

Hunyuan3D-Buffer 1.0

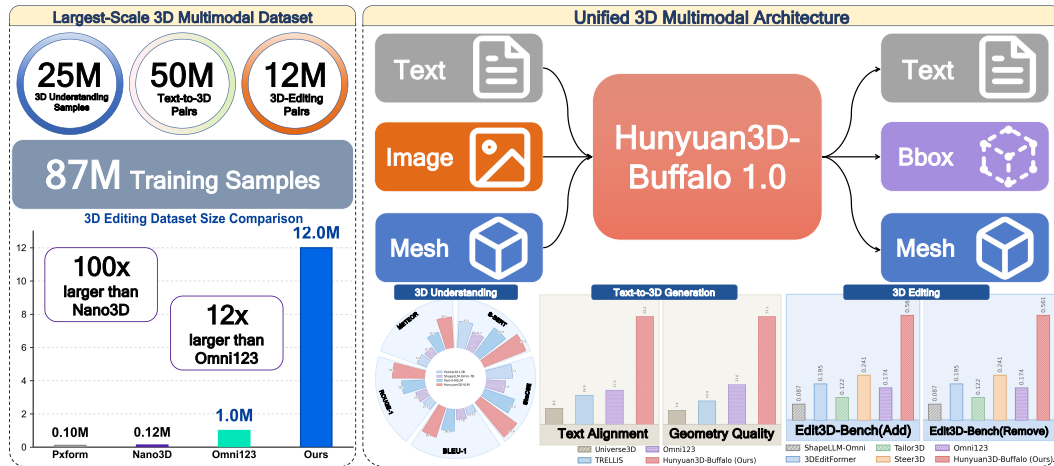
A Unified Multimodal Model for Scalable 3D Generation, Understanding, and Editing

Tencent Hunyuan*

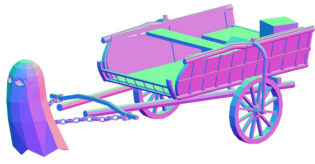
Abstract

Recent advances in image generation have demonstrated the potential of unified multimodal models that integrate understanding, generation, and editing. However, unified 3D modeling remains constrained by scarce multimodal data, particularly the lack of large-scale and geometrically consistent editing data. To address this limitation, we propose **Hunyuan3D-Buffer 1.0**, a unified framework supporting 3D understanding, text-to-3D generation, instruction-guided 3D editing, and text-grounded part generation within a single architecture. To enable scalable training, we construct an 87M-scale 3D multimodal corpus, comprising 25M understanding samples, 50M text-to-3D pairs, and 12M editing pairs generated using Nano3D-v2. Architecturally, the framework combines Hunyuan3D-VLM for semantic, structural, and spatial understanding with Hunyuan3D DiT for high-fidelity 3D synthesis. The VLM provides multimodal semantic conditions for generation, while editing and part generation additionally condition the diffusion process on the source object representation to preserve its overall structure and unedited regions. Extensive experiments show that Hunyuan3D-Buffer 1.0 achieves state-of-the-art or leading performance on text-to-3D generation and 3D editing benchmarks, while exhibiting strong understanding and part-generation capabilities. Our analysis further shows that both generation and understanding improve editing, demonstrating the effectiveness of unified 3D multimodal training.

Project Page: <https://tencent-hunyuan.github.io/Hunyuan3D-Buffer1.0/>



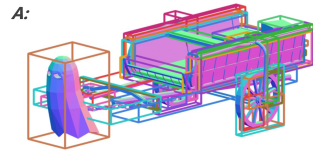
3D Understanding



Q: What is this 3D shape?

A: The model portrays a traditional wooden cart or wagon, complete with spoked wheels and an open cargo bed. Attached to the front shafts by a set of chains is a small, low-poly ghost-like figure that appears to be pulling the vehicle.

Q: Show the box of main components



Text to 3D

"A seated anthropomorphic cat on a cylindrical pedestal with circular base, wearing a Santa hat, polka-dotted dress with bow, thigh-high stockings, and high heels, holding a candy canero"



Seed 1



Seed 2



Seed 3



Seed 4



Seed 5

Shape Editing



Input Shape



"Replace the head with a horse head"



"Put on heavy armored gauntlets with spiked knuckles"



"Replace the head with a muscular bull head with thick curved horns"



"Put on a cape"

Part Generation

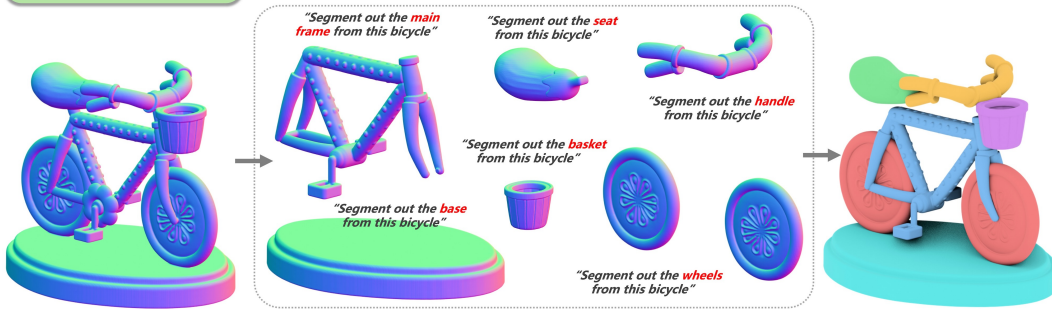


Fig. 1: **Hunyuan3D-Buffer 1.0** is an unified 3D multimodal framework that combines autoregressive modeling with diffusion-based generation, enabling 3D understanding, text-to-3D generation, 3D editing, and text-grounded part generation within a single architecture.

Table of Contents

1	Introduction.....	4
2	Related Work.....	5
2.1	3D Generation.....	5
2.2	3D Editing	5
2.3	3D Part Generation	6
2.4	Unified Multimodal Models	6
3	Data curation.....	6
3.1	Overview	6
3.2	3D Understanding Data	7
3.3	Text-to-3D Data.....	7
3.4	Editing Data.....	9
3.5	Text-grounded Part Generation Data	13
4	Method	14
4.1	Architecture	14
4.2	Training Procedure.....	15
5	Experiments	16
5.1	3D Understanding	16
5.2	Text-to-3D.....	17
5.3	3D Editing	20
5.4	Part Generation.....	23
6	Conclusion & Future Work	24
7	Author List.....	25
A	Additional Results	26

1 Introduction

Recent advances in generative artificial intelligence have driven 2D vision models toward unified multimodal systems that integrate understanding, generation, and instruction-guided editing. Models such as GPT-4o, FLUX [38], Seedream [26, 67], Qwen-Image [86], Nano-Banana, and the GPT-Image series have shown that visual generation is no longer an isolated synthesis task, but can serve as a general interface for multimodal reasoning, content creation, and interactive editing [9, 14, 19, 20, 81, 87]. More recently, works such as Vision-Banana further suggest that dense perception and part-level understanding can also be formulated through generative modeling, revealing the potential of unified architectures to transfer capabilities across tasks.

However, analogous progress in the 3D domain remains limited. Unlike 2D images, 3D assets are much harder to collect, annotate, and edit at scale. In particular, large-scale and geometrically consistent 3D editing data is scarce, making it difficult to train models that can modify existing 3D assets while preserving identity, structure, and unedited regions. As a result, 3D understanding models [62, 63, 93], 3D generation models [39, 46, 92, 108], and 3D editing models [12, 42, 103] are still largely developed as separate systems. This fragmentation prevents the model from learning a unified semantic-visual-geometric representation, and leaves it unclear whether 3D understanding, generation, and editing can mutually reinforce one another under a unified training paradigm.

To address the data bottleneck, we first construct a large-scale 3D multimodal training corpus with 87M samples, covering 25M 3D understanding samples, 50M text-to-3D pairs, and 12M 3D editing pairs. A key component of this data engine is Nano3D-v2, an agent-based 3D editing data construction algorithm that extends Nano3D [103]. Nano3D-v2 combines anchor-view selection, learned 3D edit-region localization, voxel-level editing, fine-grained geometry and texture refinement, and VLM-based annotation and filtering. This pipeline enables scalable construction of high-quality, geometrically consistent editing pairs, substantially mitigating the scarcity of 3D editing supervision.

On top of this data foundation, we propose Hunyuan3D-VLM, a 3D vision-language model designed for fine-grained semantic, structural, and spatial understanding of 3D objects. Hunyuan3D-VLM encodes both geometric structure and appearance cues, enabling object-level captioning, part-level question answering, 3D grounding, edit-instruction synthesis, and edit-outcome reasoning. By equipping the model with explicit 3D perception and grounding ability, Hunyuan3D-VLM provides the semantic and spatial reasoning foundation required for unified 3D multimodal modeling.

Building upon Hunyuan3D-VLM, we further introduce **Hunyuan3D-Buffer 1.0**, a unified 3D multimodal framework that combines autoregressive modeling with diffusion-based 3D generation. The framework connects Hunyuan3D-VLM with a generative 3D-DiT backbone initialized from Hunyuan3D-2.1 [35], allowing high-level multimodal reasoning to guide 3D synthesis and editing through a unified conditional interface. As a result, Hunyuan3D-Buffer 1.0 supports 3D understanding, text-to-3D generation, instruction-guided 3D editing, and text-grounded part generation within a single architecture. Inspired by the generative formulation of dense perception in Vision-Banana, we also cast part generation as a native instruction-following and conditional generation task, enabling the model to decompose and generate language-referred 3D parts without relying on a specialized part-generation pipeline.

Extensive experiments demonstrate that Hunyuan3D-Buffer 1.0 achieves state-of-the-art or leading performance on both text-to-3D generation and 3D editing benchmarks, while also showing strong 3D understanding and part-generation abilities. More importantly, our analysis reveals two clear cross-task synergies. First, stronger text-to-3D generation improves 3D editing, as a better generative prior leads to more complete and natural edited geometry. Second, stronger 3D understanding improves 3D editing, as fine-grained semantic and spatial reasoning helps the model localize editing regions, interpret instructions, and preserve unedited areas. These findings show that unified 3D multimodal training is not merely a combination of multiple tasks, but can induce meaningful capability transfer across understanding, generation, and editing.

In summary, our contributions are as follows:

1. We construct a large-scale 3D multimodal training corpus with 87M samples, including 25M 3D understanding samples, 50M text-to-3D pairs, and 12M 3D editing pairs. To build the editing data, we introduce **Nano3D-v2**, an agent-based 3D editing data construction algorithm that produces high-quality and geometrically consistent editing pairs at scale.
2. We propose Hunyuan3D-VLM, a 3D vision-language understanding architecture capable of fine-grained semantic, structural, and spatial understanding of 3D objects, providing a strong foundation for 3D grounding, part-level reasoning, and edit-aware understanding.
3. We develop **Hunyuan3D-Buffalo 1.0**, a unified 3D multimodal framework that combines autoregressive modeling with diffusion-based generation, enabling 3D understanding, text-to-3D generation, 3D editing, and text-grounded part generation within a single architecture.
4. Hunyuan3D-Buffalo 1.0 achieves state-of-the-art or leading performance on text-to-3D generation and 3D editing benchmarks. Through unified training, we further reveal two synergistic relationships among 3D multimodal tasks: generation improves editing, and understanding improves editing.

2 Related Work

2.1 3D Generation

Recent advances in 3D content generation have explored three major paradigms. Early works, represented by DreamFusion [61], mainly rely on optimization-based methods that distill visual priors from pretrained 2D diffusion models into 3D representations, enabling text-to-3D generation without large-scale 3D supervision [13, 51, 53, 65, 70, 71, 82, 89, 101, 105]. To improve generation efficiency and multi-view consistency, subsequent works, represented by 3DShape2VecSet [107] and TRELLIS [92], construct 3D-native VAE and train generative models in 3D latent spaces, enabling feed-forward generation with faster inference and improved geometric fidelity [16, 30, 35, 40, 44, 46, 49, 113]. Another line of work, represented by MeshAnything [15], focuses on low-poly mesh generation by tokenizing vertices and faces and modeling them with autoregressive or flow-based generative models, aiming to produce compact and production-friendly 3D meshes [17, 28, 37, 45, 72, 78, 79, 83, 84, 110, 112]. Despite these advances, achieving high fidelity, structural consistency, and controllable editing remains an open challenge.

2.2 3D Editing

Text-guided 3D editing aims to modify existing 3D assets according to natural language instructions while preserving unedited regions and maintaining 3D consistency. Early methods typically rely on per-instance optimization, using Score Distillation Sampling (SDS) or pretrained 2D diffusion priors to align 3D representations with editing instructions [12, 21, 22, 29, 52, 68, 109, 117]. Another line of work edits rendered 2D views and subsequently fuses or reconstructs them into edited 3D assets [2, 3, 6, 8, 23, 25, 43, 64, 114]. While the former incurs costly per-instance optimization, the latter depends heavily on projection and reconstruction consistency and may introduce cross-view inconsistencies or geometric distortions. More recently, training-free methods, exemplified by Nano3D [103] and VoxHammer [42], directly edit the latent or structured representation spaces of pretrained 3D generative models by reusing frozen 3D priors through inversion, latent replacement, flow-based editing, or agentic tool chains [5, 18, 31, 50, 54, 69, 115]. These methods demonstrate the editability of pretrained 3D priors, but their performance can remain unstable across diverse objects, edit types, and instructions. In parallel, training-based approaches construct paired or self-generated 3D editing data and lightly fine-tune pretrained 3D generators to learn feed-forward edit transformations [11, 27, 32, 58, 85, 91, 94, 106]. Despite these advances, achieving precise and efficient 3D editing while ensuring structural consistency and faithful preservation of unedited regions remains challenging.

2.3 3D Part Generation

3D part generation aims to produce structured assets where each semantic component is a distinct mesh. Early methods [59, 95] are restricted to fixed part taxonomies, while recent multi-stage pipelines depend on 2D segmentation and suffer from view-dependent inconsistencies. More recent 3D-native approaches adopt multi-diffusion-path architectures that synthesize parts through coordinated diffusion branches [34, 47, 48, 57, 97, 98, 116], or learn continuous feature fields for part segmentation [55]. Another line incorporates physics simulation to ensure inter-part plausibility [56, 96, 99]. Despite this progress, existing methods either assume a fixed part vocabulary or infer part structure implicitly. CubePart [116] introduces an open-vocabulary, text-grounded framework for part generation. In this work, we demonstrate for the first time that text-grounded part generation can be accomplished within a native 3D multimodal large model, unifying part generation with other 3D tasks under a single architecture.

