FLOW MATCHING REINFORCEMENT FOR 3D MESH GENERATION VIA DYNAMIC HOMING OPTIMIZATION

Anonymous authors
Paper under double-blind review



Figure 1: We generate diverse 3D assets with plausible global shapes and detailed geometry.

ABSTRACT

Flow matching is central to 3D generation, yet in practice its reinforcement learning (RL) methods are largely adapted from 2D visual generation. Representative DPO-, GRPO-, and NFT-style objectives, when applied to negative trajectories, mainly steer predicted velocities away from the corresponding directions without explicitly specifying a target velocity field toward preferred samples. In 3D generation, constrained by pretrained model capabilities, rollout diversity, and reward-distribution complexity, directly applying these RL methods yields limited gains in geometric quality. We introduce a forward-process RL method **Dynamic Homing Optimization (DHO)**, which reformulates negative-trajectory optimization as positive-sample attraction-guided dynamic homing. Specifically, Minimum-Cost Attractive Matching (MAM) assigns each negative sample a distinct positive target, and Time-Aware Dynamic Correction (TDC) then redirects its trajectory toward the target using a remaining-time-aware corrective velocity. Building on asynchronous online DHO, we develop **Flow3D-Pro**, an image-to-3D geometry generation framework. Experiments show that DHO outperforms representative DPO-, GRPO-, and NFT-style objectives in 3D generation, while Flow3D-Pro produces higher-quality 3D geometry than existing mesh generation methods.

1 INTRODUCTION

3D generation has become a cornerstone of modern content creation, powering high-fidelity asset synthesis Zhang et al. (2024); Lai et al. (2025a); Liu et al. (2026); Zhou et al. (2026) for gaming, and increasingly underpinning embodied intelligence. At the heart of this progress lie continuous-time generative models—particularly flow matching and its rectified flow instantiation Lipman et al. (2023); Liu et al. (2023), which learn velocity fields from linear conditional paths between paired prior and data samples and now serve as the backbone of recent 3D generators Xiang et al. (2026); Lai et al. (2025b); Li et al. (2026). Yet 3D generation advances have concentrated mostly on pre-training, while reinforcement learning (RL) post-training remains comparatively underexplored in

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

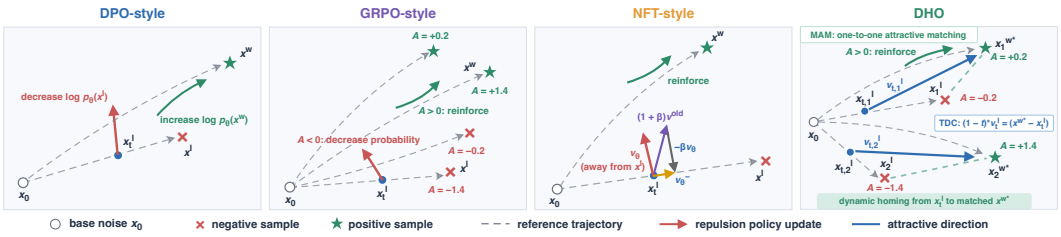

Figure 2: Negative-sample branch objectives across flow matching RL methods.

the flow-based setting. For want of dedicated formulations, current practice Ning et al. (2026) transplants RL algorithms developed for 2D visual generation, chiefly DPO-style Wu et al. (2025a) (e.g., Flow-DPO Liu et al. (2025b)), GRPO-style Li et al. (2025a); Xue et al. (2025) (e.g., Flow-GRPOLiu et al. (2025a)) and NFT-style Li et al. (2025d) (e.g., DiffusionNFT Zheng et al. (2026)).

In Figure 2, for the negative-sample branch objective, DPO-style methods implicitly suppress the likelihood of the negative trajectory, and GRPO-style methods reduce the probability of samples with negative advantages. NFT-style methods reverse the gradient through an implicit negative policy, mainly driving the predicted velocity away from the negative-trajectory target. Despite differing in mechanics, these paradigms share one trait in how they treat a negative sample: these objectives mainly repel the velocity directions that produce the corresponding trajectories, but do not explicitly specify a target velocity field that would instead generate preferred samples. They mainly rely on the positive-sample branch objective to guide the velocity field toward generating desirable trajectories. When applied to post-training for 3D generation, our empirical results find that, constrained by the capabilities of the pretrained foundation model, rollout diversity, and the complexity of the reward distribution, these RL methods yield limited gains in geometric quality.

