Supplementary Material: Are We Ready for RL in Text-to-3D Generation? A Progressive Investigation

Yiwen Tang[1,2]*, Zoey Guo[3]*, Kaixin Zhu[4]*, Ray Zhang[3]*, Qizhi Chen[1], Dongzhi Jiang[3], Junli Liu[1,2], Bohan Zeng[4], Haoming Song[1], Delin Qu[1], Tianyi Bai[6], Dan Xu[6], Wentao Zhang[4,5], Bin Zhao[1,2]†
[1]Shanghai AI Laboratory [2]Northwestern Polytechnical University
[3]The Chinese University of Hong Kong [4]Peking University [5]Zhongguancun Academy [6]HKUST

1. Overview

- Sec. 2: Related work.
- Sec. 3: Experimental settings.
- Sec. 4: Details of Hi-GRPO.
- Sec. 5: Ablation study.
- Sec. 6: Additional visualizations.

2. Related Work

2.1. RL for LLM

Advanced LLMs such as OpenAI o3 [14] and DeepSeek-R1 [7] have demonstrated strong reasoning abilities by combining Chain-of-Thought (CoT) reasoning with reinforcement learning (RL). DeepSeek-R1 introduces rule-based rewards and GRPO [18], enabling models to conduct extensive internal reasoning before producing an answer, with rewards guiding correctness and format compliance. This paradigm has also been extended to multimodal LLMs [6, 9, 19, 34], where RL is adapted for visual understanding by jointly processing images and text for step-by-step reasoning. These RL-driven approaches have proven effective across mathematical problem-solving [18, 32] and code generation [17, 21], establishing RL as a key technique for eliciting advanced capabilities in large-scale models.

2.2. RL for 2D Generation

RL has also been effectively applied to text-to-image generation. Image-Generation-CoT [8] first frames progressive image token generation as a reasoning process and applies DPO [16] accordingly. T2I-R1 [10] extends this idea by distinguishing two levels of CoT—semantic-level planning and token-level patch generation—and introduces BiCoT-GRPO to jointly optimize both using an ensemble of vision experts as reward models. Recent work [11, 23] comparing DPO and GRPO shows that GRPO offers better text–image alignment and aesthetic quality through group-relative policy updates. Together, these studies highlight that well-designed sequence-level rewards and multi-dimensional evaluation are essential for producing semantically consistent and visually appealing images in autoregressive models. For diffusion models, Dance-GRPO [29] introduces a stepwise, motion-aware reward that aligns policy updates with temporal dynamics, enabling more coherent and physically plausible generation. Flow-GRPO [13] extends GRPO to flow-matching models by coupling policy optimization with flow objectives, yielding smoother training and improved stability. These methods show that RL can effectively enhance controllability and consistency in diffusion and flow-based generative models.

2.3. Text-to-3D Generation

Text-to-3D generation has progressed from two-stage pipelines [28, 30] to native diffusion models [3, 27] and, more recently, autoregressive approaches [2, 20, 31, 33]. Two-stage methods, like Dream3D [28], first generate a high-quality 3D shape prior from text using a text-to-image diffusion model and then refine it as a neural radiance field, but this pipeline suffers from error accumulation between stages and limited 3D consistency inherited from the 2D diffusion backbone. Native diffusion models, like Trellis [27], leverage structured 3D latent representations to directly generate high-fidelity 3D content, but their strong performance comes at the cost of significant computational demands. Autoregressive models alleviate these limitations by discretizing 3D content into token sequences. MeshGPT [20] uses decoder-only transformer to model triangle meshes as sequences, and MeshAnything [2] demonstrates scalable artist-grade mesh generation using autoregressive transformers. DeepMesh [33] provides an early attempt to incorporate DPO [16] into autoregressive 3D cre-

*Equal Contribution.
†Corresponding Author.

Table 1. **Quantitative comparisons using Toys4k for Reward Analysis.**

Step 1 Reward			Step 2 Reward				Metrics	
R_1^{HPM}	R_1^{unified}	R_1^{consist}	R_2^{HPM}	R_2^{unified}	R_2^{consist}	R_2^{part}	CLIP Score↑	KD_{incep}↓
-	-	-	✓	✓	-	-	25.0	0.235
-	-	-	✓	✓	✓	-	25.7	0.223
✓	✓	-	✓	✓	✓	-	27.8	0.194
✓	✓	✓	✓	✓	✓	-	28.3	0.182
✓	✓	-	✓	✓	✓	✓	28.6	0.178
✓	✓	✓	✓	✓	✓	✓	**29.3**	**0.156**

ation, and LLaMA-Mesh [25] represents 3D OBJ files as text to unify language and 3D representations. ShapeLLM-Omni [31] proposes a unified multimodal LLM for 3D generation and understanding by discretizing 3D shapes into tokens with a 3D VQVAE. The model follows a text→voxel pipeline, where the LLM predicts discrete 3D latent tokens that are decoded by the VQVAE into voxel grids, which are then further converted into meshes using Rectified Flow model [27] for rendering. This design enables a single LLM to support text-to-3D generation, 3D understanding, and editing within one coherent framework. Despite these developments, RL training for 3D autoregressive models remains largely unexplored. In contrast to 2D generation, where RL has shown clear benefits, 3D generation introduces additional challenges—greater spatial complexity, stricter global geometry constraints, and fine-grained local details—making it more sensitive to reward design and optimization choices. These factors highlight the need for systematic RL strategies tailored to text-to-3D generation.

3. Experimental Settings

We employ ShapeLLM-Omni [31] as the base model with a learning rate of 1×10^{-6}, β of 0.01, and group size of 8. Training is conducted on 8 GPUs with a batch size of 1 per device and gradient accumulation over 2 steps. The model is trained for 1,200 steps. The configurable weight λ for supervising global planning with final quality is set to 1.0. Our reward models are deployed via the vLLM API framework. We select training prompt from Objaverse-XL [5], HSSD [12], and ABO [4], and evaluate our method on Toys4K [22].

• **Objaverse-XL:** Objaverse-XL is one of the largest 3D object datasets currently available, comprising over 10 million 3D objects sourced from diverse platforms including GitHub, Thingiverse, Sketchfab and Polycam. The dataset undergoes rigorous deduplication and rendering validation, covering a range of categories and fine-grained attributes.

• **HSSD:** HSSD contains 211 high-quality indoor synthetic 3D scenes with approximately 18,656 real-world object models, emphasizing indoor layouts, semantic structures, and object relationships.

• **ABO:** ABO focuses on real-world household objects and provide approximately 147,000 product listings, nearly 400,000 catalog images, and about 8,000 3D models with rich material, geometric, and attribute annotations.

