CIRCULAR-DPO: ALIGNING MULTI-STAGE 3D GENERATIVE MODELS VIA PREFERENCE FEEDBACK LOOP

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-stage generative models have shown great promise in 3D content creation due to focused generation of structure or texture in different stages, but their outputs often fail to align with human preferences. The key bottleneck to apply alignment methods is the presence of non-differentiable operations between generative stages. This disconnection stops preference signals applied to the final output from being backpropagated to the crucial, early stages of generation, while simple separated stage-wise alignment leads to texture-geometry inconsistency. To address this challenge, we introduce Circular-DPO, which builds a preference feedback loop to align multi-stage 3D generation models to human preference. Our method first applies Direct Preference Optimization (DPO) to refine the final 3D asset. We then construct new preference pairs by sampling and decoding the assets generated by the optimized model. These newly-formed pairs are used to train the preceding generative stage, effectively creating a feedback loop that bridges the non-differentiable gap. Furthermore, to enhance robustness against noisy data, we introduce a quality-aware weighting mechanism that prioritizes reliable preference pairs during training. Experiments demonstrate that our approach improves the alignment of generated 3D content with human preferences by enabling holistic, multi-stage optimization.

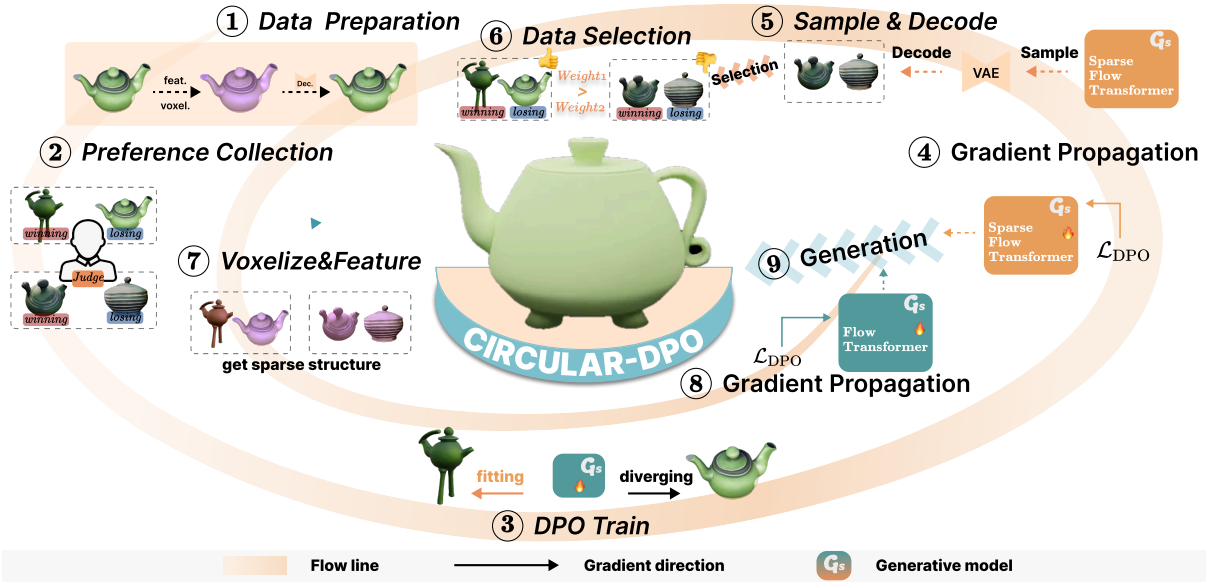

Figure 1: Overview of the Circular-DPO. Our method creates a 'preference feedback loop' using a back-to-front optimization strategy, ultimately generating 3D assets that are better aligned with human preferences.

1 INTRODUCTION

3D content generation technology benefits practical applications in various domains, including film (Wang et al., 2025), animation (Jiang et al., 2024), game design (Hao et al., 2021), and architectural design (Li et al., 2024a). In recent years, revolutionary advancements in Diffusion Models (Hunyuan3D Team, 2025; Rombach et al., 2022) have significantly propelled the development of automated 3D content generation. Current mainstream techniques primarily fall into two categories: methods based on 3D native data (Li et al., 2024b) and lift-off methods that optimize 3D representations from 2D generative models (Poole et al., 2022). These technologies can generate highly complex, high-quality, and view-consistent 3D content from text or image prompts. Despite these rapid advancements, existing research indicates that the 3D content produced by these state-of-the-art models often fails to align effectively with human aesthetic and functional preferences (Ye et al., 2024). This discrepancy between

the generated results and human expectations limit their application potential in scenarios that demand high fidelity to user intent.

To enhance the alignment of 3D generation with human preferences, a natural approach is to apply Direct Preference Optimization (DPO) (Rafailov et al., 2023), which iteratively refines an initial 3D representation to obtain a final asset that better conforms to preferences. However, the application of DPO encounters a limitation, a deeper analysis of existing technical frameworks reveals that mainstream 3D generation pipelines commonly involve multiple stages. Other advanced methods like MVDream (Shi et al., 2023), DreamFusion (Poole et al., 2022) and Magic3D (Lin et al., 2023) also involve multiple stages. For instance, 3D native methods like Trellis (Xiang et al., 2024) employ a two-stage generation pipeline to produce structured 3D latent features: the first stage generates a sparse structure, and the second stage populates it with local details, with each stage corresponding to a distinct flow model. This introduces a critical challenge: these multi-stage processes often contain non-differentiable operations between stages. Consequently, gradient information from later stages particularly from the final preference optimization applied to the local details stage cannot be effectively back propagated to preceding stages, such as the sparse 3D structure generation stage. As a result, existing methods can only perform preference optimization on the final output, failing to guide the generation in crucial early-to-intermediate stages, which often determine the geometry or structure of the generated content. At the same time, separate alignment of each stage leads to geometry texture inconsistency. This fundamentally limits the potential for aligning the overall generation quality with human preferences.

To address the issues of stage separation, we propose a new paradigm centered around a human preference-driven closed-loop optimization framework. This framework extends the application of DPO from the final 3D asset to the entire generation chain. Taking Trellis as an example, our method consists of three main steps: (1) Optimizing the Final Stage. We leverage a human preference dataset to fine-tune the generative model of the second stage via DPO. This directly optimizes the generated details of the final 3D asset, yielding a high-quality output aligned with human preferences. (2) Constructing New Preference Pairs. We pair the optimized, high-quality 3D asset with its un-optimized original counterpart to form a new set of preference pairs. (3) Guiding the Preceding Stage. We then apply DPO using these new preference pairs to guide and optimize the model responsible for generating the sparse 3D structure in the first stage. This process establishes a data loop, allowing final human preferences to continuously feed back and enhance the earlier stages of the multi-stage generation pipeline, thereby achieving joint optimization.

When applying DPO in these stages, we observed that both the original human preference data and our newly constructed preference pairs suffer from two types of noise: inherent data noise and preference ranking noise, which can significantly impair optimization performance. To enhance the robustness of the preference alignment, we introduce a quality-aware preference pair weighting mechanism. This mechanism dynamically calculates weights for each preference pair based on its quality and reliability, assigning higher importance to more credible, high-quality pairs. A reward loss, computed based on these weighted preference pairs, is then used to guide the updates of the 3D representations. This approach effectively suppresses noise interference, ensuring a more stable and precise optimization process.

Building upon this framework, we conduct comprehensive experiments demonstrating that our Circular-DPO approach successfully bridges the non-differentiable gap between generative stages through iterative preference propagation. Our experimental validation encompasses both quantitative metrics (ImageReward, HPSv2, Reward3D, CLIP-score) and qualitative human evaluations across 100 diverse prompts, showing competitive performance against baseline methods including Trellis, MVDream, and DreamReward, with particular improvements in multi-stage coordination. Furthermore, our ablation studies reveal that the quality-aware weighting mechanism effectively mitigates noise interference from both human-annotated and constructed preference pairs, while the circular feedback loop enables holistic optimization across all generation stages. These results demonstrate the viability of preference-driven alignment in multi-stage 3D generation and provide a systematic framework for bridging non-differentiable operations in complex generative pipelines.

The main contributions of this paper can be summarized as follows:

(1) A Closed-Loop Paradigm for Multi-Stage Joint Optimization.We are the first to propose a data loop constructed via DPO that feeds back human preferences from the final asset to optimize preceding generation stages. This approach alleviates the critical bottleneck of interrupted gradient flow in multi-stage 3D generation, enabling joint optimization.

(2) A Robust Preference Weighting Mechanism. We designed a quality-aware weighting strategy for preference pairs that effectively mitigates the impact of noise from both human-annotated and newly constructed preference data. This method enhances the stability and effectiveness of DPO while preserving data diversity.

(3) Effective Improvement in Generation Quality.We successfully constructed a 3D preference dataset based on human-ranked annotations. By applying our proposed closed-loop optimization paradigm and weighting mechanism, we significantly improved the quality of generated 3D models and their alignment with human preferences.