Inspired by recent positive-sample attraction approaches Deng et al. (2026b), we propose **Dynamic Homing Optimization (DHO)**, which constructs negative-sample branch objectives with attraction-guided dynamic homing. DHO explicitly specifies both a preferred-sample destination for each negative trajectory and the corrective dynamics required to reach it. Specifically, we introduce Minimum-Cost Attractive Matching (MAM) to formulate preferred-target selection as a batchwise minimum-cost one-to-one assignment, pairing each negative sample with a distinct positive sample. Given the assigned target, Time-Aware Dynamic Correction (TDC) constructs a remaining-time-aware velocity that redirects each intermediate state toward its target.

DHO performs policy updates on forward-process interpolants constructed from rollout endpoints, is sampler-flexible, and does not require storing entire sampling trajectories. In addition, we develop **Flow3D-Pro**, an image-to-3D mesh generation framework, which leverages asynchronous online DHO to improve geometric fidelity and structural plausibility. Our main contributions are as follows:

- We present Dynamic Homing Optimization (DHO), a forward-process RL method for flow matching that equips the negative-sample branch objectives with positive-sample attraction-guided dynamic homing, and introduce Flow3D-Pro for high-quality image-to-3D mesh generation with asynchronous online DHO.
- In DHO, we introduce Minimum-Cost Attractive Matching (MAM) to assign each negative sample a distinct positive target through batchwise minimum-cost one-to-one matching, and Time-Aware Dynamic Correction (TDC) to construct a remaining-time-aware corrective velocity toward that preferred target.
- Experiments demonstrate that DHO outperforms representative DPO-, GRPO-, and NFT-style objectives in 3D generation, while Flow3D-Pro achieves higher geometry generation quality than current mesh generation methods.

2 Related Work

2.1 3D Mesh Generation

Continuous-time generative models underpin modern synthesis: diffusion models established high-fidelity stochastic generation Ho et al. (2020); Song et al. (2021); Karras et al. (2022), while flow

matching Lipman et al. (2023) and its rectified flow instantiation Liu et al. (2023) learn velocity fields for transport, with related stochastic-interpolant and fast-sampling variants further improving efficiency Albergo et al. (2025); Song et al. (2023); Liu et al. (2024); Yan et al. (2024). In 3D generation, DreamFusion Poole et al. (2023) and Magic3D Lin et al. (2023) popularized 3D mesh optimization via 2D diffusion distillation, followed by higher-quality variants Wang et al. (2023); Chen et al. (2023). Native 3D generators have progressed from point or implicit representations Nichol et al. (2022); Jun & Nichol (2023) to SDF and shape-latent models Cheng et al. (2023); Zhang et al. (2023); Zhao et al. (2023), as well as textured-mesh and controllable asset pipelines Gao et al. (2022); Zhang et al. (2024); Li et al. (2025b). Recent rectified-flow 3D systems—including TRELLIS Xiang et al. (2025; 2026), Hunyuan3D Zhao et al. (2025); Yang et al. (2025); Lai et al. (2025a), TripoSG Li et al. (2025c), LATTICE Lai et al. (2025b) and other mesh flow generators Sun et al. (2026b); Li et al. (2026)—mainly study pre-training, representation, or sampling, whereas DHO studies RL post-training for 3D rectified-flow geometry generators.

2.2 Reinforcement Learning for Flow Matching

RL was adopted early for language-model alignment Guo et al. (2025); Rafailov et al. (2023), and later extended to visual generation using human feedback Kirstain et al. (2023) and reward-guided fine-tuning algorithms Xu et al. (2023); Lee et al. (2023). Diffusion reinforcement spans policy-gradient tuning (DDPO Black et al. (2024)), direct preference optimization (Diffusion-DPO Wallace et al. (2024)), and preference-tuned visual generation variants Fan et al. (2023); Clark et al. (2024). For flow matching, representative objectives: GRPO-style methods Liu et al. (2025a); Xu et al. (2026), represented by Flow-GRPO Liu et al. (2025a), treat generation as online RL and penalize low-reward rollouts, with related variants refining stability, reward density or policy granularity Xue et al. (2025); Deng et al. (2026a); Luo et al. (2026); Li et al. (2025a). DPO-style adaptations suppress negative trajectories through pairwise preference ratios Liu et al. (2025b); Wu et al. (2025a). NFT-style methods Li et al. (2025d), represented by DiffusionNFT Zheng et al. (2026), inject negative velocity-space gradients on the forward process. In 3D generation, flow-based RL methods (e.g., Flow3D-OPD Ning et al. (2026)) are primarily adapted from these 2D visual generation algorithms. However, these objectives mainly push the velocity away from directions that produce negative trajectories, without explicitly specifying a target velocity field that guides generation toward preferred samples, limiting gains in geometric quality. DHO addresses this by dynamically homing negative samples toward preferred regions. PFM Kim et al. (2024) trains an additional preference-flow model on rejected-to-preferred interpolants, and applies it to outputs from the frozen base model at inference, potentially introducing a source-distribution mismatch. DHO directly updates the original generative base field on noise-to-negative interpolants using matched positive endpoints.