• **Toys4K:** Toys4K includes approximately 4,000 3D object instances spanning around 105 categories, featuring diverse categories and significant shape variations.

4. Details of Hi-GRPO

4.1. Two-Step Generation Process

Step 1: The model generates semantic reasoning tokens $\mathbf{s}_i = \{s_{i,1}, \ldots, s_{i,|s_i|}\}$ for global geometric planning, followed by 3d tokens $\mathbf{t}_i = \{t_{i,1}, \ldots, t_{i,M}\}$, where M is the number of compressed grids. The coarse triangular mesh $\mathcal{M}_i^{(1)}$ is decoded through the VQVAE decoder.

Step 2: Conditioned on semantic reasoning, the model generates visual reasoning tokens $\mathbf{v}_i = \{v_{i,1}, \ldots, v_{i,|v_i|}\}$ focused on local details, followed by 3d tokens $\mathbf{o}_i = \{o_{i,1}, \ldots, o_{i,M}\}$, which are decoded into mesh $\mathcal{M}_i^{(2)}$.

4.2. Hierarchical Reward Ensemble Design

4.2.1. Human Preference Model

We adopt HPS V2.1 [26] in both steps. For generated 3D objects rendered from 6 uniformly distributed viewpoints $\{\mathbf{v}_1, \ldots, \mathbf{v}_6\}$, we compute text-image similarity at each viewpoint and take the maximum:

$$R_i^{\text{HPM},k} = \max_{j=1,\ldots,6} \text{HPS}(\mathbf{x}, \text{Render}(\mathcal{M}_i^{(k)}, \mathbf{v}_j)) \quad (1)$$

This reward evaluates human preference with range [0, 1].

4.2.2. Unified Reward Model

Step 1: UnifiedReward Think-qwen-7B [24] evaluates geometric alignment between prompts and coarse shapes. Each of the 6 viewpoints is scored (1-5), and the maximum is:

$$R_i^{\text{unified},1} = \max_{j=1,\ldots,6} f_{\text{UR-Think}}(\mathbf{x}, \text{Render}(\mathcal{M}_i^{(1)}, \mathbf{v}_j)) \quad (2)$$

This reward evaluates prompt alignment with range [1, 5].

Step 2: UnifiedReward-2.0-qwen-7b [24] performs three-dimensional evaluation of textured objects: (1) prompt alignment (1-5), (2) logical coherence (1-5), (3) style appeal (1-5). The maximum sum across 6 viewpoints:

$$R_i^{\text{unified},2} = \max_{j=1,\ldots,6} \sum_{\ell \in A_{\text{app}}} f_{\text{UR}}^{(\ell)}(\mathbf{x}, \text{Render}(\mathcal{M}_i^{(2)}, \mathbf{v}_j)) \quad (3)$$

Table 2. **Quantitative comparisons using Toys4k for Different RL Paradigms.**

Training Strategy					Metrics	
GRPO	Textual Reasoning	Step1 Reward	Step2 Reward	Hi-GRPO	CLIP Score↑	KD_{incep}↓
-	-	-	-	-	22.7	0.249
✓	-	-	-	-	24.3	0.237
✓	✓	-	-	-	25.2	0.228
✓	✓	✓	-	-	24.8	0.235
✓	✓	-	✓	-	26.0	0.214
-	-	-	-	✓	**28.7**	**0.182**

$\mathcal{A}_{\text{app}} = \{\text{logic, style, align}\}$. This reward evaluates 3 dimensions with range [3, 15].

4.2.3. 2D Large Multi-modal Model

Step 1: Qwen2.5-VL-7B [1] verifies whether the generated shape matches the object category in the prompt based on joint observation of 6 viewpoints:

$$R_i^{\text{consist},1} = f_{\text{Qwen}}^{\text{category}}(\mathbf{x}, \{\text{Render}(\mathcal{M}_i^{(1)}, \mathbf{v}_j)\}_{j=1}^{6}) \quad (4)$$

This reward evaluates category matching with range $\{0, 1\}$.

Step 2: Qwen2.5-VL-7B [1] evaluates three dimensions of cross-view appearance consistency: (1) color smoothness (0-1), (2) material realism and coherence (0-1), (3) texture rationality (0-1):

$$R_i^{\text{consist},2} = \sum_{\ell \in \mathcal{A}_{\text{app}}} f_{\text{Qwen}}^{(\ell)}\big(\mathbf{x}, \{\text{Render}(\mathcal{M}_i^{(2)}, \mathbf{v}_j)\}_{j=1}^{6}\big). \quad (5)$$

$\mathcal{A}_{\text{app}} = \{\text{color, material, texture}\}$. This reward evaluates 3 dimensions with range [0, 3].

4.2.4. 3D Large Multi-modal Model

2D LMMs struggle to accurately detect 3D components from multi-view observations. To obtain accurate component completeness assessment, we employ direct evaluation based on 3D point clouds in step 2.

1) Mesh to Dense Point Cloud Sampling: The refined triangular mesh $\mathcal{M}_i^{(2)} = (\mathcal{V}^{(2)}, \mathcal{F}^{(2)}, \mathcal{T})$ is converted to dense point cloud $\mathcal{P}_i$. The sampling process:

1. **Area-Weighted Sampling**: For each triangle face $f \in \mathcal{F}^{(2)}$, allocate sample points $n_f = \lceil \rho \cdot A_f \rceil$ based on area A_f, where ρ is the sampling density parameter
2. **Barycentric Uniform Sampling**: Within face $f = (v_1, v_2, v_3)$, generate random barycentric coordinates (α, β, γ) satisfying $\alpha + \beta + \gamma = 1$ and $\alpha, \beta, \gamma \geq 0$. Sample point coordinates: $\mathbf{p} = \alpha\mathbf{v}_1 + \beta\mathbf{v}_2 + \gamma\mathbf{v}_3$
3. **Texture Color Sampling**: Interpolate UV coordinates using barycentric coordinates $\mathbf{uv} = \alpha\mathbf{uv}_1 + \beta\mathbf{uv}_2 + \gamma\mathbf{uv}_3$, and sample RGB color from texture map $\mathcal{T}$

The result is point cloud $\mathcal{P}_i = \{(\mathbf{p}_k, \mathbf{c}_k)\}_{k=1}^{N_p}$, where $\mathbf{p}_k \in \mathbb{R}^3$ is position, $\mathbf{c}_k \in \mathbb{R}^3$ is RGB color.