2 Related Work

2.1 Advances in 3D Content Generation

The field of 3D content generation has shifted from early methods using Generative Adversarial Networks (GANs) (Goodfellow et al., 2014; Heusel et al., 2017) for representations like voxels and meshes (Wu et al., 2016; Chan et al., 2022; Skorokhodov et al., 2024) to the now-dominant paradigm of diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Luo & Hu, 2021; Nichol et al., 2022). A key catalyst has been score distillation sampling (SDS) (Rombach et al., 2022), which enables text-to-3D synthesis by distilling knowledge from pre-trained 2D models without requiring large-scale 3D datasets (Poole et al., 2022; Lin et al., 2023; Chen et al., 2023; Wang et al., 2023b).

To enhance efficiency and quality, recent works have focused on generation within a compact latent space. State-of-the-art models like Hunyuan3D-DiT (Hunyuan3D Team, 2025) and Trellis (Xiang et al., 2024) employ two-stage pipelines that first generate a latent shape and then synthesize texture. Trellis introduced a unified Structured LATent (SLAT) representation, enabling decoding into diverse formats like Radiance Fields (Mildenhall et al., 2020) and 3D Gaussians (Kerbl et al., 2023), and leverages large-scale flow-based transformers (Lipman et al., 2022; Esser et al., 2024). Other modern approaches explore alternative representations such as vector sets (Zhang et al., 2023; Li et al., 2024b; Zhao et al., 2024) and triplanes (Wang et al., 2023a; Lan et al., 2024). Our work aims to refine these powerful generative models by aligning their outputs with human preferences.

2.2 Preference Optimization for Generative Models

Aligning generative models with human preferences is a critical challenge, traditionally addressed in NLP by Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Stiennon et al., 2020). This paradigm, which involves training a reward model and fine-tuning with RL, was later adapted for text-to-image models (Xu et al., 2023; Fan et al., 2023; Black et al., 2023) and 3D generation (Ye et al., 2024). However, RLHF pipelines are often complex and unstable to train (Rafailov et al., 2023).

Direct Preference Optimization (DPO) (Rafailov et al., 2023) emerged as a more stable alternative, reframing alignment as a simple classification task that bypasses explicit reward modeling. This approach and its variants, like Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024), have proven effective for LLMs. The application of DPO to visual synthesis is a nascent but promising area. Diffusion-DPO (Wallace et al., 2024) adapted the objective for models like SDXL (Podell et al., 2023), while others have extended it to flow-based models for scientific applications (Jiao et al., 2024) and to 3D generation with DreamDPO (Zhou et al., 2025), which leverages Large Multimodal Models (LMMs) for preference data.

Our work builds on these advancements by applying a DPO-based framework to a versatile, multi-stage 3D model.

3 Preliminaries

3.1 A brief introduction of Flow Matching DPO

Flow Matching (Jiao et al., 2024) is a powerful generative modeling framework that learns a vector field to map a simple prior distribution, $p_0(x)$, to a complex data distribution, $p_1(x)$. The core idea is to define a time-varying probability path, ψ_t, that transports a sample $x_0 \sim p_0(x)$ from the prior to a data point $x_1 \sim p_1(x)$. This transport is governed by a vector field v_t according to the ordinary differential equation:

$$\frac{d\psi_t(x_0)}{dt} = v_t(\psi_t(x_0)) \tag{1}$$

In Conditional Flow Matching (CFM), a specific type of flow model, a simple linear interpolation is used to define the path between a prior sample x_0 and a data sample x_1: $x(t) = (1-t)x_0 + tx_1$. The generative process is then learned as a time-varying vector field, approximated by a neural network $v_\theta(x, t)$. The network is trained to predict the direction toward the data sample by minimizing the CFM objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_0\sim p_0,x_1\sim p_1}\|v_\theta(x(t), t) - (x_1 - x_0)\|_2^2 \tag{2}$$

For conditional tasks, such as generating a 3D structure x_1 given a condition c, this principle holds. We can explicitly define the time-dependent Mean Squared Error (MSE) at a specific time t as the squared L2 norm of the difference between the predicted and target vector fields:

$$\text{MSE}_t(x_0, x_1; \theta) = \|v_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)\|_2^2 \tag{3}$$

The overall training objective is then the expectation of this quantity over all possible samples and time steps:

$$\mathcal{L} = \mathbb{E}_{t,x_0\sim p_0,x_1\sim p_1}[\text{MSE}_t(x_0, x_1; \theta)] \tag{4}$$

Flow-DPO adapts the Direct Preference Optimization (DPO) framework to flow models, particularly for tasks like text-to-3D generation. Given a preference pair where a sample x^w is preferred over a sample x^l (denoted as

$x^w \succ x^l$), the DPO objective is formulated to increase the likelihood of the preferred sample while decreasing that of the dispreferred one. The general form of the objective is:

$$\begin{aligned}\mathcal{L}_{\text{DPO}} &= -\mathbb{E}_{x^w,x^l,i} \log \sigma \left(\beta T \mathbb{E} \left[\log \frac{p_{\text{opt}}(x^w_{i-1}|x^w_i)}{p_{\text{ref}}(x^w_{i-1}|x^w_i)} - \log \frac{p_{\text{opt}}(x^l_{i-1}|x^l_i)}{p_{\text{ref}}(x^l_{i-1}|x^l_i)} \right] \right) \\ &= -\mathbb{E}_{x^w,x^l,i} \log \sigma \left(\beta T \left[\mathcal{J}(x^w_i; p, p_{\text{ref}}) - \mathcal{J}(x^w_i; p, p_{\text{opt}}) - \mathcal{J}(x^l_i; p, p_{\text{ref}}) + \mathcal{J}(x^l_i; p, p_{\text{opt}}) \right] \right)\end{aligned} \tag{5}$$

As demonstrated by Jiao et al. (2024), the abstract log-probability ratios in the DPO objective can be effectively approximated using the MSE from the Flow Matching framework. This insight simplifies the final training objective for Flow-DPO into the following practical form:

$$\begin{aligned}\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{x^w_{0,1},x^l_{0,1},t} \log \sigma \Big(\beta T \Big[&\left(\text{MSE}_t(x^w_0, x^w_1; \theta_{\text{ref}}) - \text{MSE}_t(x^w_0, x^w_1; \theta_{\text{opt}})\right) \\ &- \left(\text{MSE}_t(x^l_0, x^l_1; \theta_{\text{ref}}) - \text{MSE}_t(x^l_0, x^l_1; \theta_{\text{opt}})\right) \Big] \Big)\end{aligned} \tag{6}$$

3.2 A BRIEF INTRODUCTION OF TRELLIS

Trellis architecture (Xiang et al., 2024) exemplifies a multi-stage approach by explicitly decoupling the generation of heterogeneous 3D representations into two distinct stages. In the first stage, a Rectified Flow Transformer θ_{sparse} generates a sparse voxel structure, x_{sparse}, to outline the object's active surface. For efficiency, these voxels are then compressed by a 3D convolutional VAE into a low-dimensional feature grid. Conditioned on this sparse structure, the second stage employs a sparse convolutional Transformer θ_{slat} to generate local latent variables, x_{slat}, which encode fine-grained geometric details and texture information. The framework's core innovation lies in its Structured LATents (SLAT), which couple the sparse topology from the first stage with the dense visual features from the second. This hybrid representation enables the generation of diverse outputs—such as radiance fields, 3D Gaussian splats, and meshes—through independent, specialized decoders.

3.3 GRADIENTS FAIL TO PROPAGATE FROM A LATER STAGE BACK TO AN EARLIER ONE

Texture-geometry inconsistency in Trellis multi-stage training pipelines, which manifests as non-manifold geometry or lost texture detail, arises from a divergence in training gradients. This issue stems from two problems: 1) a non-differentiable gap prevents gradients related to texture detail from propagating back from the second stage to the first, and 2) gradients from the flow-matched sparse structure do not effectively contribute to the training objective. Consequently, the optimization of geometric structure and texture detail becomes decoupled.

To resolve this, we adopt a back-to-front optimization strategy. We first optimize the second-stage (detail) model and then use its outputs to construct new preference pairs. These pairs implicitly carry the preference gradients from the detail stage. Since this process preserves the initial geometric structure, it creates a "data loop" that bridges the optimization gap, allowing human preferences to guide both stages of the pipeline holistically.

In complex multi-stage generative pipelines, preference feedback loops and preference-driven alignment provide a systematic framework for bridging non-differentiable operations.

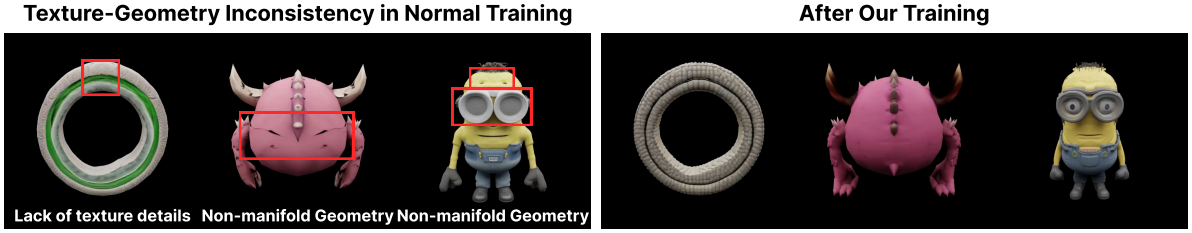

Figure 2: Demonstration of texture and geometry inconsistency.