2.4 Unified Multimodal Models

Recent studies on unified image understanding and generation have developed along several architectural directions. One line of work, represented by Chameleon [66, 73, 81, 88], converts images into discrete visual tokens with VQVAE [76], allowing text and images to be modeled within a single autoregressive Transformer under the next-token prediction objective. Another line adopts a more decoupled design, as exemplified by MetaQuery [9, 10, 60, 74, 86, 87, 111], where a Multimodal Large Language Model [1, 75] serves as a semantic encoder for complex inputs and provides conditional features to a Diffusion Transformer for high-fidelity image synthesis. A third direction seeks tighter integration between understanding and generation. For example, Bagel [7, 14, 20, 33] incorporates language modeling and flow-based generation within a unified Transformer backbone, enabling both tasks to be handled in a more native architecture. In contrast, unified modeling in the 3D domain remains relatively underexplored. Existing efforts such as ShapeLLM-Omni [4, 102] attempt to incorporate 3D data into a unified framework through VQVAE-based tokenization, but are still limited in capturing fine-grained geometry and complex spatial structures.

3 Data curation

3.1 Overview

A central obstacle to unified 3D multimodal modeling is the scarcity of large-scale, high-quality, and geometrically consistent training data that jointly covers understanding, generation, and editing. To support the three capabilities of our framework within a single training pipeline, we construct a comprehensive 3D data engine that produces three complementary corpora, summarized in Table 1.

First, a **3D understanding** corpus pairs object point clouds with instruction-style dialogues spanning captioning, question answering, grounding, and editing-related tasks, interleaved with general text and image-text data to preserve the model’s language and 2D multimodal abilities. Second, a **text-to-3D** corpus provides large-scale text-asset pairs, built by a fully automated pipeline that synthesizes compositional prompts, generates and renders the corresponding 3D assets, and attaches multi-tier, geometry-grounded captions under strict quality filtering. Third, a **3D editing** corpus supplies geometrically consistent (source, edited asset, instruction) triplets, produced by our **Nano3D-v2** pipeline and refined through a vision-language-model-based annotation and filtering procedure. In total, the engine yields roughly 25M understanding samples, 50M text-to-3D pairs, and 12M editing pairs. The remainder of this section describes the construction of each corpus in turn.

Capability	Subset	#Samples
3D understanding	Text / image / 3D instruction data	~25M
Text-to-3D	Text-asset pairs	~50M
3D editing	Human edits	~7M
	Object edits	~3M
	Part generation	~2M
	<i>Subtotal</i>	~12M

Table 1: Overview of the full training corpus across 3D understanding, text-to-3D generation, and instruction-guided 3D editing.

3.2 3D Understanding Data

Hunyuan3D-VLM is trained on a large-scale instruction-tuning corpus that interleaves three modalities: pure text, image-text, and 3D point cloud-text data. The general text-only and image-text data are drawn from a mixture of public and in-house instruction corpora and serve to preserve the model’s language reasoning and 2D multimodal perception during 3D adaptation. The text-only and image-text splits contribute approximately 7M and 3M conversation samples, respectively. The 3D point cloud-text data forms the core of the corpus and accounts for the largest share at roughly 15M samples, bringing the total mixture to about 25M samples.

The 3D split is built upon the part-level and object-level 3D dialogue data provided by Part-X-MLLM [77] and ShapeLLM-Omni [102], together with data we collect and synthesize in-house from our own asset pool. Across these sources, the 3D understanding tasks cover:

- **3D captioning:** multi-tier natural-language descriptions of an object’s geometry, structure, and salient parts.
- **3D question answering:** free-form questions about an object’s category, attributes, parts, and spatial relations.
- **3D grounding:** localizing referred parts or regions in an object, with answers expressed as quantized axis-aligned bounding boxes via `<boxs>/<boxe>` token sequences.
- **Edit-instruction synthesis:** given the source point cloud and an original, often coarse editing request, producing a precise, executable editing instruction grounded in the object’s geometry.
- **Edit-outcome captioning:** given the source object and an editing operation, describing the object that results from the edit, so that the model learns the correspondence between an editing operation and its geometric outcome.

3.3 Text-to-3D Data

We build our text-to-3D training corpus with a fully automated, five-stage pipeline: (i) hierarchical prompt taxonomy construction; (ii) compositional prompt synthesis; (iii) image-to-3D asset generation and multi-view rendering; (iv) multi-tier captioning and geometry-quality scoring; and (v) quality filtering and dataset packaging. Figure 2 gives an overview.

Stage 1: Hierarchical Prompt Taxonomy. Prompts are sampled from a four-level taxonomy that decouples *what* to generate from *how* it is composed:

- **L0 – Generation mode:** `object_single`, `hybrid_composite` (a primary subject interacting with a secondary object), and `abstract_procedural`. Mode weights are category-dependent; e.g. *Humanoids* and *Animals* are biased toward interaction scenes.

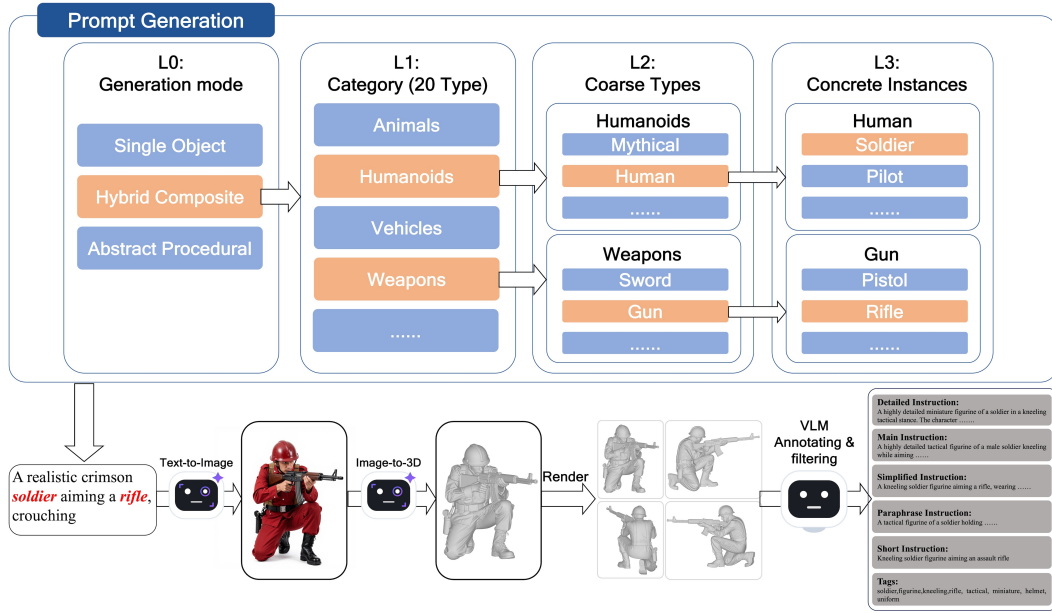


Fig. 2: Pipeline of constructing text-to-3D training corpus.

- **L1 – Category** (20 categories, weighted): e.g. *Humanoids*, *Animals*, *Vehicles*, *Architecture*, *Furniture*, *Weapons*, *Plants*, *Food*, etc. Weights skew the distribution toward the categories that matter most for our target use cases.
- **L2 / L3 – Subtype and fine-grained type**: each L1 is expanded into coarse types (L2) and concrete instances (L3), yielding thousands of leaf concepts (e.g. *Humanoids* → *fantasy warrior* → *samurai*).

Each leaf is further decorated with *attributes* sampled from controlled vocabularies: *style* (15 options, weighted toward render-friendly looks such as *realistic*, *product*, *3D cartoon*), *color*, *material*, *condition*, and *pose*. Styles that are unsuitable as image-to-3D inputs (wireframe, blueprint, watercolor, isometric, etc.) are explicitly removed, and a per-category override table further constrains which attributes apply to which concepts.

Stage 2: Compositional Prompt Synthesis. A single prompt is assembled by sampling (L0, L1, L2, L3) and attributes, then composing a natural, user-style request. For hybrid_composite prompts, a *semantic-role-aware* interaction sampler selects a secondary object that the primary subject can *plausibly* interact with (e.g. a knight *wielding* a sword), governed by role interaction tables, whitelists, and anthropomorphism guards (e.g. real animals are prevented from performing physically implausible, finger- or speech-requiring actions, and are down-styled away from photoreal looks when forced into stylized interactions). When no valid contact template exists, the prompt gracefully degrades to a single object.

To inflate the effective unique-prompt space without altering semantics, the synthesizer (a) randomly omits optional attribute axes (material/pose/condition, each with $\sim 30\%$ drop probability) and (b) appends a *quality suffix* drawn component-wise from five paraphrase axes (view angle, framing, background, subject visibility, detail level). Each record also carries a shared negative-prompt. The resulting per-record schema is:

```
{ "L0", "L1", "L2", "L3", "attributes", "prompt", "negative_prompt", "secondary"? }
```

Stage 3: Asset Generation and Rendering. Prompts are turned into images and then into 3D assets via an image-to-3D model, after which each asset is rendered to multiple canonical views (front-facing, white background) and stored alongside a sampled surface point cloud.

Tier	Length	Emphasis
Detailed	4–6 sentences (≤ 120 w)	subject, parts, pose, features
Main	24–30 tokens	structure + key parts
Simplified	15–20 tokens	main parts and fit
Paraphrase	15–20 tokens	reworded Simplified
Short	6–10 tokens	subject + ≤ 1 attribute
Tags	≤ 8 keywords	disentangled keywords

Table 2: Six caption tiers produced per asset.

For the edit-augmented split, we run an automated quality-control step: a *difference-detection* module compares the pre-edit (background-removed, white-matted) and post-edit images to produce a binary diff mask, and a *consistency check* verifies that (a) regions *outside* the mask remain unchanged between pre- and post-edit (sharpness-aligned to avoid false positives) and (b) the mask does not cover an implausibly large fraction of the foreground. Both steps are sharded across machines for million-scale throughput, and failing samples are discarded.

Stage 4: Multi-Tier Captioning and Geometry-Quality Scoring. To attach training captions to each generated asset, we query a vision-language model (Gemini, via an internal endpoint) with *the reference image as context* and *the multi-view renders of the actual generated mesh as the source of truth*. The model is instructed to (i) describe the rendered untextured white mesh in a natural text-to-3D request voice—never mentioning color, material, lighting, or the rendering/generation process, and never comparing against the reference image—and (ii) assign a strict integer *geometry-quality* score. Concretely it emits six caption tiers that are *derived* from the most detailed tier (shorter tiers may drop information but never add it), ~~the geometry-quality score~~ ^{thus one score} lies in $[0, 10]$ and penalizes truncation/cropping, duplicated parts, detached/floating chunks, broken topology, implausible proportions, and missing parts, with calibrated anchors (0: severely broken/unusable; 3: recognizable but multiply defective or truncated; 7: mostly clean and fully framed; 10: clean, complete, and correctly assembled in every view). The model is explicitly told to use the full range rather than default to high scores.

Stage 5: Quality Filtering and Dataset Packaging. Finally, we parse the model responses, extract the six caption tiers and the geometry-quality score, and filter assets by a score threshold (we keep the highest-quality tier, $\text{geometry_quality} \geq \tau$, with $\tau=10$ for our cleanest split). Surviving assets are packaged into the training format as tuples of (surface point-cloud path, caption dictionary, L1 category), yielding the final dataset of $\sim 5 \times 10^7$ high-quality text-asset pairs.

3.4 Editing Data

This section details our automated pipeline for constructing high-fidelity 3D editing data. Given a source 3D asset and a natural-language instruction, the objective is to generate a target asset that faithfully executes the requested edit while preserving the geometry, identity, and multi-view consistency of non-target regions. To achieve this, we present **Nano3D-v2**, a comprehensive framework that integrates anchor-based viewpoint selection, learned 3D edit-region localization, voxel-level local modification, fine-grained geometry refinement, and multimodal quality filtering.

Current methods for constructing 3D editing data can be broadly categorized into two paradigms. The first follows a 2D-to-3D lifting strategy, where edits are first performed in the image domain and then lifted back into 3D space. For example, ShapeLLM-Omni [102] leverages off-the-shelf 2D diffusion models to generate target visual conditions, followed by multi-view reconstruction. Although this paradigm benefits from the strong generative capability of 2D models, it lacks explicit 3D correspondences between the source asset and the edited target. As a result, the reconstructed 3D assets often suffer from identity

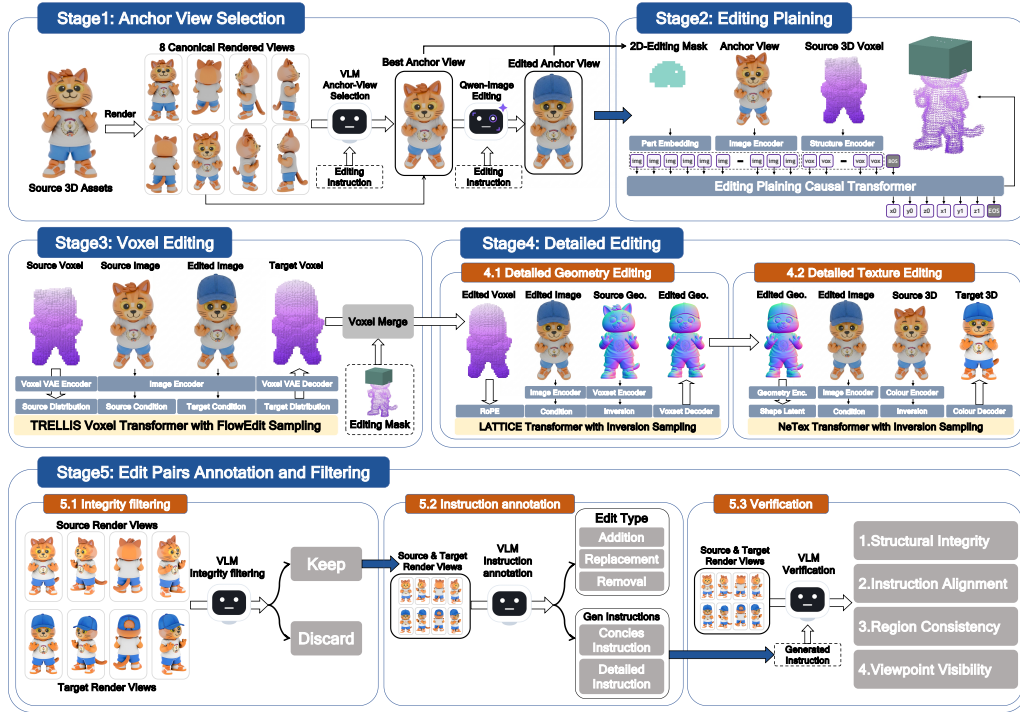


Fig. 3: Pipeline of constructing 3D editing training corpus (Nano3D-v2).

drift, geometric hallucinations, multi-view inconsistency, and unintended modifications to regions that should remain unchanged.