2) Per-Component Evaluation: Parse prompt $\mathbf{x}$ to extract component list $\mathcal{C} = \{c_1, \ldots, c_{N_c}\}$ and expected quantities $\{n_1, \ldots, n_{N_c}\}$. ShapeLLM [15] processes point cloud $\mathcal{P}_i$ and component queries. For each component c_p:

- Existence: $e_p \in \{0, 1\}$ determines the existence.
- Completeness: $q_p \in [0, 1]$ evaluates geometric completeness, shape correctness, and quantity matching

Average scores across N_c components:

$$R_i^{\text{part},2} = \frac{1}{N_c} \sum_{p=1}^{N_c} (e_p + q_p) \quad (6)$$

This reward evaluates 2 dimensions per component (existence + completeness), with averaged range [0, 2].

4.2.5. Dimension-Normalized Reward Ensemble

Each reward is normalized by its number of evaluation dimensions to ensure fair contribution:

Step 1 Total Reward:

$$R_i^{\text{high}} = R_i^{\text{HPM},1} + R_i^{\text{unified},1} + R_i^{\text{consist},1} \quad (7)$$

Step 2 Total Reward:

$$R_i^{\text{low}} = R_i^{\text{HPM},2} + \frac{R_i^{\text{unified},2}}{3} + \frac{R_i^{\text{consist},2}}{3} + \frac{R_i^{\text{part},2}}{2} \quad (8)$$

This normalization strategy ensures: (1) each reward's contribution is proportional to its number of evaluation dimensions; (2) multi-dimensional evaluations do not dominate through simple summation; (3) the system remains stable when adding or removing rewards. Rewards from step 2 are backpropagated to step 1 through weight λ:

$$\tilde{R}_i^{\text{high}} = R_i^{\text{high}} + \lambda \cdot R_i^{\text{low}} \quad (9)$$

When $\lambda = 1.0$, the high-level step is directly supervised by final output quality. For each step, advantages are normalized within prompt groups to eliminate reward scale differences across prompts:

$$A_i^{(1)} = \frac{\tilde{R}_i^{\text{high}} - \mu_g^{(1)}}{\sigma_g^{(1)} + \epsilon}, \quad A_i^{(2)} = \frac{R_i^{\text{low}} - \mu_g^{(2)}}{\sigma_g^{(2)} + \epsilon} \quad (10)$$

where $\mu_g^{(k)} = \frac{1}{G}\sum_{j=1}^{G} R_j^{(k)}$, $\sigma_g^{(k)} = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(R_j^{(k)} - \mu_g^{(k)})^2}$, $\epsilon = 10^{-4}$.

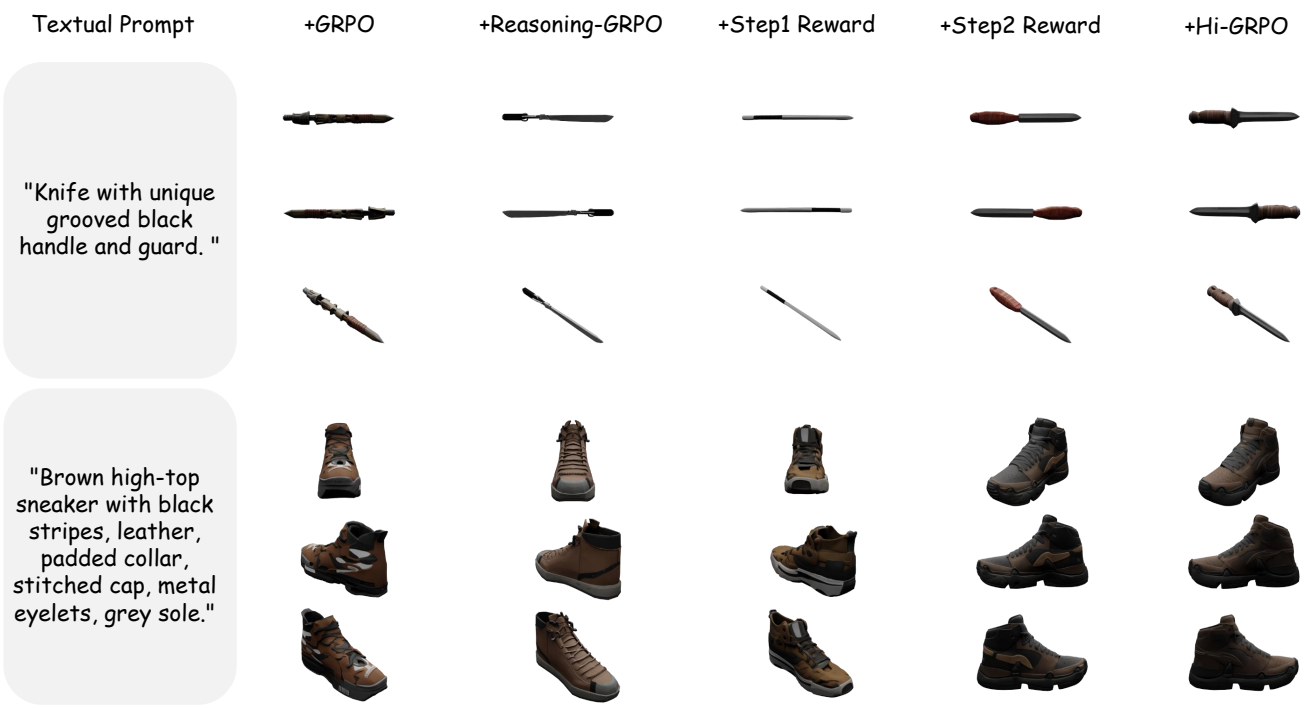

Figure 1. **Visualization Results of Small Objects for Different RL Paradigms.**

4.3. Loss Computation

For each step, we compute token-level log probabilities by concatenating the log probabilities of reasoning tokens and mesh tokens. In step 1, the complete sequence log probability concatenates semantic reasoning and coarse mesh generation: $\log \pi_\theta(\mathbf{y}_i^{(1)}) = \text{concat}(\log \pi_\theta(\mathbf{s}_i \mid \mathbf{x}), \log \pi_\theta(\mathbf{t}_i \mid \mathbf{x}, \mathbf{s}_i))$. In step 2, it concatenates visual reasoning and refined mesh generation: $\log \pi_\theta(\mathbf{y}_i^{(2)}) = \text{concat}(\{\log \pi_\theta(v_{i,t} \mid \mathbf{x}, \mathbf{s}_i, v_{i,<t})\}_{t=1}^{|\mathbf{v}_i|}, \{\log \pi_\theta(o_{i,t} \mid \mathbf{x}, \mathbf{v}_i, o_{i,<t})\}_{t=1}^{M})$. Reference policy log probabilities $\log \pi_{\text{ref}}(\mathbf{y}_i^{(k)})$ are computed similarly.