3 Methodology

3.1 Rectified Flow as Policy Optimization

We view rectified flow as a continuous-time policy $v_\theta(x_t, t, c)$ that predicts a velocity (action) given state x_t, time $t \in [0, 1]$, and condition c. Starting from a Gaussian prior $x_0 \sim p_0 = \mathcal{N}(0, \mathbf{I})$, generation follows the velocity field v_θ and returns the terminal state $x_1 \sim p_1$ (target data distribution). The policy is trained with the flow-matching objective on the linear path $x_t = (1-t)\,x_0 + t\,x_1$,

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\, x_0\sim p_0,\, x_1\sim p_1}\left[\left\|v_\theta(x_t, t, c) - (x_1 - x_0)\right\|_2^2\right]. \tag{1}$$

In preference optimization, each condition c is associated with positive samples $\mathcal{P}$ and negative samples $\mathcal{N}$, and policy optimization reshapes v_θ so that trajectories terminate on $\mathcal{P}$ rather than $\mathcal{N}$.

3.2 Dynamic Homing Optimization

Let $x^w \in \mathcal{P}$ and $x^l \in \mathcal{N}$ denote a positive sample and a negative sample, and let $x_t^l = (1-t)x_0 + t\,x^l$ be an intermediate state along the negative trajectory. We propose **Dynamic Homing Optimization (DHO)**, an attraction-guided navigation paradigm (as illustrated in Figure 2 and Figure 3) with two main components: Minimum-Cost Attractive Matching (MAM), which assigns each negative sample a distinct attractive target within the batch, and Time-Aware Dynamic Correction (TDC),

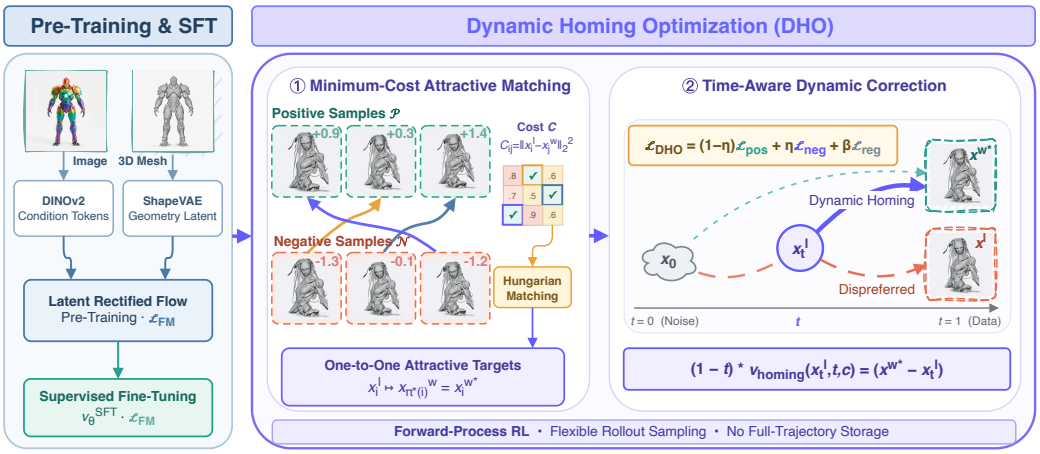

Figure 3: **Method Overview.** After pre-training and supervised fine-tuning (SFT), DHO post-trains the rectified flow model in latent feature space using reward-ranked rollout samples. In DHO, MAM assigns each negative sample an attractive positive-target destination, and TDC then constructs a time-aware homing velocity toward the assigned target. We visualize only the negative-sample branch, as the positive-sample branch uses the same MAM rule and homing-velocity construction.

which constructs a remaining-time-aware velocity toward that target. Although our analysis focuses primarily on negative-sample branch objectives, the same target-assignment and velocity-target construction also apply to positive-sample branch objectives.