4 METHOD

4.1 OVERVIEW

Our aim is to post-train a multi-stage generation pipeline (such as Trellis) with a predefined preference dataset. The gradient of DPO over the second stage of Trellis cannot backpropagate to the first stage. We propose to embed the supervision signal as preference pairs and forward them to the first stage, forming a preference feedback loop. As illustrated in the overview Figure 3, our method achieves direct preference optimization for each stage within a two-stage 3D generation framework. The process unfolds in three main steps.

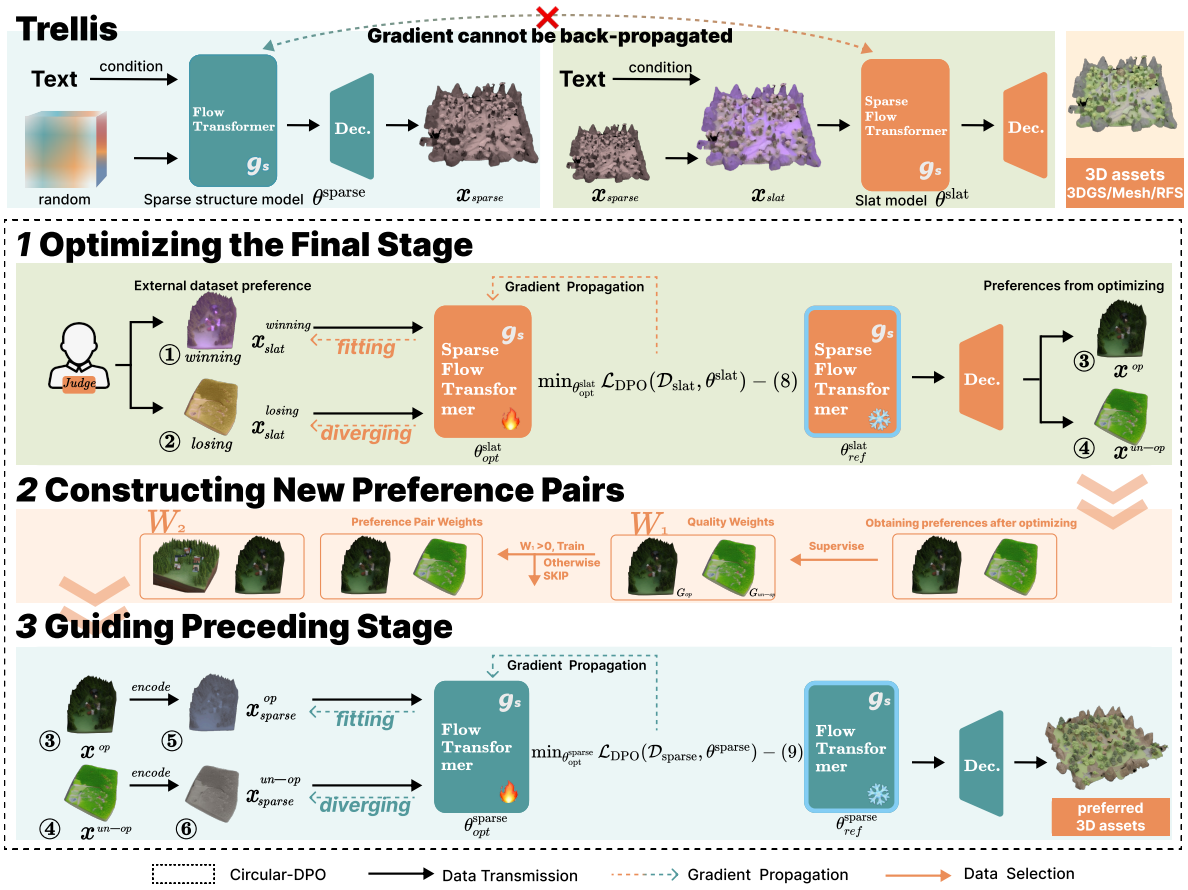

Figure 3: The overall framework of our Circular-DPO.In the first line, we review the 3D generation process of Trellis. Our method is shown in the dashed box.

Step 1: Optimizing the Final Stage. We first fine-tune the generative flow model of the final 3D asset with the DPO objective to directly embed human preferences.

Step 2: Constructing New Preference Pairs. We then form new preference pairs using the optimized asset from Step 1 as the positive sample (x^{op}) and the original as the negative sample ($x^{\text{un-op}}$). This process is guided by a quality weight w_1 and a preference weight w_2 to supervise the optimization.

Step 3: Guiding the Preceding Stage. Finally, these new preference pairs are used to fine-tune the sparse structure model from the initial generation stage, propagating the preference signal frontward.

4.2 Initial Construction of Human Preferences

We construct our initial preference dataset from DreamReward (Xu et al., 2023), a human-labeled collection designed for evaluating preference alignment. The public dataset, $\mathcal{D}$, contains 1,000 prompts, each associated with nine ranked 3D assets. To form a preference pair (x^w, x^l), we sample two assets generated from the same prompt, designating the higher-ranked asset as the positive sample x^w and the lower-ranked asset as the negative sample x^l. Further details of the dataset are provided in the appendix.

4.3 DPO as a Bridge Between Generative Stages

As detailed in Section 3.2, the Trellis 3D generation pipeline consists of two distinct stages. The transition between these stages is non-differentiable, which prevents gradient-based optimization from propagating from the second stage back to the first. We overcome this limitation by employing DPO as a bridge between the stages. This strategy enables a unified preference optimization framework across the entire pipeline, ensuring that preferences applied to the final asset can effectively guide the initial, sparse-structure generation.

Our objective is to post-train a multi-stage generative pipeline using a predefined preference dataset, $\mathcal{D}$. The core of our method is a multi-step application of a general Direct Preference Optimization (DPO) objective for flow models. After initial data preprocessing (detailed in the appendix), we define this general loss function for a given

dataset $\mathcal{D}'$ and preference weights w_2 as:

$$\mathcal{L}_{\text{DPO}}(\mathcal{D}';\theta) = -\mathbb{E}_{(x_0^w,x_1^w),(x_0^l,x_1^l)\sim\mathcal{D}',t}\log\sigma\Bigg(\beta T w_2\Big[\left(\text{MSE}_t(x_0^w,x_1^w;\theta_{\text{ref}}) - \text{MSE}_t(x_0^w,x_1^w;\theta_{\text{opt}})\right) - \left(\text{MSE}_t(x_0^l,x_1^l;\theta_{\text{ref}}) - \text{MSE}_t(x_0^l,x_1^l;\theta_{\text{opt}})\right)\Big]\Bigg) \tag{7}$$

Step 1: Optimizing the Final Stage. First, we optimize the final stage of the pipeline, which generates the structured latent (x_{slat}) representation. Using the preprocessed human preference dataset $\mathcal{D}_{\text{slat}}$, we fine-tune the policy model $\theta_{\text{opt}}^{\text{slat}}$ against a reference model $\theta_{\text{ref}}^{\text{slat}}$. This is achieved by minimizing our general DPO objective:

$$\min_{\theta_{\text{opt}}^{\text{slat}}} \mathcal{L}_{\text{DPO}}(\mathcal{D}_{\text{slat}}, \theta^{\text{slat}}) \tag{8}$$

Step 2: Constructing New Preference Pairs. Next, we construct a new preference dataset to bridge the stages. We generate two assets for each prompt: an optimized asset x^{op} from the fine-tuned model in Step 1, and an un-optimized asset $x^{\text{un-op}}$ from the original reference model. These form new preference pairs $(x^{\text{op}}, x^{\text{un-op}})$. This step also defines a quality weight w_1 and a preference alignment weight w_2, which are used to guide the optimization.

Step 3: Preceding Stage Guidance. Finally, we propagate the preference signal to the preceding stage, which generates the sparse structure (x_{sparse}). The newly constructed asset pairs $(x^{\text{op}}, x^{\text{un-op}})$ are passed through an encoder E to create a new dataset $\mathcal{D}_{\text{sparse}}$. We then fine-tune the sparse model's parameters $\theta_{\text{opt}}^{\text{sparse}}$ by minimizing the DPO objective again:

$$\min_{\theta_{\text{opt}}^{\text{sparse}}} \mathcal{L}_{\text{DPO}}(\mathcal{D}_{\text{sparse}}, \theta^{\text{sparse}}) \tag{9}$$

4.4 Data Selection

When using DPO to construct preference pairs between generative stages, a supervised data selection method is necessary to filter the pairs, ensuring they contribute to an overall improvement in model quality.