For stage $k \in \{1, 2\}$, the complete loss function is:

$$\mathcal{L}^{(k)} = -\mathbb{E}_{q\sim\mathcal{D},\{y_i^{(k)}\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{\sum_{i=1}^{G} T_i^{(k)}} \sum_{i=1}^{G} \sum_{t=1}^{T_i^{(k)}} m_{i,t}^{(k)} \left(\min \left(r_{i,t}^{(k)}(\theta) A_i^{(k)}, \text{clip}(r_{i,t}^{(k)}(\theta), 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}}) A_i^{(k)} \right) - \beta \cdot \text{KL}_{i,t}^{(k)} \right) \right] \tag{11}$$

We highlight and describe key components as follows:

• **Policy Ratio**:

$$r_{i,t}^{(k)}(\theta) = \frac{\pi_\theta(y_{i,t}^{(k)} | \mathbf{y}_{i,<t}^{(k)})}{\pi_{\theta_{\text{old}}}(y_{i,t}^{(k)} | \mathbf{y}_{i,<t}^{(k)})} \tag{12}$$

• **Decoupled Clipping**: Asymmetric clipping thresholds ε_{low} and $\varepsilon_{\text{high}}$. The higher threshold allows low-probability tokens greater probability increase space, promoting exploration and preventing entropy collapse.

• **Token-Level Averaging**: The loss is normalized by the token count $\sum_{i=1}^{G} T_i^{(k)}$, where $T_i^{(k)} = \sum_{t=1}^{T_{\max}} m_{i,t}^{(k)}$ is the number of valid tokens and $m_{i,t}^{(k)}$ is the completion mask.

• **KL Regularization**: Token-level KL divergence

$$\text{KL}_{i,t}^{(k)} = \frac{\pi_{\text{ref}}(y_{i,t}^{(k)} | \mathbf{y}_{i,<t}^{(k)})}{\pi_\theta(y_{i,t}^{(k)} | \mathbf{y}_{i,<t}^{(k)})} - \log \frac{\pi_{\text{ref}}(y_{i,t}^{(k)} | \mathbf{y}_{i,<t}^{(k)})}{\pi_\theta(y_{i,t}^{(k)} | \mathbf{y}_{i,<t}^{(k)})} - 1 \tag{13}$$

with penalty coefficient $\beta = 0.01$ prevents the policy from deviating too far from the reference. The two steps compute losses independently. The total optimization objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}^{(1)} + \mathcal{L}^{(2)} \tag{14}$$

4.4. Prompt Template for Reward Model

4.4.1. 3D Consistency

In the section 3 of main text, Qwen2.5-VL used for evaluating 3D consistency is driven by the prompt in Figure 11.

4.4.2. Shape Semantic Consistency

In the Hi-GRPO training, during step 1 we use Qwen2.5-VL-7B to assign a binary (0/1) score evaluating whether the generated mesh shape matches the category specified in the prompt. The exact prompt used is shown in Figure 13.

Figure 2. **Visualization Results of Large Objects for Different RL Paradigms.**

4.4.3. Appearance Consistency

In the second step, for the refined 3D object, we use Qwen2.5-VL-7B to evaluate three dimensions of cross-view appearance consistency: (1) color smoothness (0–1), (2) material realism and coherence (0–1), and (3) texture rationality (0–1). The prompt template is shown in Figure 12.

4.4.4. Part Semantic-Visual Consistency

We adopt ShapeLLM-13B to evaluate the existence (binary 0/1) and completeness (0–1) of object parts. Since multi-view images can easily introduce ambiguities, performing part detection directly on the 3D point cloud substantially improves accuracy, especially for complex objects. The prompt is shown in Figure 14.

5. Ablation Study

5.1. Reward Analysis.

This section investigates reward function selection and combination for AR3D-R1. Table 1 presents our findings. We first examine whether Step-2 rewards can simultaneously optimize both generation steps. Results show that rewards from refined objects struggle to control both coarse geometry and fine texture effectively. Even with combined rewards (R_2^{HPM}+R_2^{unified}+R_2^{consist}), improvements remain marginal. However, introducing step-specific rewards, adding R_1^{HPM}+R_1^{unified} for Step 1, yields substantial gains, improving CLIP scores by 2.1 point. Notably, component-level rewards prove critical for ensuring correct part positioning, quantity, and structural plausibility.

5.2. Effectiveness of Hi-GRPO.

To validate Hi-GRPO, we conduct ablation studies using the baseline GRPO algorithm. Table 2 presents quantitative results, while Figures 1 and 2 provide extensive visualizations. We first compare direct 3D token optimization against textual reasoning-guided GRPO, both evaluated with HPSV2.1+UnifiedReward+Qwen2.5-VL reward system. Quantitatively, textual reasoning yields a 0.9-point CLIP score improvement, while qualitatively it enables effective global planning for both large and small objects. We then examine Hi-GRPO's hierarchical reward ensemble by separately applying Step-1 and Step-2 reward systems. Since Step-1 rewards focus on high-level geometric structure, Table 2 shows performance degradation, with visualizations revealing noticeably reduced texture fidelity. Ultimately, the hierarchical RL paradigm of Hi-GRPO, combining global-to-local generation with step-specific reward ensembles, achieves substantial improvements across geometry, fine-grained textures, and prompt alignment.

6. Additional Visualizations

Figures 3, 4, 5, 6, and 7 visualize the generation results of our proposed AR3D-R1, ShapeLLM-Omni, and Trellis across the five categories in our MME-3DR benchmark. Figures 8, 9, and 10 visualize AR3D-R1's hierarchical generation process across different object categories.