3.2.1 Minimum-Cost Attractive Matching

We formulate attractive target selection as a minimum-cost one-to-one assignment problem within each mini-batch. In latent feature space, given B negative states $\hat{\mathcal{N}} = \{x_i^l\}_{i=1}^B$ and B positive states $\hat{\mathcal{P}} = \{x_j^w\}_{j=1}^B$, we form the cost matrix $\mathbf{C} \in \mathbb{R}^{B\times B}$,

$$C_{i,j} = \mathcal{D}(x_i^l, x_j^w) = \left\|x_i^l - x_j^w\right\|_2^2, \tag{2}$$

and seek the permutation $\boldsymbol{\pi}^*$ that minimizes the total assignment cost:

$$\begin{aligned} \boldsymbol{\pi}^* &= \arg\min_{\boldsymbol{\pi}\in\Pi_B} \sum_{i=1}^{B}\sum_{j=1}^{B} C_{i,j}\,\pi_{i,j}, \\ \Pi_B &= \left\{\boldsymbol{\pi} \in \{0,1\}^{B\times B} : \textstyle\sum_i \pi_{i,j} = \sum_j \pi_{i,j} = 1\right\}. \end{aligned} \tag{3}$$

Here Π_B denotes the set of $B \times B$ permutation matrices. This assignment can be viewed as mini-batch optimal transport between the uniform empirical distributions of negative and positive rollout endpoints Tong et al. (2024). Eq. (3) is a linear sum assignment problem solved in $\mathcal{O}(B^3)$ by the Hungarian algorithm. For a selected optimal permutation, $\pi^*(i)$ denotes the unique column index j with $\pi_{i,j}^* = 1$. The assigned attractive target of a negative sample is

$$x_i^{w*} = x_{\pi^*(i)}^w. \tag{4}$$

Moreover, the same assignment principle extends to the positive-sample branch objective: applying Eq. (3) between $\hat{\mathcal{P}}$ and itself admits the identity permutation as an optimal assignment because the diagonal has zero cost, $C_{i,i} = \|x_i^w - x_i^w\|_2^2 = 0$. Choosing this assignment matches every positive sample to itself, $x_i^{w*} = x_i^w$, so both branches are governed by the single assignment rule of Eq. (4).

Given a rollout group of G samples with rewards $\{r_i\}_{i=1}^G$, we compute the group-normalized advantage $A_i = (r_i - \bar{r})/s_r$, where $\bar{r}$ and s_r are the group mean and standard deviation. Samples with $A_i > 0$ are designated as positive, whereas those with $A_i < 0$ are designated as negative. Zero-advantage samples are omitted. To construct equally sized sets for MAM, let B be the smaller set size. If positive samples are more numerous, we retain the B samples with the highest advantages. If negative samples are more numerous, we randomly select B of them. For each retained positive sample, we define the normalized advantage weight $\omega_i^w = A_i^w / (\frac{1}{B}\sum_{j=1}^B A_j^w)$. After MAM, a negative sample inherits the weight of its assigned positive target, $\omega_i^{w*} = \omega_{\pi^*(i)}^w$.

3.2.2 TIME-AWARE DYNAMIC CORRECTION

Given the assigned target x_i^{w*}, a naive choice is the static parallel action $x^{w*} - x_0$. This is incorrect for a negative trajectory: the state x_t^l already lies off the straight path to x^{w*}, so integrating a constant velocity accumulates a terminal error proportional to $t\,(x^l - x^{w*})$. To account for this residual displacement, the action must depend on both the current state and the remaining time.

We therefore define a time-aware corrective field that points from the current state to the target and rescales by the remaining horizon $1 - t$, so that the residual displacement is covered at $t = 1$:

$$v_{\text{target}}^{\text{correction}}(x_t^l, t, c) = \frac{x^{w*} - x_t^l}{1 - t}. \tag{5}$$

Integrating Eq. (5) directs the state to the endpoint x^{w*}.

The factor $1/(1 - t)$ in Eq. (5) diverges as $t \to 1$. To obtain a numerically stable regression target for the negative branch, we adopt the time-dependent weight $w_{\text{correction}}(t) = (1 - t)^2$, which cancels the singularity: weighting the squared residual of $(v_\theta - v_{\text{target}}^{\text{correction}})$ by $w_{\text{correction}}(t)$ is equivalent to matching the rescaled velocity $(1 - t)\, v_\theta$ against a bounded target. Substituting Eq. (5) and $x_t^l = t\, x^l + (1 - t)x_0$ gives

$$(1 - t)\, v_{\text{target}}^{\text{correction}}(x_t^l, t, c) = x^{w*} - x_t^l = (x^{w*} - x_0) - t\,(x^l - x_0), \tag{6}$$

which is smooth on all of $[0, 1]$. Weighting the correction by the normalized advantage of its assigned positive target yields the policy optimization objective for the negative-sample branch:

$$\mathcal{L}_{\text{neg}}(\theta) = \mathbb{E}_{t, x_0, x^l}\left[\omega^{w*} \left\| (1 - t)\, v_\theta(x_t^l, t, c) - \big((x^{w*} - x_0) - t\,(x^l - x_0)\big) \right\|_2^2\right]. \tag{7}$$

Eq. (7) gives stronger correction weights toward higher-advantage positive targets while keeping the signal well-conditioned across the whole time horizon. Since $\omega^{w*} > 0$ only rescales the regression loss, it preserves the homing target of Eq. (6).