The second paradigm introduces explicit 3D spatial constraints to regularize the editing process, as exemplified by Nano3D [103] and VoxHammer [42]. By restricting modifications to localized 3D regions, these methods improve spatial consistency and better preserve unedited areas. However, they are still limited in several aspects. First, the geometric quality of the edited results is largely bounded by the underlying 3D backbone, such as TRELLIS [92], which often leads to over-smoothed or inaccurate local structures. Second, methods such as VoxHammer require manually specified 3D bounding boxes, making them difficult to scale to large-scale automatic data construction. Third, heuristic region estimation, as adopted in the original Nano3D [103], often fails to precisely localize small, occluded, or view-dependent edits.

Nano3D-v2 addresses these shortcomings by introducing a dedicated, learned 3D localization module and a high-fidelity refinement stage. As illustrated in Fig. 3, the pipeline comprises three core modules—Editing Planning, Voxel Editing, and Detailed Geometry Refinement—together with rendering-based filtering and annotation, executed across 5 distinct stages.

Stage 1: Anchor View Selection. Starting with a source 3D asset, we render eight canonical views. Instead of arbitrary view selection, we employ a vision-language model to identify the optimal *editing anchor*—the viewpoint that provides the highest semantic salience for the requested edit. We then utilize Qwen-Image to perform instruction-guided editing on this anchor view. This strategy ensures that the subsequent 3D optimization is grounded in the most informative 2D visual condition rather than a randomly selected perspective.

Stage 2: Editing Planning. To surpass the limitations of heuristic masking, we develop a dedicated **Editing Planning** model. We first derive a 2D editing mask by computing the pixel-wise discrepancy between the source and edited anchor images. We then train an autoregressive Transformer to predict the corresponding 3D editing region. By feeding the



Fig. 4: Examples of editing pairs in the training corpus created by Nano3D-v2.

2D mask and the source voxel representation into this model, it learns to output a precise 3D bounding box. This box serves as a hard spatial constraint: voxels within the volume are mutable, while those outside remain frozen to preserve the original geometry.

Stage 3: Voxel Editing. Conditioned on the edited anchor image and the predicted 3D box, we employ the voxel Transformer from TRELIS [92] to perform voxel-level FlowEdit. To further mitigate spurious deformations in unedited areas, we follow the local editing paradigm introduced in Nano3D [103] by implementing a *voxel-merge* operation. By replacing the edited voxels outside the predicted box with their original source counterparts, we explicitly enforce local editing and prevent global identity drift.

Stage 4: Fine-grained Geometry and Texture Editing. While voxel-level editing establishes a consistent global structure, it lacks the resolution required for fine surface details and texture editing. We therefore employ LATTICE [40] for sub-voxel geometry refinement. Using the merged voxel as a geometric prior, the LATTICE Transformer performs inversion-based inpainting. This process refines the edited manifold while seamlessly stitching it with

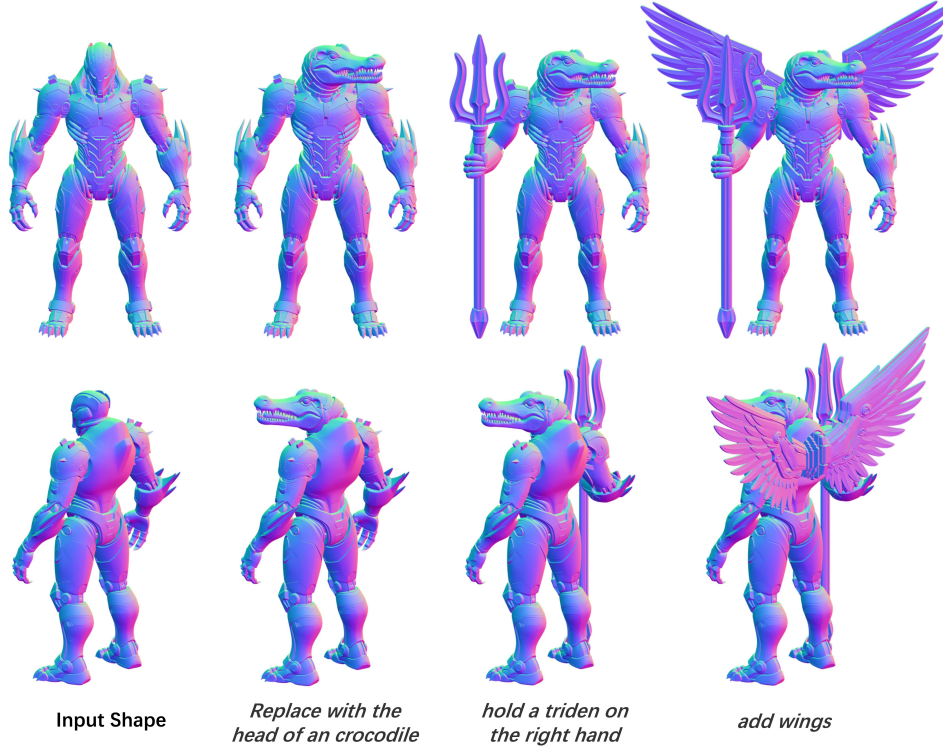


Fig. 5: Examples of multi-round editing by Nano3D-v2.

the source mesh, resulting in high-resolution surfaces and artifact-free transitions around the editing boundary. We further employ NaTex [41] for native texture editing. Given the source textured mesh, the edited mesh from LATTICE, the edited image, and the 3D editing region, the NaTex Transformer performs inversion-based texture inpainting. Since inpainting alone may still leave discrepancies around the editing boundary, we additionally apply alpha blending within a 7-voxel band along the boundary (the latent resolution of NaTex is set to 128 voxels), which ensures a smooth transition.

Stage 5: Edit Pairs Annotation and Filtering. The raw edit pairs produced by the preceding stages are noisy: some edited meshes are structurally broken, and the original user prompts are often coarse, ambiguous, or only loosely aligned with the realized geometric change. To turn these pairs into reliable supervision, we apply a vision-language-model-based annotation and filtering procedure that operates on multi-view renders of the source and edited assets, covering three aspects:

- **Integrity filtering.** A VLM classifies each subject as a *Human* or an *Object* and scores its structural completeness under category-specific criteria, for instance applying stricter body and face requirements to humans while focusing on the presence and connectivity of major components for objects. Pairs whose edited asset fails to meet the integrity requirement are discarded.
- **Instruction annotation.** For the surviving pairs, we re-derive the editing instruction directly from the geometry rather than trusting the original prompt. The VLM compares the before and after assets, labels the edit as one of *Addition*, *Replacement*, or *Removal*, and produces a concise and a detailed instruction describing the most prominent geometric change first, constrained to geometry alone and ignoring color, material, and texture.
- **Verification.** We verify each annotated pair along four dimensions: the structural integrity of both assets, the alignment between the generated instruction and the

actual visible change, the geometric consistency of the non-edited regions, and the visibility of the edit under the given viewpoints.

Overall, Nano3D-v2 harmonizes the semantic flexibility of 2D generative models with the structural rigor of learned 3D spatial constraints. By bypassing the pitfalls of direct lifting and manual intervention, our pipeline provides a scalable solution for generating consistent, high-quality 3D editing pairs. Examples of editing pairs in the training corpus created by Nano3D-v2 are shown in Fig. 4. Multi-round editing is also supported by Nano3D-v2, as shown in Fig. 5.

3.5 Text-grounded Part Generation Data

Vision Banana [24] shows that casting dense perception (e.g., semantic, instance, and referring segmentation) as a conditional *generation* problem yields strong zero-shot transfer, making image generation a *universal interface* for heterogeneous tasks. This motivates the same recipe in 3D: *3D segmentation via instruction tuning on a 3D multimodal large language model (MLLM)*, where part understanding is one instruction-following *subtask* prompted in natural language (e.g., “segment the wheels”).

Unlike prior part generation methods [57, 97, 116], which are specialized pipelines relying on geometry-specific modules and bounding-box supervision, we treat part understanding as a native subtask of our unified 3D MLLM: we build instruction-tuning data pairing 3D assets with natural-language queries and part-level targets, so the same model used for understanding, generation, and editing can also localize and decompose parts on demand.

PartNeXt [80] provides part-level decompositions with semantic grounding, where each segmented region is linked to a part concept that can be referred to in language. Such labels can be directly used to build instruction-following data, e.g., “segment the wheels” or “remove the handle”. However, assets from datasets, such as HY3D-Bench [36], only provide raw mesh-level part decompositions. These parts are usually geometric pieces created during modeling or export, and they do not have semantic labels. They are also often over-segmented: one semantic part, such as a wheel, wing, handle, or head, may be split into several disconnected mesh components. As a result, each raw component is often too small or ambiguous to be named by a clear semantic noun, making it difficult to directly convert these decompositions into part-level instruction-tuning data.

To address this issue, we design a **semantic mesh merging tool** that turns over-segmented mesh components into semantic macro parts. Here, a macro part means a coarse semantic part group: it may contain multiple raw mesh components, but corresponds to one part concept that can be referred to in language. The tool consists of three stages: semantic part vocabulary discovery, component-to-part grounding and merging, and quality filtering with instruction packaging.

Stage 1: Semantic Part Vocabulary Discovery. For each object, we first render the full mesh from multiple canonical views and ask a VLM to infer a compact set of meaningful part concepts, such as wheels, wings, handles, or head. These concepts form an object-specific part vocabulary. We constrain the vocabulary to use simple and mutually exclusive names, so that each concept can be used directly in natural-language instructions.

Stage 2: Component-to-Part Grounding and Merging. We then render each raw mesh component with a red highlight from multiple views. For each highlighted component, we provide the VLM with the object-specific part vocabulary obtained in Stage 1 and ask which semantic part the component belongs to. The model selects from this closed vocabulary, with an extra *unmatched* option for unclear fragments. Components assigned to the same semantic concept are grouped together as one macro part. This step converts low-level geometric fragments into language-referable part targets.

Stage 3: Quality Filtering and Instruction Packaging. Before using the merged parts as training data, we re-render each macro part as a red-highlighted region on the full object and use a VLM to verify its quality. The model checks whether the highlighted region matches

the intended concept, whether the grouped components form a coherent part, and whether the part looks visually natural. We keep only clean object-part pairs that pass these checks. For each kept pair, we compute one normalization transform from the full object and apply the exact same scale and translation to all three exported assets: the full object, the macro part alone, and the remaining object after removing that macro part. These triplets are then converted into instruction-tuning samples for part segmentation and part removal, such as “segment the wheels from this vehicle model” and “remove the wheels from this vehicle model”.

4 Method

4.1 Architecture

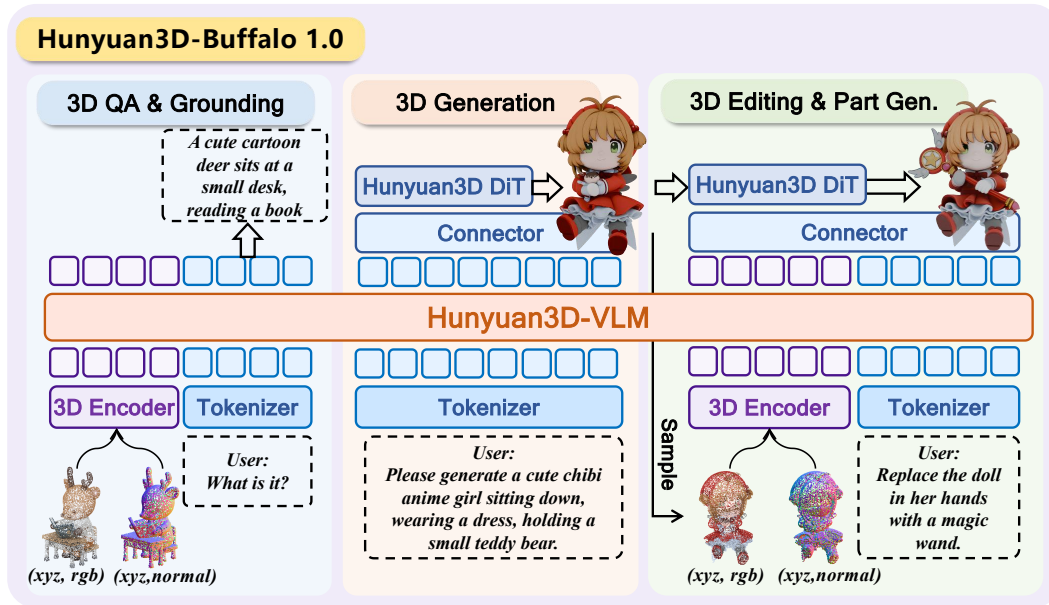


Fig. 6: **Hunyuan3D-Buffer 1.0 pipeline.** The framework unifies 3D QA and grounding, text-to-3D generation, and 3D editing through a shared Hunyuan3D-VLM backbone, which connects language, 3D representations, and generative Hunyuan3D DiT modules for multi-modal understanding, generation, and editing.

Inspired by hybrid frameworks such as Qwen-Image [86], our architecture synergizes two specialized modules: (1) Hunyuan3D-VLM for multimodal understanding and part-level reasoning, and (2) 3D-DiT (initialized from Hunyuan3D-2.1) for 3D synthesis. While the VLM serves as the semantic engine, the 3D-DiT functions as the generative module. To integrate them, a lightweight MLP-Connector aligns VLM hidden states with the DiT’s conditional space, ensuring high-level reasoning effectively guides the diffusion process without disrupting pretrained priors.

3D-aware Vision Language Model. To endow the VLM with fine-grained 3D perception, Hunyuan3D-VLM encodes 3D assets through a structure-and-appearance representation. Given a colored point cloud, the structural pathway processes geometric signals, including XYZ coordinates and surface normals, to capture object shape, spatial layout, and part boundaries. In parallel, the semantic pathway encodes RGB appearance cues, which helps the model distinguish parts that may be geometrically similar but visually different. The resulting 3D representations are encoded into latent tokens with a VecSet encoder. Before being injected into the VLM, these tokens are further compressed by a Q-Former into a fixed-length sequence of 512 tokens, enabling efficient fusion with text and image tokens.