Figure 3. **Visualization Results of Spatial & Structural Geometry in MME-3DR.**

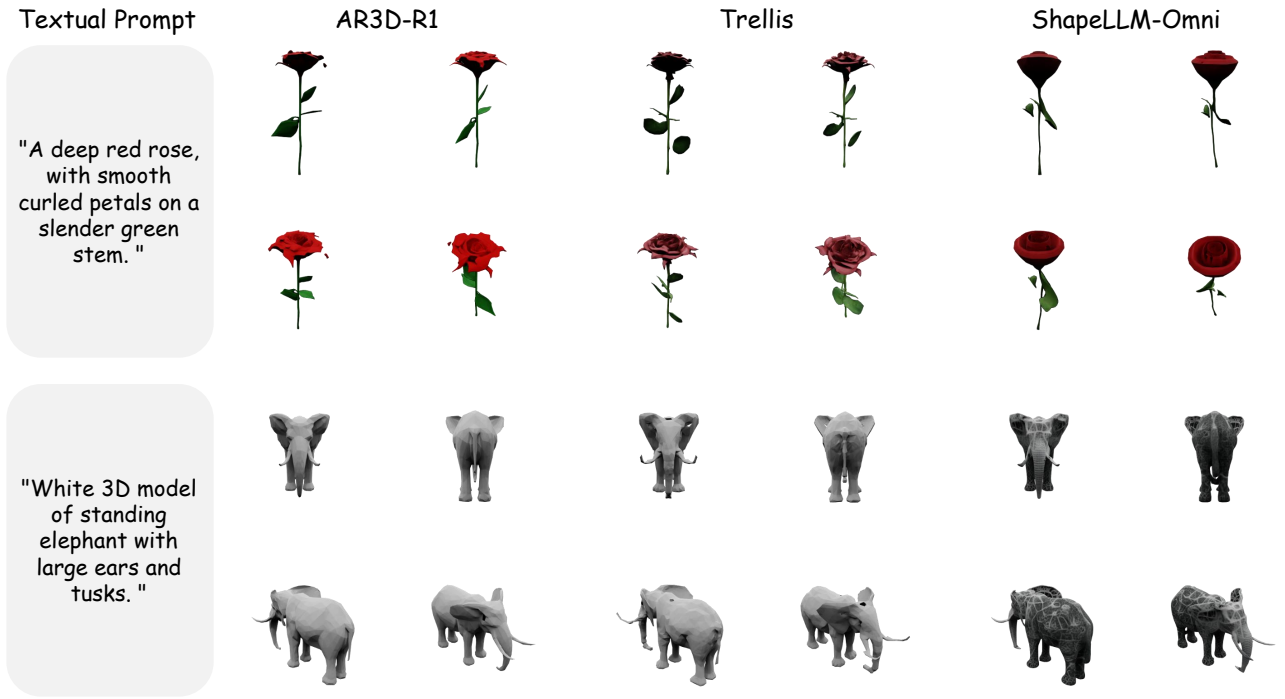

Figure 4. **Visualization Results of Biological & Organic Shapes in MME-3DR.**

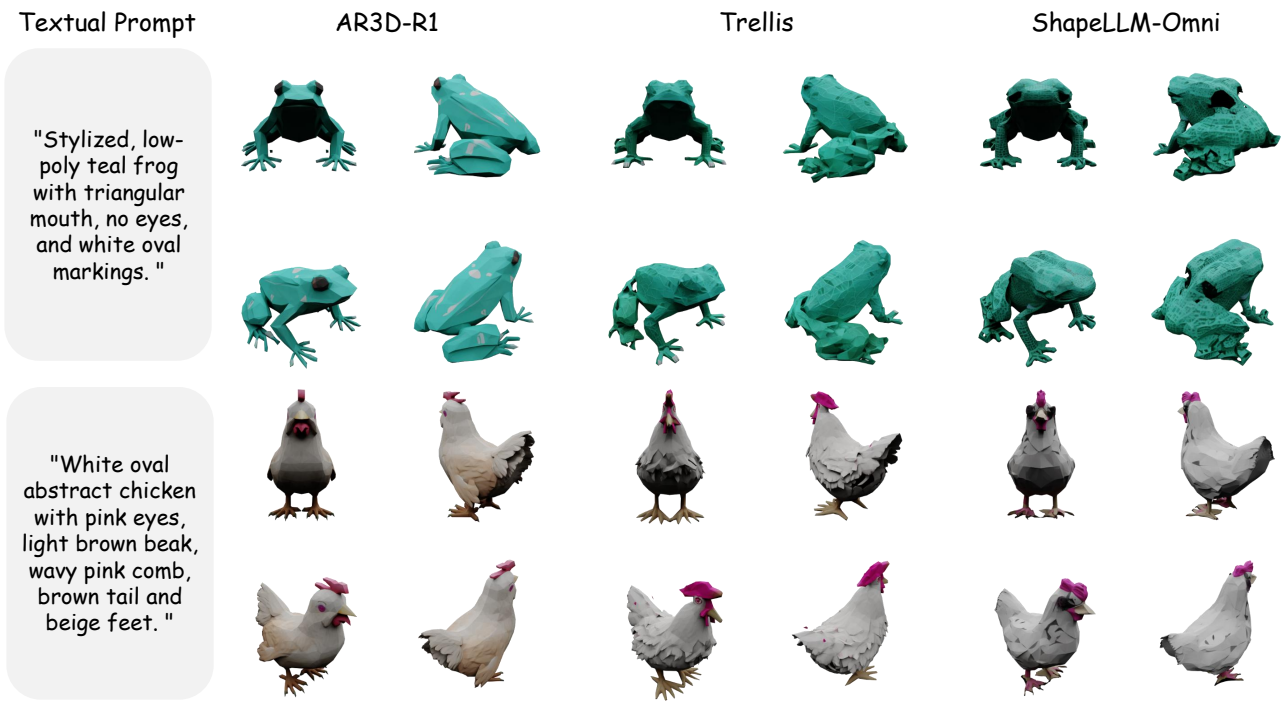

Figure 5. **Visualization Results of Stylized Representations in MME-3DR.**

Figure 6. **Visualization Results of Mechanical Affordances in MME-3DR.**