The positive-sample branch shares the MAM rule and homing-velocity construction with the negative-sample branch. Self-matching the positive set returns the identity, so a positive sample is assigned to itself, $x^{w*} = x^w$. Substituting this target into the time-aware field of Eq. (5) and using $x_t^w = (1 - t)x_0 + t\, x^w$ collapses the correction to the standard rectified-flow action:

$$v_{\text{target}}^{\text{correction}}(x_t^w, t, c) = \frac{x^w - x_t^w}{1 - t} = \frac{(1 - t)\,(x^w - x_0)}{1 - t} = x^w - x_0. \tag{8}$$

Since the self-matched velocity target has no endpoint singularity, we retain the rectified-flow action with positive-advantage weighting and no additional temporal weighting:

$$\mathcal{L}_{\text{pos}}(\theta) = \mathbb{E}_{t, x_0, x^w}\left[\omega^w \left\| v_\theta(x_t^w, t, c) - (x^w - x_0) \right\|_2^2\right]. \tag{9}$$

3.2.3 FULL OPTIMIZATION OBJECTIVE

To constrain policy deviation during post-training, we regularize the learned velocity field toward a frozen reference policy $v_{\theta_{\text{ref}}}$. Specifically, we use the squared velocity deviation as a likelihood-free reference-policy regularizer:

$$\mathcal{L}_{\text{reg}}(\theta) = \mathbb{E}_{t,\, x_0,\, x \in \hat{\mathcal{P}} \cup \hat{\mathcal{N}}}\left[\left\| v_\theta(x_t, t, c) - v_{\theta_{\text{ref}}}(x_t, t, c) \right\|_2^2\right]. \tag{10}$$

Combining Eqs. (7), (9), and (10), the full DHO objective jointly anchors positive trajectories, dynamically homes negative ones, and regularizes toward the reference policy:

$$\mathcal{L}_{\text{DHO}}(\theta) = \eta\, \mathcal{L}_{\text{neg}}(\theta) + (1 - \eta)\, \mathcal{L}_{\text{pos}}(\theta) + \beta\, \mathcal{L}_{\text{reg}}(\theta), \tag{11}$$

where $\eta \in (0, 1)$, $\beta > 0$ control the trade off between homing strength and reference regularization.

3.2.4 ENDPOINT BEHAVIOR ANALYSIS

DHO explicitly defines a homing field toward a positive target. We analyze the endpoint behavior of DHO's homing field from two perspectives: the converged endpoint and the training dynamics. Since optimizing x^w is standard flow matching, we mainly focus on the negative branch x^l.

Proposition 1. *Let $x_0 \sim p_0$ be a base noise sample that deterministically maps to a negative sample x^l under the pre-trained reference policy $v_{\theta_{ref}}$. As the policy v_θ converges to the optimal dynamic homing field $v^*(x_t, t, c) = \frac{x^{w*} - x_t}{1-t}$ via training, the generative mapping $F_\theta : x_0 \mapsto x_1$ shifts its terminal endpoint such that $F_{\theta^*}(x_0) = x^{w*}$.*

This characterizes the converged endpoint: as the policy converges to the optimal homing field, the deterministic map F_θ is redirected so that the specific prior sample x_0 that previously generated x^l is routed to its assigned positive mode x^{w*}.

Proposition 2. *For a fixed x^l and its assigned x^{w*}, assume $v_\theta^{init}(x, t) = \frac{x^l - x}{1-t}$ for $t \in [0, 1)$. Let the selected policy snapshots satisfy the idealized velocity update $v_\theta^{\tau+1}(x, t) = v_\theta^\tau(x, t) + \alpha\big(\frac{x^{w*} - x}{1-t} - v_\theta^\tau(x, t)\big)$ pointwise over (x, t), where $\tau \geq 0$ indexes successive policy snapshots rather than necessarily every raw SGD step and $\alpha \in (0, 1)$ is an effective contraction factor between consecutive snapshots. Then the terminal generation is $x_1(\tau) = \big(1 - (1-\alpha)^\tau\big)x^{w*} + (1-\alpha)^\tau x^l$, and hence $\lim_{\tau\to\infty} x_1(\tau) = x^{w*}$.*

This is an idealized field-space analysis of how the terminal endpoint of a particular negative generation evolves when selected policy snapshots contract pointwise toward its homing field. $x_1(\tau)$ interpolates between x^l and x^{w*} with weight $(1-\alpha)^\tau$, closing the residual gap at a geometric rate. Figure 4 illustrates this behavior qualitatively.

