D understanding

We compare our 3D-VLM with Part-X-MLLM [13] on UniPart-Bench [13] across 7 tasks. As shown in the Table 2, our Universe3D achieves superior performance on 3D Grounding, 3D Caption, Part Caption, and several other tasks. As shown in Table 3, our Universe3D demonstrates text and image understanding capabilities comparable to Qwen2.5-VL-7B-Instruct [1], and significantly outperforms other 3D multimodal models.

Table 1. Image-to-3D generation task comparison.

Model	CLIP	FD_{incep}	KD_{incep}	FD_{dinov2}	KD_{dinov2}
SAR3D [4]	75.86	88.18	3.37	750.3	181.7
TripoSG [9]	93.55	13.41	0.15	141.7	6.47
ShapeLLM-Omni [15]	87.18	29.66	1.01	449.6	55.9
Hunyuan3D-2.1 [8]	93.50	11.61	0.12	113.2	5.29
TRELLIS [14]	**93.64**	**11.27**	**0.08**	**110.51**	**5.07**
Ours	93.10	13.04	0.09	119.2	7.94

2.2. Image-to-3D

According to the results in Table 1, our model is highly competitive on the image-to-3D task on the Toys4K [12] dataset, achieving top-three rankings. However, it does not reach state-of-the-art (SOTA) status. A primary reason for this is the architectural choice of our image encoder. We posit that the ViT from Qwen-VL [1] lacks the specialized capability to fully resolve the structural information from the input image, which is critical for generating geometri-

Table 2. Comparison with Part-X-MLLM on **UniPart-Bench** across seven part understanding and grounding tasks. Q1 denotes the coarse query setting, and Q2 denotes the fine-grained query setting.

Task	Model	IoU ↑	SBERT ↑	SimCSE ↑	BLEU-1 ↑	ROUGE-L ↑	METEOR ↑
Pure box listing	Part-X-MLLM	0.75	-	-	-	-	-
	Universe3D (Our)	**0.76**	-	-	-	-	-
	Δ Gain	*+0.01*	-	-	-	-	-
Multi-Part Grounding (Q1)	Part-X-MLLM	0.73	55.60	54.19	35.55	35.58	18.09
	Universe3D (Our)	**0.76**	**60.70**	**60.30**	**42.20**	**42.24**	**21.66**
	Δ Gain	*+0.03*	*+5.10*	*+6.11*	*+6.65*	*+6.66*	*+3.57*
Multi-Part Grounding (Q2)	Part-X-MLLM	0.73	63.68	60.68	31.01	**33.68**	27.72
	Universe3D (Our)	**0.74**	**64.13**	**62.23**	**31.57**	33.36	**29.33**
	Δ Gain	*+0.01*	*+0.45*	*+1.55*	*+0.56*	*-0.32*	*+1.61*
Single-Part Grounding (Q1)	Part-X-MLLM	0.53	73.28	71.70	36.29	38.94	33.21
	Universe3D (Our)	**0.57**	**76.16**	**75.08**	**39.85**	**41.18**	**37.30**
	Δ Gain	*+0.04*	*+2.88*	*+3.38*	*+3.56*	*+2.24*	*+4.09*
Single-Part Grounding (Q2)	Part-X-MLLM	0.44	-	-	-	-	-
	Universe3D (Our)	**0.52**	-	-	-	-	-
	Δ Gain	*+0.08*	-	-	-	-	-
Box-to-Text (Q1)	Part-X-MLLM	-	57.35	56.49	38.12	38.14	19.49
	Universe3D (Our)	-	**64.41**	**64.07**	**46.41**	**46.60**	**23.87**
	Δ Gain	-	*+7.06*	*+7.58*	*+8.29*	*+8.46*	*+4.38*
Box-to-Text (Q2)	Part-X-MLLM	-	64.64	61.96	31.35	33.73	28.13
	Universe3D (Our)	-	**70.65**	**68.97**	**37.20**	**38.62**	**35.38**
	Δ Gain	-	*+6.01*	*+7.01*	*+5.85*	*+4.89*	*+7.25*
Part QA	Part-X-MLLM	0.55	78.98	84.25	40.54	**42.26**	34.24
	Universe3D (Our)	**0.56**	**83.11**	**87.17**	**46.79**	40.94	**42.05**
	Δ Gain	*+0.01*	*+4.13*	*+2.92*	*+6.25*	*-1.32*	*+7.81*

Table 3. Performance comparison on text-only and image understanding benchmarks.