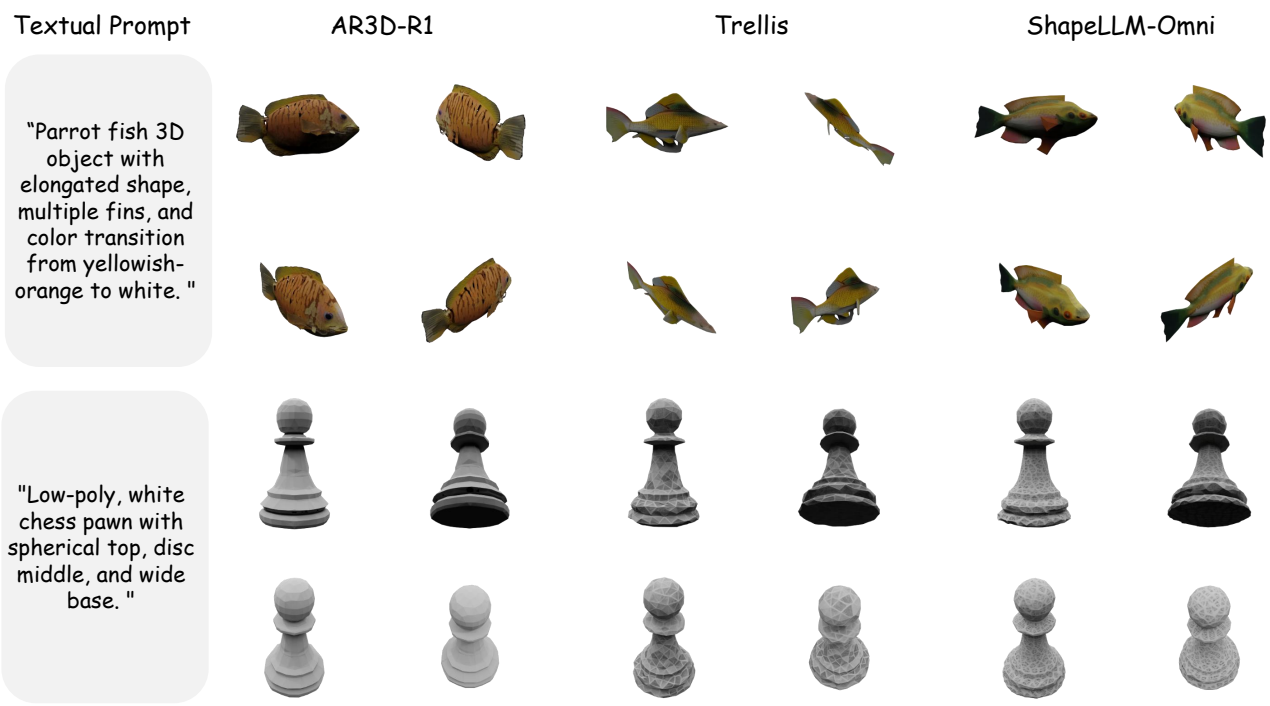

Figure 7. **Visualization Results of World-Knowledge Rare Objects in MME-3DR.**

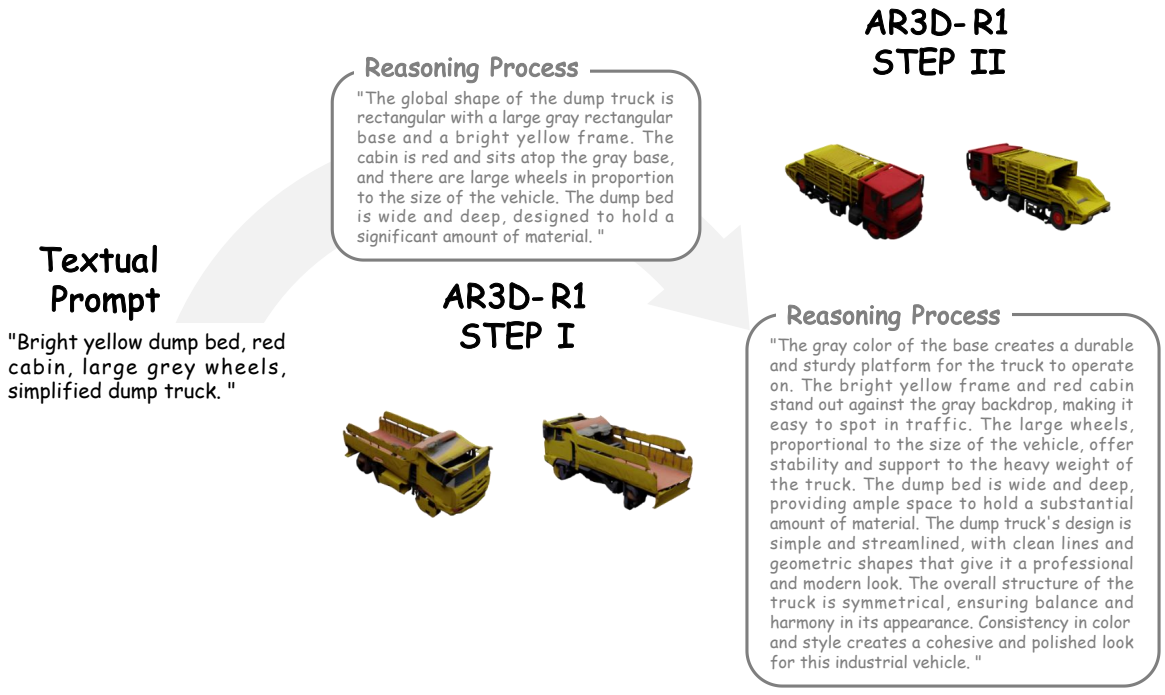

Figure 8. **Visualization of the Two-Step Reasoning Generation Process in Cabin.**

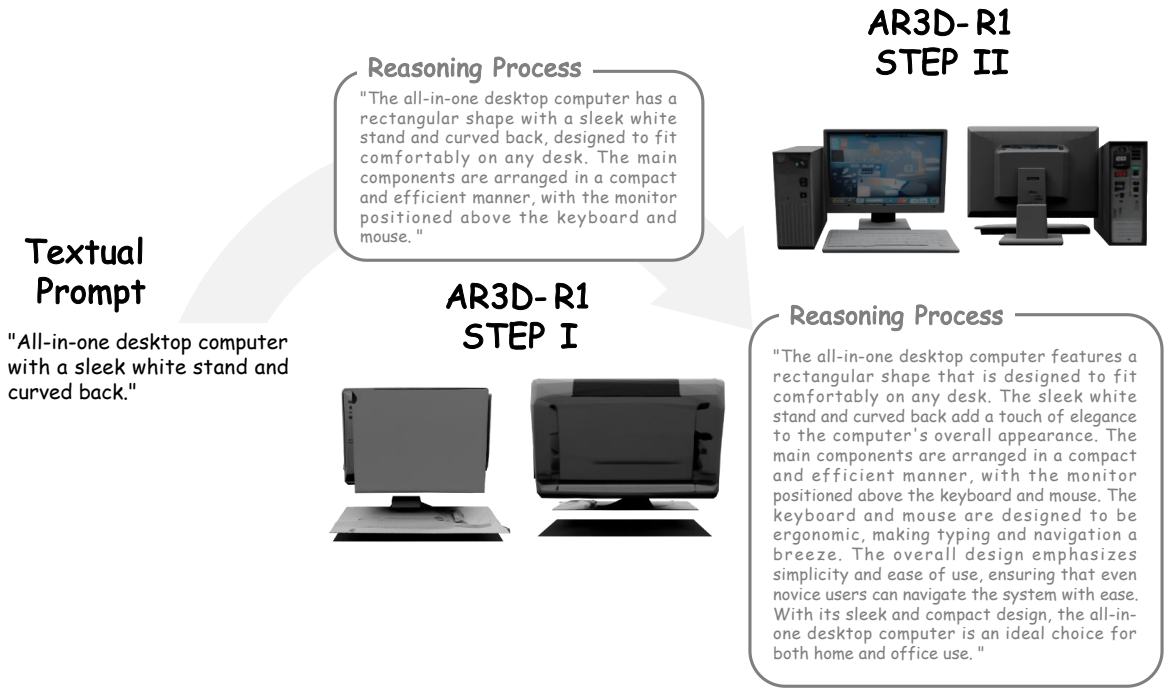

Figure 9. **Visualization of the Two-Step Reasoning Generation Process in Desktop Computer.**