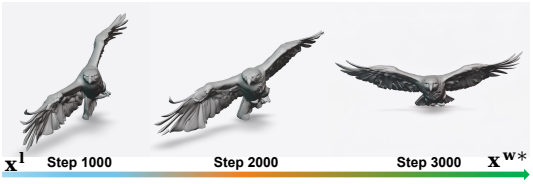

Figure 4: The generated owl gradually becomes canonically aligned during DHO training.

Moreover, MAM is recomputed across mini-batches and may reassign the positive target. These propositions therefore apply conditionally to a fixed assignment and piecewise to intervals with different assignments: each interval supplies a constructive correction toward its current positive target. If the assignments eventually stabilize at x^{w*}, the stated recurrence predicts convergence to that target. As training progresses, the velocity field within each interval moves toward a more positive direction. Detailed proofs for these propositions are in the appendix.

3.2.5 Classifier-Free Guidance for DHO

DHO is compatible with Classifier-Free Guidance (CFG), whose guidance scale balances exploration diversity and generation quality. During pre-training and SFT, the conditioning embedding $\mathbf{c}$ is randomly replaced with a learnable null embedding $\mathbf{c}_\varnothing$ with a fixed probability, enabling the model to learn both conditional and unconditional velocity fields. Given a guidance scale $s \geq 1$ ($s = 1$ corresponding to the CFG-free setting), the guided velocity is

$$\hat{v}_\theta(x_t, t, \mathbf{c}) = v_\theta(x_t, t, \mathbf{c}_\varnothing) + s\big(v_\theta(x_t, t, \mathbf{c}) - v_\theta(x_t, t, \mathbf{c}_\varnothing)\big). \tag{12}$$

We consistently apply Eq. (12) to rollout generation, policy updates, and inference using the same guidance scale s. This consistency ensures that policy optimization is performed under the same guided dynamics used for exploration and exploitation. The DHO optimization objective remains unchanged except that v_θ is replaced by $\hat{v}_\theta$.

3.3 Flow3D-Pro

Building on DHO, we develop **Flow3D-Pro**, a two-stage framework for image-to-3D mesh generation. Directly recovering detailed 3D geometry from a 2D reference image remains challenging. Following prevailing multi-stage paradigms Lai et al. (2025b); Xiang et al. (2026); Sun et al. (2026a), Flow3D-Pro decomposes generation into a lifting stage (Stage I) and a refinement stage (Stage II). Stage I lifts the 2D reference image into 3D space to establish the global geometry and spatial arrangement of the mesh, whereas Stage II refines the resulting mesh with finer geometric details. The lifting stage is built upon Hunyuan3D 2.1 Yang et al. (2025), while the refinement stage is derived from LATTICE Lai et al. (2025b).

Stage I establishes the global geometry and structural initialization—including the overall shape, pose, and spatial arrangement—that constrain subsequent refinement. We therefore focus post-

training on Stage I, whose high output diversity during 2D-to-3D lifting provides a broader exploration space for RL to discover improved policies. Specifically, Stage I is initialized from pretrained weights and then optimized in two sequential post-training phases: SFT, followed by DHO. Stage II remains frozen at its pretrained weights and is used only at inference time to refine the geometry produced by Stage I, primarily by enhancing local surface details.

3.3.1 Asynchronous Online DHO

Flow3D-Pro rolls out G meshes conditioned on each reference image. For each mesh, we render six views and provide them together with the reference image to Gemini 3.1 Pro, which serves as the reward model and assigns a scalar reward based on generation quality. Gemini evaluates each mesh along three dimensions: reconstruction fidelity, geometric plausibility, and orientation alignment. Reconstruction fidelity assesses the consistency of silhouette proportions and surface appearance with the reference image. Geometric plausibility considers the validity of the overall shape and its consistency across views. Orientation alignment evaluates whether the generated mesh is aligned with the coordinate axes, which helps the learning of symmetric structures. By jointly considering these three dimensions, Gemini assigns each mesh a reward score from 1 to 5 (higher is better).