To support explicit 3D grounding and part-level reasoning, we further augment the VLM vocabulary with 133 special tokens, following Part-X-MLLM [77]. Three of them, $\langle |point_start| \rangle$, $\langle |point_end| \rangle$, and $\langle |point_pad| \rangle$, delimit the encoded point-cloud token sequence and mark its placeholder positions within the multimodal input. The remaining tokens encode 3D bounding boxes: $\langle boxs \rangle$ and $\langle boxe \rangle$ act as box delimiters, while 128 discrete coordinate tokens, from $\langle box-0 \rangle$ to $\langle box-127 \rangle$, represent quantized coordinate values in the range $[0, 127]$. Each 3D bounding box is therefore represented as six quantized coordinate tokens wrapped by the box delimiters. A decoder-only transformer, initialized from a pretrained LLM, takes the fused sequence of structural, semantic, visual, and textual tokens as input and autoregressively predicts the textual and coordinate-token output. In this way, diverse 3D understanding tasks, including 3D captioning, 3D question answering, 3D grounding, edit-instruction synthesis, and edit-outcome captioning, are unified as an instruction-following sequence prediction problem.

Conditioning 3D-DiT with VLM Hidden States. For 3D generation, Hunyuan3D-VLM processes multimodal prompts—comprising interleaved text, images, and 3D shapes—to extract contextual hidden states. These high-level semantic features are then projected via the MLP-Connector into the 3D-DiT’s conditional embedding space. By initializing the 3D-DiT from Hunyuan3D-2.1, our architecture leverages robust 3D generative priors while utilizing the connector as a flexible interface for VLM-driven reasoning. This synergy effectively bridges the instruction-following and multimodal reasoning of an autoregressive VLM with the high-fidelity synthesis of a diffusion transformer.

3D Editing and Part Generation with Source-Object Conditioning. To ensure structural consistency during 3D editing and part generation, we condition the diffusion process on both VLM-derived semantic embeddings and the original object’s representation. Specifically, the source 3D representation is concatenated with the noisy latent map as input to the 3D-DiT’s self-attention layers. This explicit conditioning grants the model direct access to the original geometry during denoising, thereby facilitating the faithful preservation of unedited regions while enabling precise modifications to the target parts as instructed. Note that 3D editing and part generation share the same data preparation and training pipeline: both require a source shape along with an instruction (either an editing instruction or a part segmentation instruction), and their data flows during training are identical.

4.2 Training Procedure

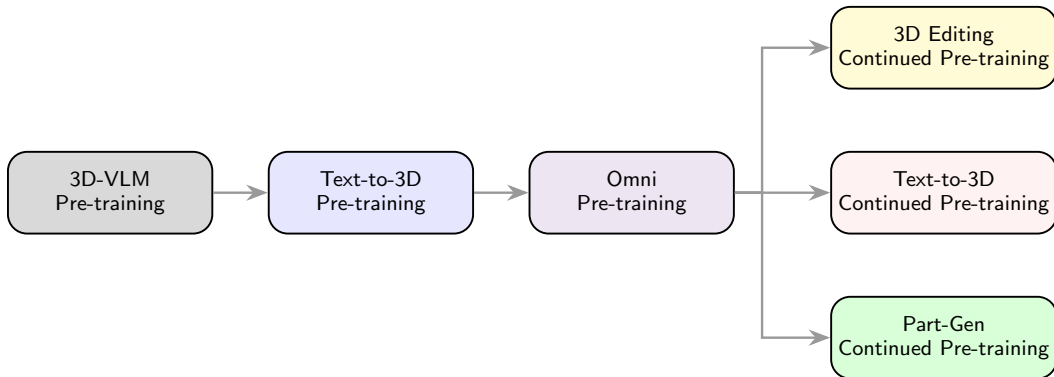


Fig. 7: Overview of the training pipeline. The shared trunk proceeds through 3D-VLM pre-training, text-to-3D pre-training, and omni pre-training. It then branches into three continued pre-training paths: 3D editing, text-to-3D, and part generation.

As illustrated in Fig. 7, we train our framework in four stages: 3D-VLM pre-training (Stage 1), text-to-3D pre-training (Stage 2), omni pre-training (Stage 3), and continued pre-training

(Stage 4) where the model branches into three task-specific paths for 3D editing, text-to-3D, and part generation respectively. All generative stages optimize the 3D-DiT with a flow-matching objective that predicts the velocity field transporting a Gaussian prior to the target 3D latent distribution.

Stage 1: 3D-VLM Training. We first endow Hunyuan3D-VLM with fine-grained 3D perception in two phases. In the *alignment* phase, we freeze the Qwen-VL[1] backbone and train only the 3D-token connector (VecSet encoder, Q-Former, and projection) to align 3D latent tokens with the language model’s input embedding space. In the *instruction-tuning* phase, we unfreeze the full model and jointly train it on text-to-text, image-to-text, 3D grounding, 3D captioning, 3D question answering, and edit-instruction synthesis. The resulting VLM serves as a unified semantic engine and is kept frozen in all subsequent stages, so that generative training focuses on the 3D-DiT and the connector without disturbing the learned 3D understanding.

Stage 2: Text-to-3D Pretraining. We then couple the VLM with the 3D-DiT for text-conditioned generation. The VLM hidden states are first processed by the MLP-Connector and then injected into the 3D-DiT, where the resulting per-token features condition the denoiser through cross-attention. Initialized from Hunyuan3D-2.1, the 3D-DiT is trained on our $\sim 50\text{M}$ text-asset pairs, establishing broad coverage of categories, structures, and compositions as a strong initialization for both branches.

Stage 3: Omni Pretraining. We pretrain the model on text-to-3D, 3D editing, and part generation tasks in a unified manner. Overall, the sampling ratio is set to text-to-3D : (3D editing + part generation) = 1 : 1. This balance is essential: it lets the editing capability emerge without sacrificing generation, which in turn underpins editing that generalizes beyond the training distribution. Since the editing and part generation corpus is far smaller than the text-to-3D one, we repeat the editing and part generation data by $4\times$ to match the text-to-3D data within each batch. Part generation is regarded as a special case of 3D editing, where the target region is the queried part itself.

Stage 4: Continued Pre-training. In the final stage, we decouple the three tasks and train them separately. For 3D editing and part generation, we still mix in half text-to-3D data during training to preserve generative quality. For text-to-3D, we train exclusively on text-to-3D data.

5 Experiments

5.1 3D Understanding

Evaluation setting. We evaluate the 3D understanding capability of Hunyuan3D-VLM on UniPart-Bench [102], a part-centric benchmark for structured 3D perception and language understanding. Each 3D object is represented as an RGB point cloud and annotated with axis-aligned part bounding boxes, part-level texts, object-level captions, and part-aware question-answer pairs. Following the benchmark setting, part annotations are provided at two semantic granularities: Q1 denotes the coarse part category or name, while Q2 denotes a fine-grained natural-language description of the part. For comparison, we include representative 3D multimodal baselines, including GPT4Point [63], PointLLM [93], ShapeLLM [62], ShapeLLM-Omni [102], Part-X-MLLM [77], and UniVerse3D [104].

Tasks and metrics. The all-task evaluation covers the understanding-related tasks in UniPart-Bench. Task 0, *Pure box listing*, requires the model to predict all part bounding boxes without text generation. Tasks 1–2 are *Multi-Part Grounding*, where the model outputs all part boxes together with Q1 labels or Q2 descriptions. Tasks 3–4 are *Single-Part Grounding*, where the model localizes a queried part specified by either a Q1 part name or a Q2 fine-grained description. Tasks 5–6 are *Box-to-Text*, where the model generates the corresponding Q1 label or Q2 description given a part box. Task 7 is *Part QA*, which evaluates part-level

question answering with grounding. We report IoU for bounding-box localization and use SBERT, SimCSE, BLEU-1, ROUGE-L, and METEOR for textual outputs.

Table 3: Comparison of part-level question answering and object-level captioning on UniPart-Bench [102]. The two tasks assess complementary aspects of 3D understanding, including localized part-aware reasoning and holistic object-level semantic description.

Model	Part Understanding Q&A					Overall 3D Object Captioning				
	SBERT	SimCSE	BLEU-1	ROUGE-L	METEOR	SBERT	SimCSE	BLEU-1	ROUGE-L	METEOR
GPT4Point [63]	48.32	45.17	15.16	22.55	16.19	25.60	27.00	11.50	12.00	12.70
PointLLM-7B [93]	61.30	58.48	21.78	29.26	22.45	42.79	42.44	11.58	14.39	16.90
PointLLM-13B [93]	56.36	51.47	21.40	29.16	21.80	43.51	43.12	13.54	15.74	17.45
ShapeLLM-13B [62]	61.19	57.26	23.32	32.56	24.45	25.15	27.14	11.77	12.14	12.84
ShapeLLM-Omni-7B [102]	57.35	51.16	22.77	29.57	23.24	31.18	31.93	17.79	19.04	14.30
Part-X-MLLM [77]	78.98	84.25	40.54	42.26	34.24	53.82	51.97	36.04	38.11	30.71
UniVerse3D [104]	83.11	87.16	46.79	43.94	42.05	65.18	66.25	42.75	44.17	41.11
Hunyuan3D-VLM (Ours)	85.47	89.06	49.95	45.01	45.79	72.94	73.60	50.93	52.84	50.47

Table 4: Detailed all-task evaluation of Hunyuan3D-VLM on UniPart-Bench [102]. The benchmark covers pure box listing, multi-part grounding, single-part grounding, box-to-text generation, and part-level question answering.

Task Name	IoU	SBERT	SimCSE	BLEU-1	ROUGE-L	METEOR
0 Pure box listing	0.864	-	-	-	-	-
1 Multi-Part Grounding (Q1)	0.880	68.00	68.55	52.06	52.09	26.35
2 Multi-Part Grounding (Q2)	0.844	70.92	69.47	40.04	41.86	38.19
3 Single-Part Grounding (Q1)	0.626	78.95	77.92	45.74	47.47	44.07
4 Single-Part Grounding (Q2)	0.525	-	-	-	-	-
5 Box-to-Text (Q1)	-	67.64	68.27	49.89	50.00	25.45
6 Box-to-Text (Q2)	-	74.13	72.99	42.01	44.28	40.96
7 Part QA	0.633	85.47	89.06	49.95	45.01	45.79

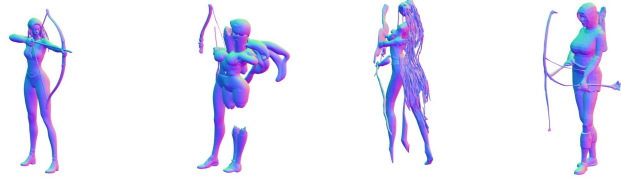
Results. As shown in Table 3, Hunyuan3D-VLM consistently achieves the best performance on both part-level Q&A and overall 3D object captioning. For part understanding Q&A, our model obtains 85.47 SBERT and 89.06 SimCSE, with notably stronger lexical scores as well, including 49.95 BLEU-1 and 45.79 METEOR. These results indicate that Hunyuan3D-VLM can produce semantically accurate and linguistically faithful answers for fine-grained part-centric questions. For holistic object captioning, our model further reaches 72.94 SBERT and 52.84 ROUGE-L, showing a clear advantage over previous 3D MLLMs in object-level semantic description. Overall, the results demonstrate that Hunyuan3D-VLM improves 3D understanding at both localized part level and global object level.

Table 4 further reports the detailed performance across UniPart-Bench tasks. Hunyuan3D-VLM obtains 0.864 IoU on pure box listing and achieves strong grounding performance on both multi-part and single-part settings. In addition, the model produces competitive language scores for box-to-text generation and part QA, showing its ability to unify part localization, region-conditioned description, and part-aware reasoning within a single 3D understanding framework.

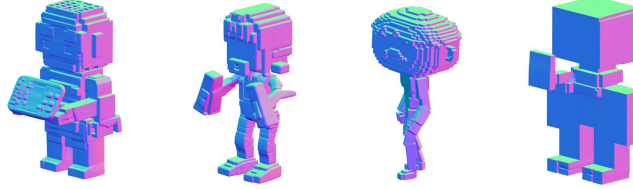
5.2 Text-to-3D

We conduct a human evaluation to compare our method against three baselines (TRELIS [92], Universe3D [104], and Omni123 [100]). All methods are evaluated on the same set of 100 text prompts, producing 100 four-way comparison groups. To ensure the evaluation is unbiased and representative, the prompts are deliberately diverse in both content and length: they span a wide range of object categories—including characters and creatures, everyday objects, weapons and tools, furniture and decorative items, and clothing—and vary from short, concise descriptions (as few as 2 words, e.g., “Sitting cat.”) to long, fine-grained specifications (up to 54 words), with an average length of roughly 28 words. About

This slender female archer stands in a confident, upright pose while holding a large, curved bow. Her athletic form features a simple top, fitted trousers, and a voluminous hairstyle that flows behind her shoulders.



This blocky, voxel-style character figure features a rigid, geometric silhouette with an oversized head and modular limbs. The figure stands in a casual posture while holding a rectangular handheld video game console equipped with a central screen and various control buttons.



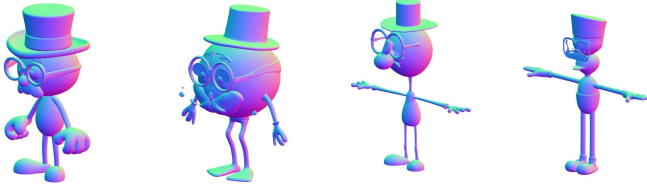
This 3D figure depicts an upright character in a mid-calf cinched by a wide belt with a circular buckle. The character wears a pointed, hooded garment and holds a tall staff topped with a circular ring and a diamond-shaped point.



This bipedal, humanoid monster features a segmented, armored body with a thick, curved tail and prominent sharp ridges along its back. It is distinctively characterized by a pair of oversized sunglasses resting on its reptilian face.



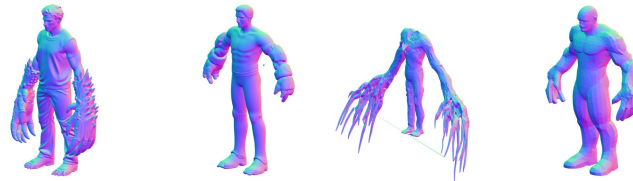
This cartoon character features a large, segmented spherical head topped with a cylindrical hat, accented by thick-rimmed circular glasses and a voluminous mustache. It has a slender torso, long thin arms with wide, fanned hands, and simple legs ending in bulbous feet.