Model	MMLU	PIQA	MMStar	RealWorldQA	MMMU
Part-X-MLLM [13]	25.7	59.4	10.1	1.18	25.7
ShapeLLM-Omni [15]	49.1	54.8	26.5	32.2	34.7
Qwen-2.5VL-7B [1]	69.1	77.3	62.5	68.5	51.7
UniVerse3D (Ours)	52.1	68.8	52.2	52.6	38.7

cally faithful 3D models.

3. Limitation and Discussion

Although Universe3D unifies generation, understanding, and editing with strong performance, our current architecture has two main limitations.

1. Limited consistency in image-to-3D generation. As shown in Table 1, our model is highly competitive on the Toys4K [12] dataset and achieves top-three rankings. However, it does not yet reach state-of-the-art status. We attribute this primarily to the architectural choice of our image encoder. The ViT backbone employed in Qwen-VL is optimized for high-level semantic understanding rather than the fine-grained spatial resolution required for geometrically faithful 3D reconstruction. Consequently, the system occasionally fails to fully resolve structural details and alignment from the input image. To address this in future iterations, we plan to incorporate stronger vision encoders such as Bagel [6] or Hunyuan-Image 3.0 [2]. These models provide denser visual representations that can substantially improve the geometric and visual consistency between input images and the reconstructed 3D outputs.

2. Lack of unified texture modeling. The current framework requires an external texture module to generate surface appearance, which compromises the goal of a fully unified system. This reliance separates the geometry generation from texture synthesis. To resolve this issue, we intend to explore unified texture–geometry representations in future work. Approaches such as trellis-based methods offer a promising pathway to integrate texture modeling directly into the generative pipeline, thereby eliminating the need for external components and further consolidating the Universe3D framework.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 2, 3

[2] Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025. 3

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[4] Yongwei Chen, Yushi Lan, Shangchen Zhou, Tengfei Wang, and Xingang Pan. Sar3d: Autoregressive 3d object generation and understanding via multi-scale 3d vqvae. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28371–28382, 2025. 2

[5] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023. 1

[6] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 3

[7] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 1

[8] Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025. 2

[9] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025. 2

[10] Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Yanwen Guo, Zhuo Chen, and Chunchao Guo. P3-sam: Native 3d part segmentation. *arXiv preprint arXiv:2509.06784*, 2025. 1

[11] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 1

[12] Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1798–1808, 2021. 2, 3

[13] Chunshi Wang, Junliang Ye, Yunhan Yang, Yang Li, Zizhuo Lin, Jun Zhu, Zhuo Chen, Yawei Luo, and Chunchao Guo. Part-x-mllm: Part-aware 3d multimodal large language model. *arXiv preprint arXiv:2511.13647*, 2025. 2, 3

[14] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 2

[15] Junliang Ye, Zhengyi Wang, Ruowen Zhao, Shenghao Xie, and Jun Zhu. Shapellm-omni: A native multimodal llm for 3d generation and understanding. *arXiv preprint arXiv:2506.01853*, 2025. 2, 3

[16] Junliang Ye, Shenghao Xie, Ruowen Zhao, Zhengyi Wang, Hongyu Yan, Wenqiang Zu, Lei Ma, and Jun Zhu. Nano3d: A training-free approach for efficient 3d editing without masks. *arXiv preprint arXiv:2510.15019*, 2025. 1