In practice, reward evaluation by Gemini introduces substantial and highly variable latency across meshes. To improve training efficiency, we adopt an asynchronous online DHO framework following Zhou et al. (2026), decoupling mesh rollout and reward evaluation from policy optimization. We allocate N GPUs to rollout and M GPUs to policy updates. Each rollout worker holds a fixed snapshot of the latest released policy weight while generating a complete mesh group. Once Gemini evaluation is complete, the scored group is inserted into a rollout replay buffer. In parallel, the trainer workers continuously sample evaluated groups from the buffer and update the current policy using the DHO objective. The trainer periodically publishes a new policy snapshot, which rollout workers fetch before starting subsequent rollouts. Data generated by outdated policy versions are discarded to limit policy staleness.

We balance policy synchronization and replay-buffer utilization through an update-to-buffer ratio. Let K denote the number of policy-update steps between two consecutive policy releases, b the per-GPU training batch size, and C the buffer capacity measured in training samples. We constrain

$$\lambda_{\min} \leq \frac{KMb}{C} \leq \lambda_{\max}, \tag{13}$$

where KMb is the number of training samples consumed between policy releases. This constraint balances insufficient learning from newly collected rewards against excessive reuse of stale rollout data, allowing rollout, reward evaluation, and DHO policy updates to proceed concurrently.

4 Experiments

4.1 Dataset

We collect artist-created and AI-generated meshes and process them through several steps: (a) data filtering, (b) watertight conversion, (c) point-cloud sampling and signed distance field (SDF) extraction, and (d) manual curation. Human annotators select high-quality meshes by considering global shape quality and local geometric details, resulting in 5K meshes for SFT. For DHO, we collect 1K diverse reference images, covering 3D and 2D cartoon styles as well as photorealistic subjects. The collection spans humans, animals, vehicles, architecture, plants and other object categories.

We assess generalization on two evaluation sets disjoint from training. The first, ***GenMesh-Test***, is our curated set of 200 challenging reference images (comparable in scale to the evaluation set of Lai et al. (2025b)) covering diverse object categories, visual styles, and levels of geometric complexity. The second, ***LATTICE-Bench***, is the evaluation set adopted by LATTICE Lai et al. (2025b).

4.2 Implementation Details

DHO is trained for 5K optimization steps with AdamW at a learning rate of 2.5×10^{-6}. Training is performed in bfloat16 with a per-GPU batch size of 2. We set $\eta = 0.1$, $\beta = 1.0$, the rollout group size to $G = 8$, and the CFG scale to 3.0. For asynchronous optimization, we allocate $N = 28$ GPUs

Table 1: Comparison of RL objectives for Stage I, evaluated before and after the Stage II refiner.

Method	Stage I			Stage II		
	ULIP ↑	Uni3D ↑	User Study ↑	ULIP ↑	Uni3D ↑	User Study ↑
Base	0.143	0.347	0.60	0.143	0.350	0.71
SFT	0.144 (+0.70%)	0.351 (+1.15%)	1.93	0.143 (+0.00%)	0.352 (+0.57%)	1.91
GRPO-style	0.147 (+2.80%)	0.357 (+2.88%)	3.02	0.147 (+2.80%)	0.358 (+2.29%)	3.00
DPO-style	0.146 (+2.10%)	0.357 (+2.88%)	2.36	0.146 (+2.10%)	0.358 (+2.29%)	2.32
NFT-style	0.146 (+2.10%)	0.355 (+2.31%)	3.07	0.147 (+2.80%)	0.357 (+2.00%)	2.80
DHO	**0.149 (+4.20%)**	**0.359 (+3.46%)**	**4.02**	**0.150 (+4.90%)**	**0.360 (+2.86%)**	**4.26**

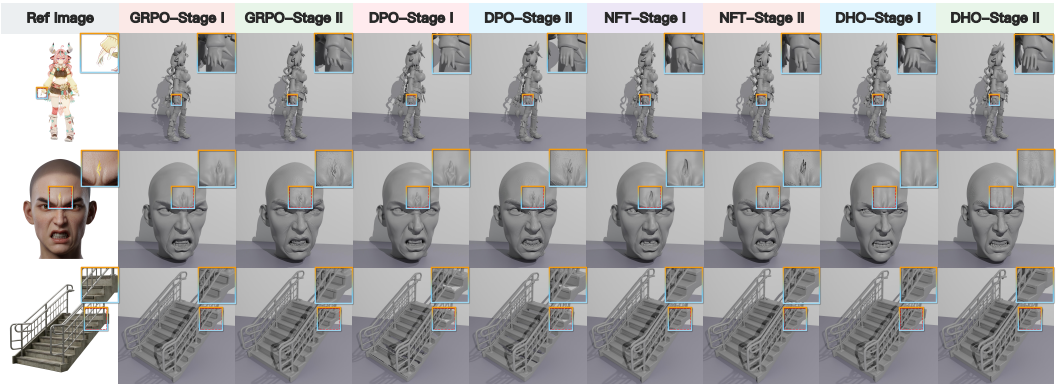

Figure 5: Qualitative comparison of RL objectives. To facilitate visualization, all generated meshes are rotated to a common orientation. The same convention is used in subsequent visualizations.

to rollout and $M = 4$ GPUs to policy updates, with $\lambda_{\min} = 4$, $\lambda_{\max} = 8$, and $K = 16$. Stage II is kept frozen during inference. Full training and inference configurations are in the appendix.