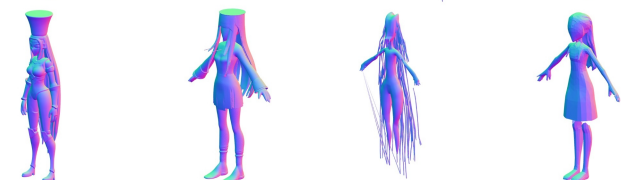
This regal knight figurine features a long, sweeping cloak and distinctive, tiered pauldrons. Its imposing silhouette is defined by a headpiece topped with a cluster of spherical ornaments on thin stalks.



This 3D model features a male figure in a neutral, human-like stance wearing a t-shirt and trousers. His hands are replaced by massive, predatory gauntlets characterized by segmented, bony plating and sharp, downward-curving talons.



This stylized, slender humanoid character features a clean, minimalist silhouette with a prominent, monolithic cylindrical headpiece and long, flowing hair. The figure is defined by sharp, angular transitions, featuring flared wrist cuffs and knee-length leg coverings that emphasize its elongated proportions.



Ours

Omni123

Universe3D

Trellis

Fig. 8: Qualitative text to 3D results.

one fifth of the prompts are short tags, while the remainder are medium-to-long detailed descriptions, so that the comparison reflects performance across both coarse and highly specific inputs. For each prompt, the four generated 3D assets are presented anonymously and in randomized order to avoid position and brand bias. Participants are asked to choose the best result along three dimensions: (i) text alignment, i.e. how faithfully the asset matches the input prompt; (ii) geometry quality, i.e. the quality and plausibility of the 3D shape; and (iii) overall preference. For each method and dimension, we report the proportion of comparisons in which it was preferred.

Table 5: Results of the user study for text-to-3D generation. We report the *preference rate* (%), i.e., the percentage of comparisons in which each model is selected as the best among the four candidates. In each comparison, the four results are shown side by side in randomized order, and participants pick the single best result for each criterion (a “tie” option is also allowed, so columns may sum to slightly below 100%). Higher is better, with a random-choice baseline of 25%. Best results are shown in **bold**.

Model	Text alignment	Geometry quality	Overall preference
Universe3D [104]	8.2	7.4	8.3
TRELLIS [92]	14.9	12.4	14.4
Omni123 [100]	17.5	21.0	18.4
Hunyuan3D-Buffalo 1.0 (Ours)	55.2	57.1	56.6

Table 6: Ablation on the quantity of training data. We report the *preference rate* (%), i.e., the percentage of comparisons in which a model is chosen as the best among the three variants. In each comparison, the three results are shown side by side in randomized order, and participants select the best result for each criterion (a “tie” option is also allowed, so columns may sum to slightly below 100%). Higher is better, and the random-choice baseline is 33.3%. Best results are shown in **bold**.

Num. of samples	Text alignment	Geometry quality	Overall preference
300w	9.9	8.8	8.4
1500w	28.8	29.0	28.6
5000w	54.5	57.4	57.5

As shown in Table 5, our method is preferred in the clear majority of comparisons across all three dimensions, attaining preference rates of 55.2%, 57.1%, and 56.6% for text alignment, geometry quality, and overall preference, respectively. These rates are more than double those of the strongest baseline, Omni123 [100] (17.5%, 21.0%, and 18.4%), and far exceed the 25% random-choice level, while TRELLIS [92] and Universe3D [104] trail further behind. The margin is largest on geometry quality, indicating that our method produces noticeably more faithful and higher-fidelity 3D shapes, and it remains the most preferred for text alignment, showing that this geometric gain does not come at the cost of prompt faithfulness. Notably, the ranking is consistent across every individual participant and holds for both the short tag-style prompts and the longer fine-grained descriptions, suggesting that the advantage is robust to the diversity of input content and length rather than driven by a particular subset of cases.

Ablation of different quantity of training data Table 6 ablates the effect of pretraining data scale, comparing models pretrained on 300w (3M), 1500w (15M), and 5000w (50M) samples. We observe a clear and monotonic improvement in human preference as the pretraining corpus grows: the preference rate for overall quality rises from 8.4% (300w) to 28.6% (1500w) and to 57.5% (5000w), with the same trend holding for text alignment (9.9% $\rightarrow$ 28.8% $\rightarrow$ 54.5%) and geometry quality (8.8% $\rightarrow$ 29.0% $\rightarrow$ 57.4%). Notably, the model trained on the full 5000w data is preferred in well over half of all comparisons—roughly twice as often as the 1500w model and more than six times as often as the 300w model—and this ordering is consistent across every participant and all three evaluation

Table 7: **Quantitative comparison of language-guided 3D shape editing on Edit3D-Bench [85].** Both CLIP-conditioned and 3D-VLM-conditioned variants of Hunyuan3D-Buffalo 1.0 are included to analyze the effect of stronger 3D instruction understanding.

Method	Add		Remove		Avg	
	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
ShapeLLM-Omni [102]	0.2546	0.0877	0.2237	0.1166	0.2392	0.1022
3DEditFormer [90]	0.1676	0.1955	0.1342	0.1836	0.1509	0.1896
Tailor3D [64]	0.1661	0.1217	0.1755	0.1352	0.1708	0.1285
Steer3D [85]	0.1404	0.2414	0.0976	0.3044	0.1190	0.2729
Omni123 [100]	0.0736	0.1743	0.0632	0.2259	0.0684	0.2001
Hunyuan3D-Buffalo 1.0 w/ CLIP (Ours)	0.0154	0.5657	0.0162	0.7015	0.0158	0.6336
Hunyuan3D-Buffalo 1.0 w/ 3D-VLM (Ours)	0.0127	0.5610	0.0054	0.7420	0.0091	0.6515

criteria. The gains are pronounced on both text alignment and geometry quality, indicating that larger-scale pretraining improves not only how faithfully the generated 3D assets follow the input prompt but also the fidelity and plausibility of the underlying geometry. These results highlight that scaling up pretraining data is a key driver of generation quality, and that performance has not yet saturated, suggesting that further data scaling may yield additional improvements.

5.3 3D Editing

As the most direct and natural interface for 3D creation, language instruction-driven 3D editing is a pivotal capability for a wide range of 3D applications. We evaluate our model on Edit3D-Bench [85], using its curated source-target mesh pairs for geometric addition and removal operations. The objective is to perform semantically faithful and structurally coherent modifications that closely follow the language instruction, while keeping the unedited regions of the source mesh as intact as possible.

To quantify editing fidelity, we report Chamfer Distance (CD) and F1 score (F1) between the generated and ground-truth edited meshes. Each test sample requires the model to translate a natural-language directive into a precise, localized geometric transformation. We compare against five recent 3D editing methods: ShapeLLM-Omni [102], Steer3D [85], 3DEditFormer [90], Tailor3D [64], and Omni123 [100]. This selection spans both native autoregressive editors and optimization-based pipelines.

Quantitative Comparison on Edit3D-Bench. Table 7 reports the quantitative results on add and remove editing tasks. Hunyuan3D-Buffalo 1.0 significantly outperforms previous 3D editing methods across both CD and F1 metrics. Compared with the strongest baseline in terms of average CD, Omni123 [100], our 3D-VLM version reduces the average CD from 0.0684 to 0.0091, corresponding to an 86.7% relative reduction. For average F1, our model improves over the strongest baseline Steer3D [85] from 0.2729 to 0.6515, achieving a $2.39\times$ improvement. These improvements are consistent across both addition and removal tasks, where Hunyuan3D-Buffalo 1.0 w/ 3D-VLM achieves 0.0127/0.5610 and 0.0054/0.7420 in CD/F1, respectively, demonstrating accurate localized editing and strong preservation of the remaining geometry.

We also compare two conditioning variants of Hunyuan3D-Buffalo 1.0: the CLIP-based version and the 3D-VLM-based version. The 3D-VLM version achieves a lower average CD than the CLIP version, reducing it from 0.0158 to 0.0091, while improving the average F1 from 0.6336 to 0.6515. This indicates that the 3D-VLM-conditioned model follows editing instructions more accurately and produces finer geometric details.

Qualitative Analysis of Localized and Consistent Editing. As shown in Fig. 9, our method performs localized geometric edits while preserving the overall structure, pose, and fine-grained details of the input shape. For addition tasks, such as adding glasses, a rugged

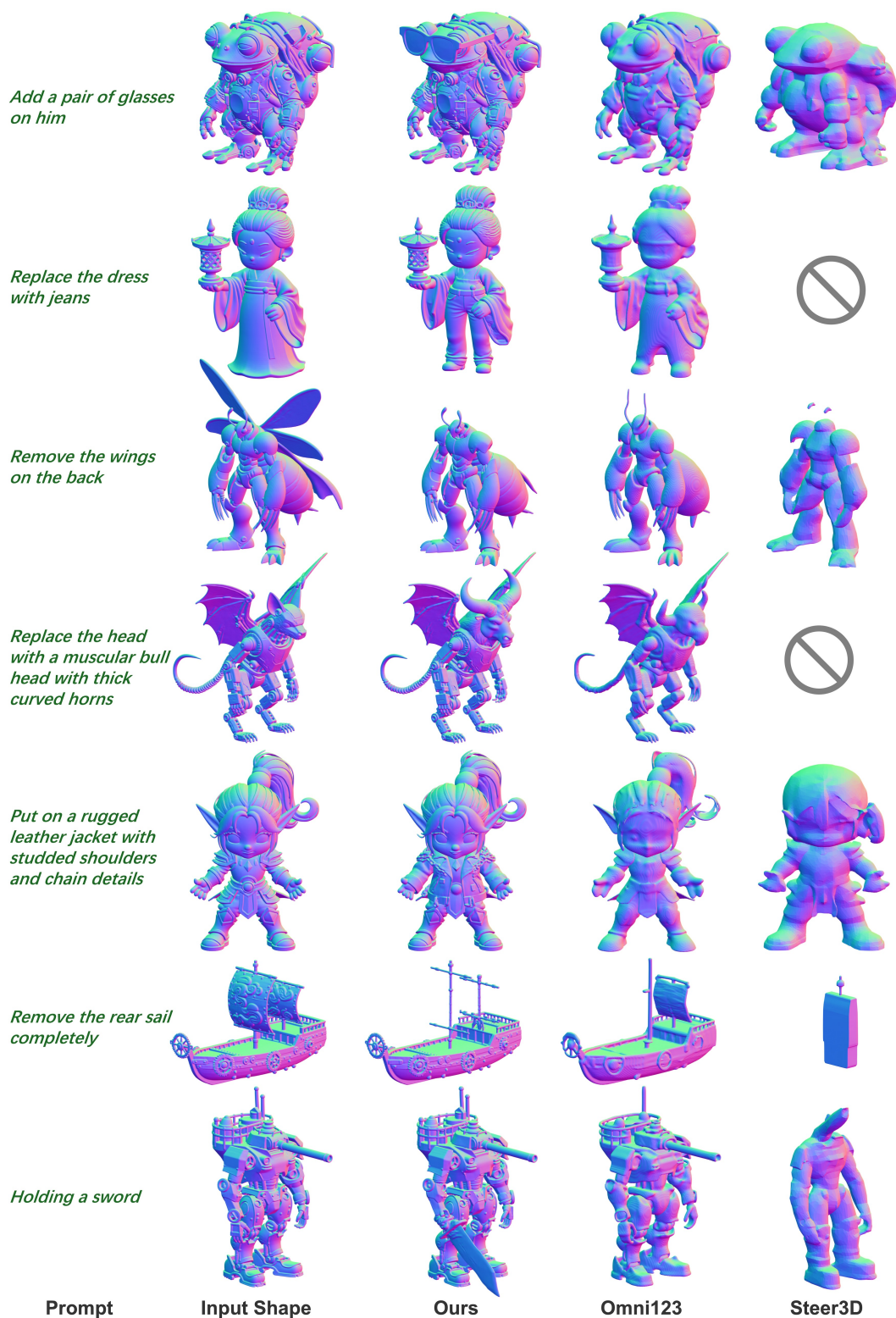


Fig. 9: **Qualitative shape editing results.** Our method significantly outperforms all base-lines in both geometric consistency before and after editing, and responsiveness to editing instructions.