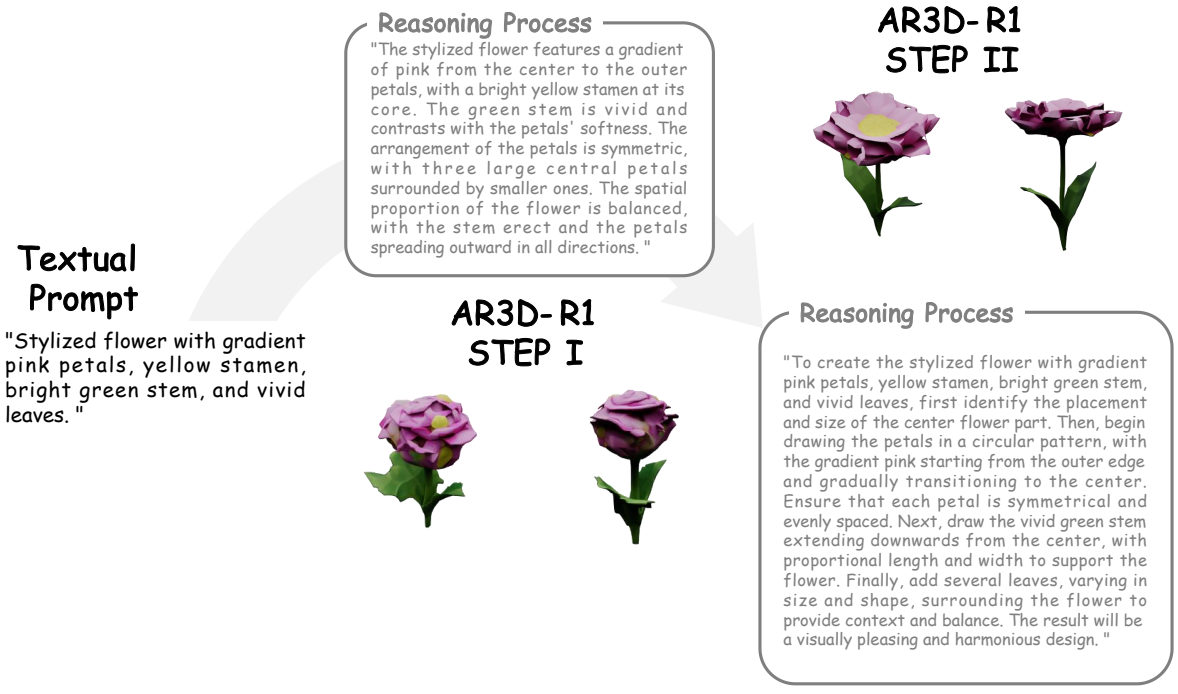

Figure 10. **Visualization of the Two-Step Reasoning Generation Process in Stylized Flower.**

```
You are an expert 3D model evaluator specializing in multi-view consistency assessment.
You are given a text prompt: "{prompt}"

Below are {num_views} rendered views of a generated 3D object from different camera angles.

Your task is to evaluate the 3D model through comprehensive multi-view analysis:

Step 1: Provide a holistic description of the 3D object
Examine all views collectively to describe the complete 3D structure.
Focus on: overall geometry, distinctive features, surface properties (materials/colors), and structural integrity.
     Base your description solely on visual observation, independent of the text prompt.

Step 2: Evaluate cross-view spatial consistency across three dimensions:

Dimension 1: Shape Outline Consistency (0-1 score)
- Assess whether the silhouette and overall geometric form remain coherent across different viewpoints
- Check for contradictions in 3D structure when comparing views (e.g., inconsistent proportions, impossible geometry)
- Verify that the shape maintains logical spatial relationships across all perspectives

Dimension 2: Appearance Consistency (0-1 score)
- Evaluate whether colors, textures, and material properties are consistent across views
- Check if surface details (patterns, textures) align properly between different angles
- Assess lighting consistency and whether visual quality is maintained across viewpoints

Dimension 3: Object Parts Consistency (0-1 score)
- Verify that all object components (limbs, features, structural elements) are present and complete across views
- Check whether parts maintain correct spatial relationships and proportions when viewed from different angles
- Identify any missing, duplicated, or incorrectly positioned parts across the multi-view set

Important: The overall 3D consistency score is the sum of the three dimension scores (range: 0-3).

Your response must follow this JSON format:
1. Include four keys: "description", "shape_consistency", "appearance_consistency", "parts_consistency", "total_score
     ", "explanation"
2. Each consistency dimension should be scored individually (0-1, float allowed)
3. "total_score" = shape_consistency + appearance_consistency + parts_consistency (range: 0-3)
4. "explanation" should justify each dimension's score with specific observations from the views

Example response:
'''json
{
"description": "The 3D model depicts a quadrupedal creature with canine characteristics. Across multiple views, the
     object shows a head with ears, a torso, four legs, and a tail. The model exhibits brown texturing on its surface
     . The overall structure suggests a mammalian body plan.",
"shape_consistency": 0.8,
"appearance_consistency": 0.7,
"parts_consistency": 0.6,
"total_score": 2.1,
"explanation": "Shape Outline Consistency (0.8): The silhouette maintains coherent proportions across most views,
     with clear quadrupedal structure. Minor inconsistencies appear in the tail curvature between side and rear views
     , preventing a full score. Appearance Consistency (0.7): The brown texture is generally uniform, but lighting
     reveals some texture misalignment on the torso when comparing front and side perspectives. The surface quality
     varies slightly across views. Parts Consistency (0.6): Major components (head, body, legs, tail) are present in
     all views. However, facial features show inconsistency - only one eye is visible in frontal views when two
     should be present, and ear positioning shifts unnaturally between angles. The legs maintain reasonable
     attachment points. Total score: 0.8 + 0.7 + 0.6 = 2.1"
}
'''
```