4.4.1 Quality-Based Filtering with w_1

Inspired by prior work Karthik et al. (2024), we first devise a quality score to filter out low-quality or mislabeled preference pairs. For a given prompt c, we sample k assets $\{I_1^{\text{op}}, \ldots, I_k^{\text{op}}\}$ from the optimized policy $\theta_{\text{opt}}^{\text{slat}}$ and $N-k$ assets $\{I_{k+1}^{\text{un-op}}, \ldots, I_N^{\text{un-op}}\}$ from the reference policy $\theta_{\text{ref}}^{\text{slat}}$. We then score every asset I_i in this collection of N candidates using a pre-trained reward model, $\mathcal{R}_\phi(I_i, c)$.

Based on these scores, we calculate the win count C_i for each asset I_i in all-pairs comparisons. The preference probability $P(I_i)$ is then computed as:

$$P(I_i) = \frac{2C_i}{N(N-1)} \tag{10}$$

To amplify score differences, we apply a transformation $G_i = 2^{P(I_i)} - 1$. The final quality gap score, w_1, for a given preference pair $(I^{\text{op}}, I^{\text{un-op}})$ is the difference in their transformed scores:

$$w_1 = G_{\text{op}} - G_{\text{un-op}} \tag{11}$$

A higher w_1 indicates a larger and more reliable quality gap. We use this score to filter our dataset, discarding any preference pair where $w_1 \leq \tau$, with a threshold τ set to zero.

4.4.2 Preference Reliability Weighting with w_2

Following Wu et al. (2024), we further introduce a dynamic weight, w_2, to down-weight samples that are likely noisy, even if they pass the w_1 filter. This weight is based on the log-likelihood of a preference pair under a reward model r:

$$h(c, x^w, x^l) = \log\sigma(r(c, x^w) - r(c, x^l)) \tag{12}$$

The weight w_2 is the normalized exponentiated log-likelihood, which prioritizes pairs that the reward model considers highly probable:

$$w_2(c, x^w, x^l) = \frac{\exp(h)}{\mathbb{E}_{\mathcal{D}}[\exp(h)]} \tag{13}$$

By integrating these two mechanisms, our final loss function, $\mathcal{L}_{\text{Circular-DPO}}$, applies the standard DPO loss only to high-quality pairs, weighted by their reliability:

$$\mathcal{L}_{\text{Circular-DPO}} = \begin{cases} \mathcal{L}_{\text{DPO}} & \text{if } w_1 > \tau \\ 0 & \text{if } w_1 \leq \tau \end{cases} \tag{14}$$

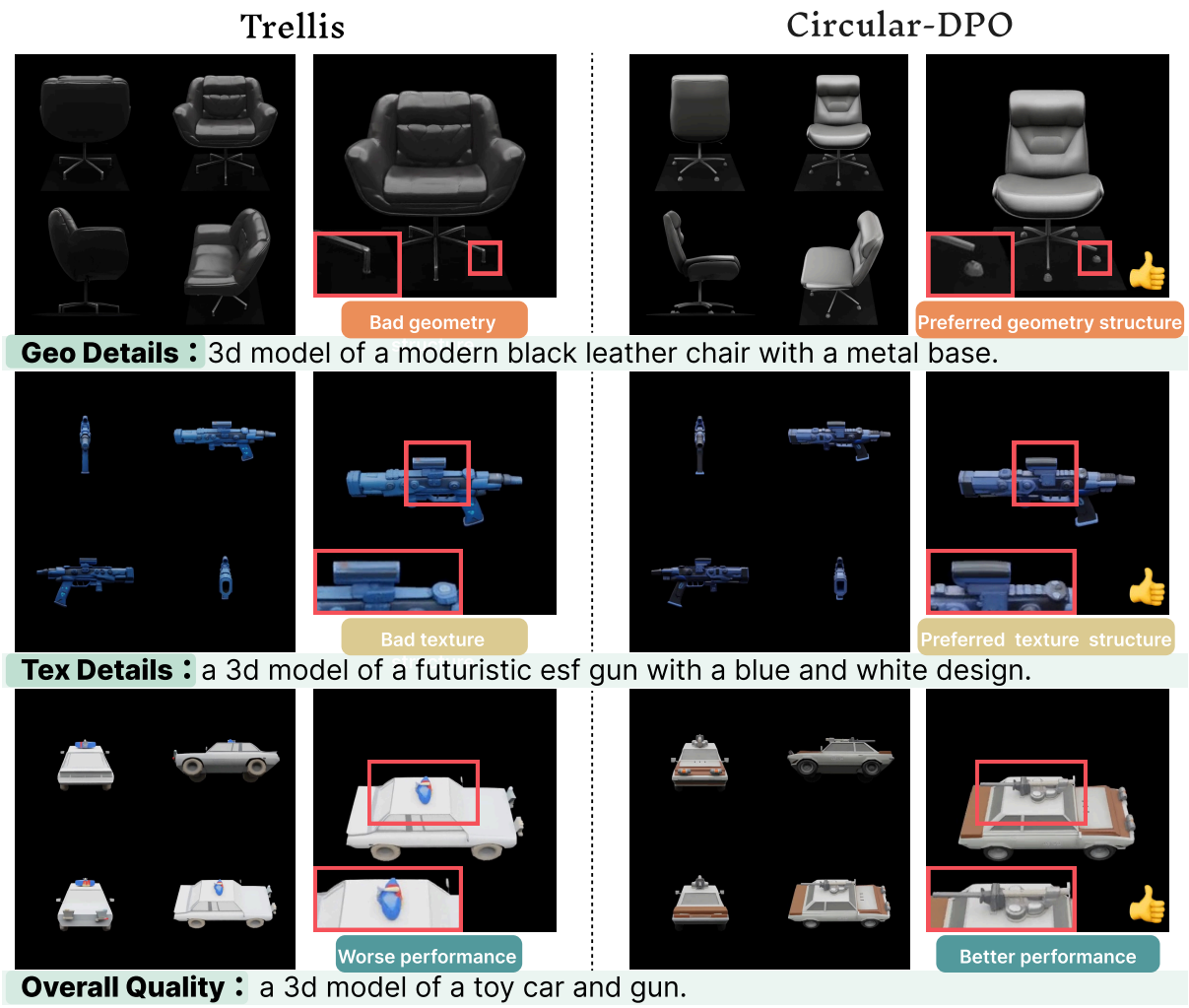

Figure 4: Qualitative Visual Results.

5 Experiment

We evaluated our proposed Circular-DPO against state-of-the-art 3D generation models. For this comparison, we curated a new test set of 100 prompts designed to cover a wide range of creativity and complexity. This set was constructed by first selecting 30 diverse prompts from the 3DRewardDB, and then using Gemini 1.5 Pro to generate 70 additional, novel prompts based on the full 3DRewardDB collection. To comprehensively evaluate our method, we conduct both qualitative visual comparisons and rigorous quantitative analysis. Our quantitative evaluation employs four distinct metrics to assess alignment with human preferences and text-image consistency ImagerewardXu et al. (2023), HPSv2, Reward3DYe et al. (2024), and CLIP ScoreRadford et al. (2021). In addition, we also conducted a detailed user study specifically to further demonstrate the consistency between our method and human preferences.

5.1 Qualitative comparison

It shows 3D assets generated by our Dreamreward and Trellis baselines for multiple prompts, allowing for an intuitive visual comparison. As illustrated in the Figure 4, our method achieves significant enhancements in geometry, texture, and overall visual quality after preference alignment. For instance, in the first example, the armchair's backrest geometry is optimized to better align with human aesthetic preferences. In the second example, given a similar geometric base, our model generates markedly more detailed and refined textures on the gun. Finally, the third example demonstrates a holistic improvement across multiple domains, where our method simultaneously enhances the model's texture, geometry, and text-to-3D consistency.

5.2 Quantitative comparison

In Tables 1 and 2, we compare our method with several benchmarks, namely LatentNeRFMildenhall et al. (2020), MVDreamShi et al. (2023), DreamRewardYe et al. (2024), DreamDPOZhou et al. (2025), and TrellisXiang et al. (2024). This shows that our results consistently outperform other baselines across multiple evaluation criteria.

The quantitative results, presented in the Table 1, demonstrate that our method achieves a comprehensive improvement over the Trellis baseline. Specifically, Circular-DPO surpasses the baseline across all 2D evaluation metrics. On the 3DReward metric, our method also shows superior performance. Furthermore, the higher CLIP Score indicates that our approach maintains high semantic consistency. While our method significantly advances upon the baseline, we note that it does not yet outperform previous score distillation-based approaches. However, the time we take to generate 3D assets is much shorter than that of the distillation method.

Table 1: Quantitative comparisons

Evaluation	Time	3D representation	Imagereward↑	Hpsv2 ↑	Reward3D ↑	CLIP-score ↑
Latent-NeRF	15min	Nerf	-0.7774	0.1722	-2.9517	0.3137
MVDream	2h	Nerf	-0.3948	0.2013	-0.4292	0.3431
DreamReward	40min	Nerf	1.3014	0.2302	2.0158	0.3428
DreamDPO	1.5h	Nerf	-0.3774	0.2013	-0.1767	0.3474
Trellis	10s	3DGS	-0.6684	0.1902	-0.6450	0.3227
OURS	14s	3DGS	-0.3684	0.2010	-0.5969	0.3290

5.3 Ablation Study

In Table 2, we present a series of eight ablation studies to analyze the contributions of our method's key components. To evaluate these ablations, we conduct both qualitative and quantitative comparisons. The quantitative results, visualized in Figure 5, are measured using four metrics.