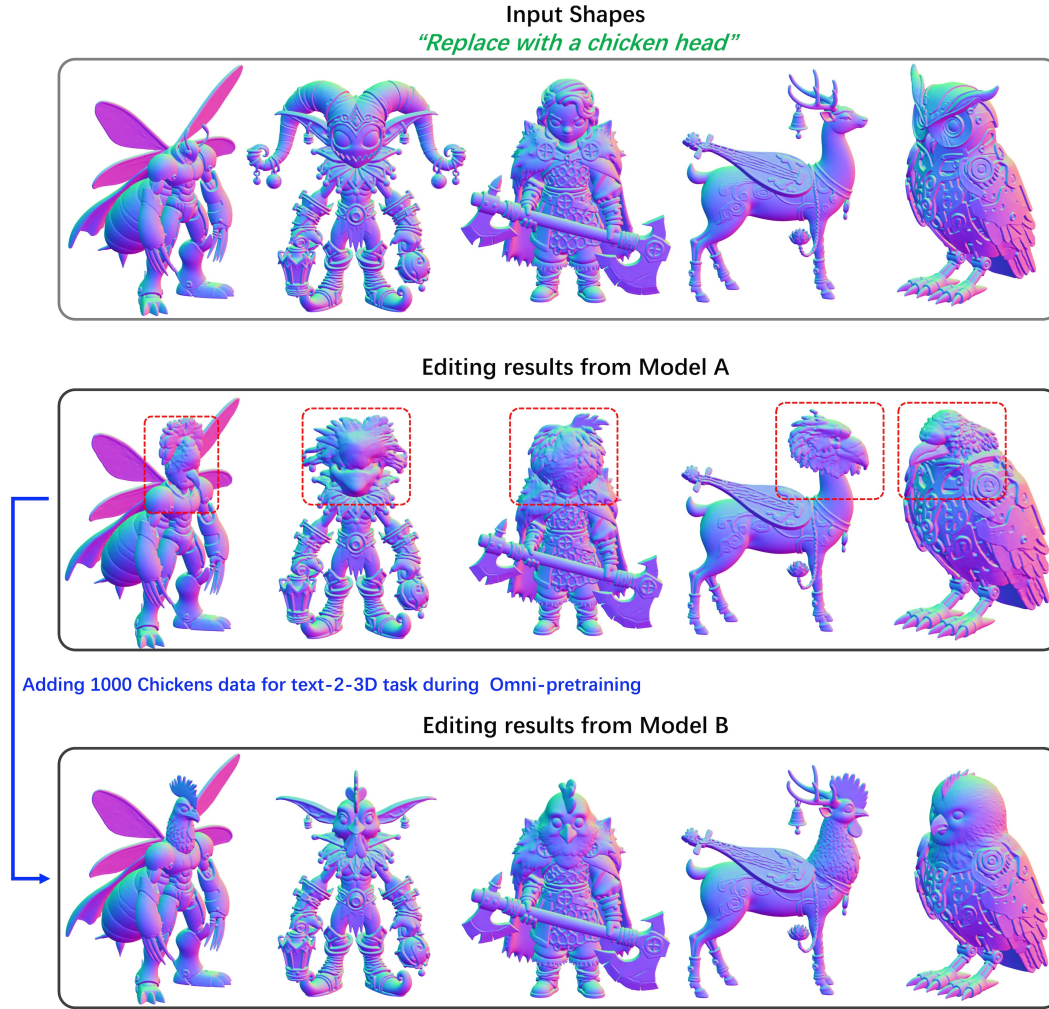


Fig. 10: **Scaling up text-to-3d data facilitates 3D editing.** Model A is our base model; when instructed to edit an object by replacing its head with a chicken head, it fails to produce a satisfactory result. Model B is built upon Model A by adding only 1,000 additional chicken samples for the text-to-3D task during the **Omni pre-training stage**—crucially, without introducing any new editing data. After incorporating this text-to-3D data, the model can successfully replace the head with a chicken head. This suggests a clear direction: to improve 3D editing, the text-to-3D generation capability should be maximized as much as possible. Since constructing text-to-3D data is far less costly than constructing 3D editing data, scaling up text-to-3D data is a relatively more feasible path toward stronger 3D editing.

leather jacket, or a sword, our model introduces the target component at the correct semantic location without disrupting unrelated regions. For removal tasks, such as removing the rear sail or the wings on the back, the edited results retain the main body geometry and avoid excessive deformation. In contrast, Omni123 [100] often produces over-smoothed shapes or incomplete edits, losing detailed geometry in the source mesh. Steer3D [85] shows less stable behavior: in several cases, it either fails to produce a valid result or changes the global structure of the object instead of applying a localized edit. These qualitative results indicate that our model better balances instruction following and source-shape preservation, which is essential for practical 3D editing.

Scaling Text-to-3D Data for Better Shape Editing. To better understand the relationship between 3D editing and 3D generation, we conduct a deeper investigation into how the

two interact. We find that 3D editing fundamentally relies on a strong text-to-3D generative foundation: when the underlying text-to-3D generation is weak, it is difficult to achieve strong editing capability. Figure 10 illustrates this with a concrete example. By adding more text-to-3D training data, the corresponding editing ability simultaneously emerges without additional editing data. This suggests a clear direction: to improve 3D editing, the text-to-3D generation capability should be maximized as much as possible. Since constructing text-to-3D data is far less costly than constructing 3D editing data, scaling up text-to-3D data is a relatively more feasible path toward stronger 3D editing.

5.4 Part Generation

As illustrated in Fig. 11, our method supports open-vocabulary, text-grounded part generation across a diverse range of objects. For each object, a natural-language query specifying the desired part is sufficient to accurately extract the corresponding geometry. The extracted parts exhibit high geometric fidelity to the input shape, whether the target is a structurally complex object such as an octopus or a geometrically simple one such as a wheel. When multiple parts are extracted, they can be reassembled to yield a compositional, part-by-part generation result. Notably, although our approach delivers open-vocabulary text-grounded part extraction, it is treated as a sub-task within a unified multimodal large language model rather than as a dedicated pipeline, demonstrating the generality of our framework.

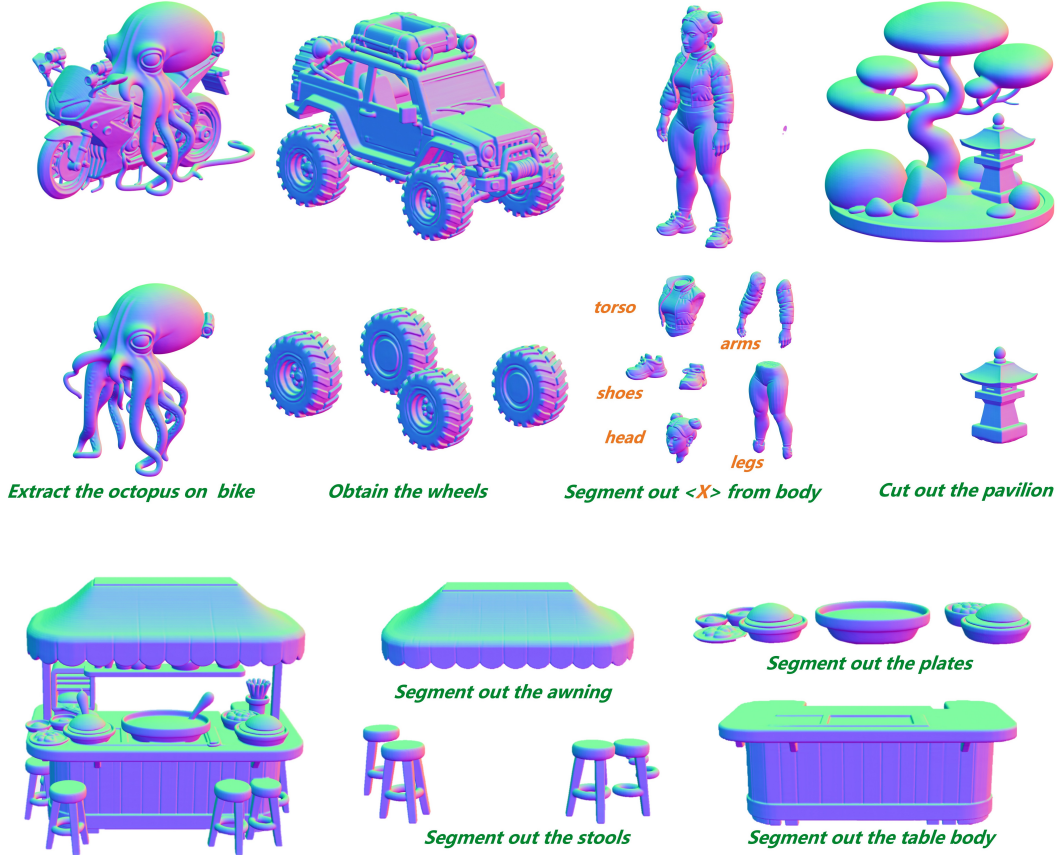


Fig. 11: Qualitative part generation results.

6 Conclusion & Future Work

In this work, we present a unified multimodal 3D model that jointly addresses 3D understanding, 3D generation, and 3D editing within a single framework. Our model achieves state-of-the-art performance across all three tasks, and notably surpasses prior methods by a significant margin on both 3D generation and 3D editing. We believe that unified 3D models represent a promising future direction, with the potential to address multiple 3D tasks within a single framework. However, several key challenges remain to be solved before this vision can be fully realized.

1. **Single-stage high-quality geometry representation.** The representation for 3D generation has not yet converged. To achieve high-quality geometry generation, current approaches often resort to multi-stage DiT pipelines—for example, TRELIS [92] employs a multi-stage architecture. This multi-stage design poses significant challenges for unified models: if we wish to perform editing, we must also carry out multi-stage editing to obtain high-quality results, which is inherently difficult to scale up. Whether the 3D domain can achieve single-stage high-quality geometry generation—analogous to what has been accomplished in the image domain—remains an open and fundamental question.
2. **Text-to-3D & 3D editing dataset captioning quality.** For text-to-3D generation, we currently rely on multimodal large language models such as Gemini for 3D asset captioning. However, these models still produce considerable ambiguity in their descriptions, leading to noisy training pairs in the text-to-3D dataset. We believe this issue will be progressively alleviated as multimodal large language models continue to advance.
3. **End-to-end texture editing.** Our current model primarily focuses on geometry editing. Texture editing remains largely unexplored due to the lack of suitable training data. This also raises a broader question: whether geometry and texture representations can be jointly modeled, and whether a unified single-stage representation for both high-quality geometry and texture can be achieved.
4. **Robust editing data construction pipeline.** Our current editing data construction pipeline, based on Nano3D-v2, maintains consistency by replacing tokens outside a box mask in the editing region. While this effectively preserves consistency in the unmasked exterior, the interior of the mask often contains non-edited regions whose consistency is difficult to guarantee. Such inconsistencies propagate into the end-to-end model during training, degrading editing quality. Developing a more robust and precise editing data construction pipeline remains an important open problem.
5. **New architecture exploration.** Our current framework adopts a cascaded AR + DiT architecture. In the image and video generation domains, Transfusion-style architectures have demonstrated remarkable effectiveness by deeply fusing information across multiple modalities. Exploring such architectures for 3D generation is a natural and promising next step.
6. **Data scaling.** We observe that both the quantity and quality of 3D data have not yet reached an ideal scale. Continuing to scale up the data—in both volume and quality—remains a critical direction for further improvement.

Acknowledgement We thank the authors of **Omni123** for providing the results. We thank **Zehuan Huang** for his contribution during his internship in Hunyuan.

7 Author List

Junliang Ye^{*}, Kenkun Liu^{*}, Guocun Wang^{*}, Yang Li^{*†}, Yansong Qu^{*}, Chunshi Wang^{*},
Jingwei Xu, Yunhan Yang, Zibo Zhao, Jiachen Xu, Jiaao Yu, Lifu Wang, Zhihao Liang,
Zhuo Chen[†], Chunchao Guo[†]

[†] **Project Leaders:** Yang Li, Zhuo Chen, and Chunchao Guo

^{*} **Core Contributors**

3D Editing: Junliang Ye, Guocun Wang, Yansong Qu, Yang Li, Chunshi Wang, and Kenkun Liu

Text-to-3D: Kenkun Liu, Junliang Ye, and Yang Li

3D Understanding: Guocun Wang, Junliang Ye, Kenkun Liu, and Yang Li

A Additional Results

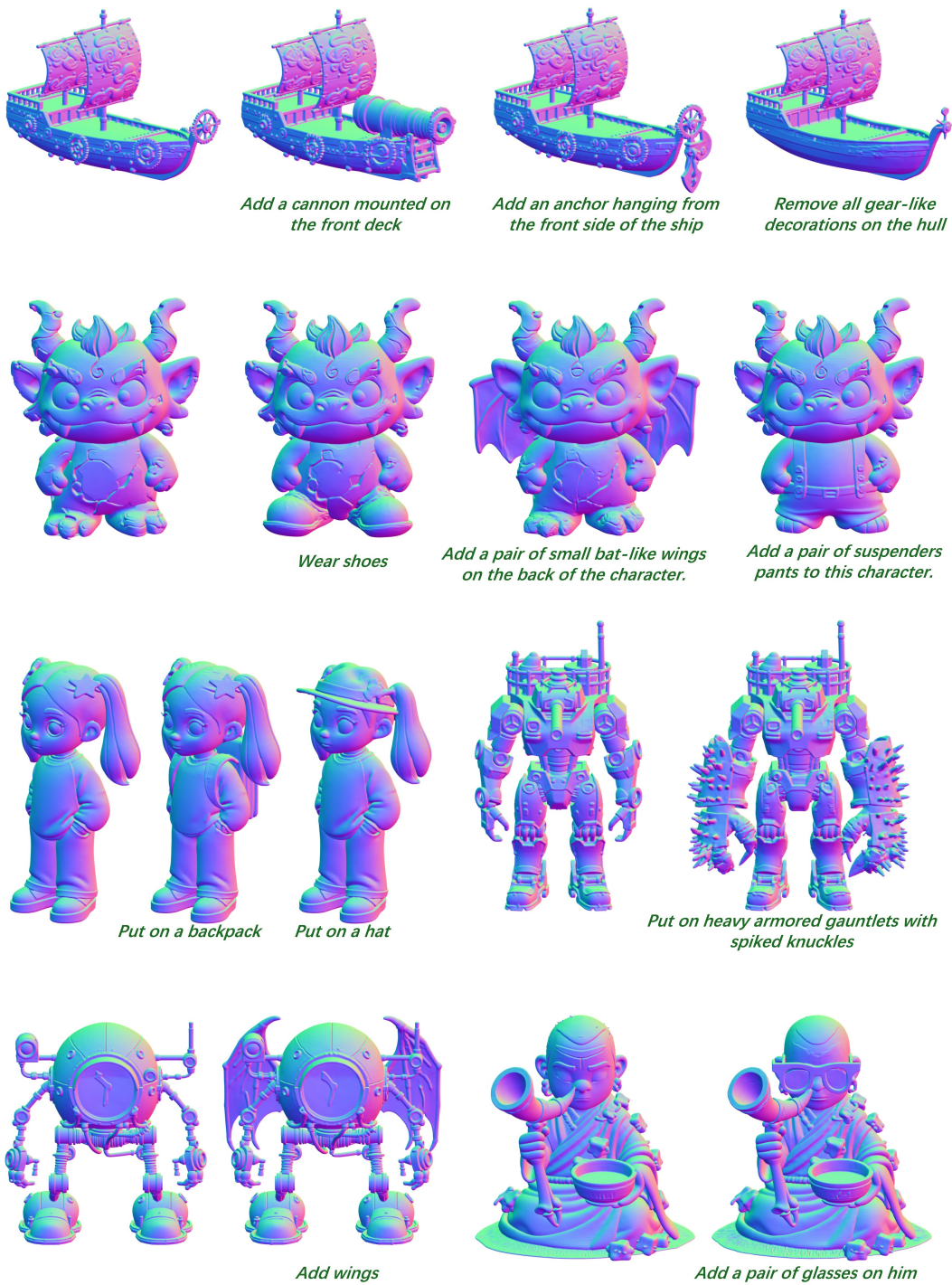
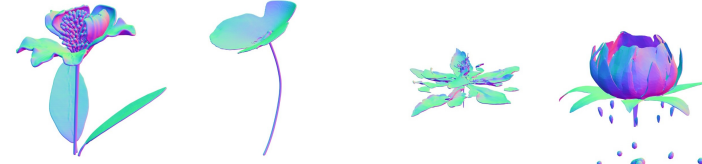
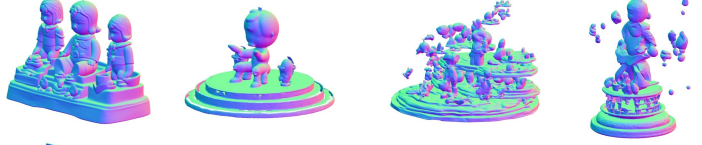


Fig. 12: Qualitative shape editing results.