Figure 11. **Prompt Template for 3D Consistency Evaluation using Qwen2.5-VL.**

```
You are tasked with assessing the appearance consistency of a 3D object rendered from multiple viewpoints.

Input:

A text description of the target object: "{prompt}"

A set of {num_views} rendered images representing distinct camera viewpoints.

Your evaluation must rely solely on visual evidence present in the images.

1. Global Appearance Summary

Provide a brief, objective summary of the object visible appearance, including:
- dominant chromatic characteristics
- material qualities (e.g., reflectance, roughness, transparency)
- texture distribution and visual complexity

Do not reference the text prompt.

2. Appearance Consistency Assessment

Quantitatively evaluate the cross-view appearance consistency along three criteria:

(a) Color Smoothness (0--1)
Assess whether chromatic properties remain stable across viewpoints:
- continuity of hue and luminance
- absence of abrupt or implausible color shifts
- overall cross-view chromatic coherence

(b) Material Realism and Coherence (0--1)
Evaluate whether material attributes are physically plausible and stable across views:
- consistent specular/roughness behavior
- stable surface reflectance characteristics
- absence of viewpoint-dependent material changes without physical justification

(c) Texture Rationality (0--1)
Examine whether texture patterns behave logically relative to the object geometry:
- alignment and continuity of texture patterns
- absence of distortion, stretching, or UV inconsistencies
- cross-view stability of fine-grained surface details

3. Output Format

Return a JSON dictionary containing:
- "summary" : global appearance summary
- "color_smoothness" : float in [0,1]
- "material_realism" : float in [0,1]
- "texture_rationality" : float in [0,1]
- "total_score" : sum of the three metrics (range: 0--3)
- "justification" : concise explanation referencing specific visual observations supporting each score

Example JSON Structure:
{
  "summary": "",
  "color_smoothness": 0.0,
  "material_realism": 0.0,
  "texture_rationality": 0.0,
  "total_score": 0.0,
  "justification": ""
}
```

Figure 12. **Complete Prompt Template for Appearance Consistency Evaluation using Qwen2.5-VL.**

```
You are an expert 3D model evaluator. Your task is to verify whether a generated 3D object matches the object
category specified in the text prompt based on joint observation of multiple viewpoints.

Text prompt: "{prompt}"

Step 1: Describe the geometric structure of the object
Examine all views collectively and describe the 3D geometry and shape characteristics you observe. Focus on overall
geometric form, structural shape, spatial proportions, and distinctive geometric features that define the object's
category.

Step 2: Extract the target category from prompt
Identify the primary object category specified in the text prompt.

Step 3: Verify category matching
Assess whether the observed geometric structure and shape characteristics match the category specified in the prompt.

Output Format:
- If the generated object's geometry matches the prompted category across all views, respond with: \boxed{1}.
- Otherwise, respond with: \boxed{0}

Only output a single number (0 or 1) inside the box.
```

Figure 13. **Prompt for Category Matching Verification.**

```
You are an expert 3D component analyzer. Your task is to evaluate the presence and completeness of specific object
parts in a generated 3D model using its point cloud representation.

Input:
- Point cloud P with N points
- Required component list: {components_list}
- Expected quantity list: {quantities_list}

Step 1: Part-focused object description
Analyze the point cloud and provide a concise description of the object, with emphasis on identifying visible parts
and their approximate quantities.

Step 2: Required component evaluation
Using both the point cloud and the description from Step 1, evaluate each required component in {components_list}
along two criteria:

2a. Existence (0 or 1):
- Score 1 if the component is present.
- Score 0 if it is entirely missing.

2b. Completeness (0--1):
Assess the geometric completeness of each component:
- Spatial coherence: whether its points form a connected, physically continuous structure.
- Shape plausibility: whether its geometry matches the expected part category.
- Quantity alignment: whether the observed quantity aligns with the expected quantity in {quantities_list}.

Final Output (JSON):
{
  "part_description": "...",
  "existence_scores": [e_1, e_2, ...],
  "completeness_scores": [c_1, c_2, ...],
  "component_scores": [e_1 + c_1, e_2 + c_2, ...]
}
```

Figure 14. **Per-component Evaluation Prompt for ShapeLLM.**

References

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

[2] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 1

[3] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26576–26586, 2025. 1

[4] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022. 2

[5] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023. 2

[6] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025. 1

[7] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1

[8] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Rui Huang, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Shanghang Zhang, Peng Gao, et al. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025. 1

[9] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025. 1

[10] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025. 1

[11] Dongzhi Jiang, Renrui Zhang, Haodong Li, Zhuofan Zong, Ziyu Guo, Jun He, Claire Guo, Junyan Ye, Rongyao Fang, Weijia Li, et al. Draco: Draft as cot for text-to-image preview and rare concept generation. *arXiv preprint arXiv:2512.05112*, 2025. 1

[12] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2024. 2

[13] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025. 1

[14] OpenAI: Introducing OpenAI o3 and o4 mini. 2025. *(2025)*, https://openai.com/o3/. 1

[15] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pages 214–238. Springer, 2024. 3

[16] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023. 1

[17] ByteDance Seed, Yuyu Zhang, Jing Su, Yifan Sun, Chenguang Xi, Xia Xiao, Shen Zheng, Anxiang Zhang, Kaibo Liu, Daoguang Zan, et al. Seed-coder: Let the code model curate data for itself. *arXiv preprint arXiv:2506.03524*, 2025. 1

[18] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 1

[19] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025. 1

[20] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19615–19625, 2024. 1

[21] Haoming Song, Delin Qu, Yuanqi Yao, Qizhi Chen, Qi Lv, Yiwen Tang, Modi Shi, Guanghui Ren, Maoqing Yao, Bin Zhao, et al. Hume: Introducing system-2 thinking in visual-language-action model. *arXiv preprint arXiv:2505.21432*, 2025. 1

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

[23] Chengzhuo Tong, Ziyu Guo, Renrui Zhang, Wenyu Shan, Xinyu Wei, Zhenghao Xing, Hongsheng Li, and Pheng-Ann

Heng. Delving into rl for image generation with cot: A study on dpo vs. grpo. *arXiv preprint arXiv:2505.17017*, 2025. 1

[24] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025. 2

[25] Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. Llama-mesh: Unifying 3d mesh generation with language models. *arXiv preprint arXiv:2411.09595*, 2024. 2

[26] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 2

[27] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 1, 2

[28] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023. 1

[29] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025. 1

[30] Xianghui Yang, Huiwen Shi, Bowen Zhang, Fan Yang, Jiacheng Wang, Hongxu Zhao, Xinhai Liu, Xinzhou Wang, Qingxiang Lin, Jiaao Yu, et al. Hunyuan3d 1.0: A unified framework for text-to-3d and image-to-3d generation. *arXiv preprint arXiv:2411.02293*, 2024. 1

[31] Junliang Ye, Zhengyi Wang, Ruowen Zhao, Shenghao Xie, and Jun Zhu. Shapellm-omni: A native multimodal llm for 3d generation and understanding. *arXiv preprint arXiv:2506.01853*, 2025. 1, 2

[32] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024. 1

[33] Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun 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. 1

[34] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025. 1