Our ablation study comprises eight experimental groups to isolate the effects of our proposed components. **Groups (b), (c), and (d)** evaluate different applications of Direct Preference Optimization (DPO): (b) trains both stages independently without our data loop, while (c) and (d) apply DPO exclusively to the first-stage sparse model and the second-stage detail model, respectively. **Groups (e) and (f)** serve as non-DPO baselines, which are directly fine-tuned on the preference data to assess the impact of the DPO objective itself. **Groups (g) and (h)** analyze the contribution of our proposed data selection weights by ablating the sample quality weight (w_1) and the preference pair reliability weight (w_2), respectively. Finally, **Group (i)** represents the baseline performance of the original Trellis model without any preference alignment fine-tuning.

Impact of DPO on Model Components Our ablations demonstrate that DPO significantly enhances both geometric and textural generation. Observing groups (a), (c), and (i) in Figure 5, DPO training on the sparse structure model corrects the geometric errors and poor structures common in the untrained baseline (i). Compared to standard post-training (e), our DPO approach (c) achieves superior geometric quality and text-alignment. Furthermore, our full method with its data propagation strategy (a) improves fine-grained structures, reducing artifacts like holes and broken faces compared to isolated DPO training (c).

Similarly, for the latent feature model, DPO training enriches texture generation. As seen in Figure 5, our method overcomes the monotonous textures of the baseline (i) and handles complex details, such as facial features, more

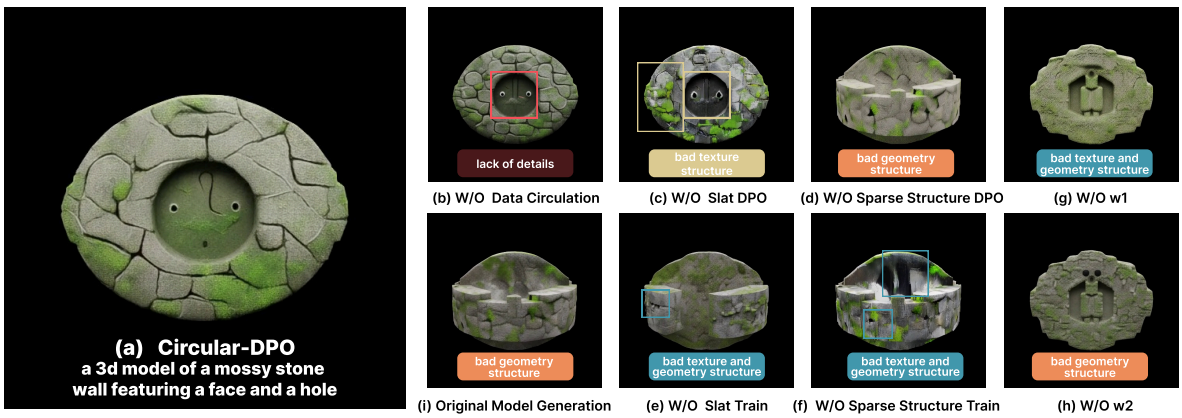

Figure 5: Visual results of the ablation study.

effectively than standard post-training (e). Again, our full method (a) shows superior detail and handling of complex textures compared to isolated DPO on the second stage (d).

DPO vs. Standard Post-Training With an equivalent amount of data, DPO provides a more substantial quality improvement than standard post-training. A comparison of DPO-trained models (c, d) against conventionally trained models (e, f) in Figure 5 reveals this advantage. While standard training offers some enhancements, it struggles with error correction. In contrast, DPO yields superior overall quality, greater texture richness.

Effect of Quality and Preference Weights The quality (w_1) and preference (w_2) weights are crucial for guiding the model toward optimal results. As shown in groups (a), (g), and (h) in Figure 5, ablating either of these weights leads to suboptimal outcomes. While some improvements over the baseline are still present, the final generation quality is clearly diminished without the filtering and weighting mechanisms.

Table 2: Ablation Study

Evaluation	label	Imagereward↑		Hpsv2 ↑		Reward3D ↑		CLIP-score ↑	
		rank	score	rank	score	rank	score	rank	score
Circular-DPO	(a)	2	-0.3684	1	0.2010	1	-0.5969	1	0.3290
Trellis-w/o-bridge-dpo	(b)	5	-0.5393	4	0.1952	9	-0.7674	5	0.3227
Trellis-dpo-only step1	(c)	8	-0.5880	8	0.1907	3	-0.6336	3	0.3264
Trellis-dpo-only step2	(d)	4	-0.5028	5	0.1931	8	-0.7112	9	0.3168
Trellis-normal-only step1	(e)	7	-0.5642	7	0.1915	4	-0.6427	8	0.3209
Trellis-normal-only step2	(f)	6	-0.5527	6	0.1927	5	-0.6442	2	0.3215
OURS-Trellis-w/o-w1	(g)	1	-0.3246	2	0.2009	2	-0.6199	2	0.3286
OURS-Trellis-w/o-w2	(h)	3	-0.4908	3	0.1969	7	-0.6661	4	0.3230
Trellis	(i)	9	-0.6684	9	0.1902	6	-0.6450	5	0.3227

5.4 User research

We recruited 20 participants with 3D modeling experience to take part in the user study. During the experiment, users were free to view the generated 3D models from various perspectives.Each participant evaluated six randomly selected 3D assets, scoring them on a five-point scale across five criteria and making a preference selection based on the overall appeal of the model.The results compellingly demonstrate the effectiveness of our method. Nearly 70% of participants preferred the assets generated by our approach over the baseline. Furthermore, our method achieved higher average scores across all evaluated dimensions compared to the original Trellis model, confirming its ability to enhance model quality in a way that aligns with genuine human preferences.

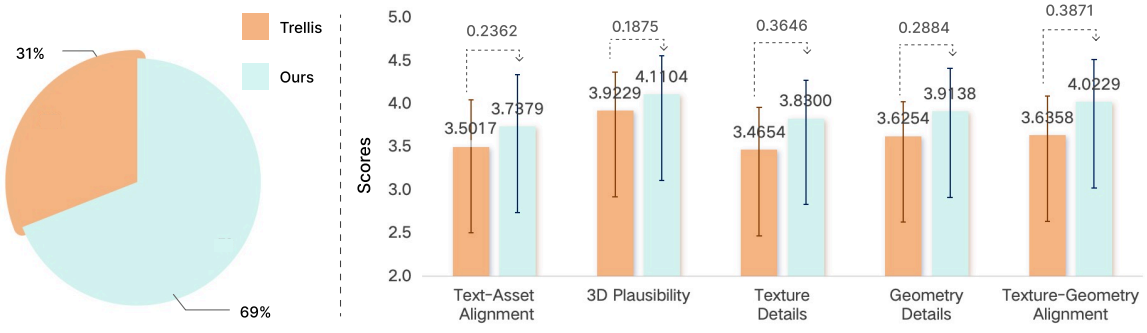

Figure 6: Display of user study results. Left: The generation effect that users prefer more. Right: The five-point scale score results for text consistency, 3D plausibility, texture details, geometric details, and texture-geometric consistency.

6 Conclusion

The proposed method creates a "preference feedback loop" using a back-to-front optimization strategy. First, it fine-tunes the final (slat) generative stage using human preferences. Then, it constructs new preference pairs by contrasting the outputs of the optimized and original models. These new pairs are used to train the preceding (sparse structure) stage, effectively propagating preference signals across the non-differentiable gap. Experiments show the method significantly improves the geometry, texture, and overall alignment with human preferences compared to the baseline.

However, our method still has limitation. Its performance on quantitative metrics has not yet surpassed previous score distillation-based methods, but this is compensated by its training time and efficiency.For future work, we will also apply this feedback loop mechanism to optimize 3D generation tasks based on score distillation, in order to explore the generalizability of our method.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16123–16133, 2022.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui 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*, pp. 22246–22256, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *arXiv preprint arXiv:2305.16381*, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, volume 33, pp. 6840–6851, 2020.

Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Yanqin Jiang, Chaohui Yu, Chenjie Cao, Fan Wang, Weiming Hu, and Jin Gao. Animate3d: Animating any 3d model with multi-view video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Rui Jiao, Xiangzhe Kong, Wenbing Huang, and Yang Liu. 3d structure prediction of atomic systems with flow-based direct preference optimization. *arXiv preprint arXiv:2405.16378*, 2024.

Shyamgopal Karthik, Huseyin Coskun, Zeynep Akata, Sergey Tulyakov, Jian Ren, and Anil Kag. Scalable ranked preference optimization for text-to-image generation. *arXiv preprint arXiv:2410.18013*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

Yushi Lan, Fangzhou Hong, Shuai Yang, Shangchen Zhou, Xuyi Meng, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation. *arXiv preprint arXiv:2404.18523*, 2024.