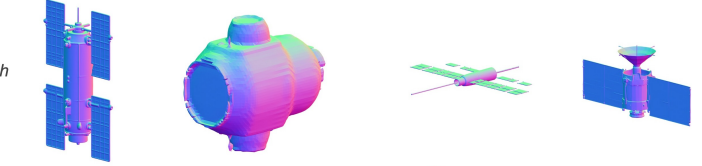
A 3D-rendered flower with a spherical stamen cluster, five curved petals with undulating edges, two symmetrical lanceolate leaves, and a slender vertical stem.



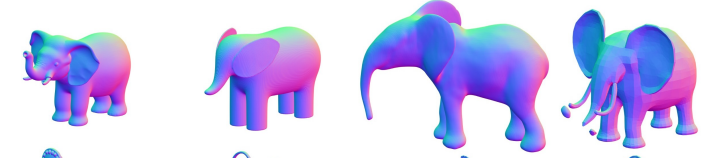
A collectible figurine assembly with chibi-style characters and objects on a tiered base, including a central seated figure, flanking figures with accessories, and floating or elevated elements above.



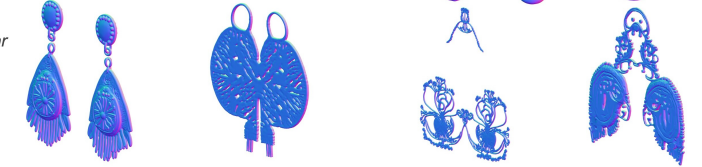
A satellite with a central cylindrical body, rod-like protrusions at one end, and two symmetrical lateral solar panel arrays each made of three rectangular panels, shown in a front-left diagonal view with clear separation between parts.



A stylized 3D elephant with a rounded body, short upward-curving trunk, large coiled ear showing inner spiral detail, and small domed tail, posed in left three-quarter view without visible legs.



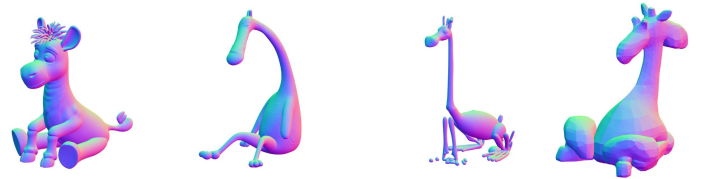
Symmetrical ornate earrings with a circular stud featuring a floral cutout, linked to a crescent pendant that has a beaded ring, openwork disc with radial filigree, central five-petaled flower, and beaded fringe.



A stylized 3D crab with a domed, segmented cephalothorax, pronounced rostral spine, symmetrical large chelipeds ending in serrated claws, eight radially arranged articulated walking legs, and a folded segmented abdomen partially visible beneath.



A cartoonish giraffe in a relaxed seated pose with bent forelimbs, splayed hindlimbs, an upward-tilted head, exaggerated proportions including an elongated neck and compact torso, and clearly defined features like large round eyes, a pink oval mouth, small ossicones, and a tufted tail.



A single-engine propeller aircraft with a streamlined fuselage, rounded nose, multi-blade propeller, framed cockpit canopy, tapered rear ending in tail surfaces, fixed landing gear, low-mounted wings with slight dihedral, and visible panel lines and rivets.



A stylized 3D humanoid in a dynamic martial arts stance with a bald head, long braid with tassels, mustache, expressive face, loose robe with cuff and hem trim, wide sash, raised left leg showing footwear, right arm extended forward palm open, left arm bent near torso.



A stylized 3D dog in a low crouch pose with exaggerated proportions, featuring a floral crown, neck bandana, white markings on muzzle, chest, paws, and tail tip, and visible joint articulation at shoulders and hips.



Ours

Omni123

Universe3D

Trellis

Fig. 13: Qualitative text to 3D results.

References

- [1] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [2] R. Bar-On, D. Cohen-Bar, and D. Cohen-Or. Editp23: 3d editing via propagation of image prompts to multi-view, 2025. URL <https://arxiv.org/abs/2506.20652>.
- [3] A. Barda, M. Gadelha, V. G. Kim, N. Aigerman, A. H. Bermano, and T. Groueix. Instant3dit: Multiview inpainting for fast editing of 3d objects, 2024. URL <https://arxiv.org/abs/2412.00518>.
- [4] K. Bhat, N. Khanna, K. Channa, T. Zhou, Y. Zhu, X. Sun, C. Shang, A. Sudarshan, M. Chu, D. Li, et al. Cube: A roblox view of 3d intelligence. *arXiv preprint arXiv:2503.15475*, 2025.
- [5] W. Cai, S. Fang, W. Ye, X. Dong, Y. Yang, X. Zhang, W. Cheng, Y. Cao, G. Yu, and T. Chen. Native 3d editing with full attention. *arXiv preprint arXiv:2511.17501*, 2025.
- [6] C. Cao, C. Yu, F. Wang, X. Xue, and Y. Fu. Mvinpainter: Learning multi-view consistent inpainting to bridge 2d and 3d editing, 2024. URL <https://arxiv.org/abs/2408.08000>.
- [7] S. Cao, H. Chen, P. Chen, Y. Cheng, Y. Cui, X. Deng, Y. Dong, K. Gong, T. Gu, X. Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.
- [8] H. Chen, R. Shi, Y. Liu, B. Shen, J. Gu, G. Wetzstein, H. Su, and L. Guibas. Generic 3d diffusion adapter using controlled multi-view editing, 2024. URL <https://arxiv.org/abs/2403.12032>.
- [9] J. Chen, Z. Xu, X. Pan, Y. Hu, C. Qin, T. Goldstein, L. Huang, T. Zhou, S. Xie, S. Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.
- [10] J. Chen, L. Xue, Z. Xu, X. Pan, S. Yang, C. Qin, A. Yan, H. Zhou, Z. Chen, L. Huang, et al. Blip3o-next: Next frontier of native image generation. *arXiv preprint arXiv:2510.15857*, 2025.
- [11] L. Chen, P. Wang, G. Zhang, Z. Ma, and L. Zhang. Omni-3dedit: Generalized versatile 3d editing in one-pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12640–12650, 2026.
- [12] M. Chen, J. Xie, I. Laina, and A. Vedaldi. Shap-editor: Instruction-guided latent 3d editing in seconds, 2023. URL <https://arxiv.org/abs/2312.09246>.
- [13] R. Chen, Y. Chen, N. Jiao, and K. Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22246–22256, 2023.
- [14] X. Chen, Z. Wu, X. Liu, Z. Pan, W. Liu, Z. Xie, X. Yu, and C. Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.
- [15] Y. Chen, T. He, D. Huang, W. Ye, S. Chen, J. Tang, X. Chen, Z. Cai, L. Yang, G. Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024.
- [16] Y. Chen, Z. Li, Y. Wang, H. Zhang, Q. Li, C. Zhang, and G. Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention. *arXiv preprint arXiv:2507.17745*, 2025.
- [17] Y. Chen, Y. Wang, Y. Luo, Z. Wang, Z. Chen, J. Zhu, C. Zhang, and G. Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13922–13931, 2025.
- [18] Y. Chi, X. Li, Z. Huang, and J. M. Rehg. Vinedresser3d: Towards agentic text-guided 3d editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12673–12683, 2026.
- [19] Y. Cui, H. Chen, H. Deng, X. Huang, X. Li, J. Liu, Y. Liu, Z. Luo, J. Wang, W. Wang, Y. Wang, C. Wang, F. Zhang, Y. Zhao, T. Pan, X. Li, Z. Hao, W. Ma, Z. Chen, Y. Ao, T. Huang, Z. Wang, and X. Wang. Emu3.5: Native multimodal models are world learners, 2025. URL <https://arxiv.org/abs/2510.26583>.
- [20] C. Deng, D. Zhu, K. Li, C. Gou, F. Li, Z. Wang, S. Zhong, W. Yu, X. Nie, Z. Song, G. Shi, and H. Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.
- [21] N. A. Dinh, I. Lang, H. Kim, O. Stein, and R. Hanocka. Geometry in style: 3d stylization via surface normal deformation, 2025. URL <https://arxiv.org/abs/2503.23241>.

- [22] S. Dong, L. Ding, Z. Huang, Z. Wang, T. Xue, and D. Xu. Interactive3d: Create what you want by interactive 3d generation, 2024. URL <https://arxiv.org/abs/2404.16510>.
- [23] Z. Erkoç, C. Gümeli, C. Wang, M. Nießner, A. Dai, P. Wonka, H.-Y. Lee, and P. Zhuang. Predictor3d: Fast and precise 3d shape editing, 2024. URL <https://arxiv.org/abs/2412.06592>.
- [24] V. Gabeur, S. Long, S. Peng, P. Voigtlaender, S. Sun, Y. Bao, K. Truong, Z. Wang, W. Zhou, J. T. Barron, K. Genova, N. Kannen, S. Ben, Y. Li, M. Guo, S. Yogin, Y. Gu, H. Chen, O. Wang, S. Xie, H. Zhou, K. He, T. Funkhouser, J.-B. Alayrac, and R. Soricut. Image generators are generalist vision learners. *arXiv preprint arXiv:2604.20329*, 2026.
- [25] W. Gao, D. Wang, Y. Fan, A. Bozic, T. Stuyck, Z. Li, Z. Dong, R. Ranjan, and N. Sarafianos. 3d mesh editing using masked lrms, 2024. URL <https://arxiv.org/abs/2412.08641>.
- [26] Y. Gao, L. Gong, Q. Guo, X. Hou, Z. Lai, F. Li, L. Li, X. Lian, C. Liao, L. Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.
- [27] I. Gat, D. Cohen-Bar, G. Levy, E. Richardson, and D. Cohen-Or. Shapeup: Scalable image-conditioned 3d editing. *arXiv preprint arXiv:2602.05676*, 2026.
- [28] Z. Hao, D. W. Romero, T.-Y. Lin, and M.-Y. Liu. Meshton: High-fidelity, artist-like 3d mesh generation at scale. *arXiv preprint arXiv:2412.09548*, 2024.
- [29] A. Haque, M. Tancik, A. A. Efros, A. Holynski, and A. Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023.
- [30] X. He, Z.-X. Zou, C.-H. Chen, Y.-C. Guo, D. Liang, C. Yuan, W. Ouyang, Y.-P. Cao, and Y. Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14822–14833, 2025.
- [31] T.-F. Hsiao, B.-K. Ruan, Y.-L. Liu, and H.-H. Shuai. Vecset-edit: Unleashing pre-trained lrm for mesh editing from single image. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2026.
- [32] S. Hu, Y. Wei, F. Zha, Y. Guo, and J. Zhang. Easy3e: Feed-forward 3d asset editing via rectified voxel flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12730–12740, 2026.
- [33] Z. Huang, D. Zheng, C. Zou, R. Liu, X. Wang, K. Ji, W. Chai, J. Sun, L. Wang, Y. Lv, T. Huang, J. Liu, Q. Guo, M. Yang, J. Chen, and J. Zhou. Ming-univision: Joint image understanding and generation with a unified continuous tokenizer. *arXiv preprint arXiv:2510.06590*, 2025.
- [34] T. Hunyuan3D. Hunyuan3d studio: End-to-end ai pipeline for game-ready 3d asset generation. *arXiv preprint arXiv:2509.12815*, 2025.
- [35] T. Hunyuan3D, B. Zhang, C. Guo, H. Liu, H. Yan, H. Shi, J. Huang, J. Yu, K. Li, P. Wang, et al. Hunyuan3d-omni: A unified framework for controllable generation of 3d assets. *arXiv preprint arXiv:2509.21245*, 2025.
- [36] T. Hunyuan3D, :, B. Zhang, C. Guo, D. Guo, H. Liu, H. Yan, H. Shi, J. Yu, J. Xu, J. Huang, K. Li, L. Wang, Linus, P. Wang, Q. Lin, R. Tang, X. Yang, Y. Li, Y. Guan, Y. Zhao, Y. Yang, Z. Lai, Z. Liang, and Z. Zhao. Hy3d-bench: Generation of 3d assets, 2026. URL <https://arxiv.org/abs/2602.03907>.
- [37] J. Kim, Y. Lan, A. Fortes, Y. Chen, and X. Pan. Fastmesh: Efficient artistic mesh generation via component decoupling. *arXiv preprint arXiv:2508.19188*, 2025.
- [38] B. F. Labs, S. Batifol, A. Blattmann, F. Boesel, S. Consul, C. Diagne, T. Dockhorn, J. English, Z. English, P. Esser, S. Kulal, K. Lacey, Y. Levi, C. Li, D. Lorenz, J. Müller, D. Podell, R. Rombach, H. Saini, A. Sauer, and L. Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL <https://arxiv.org/abs/2506.15742>.
- [39] Z. Lai, Y. Zhao, H. Liu, Z. Zhao, Q. Lin, H. Shi, X. Yang, M. Yang, S. Yang, Y. Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025.
- [40] Z. Lai, Y. Zhao, Z. Zhao, H. Liu, Q. Lin, J. Huang, C. Guo, and X. Yue. Lattice: Democratize high-fidelity 3d generation at scale. *arXiv preprint arXiv:2512.03052*, 2025.
- [41] Z. Lai, Y. Zhao, Z. Zhao, X. Yang, X. Huang, J. Huang, X. Yue, and C. Guo. Natex: Seamless texture generation as latent color diffusion. *arXiv preprint arXiv:2511.16317*, 2025.

- [42] L. Li, Z. Huang, H. Feng, G. Zhuang, R. Chen, C. Guo, and L. Sheng. Voxhammer: Training-free precise and coherent 3d editing in native 3d space. *arXiv preprint arXiv:2508.19247*, 2025.
- [43] P. Li, S. Ma, J. Chen, Y. Liu, C. Zhang, W. Xue, W. Luo, A. Sheffer, W. Wang, and Y. Guo. Cmd: Controllable multiview diffusion for 3d editing and progressive generation, 2025. URL <https://arxiv.org/abs/2505.07003>.
- [44] W. Li, J. Liu, H. Yan, R. Chen, Y. Liang, X. Chen, P. Tan, and X. Long. Craftsman3d: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024.
- [45] W. Li, A. Toisoul, T. Monnier, R. Shapovalov, R. Ranjan, P. Tan, and A. Vedaldi. Meshflow: Efficient artistic mesh generation via meshvae and flow-based diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5849–5858, 2026.
- [46] Y. Li, Z.-X. Zou, Z. Liu, D. Wang, Y. Liang, Z. Yu, X. Liu, Y.-C. Guo, D. Liang, W. Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.
- [47] Y. Li, V. Cheung, X. Liu, Y. Chen, Z. Luo, B. Lei, H. Weng, Z. Zhao, J. Huang, Z. Chen, and C. Guo. Auto-regressive surface cutting. *arXiv preprint arXiv:2506.18017*, 2026.
- [48] Z. Li, W. Li, T. Wang, Z. Wang, J. Wu, H. Wang, Y. Yang, Z. Huang, Y. Li, P. Liu, and C. Guo. Moca: Mixture-of-components attention for scalable compositional 3d generation. *arXiv preprint arXiv:2512.07628*, 2025.
- [49] Z. Li, Y. Wang, H. Zheng, Y. Luo, and B. Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *arXiv preprint arXiv:2505.14521*, 2025.
- [50] S. Lim, S. Yoon, G. Koo, H. Yun, and C. D. Yoo. Tango: Training-free 3d editing via tangent-space guidance and optimization. *arXiv preprint arXiv:2607.14927*, 2026.
- [51] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 300–309, 2023.
- [52] F. Liu, H. Wang, W. Chen, H. Sun, and Y. Duan. Make-your-3d: Fast and consistent subject-driven 3d content generation, 2024. URL <https://arxiv.org/abs/2403.09625>.
- [53] F. Liu, J. Ye, Y. Wang, H. Wang, Z. Wang, J. Zhu, and Y. Duan. Dreamreward-x: Boosting high-quality 3d generation with human preference alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [54] H. Liu, Y. Lin, J. Guo, R. Chu, J. Wang, R. Li, and Y. Yang. Velocity-space 3d asset editing. *arXiv preprint arXiv:2605.07385*, 2026.
- [55] M. Liu, M. A. Uy, D. Xiang, H. Su, S. Fidler, N. Sharp, and J. Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9704–9715, 2025.
- [56] Z. Luo, Y. Li, M. Zhang, S. Wang, H. Yan, X. Song, T. Shang, W. Mao, H. Li, X. Han, and P. Ji. Bag: Body-aligned 3d wearable asset generation. *IEEE Transactions on Visualization and Computer Graphics*, 2026.
- [57] C. Ma, Y. Li, X. Yan, J. Xu, Y. Yang, C. Wang, Z. Zhao, Y. Guo, Z. Chen, and C. Guo. P3-sam: Native 3d part segmentation. *arXiv preprint arXiv:2509.06784*, 2025.
- [58] Z. Ma, H. Chen, Y. Yue, and G. Gkioxari. Feedforward 3d editing via text-steerable image-to-3d, 2025. URL <https://arxiv.org/abs/2512.13678>.
- [59] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. 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.
- [60] X. Pan, S. N. Shukla, A. Singh, Z. Zhao, S. K. Mishra, J. Wang, Z. Xu, J. Chen, K. Li, F. Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.
- [61] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [62] Z. Qi, R. Dong, S. Zhang, H. Geng, C. Han, Z. Ge, L. Yi, and K. Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pages 214–238. Springer, 2024.

- [63] Z. Qi, Y. Fang, Z. Sun, X. Wu, T. Wu, J. Wang, D. Lin, and H. Zhao. Gpt4point: A unified framework for point-language understanding and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26417–26427, 2024.
- [64] Z. Qi, Y. Yang, M. Zhang, L. Xing, X. Wu, T. Wu, D. Lin, X. Liu, J. Wang, and H. Zhao. Tailor3d: Customized 3d assets editing and generation with dual-side images, 2024. URL <https://arxiv.org/abs/2407.06191>.
- [65] L. Qiu, G. Chen, X. Gu, Q. Zuo, M. Xu, Y. Wu, W. Yuan, Z. Dong, L. Bo, and X. Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9914–9925, 2024.
- [66] L. Qu, H. Zhang, Y. Liu, X. Wang, Y. Jiang, Y. Gao, H. Ye, D. K. Du, Z. Yuan, and X. Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2545–2555, 2025.
- [67] T. Seedream, Y. Chen, Y. Gao, L. Gong, M. Guo, Q. Guo, Z. Guo, X. Hou, W. Huang, Y. Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.
- [68] E. Sella, G. Fiebelman, P. Hedman, and H. Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects, 2023. URL <https://arxiv.org/abs/2303.12048>.
- [69] E. Sella, H. Phung, N. Amiel, O. Litany, O. Patashnik, and H. Averbuch-Elor. Proxe: Fine-grained 3d shape editing via primitive-based abstractions. *arXiv preprint arXiv:2604.23774*, 2026.
- [70] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.
- [71] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.
- [72] J. Tang, Z. Li, Z. Hao, X. Liu, G. Zeng, M.-Y. Liu, and Q. Zhang. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. *arXiv preprint arXiv:2409.18114*, 2024.
- [73] C. Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. <https://doi.org/10.48550/arXiv.2405.09818>. URL <https://github.com/facebookresearch/chameleon>.
- [74] S. Tong, D. Fan, J. Li, Y. Xiong, X. Chen, K. Sinha, M. Rabbat, Y. LeCun, S. Xie, and Z. Liu. Metamorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17001–17012, 2025.
- [75] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [76] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.
- [77] C. Wang, J. Ye, Y. Yang, Y. Li, Z. Lin, J. Zhu, Z. Chen, Y. Luo, and C. Guo. Part-x-mlm: Part-aware 3d multimodal large language model. *arXiv preprint arXiv:2511.13647*, 2025.
- [78] C. Wang, H. Weng, J. Ye, B. Lei, Y. Li, Z. Zhao, Z. Lai, K. Zhang, Y. Yang, Z. Chen, et al. Polyflow: Continuous topology embedding flow matching for artist-style mesh generation. *arXiv preprint arXiv:2606.30673*, 2026.
- [79] H. Wang, Y.-C. Guo, Y.-T. Liu, Z.-X. Zou, B. Zhang, W. Quan, D. Liang, Y.-P. Cao, and D.-M. Yan. Face: A face-based autoregressive representation for high-fidelity and efficient mesh generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12719–12729, 2026.
- [80] P. Wang, Y. He, X. Lv, Y. Zhou, L. Xu, J. Yu, and J. Gu. PartNeXt: A next-generation dataset for fine-grained and hierarchical 3d part understanding. In *Advances in Neural Information Processing Systems*, 2025. Datasets and Benchmarks Track.
- [81] X. Wang, X. Zhang, Z. Luo, Q. Sun, Y. Cui, J. Wang, F. Zhang, Y. Wang, Z. Li, Q. Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

- [82] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36:8406–8441, 2023.
- [83] H. Weng, Y. Wang, T. Zhang, C. Chen, and J. Zhu. Pivotmesh: Generic 3d mesh generation via pivot vertices guidance. *arXiv preprint arXiv:2405.16890*, 2024.
- [84] H. Weng, Z. Zhao, B. Lei, X. Yang, J. Liu, Z. Lai, Z. Chen, Y. Liu, J. Jiang, C. Guo, et al. Scaling mesh generation via compressive tokenization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11093–11103, 2025.
- [85] J. Weng, S. Zhang, Z. Diao, P. Li, H. Zhang, J. Chen, and H. Zhao. Feedforward 3d editing learns from semantic-part transformation. *arXiv preprint arXiv:2605.27351*, 2026.
- [86] C. Wu, J. Li, J. Zhou, J. Lin, K. Gao, K. Yan, S.-m. Yin, S. Bai, X. Xu, Y. Chen, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025.
- [87] C. Wu, P. Zheng, R. Yan, S. Xiao, X. Luo, Y. Wang, W. Li, X. Jiang, Y. Liu, J. Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.
- [88] Y. Wu, Z. Zhang, J. Chen, H. Tang, D. Li, Y. Fang, L. Zhu, E. Xie, H. Yin, L. Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.
- [89] Z. Wu, P. Zhou, X. Yi, X. Yuan, and H. Zhang. Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9892–9902, 2024.
- [90] R. Xia, Y. Tang, and P. Zhou. Towards scalable and consistent 3d editing, 2025. URL <https://arxiv.org/abs/2510.02994>.
- [91] R. Xia, Y. Tang, and P. Zhou. Towards scalable and consistent 3d editing. *arXiv preprint arXiv:2510.02994*, 2025.
- [92] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025.
- [93] R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024.
- [94] Y. Xu, H. Zhu, C. Liu, T. Wang, K. Chen, S. Xu, J. Yang, Q. Zhang, et al. Beyond voxel 3d editing: Learning from 3d masks and self-constructed data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 635–646, 2026.
- [95] H. Yan, Y. Li, Z. Wu, S. Chen, W. Sun, T. Shang, W. Liu, T. Chen, X. Dai, C. Ma, H. Li, and P. Ji. Frankenstein: Generating semantic-compositional 3d scenes in one tri-plane. In *ACM SIGGRAPH Asia Conference Proceedings*, 2024.
- [96] H. Yan, M. Zhang, Y. Li, C. Ma, and P. Ji. Phycage: Physically plausible compositional 3d asset generation from a single image, 2024.
- [97] X. Yan, J. Xu, Y. Li, C. Ma, Y. Yang, C. Wang, Z. Zhao, Z. Lai, Y. Zhao, Z. Chen, and C. Guo. X-part: high fidelity and structure coherent shape decomposition. *arXiv preprint arXiv:2509.08643*, 2025.
- [98] Y. Yang, Y. Zhou, Y.-C. Guo, Z.-X. Zou, Y. Huang, Y.-T. Liu, H. Xu, D. Liang, Y.-P. Cao, and X. Liu. Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion. *arXiv preprint arXiv:2507.06165*, 2025.
- [99] Y. Yang, C. Wang, J. Ye, Y. Li, Z. Chen, Z. Huang, Y. Mu, Z. Chen, C. Guo, and X. Liu. Physforge: Generating physics-grounded 3d assets for interactive virtual world. *arXiv preprint arXiv:2605.05163*, 2026.
- [100] C. Ye, C. Cao, C. Pan, Y. Hao, Y. Zhi, Y. Hu, and X. Han. Omni123: Exploring 3d native foundation models with limited 3d data by unifying text to 2d and 3d generation. *arXiv preprint arXiv:2604.02289*, 2026.
- [101] J. Ye, F. Liu, Q. Li, Z. Wang, Y. Wang, X. Wang, Y. Duan, and J. Zhu. Dreamreward: Text-to-3d generation with human preference. In *European Conference on Computer Vision*, pages 259–276. Springer, 2024.
- [102] J. Ye, Z. Wang, R. Zhao, S. Xie, and J. Zhu. Shapellm-omni: A native multimodal llm for 3d generation and understanding. *arXiv preprint arXiv:2506.01853*, 2025.

- [103] J. Ye, S. Xie, R. Zhao, Z. Wang, H. Yan, W. Zu, L. Ma, and J. Zhu. Nano3d: A training-free approach for efficient 3d editing without masks. *arXiv preprint arXiv:2510.15019*, 2025.
- [104] J. Ye, Z. Huang, Y. Qu, C. Wang, Y. Yang, Y. Li, Y. Luo, Z. Chen, S. Lu, J. Zhu, et al. Universe3d: Emerging properties of unified multimodal models in 3d understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 613–623, 2026.
- [105] T. Yi, J. Fang, J. Wang, G. Wu, L. Xie, X. Zhang, W. Liu, Q. Tian, and X. Wang. Gaussian-dreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6796–6807, 2024.
- [106] Y. Yin, Y. Zhou, J. Wei, X. Yang, J. Zhang, J. Bai, J. Ye, W. Zhang, and G. Lin. Editverse3d: High-quality 3d object editing with region-aware learning. *arXiv preprint arXiv:2607.07187*, 2026.
- [107] B. Zhang, J. Tang, M. Niessner, and P. Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.
- [108] L. Zhang, Z. Wang, Q. Zhang, Q. Qiu, A. Pang, H. Jiang, W. Yang, L. Xu, and J. Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024.
- [109] Y. Zhang, Z. Li, M. Zhou, S. Wu, and J. Wu. The scene language: Representing scenes with programs, words, and embeddings, 2025. URL <https://arxiv.org/abs/2410.16770>.
- [110] R. Zhao, J. Ye, Z. Wang, G. Liu, Y. Chen, Y. Wang, and J. Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10612–10623, 2025.
- [111] R. Zhao, B. Li, Z. Liu, Y. Liang, J. Ye, F. Liu, D. Wu, Z. Wang, X. Yu, Y. Rao, et al. Gem: Generative supervision helps embodied intelligence. *arXiv preprint arXiv:2605.28548*, 2026.
- [112] T. Zhao, Y. Zhang, H. Long, J. Zhang, W. Li, Y. Yang, G. Zhang, J. Hladký, M. Nießner, and W. Yang. Lato: 3d mesh flow matching with structured topology preserving latents. *arXiv preprint arXiv:2603.06357*, 2026.
- [113] Z. Zhao, W. Liu, X. Chen, X. Zeng, R. Wang, P. Cheng, B. Fu, T. Chen, G. Yu, and S. Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems*, 36: 73969–73982, 2023.
- [114] Y. Zheng, M. Huang, N. Chen, and Z. Mao. Pro3d-editor : A progressive-views perspective for consistent and precise 3d editing, 2025. URL <https://arxiv.org/abs/2506.00512>.
- [115] Z. Zhou, F. Ma, C. Gui, X. Xia, H. Fan, Y. Yang, and T.-S. Chua. Anchorflow: Training-free 3d editing via latent anchor-aligned flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14387–14397, 2026.
- [116] Y. Zhu, K. Deng, J.-P. Fauconnier, I. Navarro, D. Li, A. Pun, Y. Zhang, P. Zhuang, X. Sun, M. Agrawala, K. Bhat, and T. Zhou. Cubepart: An open-vocabulary part-controllable 3d generator. *arXiv preprint arXiv:2605.28763*, 2026.
- [117] J. Zhuang, D. Kang, Y.-P. Cao, G. Li, L. Lin, and Y. Shan. Tip-editor: An accurate 3d editor following both text-prompts and image-prompts, 2024. URL <https://arxiv.org/abs/2401.14828>.