Chengyuan Li, Tianyu Zhang, Xusheng Du, Ye Zhang, and Haoran Xie. Generative ai for architectural design: A literature review. *arXiv preprint*, 2024a.

Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024b.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 300–309, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2837–2845, 2021.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pp. 405–421, 2020.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

Ivan Skorokhodov, Aliaksandr Siarohin, Yilun Du, Konstantinos G. Derpanis, Gordon Wetzstein, Sanja Fidler, and Sergey Tulyakov. ImageNet-3D: 3D-Aware generation using 2D diffusion models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265, 2015.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *Advances in Neural Information Processing Systems*, 33, 2020.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4563–4573, 2023a.

Zhecheng Wang, Jiaju Ma, Eitan Grinspun, Bryan Wang, and Tovi Grossman. Script2screen: Supporting dialogue scriptwriting with interactive audiovisual generation. *arXiv preprint arXiv:2504.14776*, 2025.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023b.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pp. 82–90, 2016.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. *arXiv preprint arXiv:2407.07880*, 2024.

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. *arXiv preprint arXiv:2412.01506*, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *arXiv preprint arXiv:2304.05977*, 2023.

JunLiang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. Dreamreward: Text-to-3d generation with human preference. *arXiv preprint arXiv:2403.14613*, 2024.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhenglin Zhou, Xiaobo Xia, Fan Ma, Hehe Fan, Yi Yang, and Tat-Seng Chua. Dreamdpo: Aligning text-to-3d generation with human preferences via direct preference optimization. *arXiv preprint arXiv:2502.04370*, 2025.

A The Use of Large Language Models

We conducted two tasks using large language models (LLMs). First, as mentioned in the main text, we leveraged an LLM to generate abundant text prompts serving as the test dataset. Specifically, the LLM first processed all prompt data in our training set, and then generated a set of prompts based on this learning process to form the test dataset. Second, we utilized the same LLM to translate, polish, and refine the textual content throughout the entire manuscript. The LLM employed in this study is Gemini 2.5 Pro.

B Reproducibility statement

The training process has been clearly and detailedly described in the main text, and the data processing process is elaborated in Appendix C.3.

C Additional Implementation

C.1 Training details

C.1.1 Training Implementation Details

Our Circular-DPO takes approximately 24 hours to train on 1000 samples. After training, the model can quickly generate 3D assets within tens of seconds based on given conditions. The generated 3D assets will display finer texture details, more realistic geometric structures, and content that is more consistent with text descriptions.

C.1.2 Pseudocode

The Circular-DPO algorithm addresses preference alignment in multi-stage 3D generative models through a systematic three-stage approach that overcomes the fundamental challenge of non-differentiable operations between generation stages.

Algorithm 1 Circular-DPO: Aligning Multi-Stage 3D Generative Models

Require: Preference dataset $D_{slat} = \{(c, x^w_{slat}, x^l_{slat})\}$
Require: Policy models θ^{slat}_{opt}, θ^{sparse}_{opt}
Require: Reference models θ^{slat}_{ref}, θ^{sparse}_{ref}
Require: Encoder E
Require: Quality weight function w_1, Preference weight function w_2
Step 1: Optimizing the Final Stage
for each batch in D_{slat} **do**
  Compute $\mathcal{L}^{slat}_{DPO}$ using preference pairs
  Update θ^{slat}_{opt} using gradient descent
end for
Step 2: Constructing New Preference Pairs
for each condition c in dataset **do**
  Sample x^{op}_{slat} from optimized model θ^{slat}_{opt}
  Sample x^{un-op}_{slat} from reference model θ^{slat}_{ref}
  Decode: $x^{op}_{sampled} = \text{Decode}(x^{op}_{slat})$, $x^{un-op}_{sampled} = \text{Decode}(x^{un-op}_{slat})$
  Encode: $x^{op}_{sparse} = E(x^{op}_{sampled})$, $x^{un-op}_{sparse} = E(x^{un-op}_{sampled})$
  Compute quality weights: $w_{1,(op,un-op)} = G_{op} - G_{un-op}$
  Compute preference weights: $w_2(c, x^{op}, x^{un-op}) = \frac{\exp(h)}{\mathbb{E}[\exp(h)]}$
  if $w_1 > \tau$ **then**
    Add $(c, x^{op}_{sparse}, x^{un-op}_{sparse})$ to D_{sparse} with weights w_2
  else
    Skip this preference pair
  end if
end for
Step 3: Guiding the Preceding Stage
for each weighted batch in D_{sparse} **do**
  Compute weighted $\mathcal{L}^{sparse}_{DPO} = w_2 \times \mathcal{L}_{DPO}$ using new preference pairs
  Update θ^{sparse}_{opt} using gradient descent
end for
return Optimized models θ^{slat}_{opt}, θ^{sparse}_{opt}

The algorithm operates as follows:

(1) The algorithm first applies DPO training to the SLAT flow model using human-annotated preference data to optimize final asset generation quality.

(2) It then constructs new preference pairs by sampling optimized examples from the trained model and unoptimized examples from the reference model, applying quality-aware filtering to retain only high-quality pairs for subsequent training.

(3) Finally, the filtered preference pairs are used to train the sparse structure flow model with weighted DPO loss, enabling preference signals to propagate from the final output back to the initial generation stage. This circular feedback mechanism achieves joint optimization across the entire multi-stage pipeline while maintaining the architectural benefits of stage separation.

C.2 Training process

We show the change in training loss of Circular-DPO as the number of training steps increases. The blue curve represents the retraining data we conducted. For the preference data provided by the external dataset, our method starts to converge when approaching 3000 steps and stops training at around 5000 steps after the training stabilizes.

C.3 Data process

The generation process of the data structure required for training is as follows. First, obtain the original 3D assets and their corresponding text conditions c. Render the 3D assets to obtain images from multiple perspectives. Perform voxelization processing on the obtained 3D assets, and at the same time, use DINOv2 to extract the features of the 3D assets through the rendered images. Input the obtained voxelized data and features into the Sparse VAE Encoder, and obtain the sparse structure x_{sparse} of the 3D assets through encoding. Using the sparse structure x_{sparse} of the 3D assets, the extracted image features, and the corresponding text conditions c, x_{slat} is obtained through encoding by the encoder, which is the initial data for the first step of training, i.e., the process

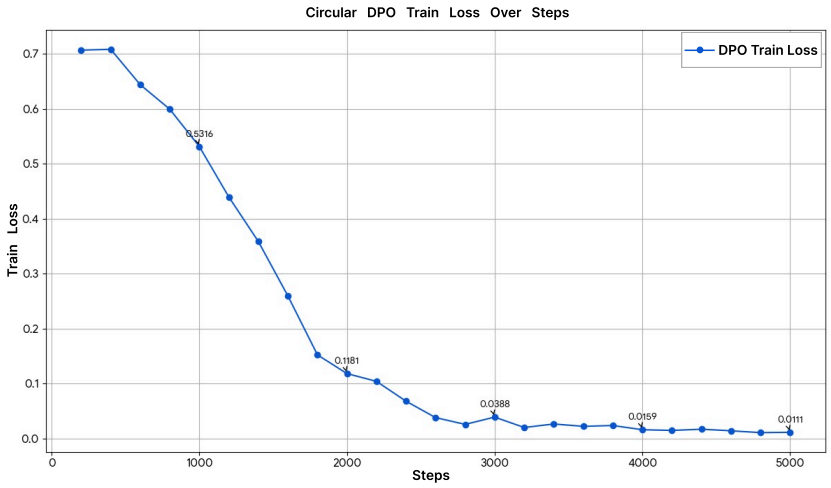

Figure 7: Training loss over steps.

from steps 1 to 5 in the Figure 8. After obtaining new sample pairs, use the trained sparse flow model to sample

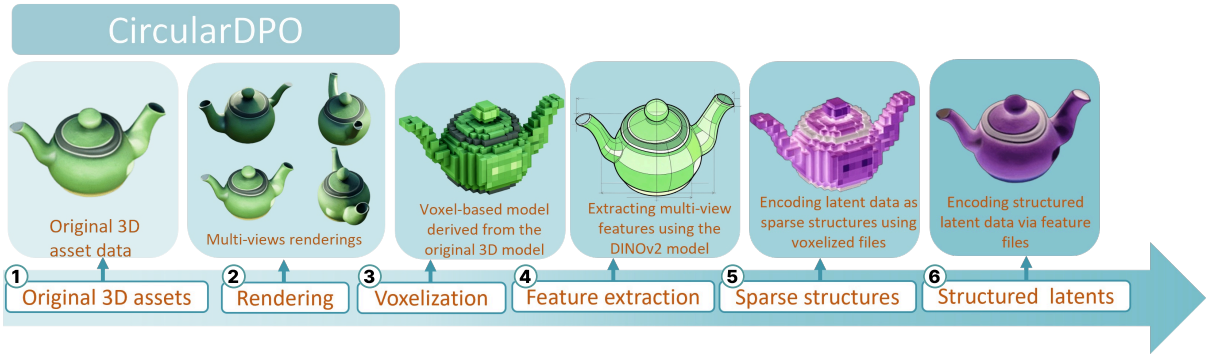

Figure 8: Training data processing process and intermediate process data.

and obtain $structure - slat$, which is step 6. Decode it to get new 3D asset preference pairs. Continue performing steps 1 to 5 on the new 3D asset preference pairs to obtain x_{sparse}, which will be used as the training data for the third step to directly optimize the preferences of the flow model.

The new preferences are sampled by the trained sparse flow model, conditioned on the x_{sparse} of the positive samples provided in the original dataset, the extracted image features, and the corresponding text conditions c. To unify the input data, the negative samples are sampled by the untrained sparse flow model. In the Figure 10, we show a comparison between the initial data and the newly sampled data. Through sampling, we can obtain the 3D assets generated after training. By pairing these generated 3D assets with the original 3D assets, we can create new preference pairs.

C.4 Rationale for the Preference Feedback Loop

When we apply Direct Preference Optimization (DPO) to the second stage (the slat model), which is primarily responsible for local details and texture, the optimization process not only refines the texture but also inevitably generates more detailed geometric information, as shown in the Figure 9. The geometric details of textures such as those of houses and bread have been improved. However, due to non-differentiable operations between the stages, this valuable geometric information learned in the later stage cannot be backpropagated via gradients to the first stage, which generates the sparse structure.

If the two stages were fully disentangled, applying DPO independently to each would suffice for the alignment process. In fact, this entanglement is an inherent consequence of the model's architecture: the output of the first stage is a compact latent representation compressed by an encoder (e.g., a VAE), which implies that the structural

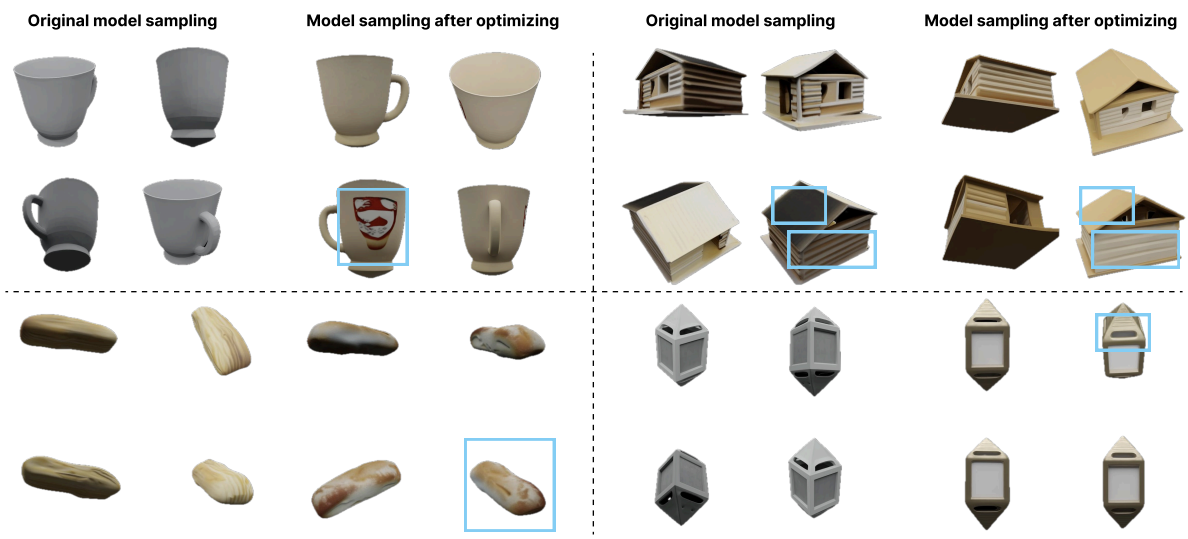

Figure 9: Differences in data before and after optimizing.

Figure 10: Differences in preference pairs before and after optimizing.

information of the original 3D asset is down-sampled. Consequently, to successfully decode the complete 3D model, the features in the second stage must contain and elaborate upon this specific structural information.

It is this unavoidable entanglement that underscores the necessity of our proposed Circular-DPO framework. The framework establishes a preference feedback loop, which, in essence, does not merely pass gradients. Instead, it propagates the preference signal—to fit the winning sample while diverging from the losing one—along with the associated optimized geometry and texture information. This is achieved by constructing new preference pairs, which effectively guide the preceding generation stage and lead to a global, consistent optimization.

D The Inefficiency of Front-to-Back Optimization

Optimizing the pipeline from back to front is significantly more efficient than a front-to-back approach. The core inefficiency of a front-to-back strategy lies in the data selection process required for constructing preference pairs.

Specifically, a front-to-back method would first require fine-tuning the initial-stage (sparse) model. To create preference pairs for the second stage, one would then need to generate full 3D assets by sampling from both the newly optimized initial-stage model and the original, unoptimized one. As illustrated in Figure 2, this process involves generating a large number of complete 3D assets simply for the purpose of ranking and selection.

In contrast, our back-to-front approach bypasses this redundant generation step. By first optimizing the final stage, we can construct preference pairs for the initial stage without needing to generate assets from an un-trained or intermediate model, thereby substantially improving training efficiency.

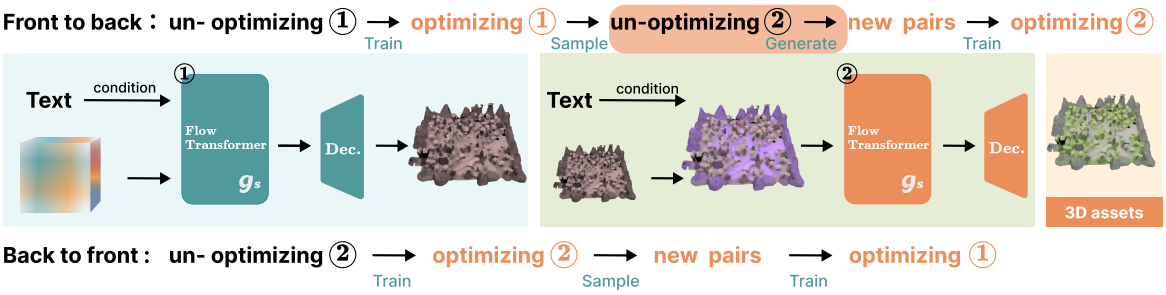

Figure 11: Front-to-back VS Back-to-front.

E HUMAN PREFERENCE DATASET 3DREWARDDB

E.1 SHOW SEVERAL EXAMPLES IN THE 3DREWARDDB

The new publicly available data from 3DrewardDB involves re-annotating the large-scale 3D dataset Objverse, and it releases 1000 candidate prompts, each accompanied by 4-10 sampled 3D assets.In the Figure 12, we show several examples of human preference rankings.

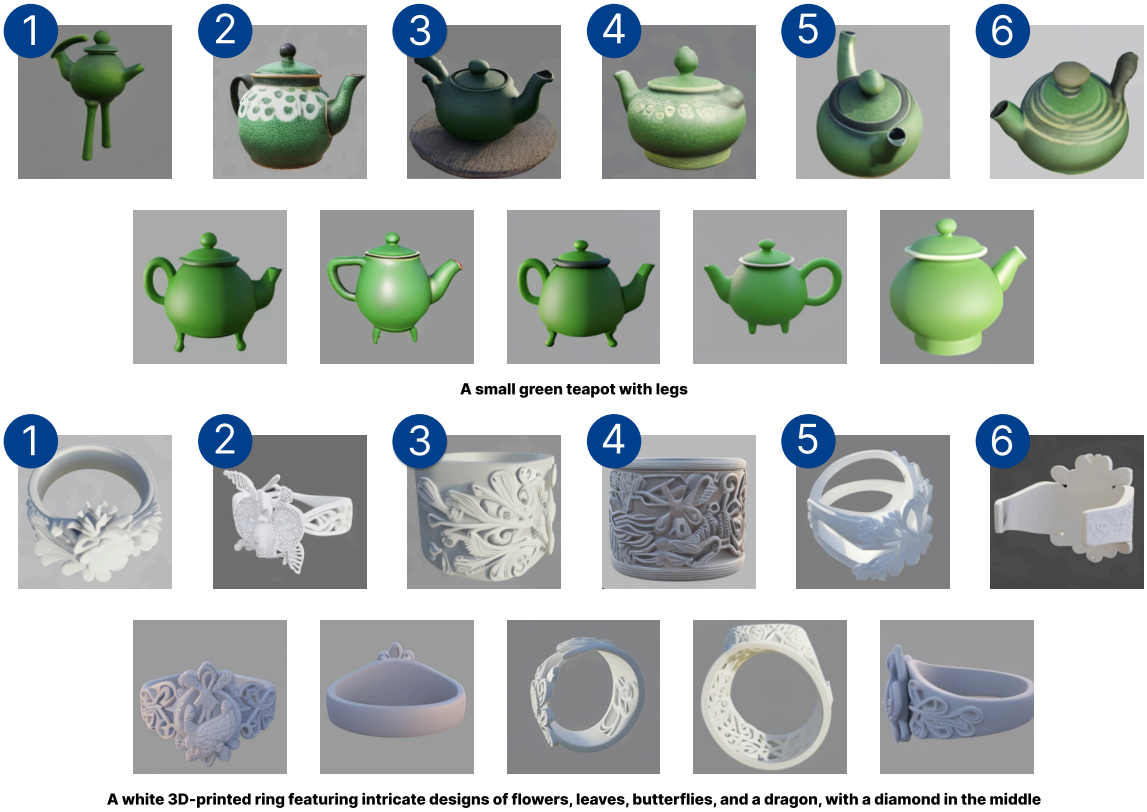

Figure 12: Examples of 3DrewardDB.

E.2 THE FITTING SITUATION OF PREFERENCES AFTER TRAINING

After training, our generation will better fit the 3D models in human preferences in terms of texture and geometric structure. For example, in the first example, the highest-ranked 3D pattern is a long chair back. Under the condition of the same seed, we can generate geometric shapes that are more in line with preferences. Other dimensions such as texture can also achieve this effect.

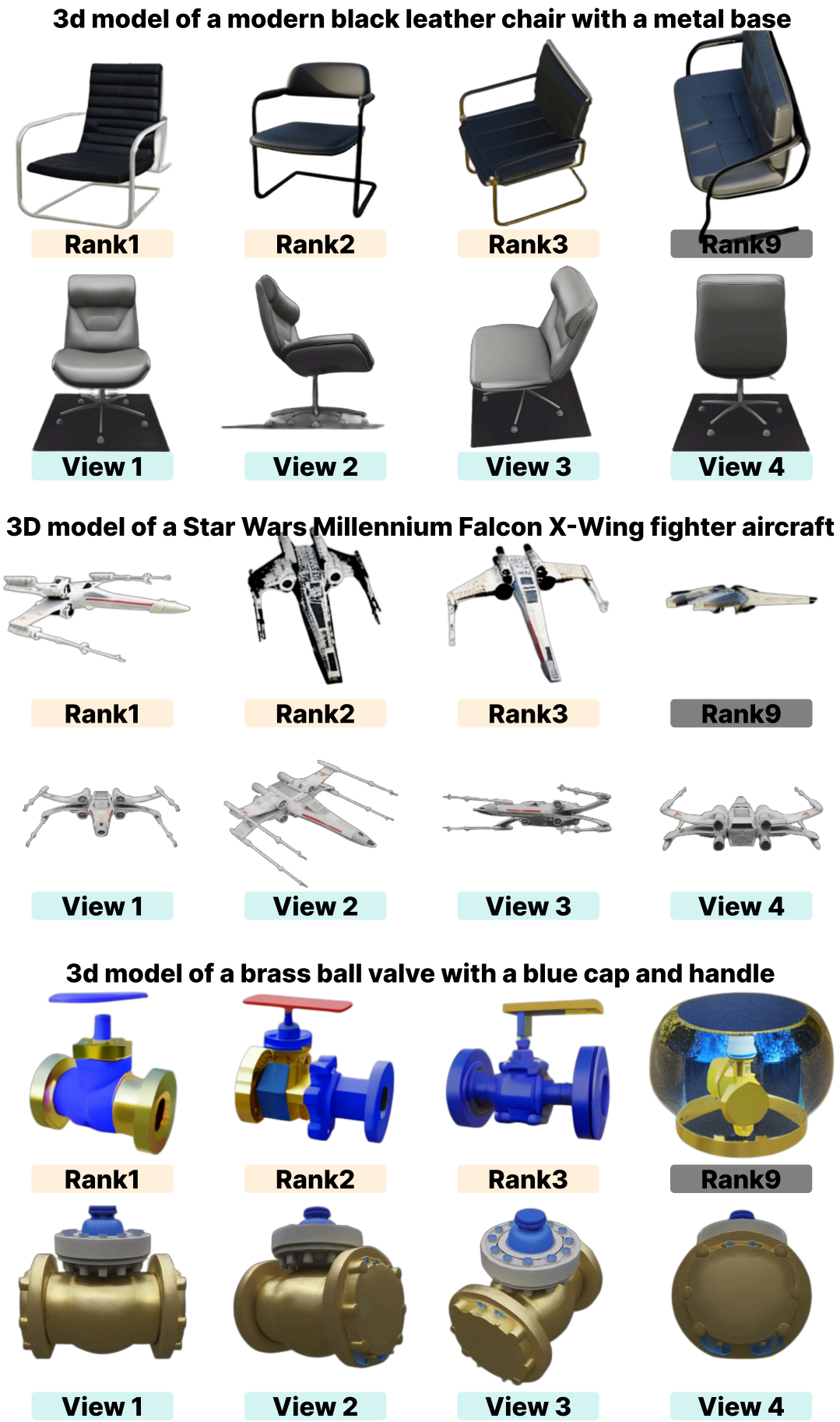

Figure 13: The fitting situation of preferences.

F MORE VISUAL RESULTS

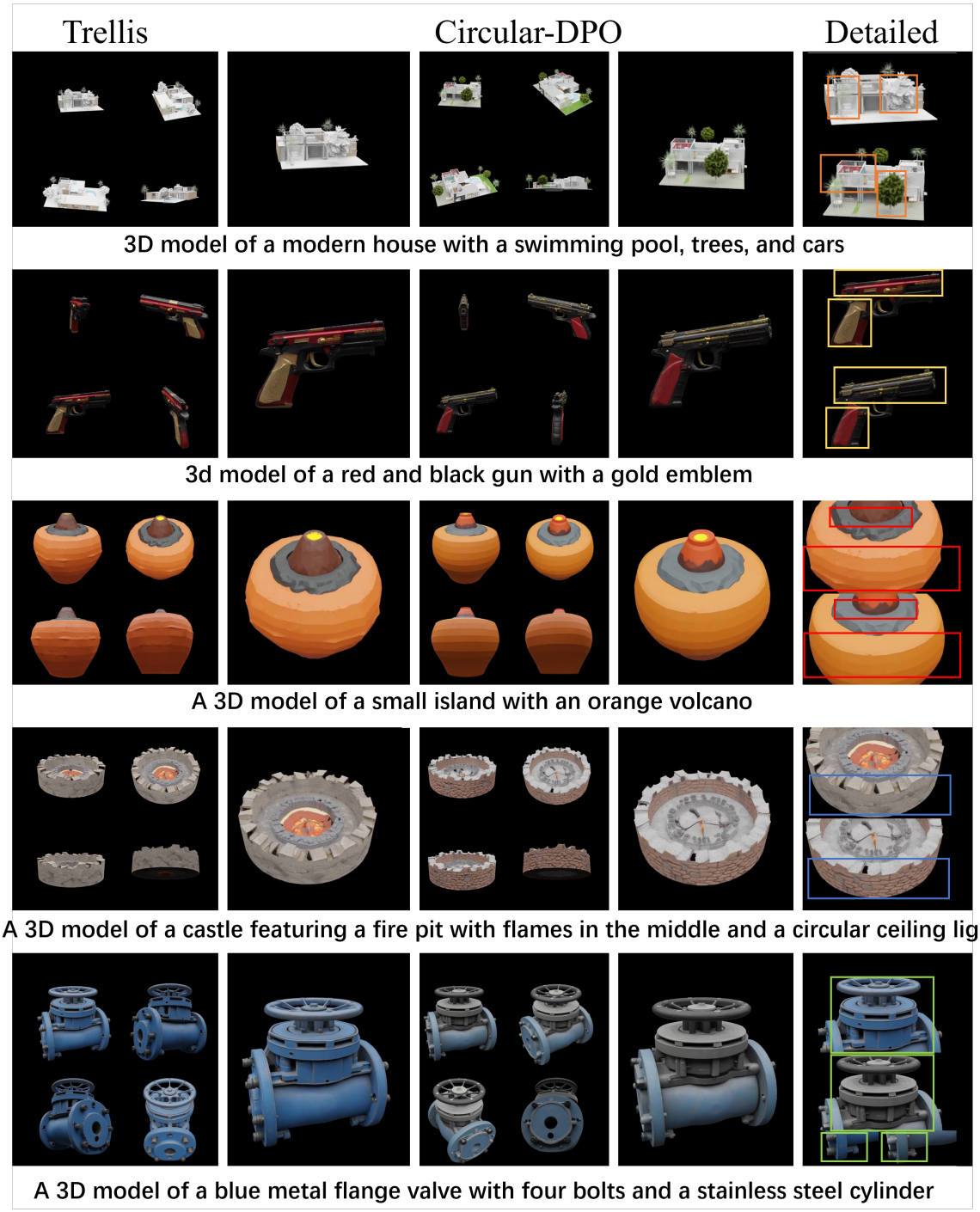

Figure 14: More Visual Results.