4.3 Evaluation Metrics

We evaluate the image-to-geometry generation through various metrics including ULIP Xue et al. (2023; 2024), Uni3D Zhou et al. (2024), following Lai et al. (2025b); Xiang et al. (2026), alongside a comprehensive User Study (US) to assess perceptual quality. Human experts rank the generated meshes of U methods from best (U - 1) to worst (0), revealing relative preference relationships. We prioritize the User Study as our primary subjective metric for mesh generation.

4.4 Comparison of Reinforcement Learning Objectives

To facilitate a controlled comparison of RL objectives for Stage I, we evaluate the base model, SFT, DHO, and representative RL methods: GRPO-style, DPO-style and NFT-style. All RL variants start from the same SFT weight, share the reward model, and are evaluated on GenMesh-Test. Every Stage I output is processed by the same frozen Stage II refiner. We repeat each experiment with three different random seeds and report the mean results. Implementation details are in the appendix.

Table 1 shows that DHO obtains higher scores than the other RL methods, especially on the User Study, our primary metric, demonstrating the advantage of positive-sample attraction-guided dynamic homing. Moreover, the improved Stage I outputs provide a more favorable basis for Stage II refinement, indicating that better Stage I geometry carries over to higher final mesh quality.

Figure 5 further illustrates qualitative differences among RL objectives. GRPO-style, DPO-style, and NFT-style produce implausible geometry (four fingers), convert image texture into unintended geometry (a flame-shaped forehead), or damage an otherwise intact structure (a broken staircase). These defects arise at Stage I and remain after Stage II refinement, suggesting that the quality of the lifting-stage geometry affects the refined output. In contrast, DHO achieves better geometric generation by attraction-guided navigation.

4.5 Comparison with Existing 3D Generation Methods



Figure 6: Visual comparison with representative 3D generation methods.

We next compare Flow3D-Pro with representative image-to-3D mesh generators on GenMesh-Test. Flow3D-Pro-Base denotes the Flow3D-Pro variant that uses the pretrained Stage I model without post-training, while retaining the same frozen Stage II refiner.

Table 2: Quantitative evaluation on 3D mesh generation.

Method	ULIP ↑	Uni3D ↑	User Study ↑
Michelangelo Zhao et al. (2023)	0.135	0.290	0.18
Craftsman 1.5 Li et al. (2025b)	0.146	0.325	1.32
Hi3DGen Ye et al. (2025)	0.144	0.335	3.77
Direct3D-s2 Wu et al. (2025b)	0.141	0.345	3.27
Hunyuan3D 2.1 Yang et al. (2025)	0.143	0.347	4.14
Trellis Xiang et al. (2025)	0.145	0.351	4.27
Flow3D-OPD Ning et al. (2026)	0.146	0.354	7.04
Trellis2 Xiang et al. (2026)	0.145	0.353	6.73
Flow3D-Pro-Base	0.143	0.350	5.64
Flow3D-Pro	**0.150**	**0.360**	**8.64**

As shown in Table 2, after Stage I post-training, Flow3D-Pro achieves the highest geometry generation performance among the compared methods on all three metrics, reaching 0.150 in ULIP, 0.360 in Uni3D, and 8.64 in User Study. In Figure 6, Flow3D-Pro achieves better reference-image adherence, more plausible geometry, and finer geometric details. These experiments demonstrate the role of DHO in improving geometric quality. The appendix provides additional component ablations, evaluations on LATTICE-Bench, and more qualitative results.

5 Conclusion

We present Dynamic Homing Optimization (DHO), a forward-process RL method for flow matching that equips negative-sample branch optimization with attraction-guided dynamic homing. DHO combines MAM to assign preferred positive targets with TDC to construct homing velocities toward them. We instantiate DHO in Flow3D-Pro using asynchronous online post-training. Experiments show that DHO outperforms representative DPO-, GRPO-, and NFT-style objectives in mesh generation, while Flow3D-Pro achieves superior image-to-3D geometry quality over existing methods.

AI USE STATEMENT

Generative AI tools were used in preparing this manuscript for language polishing and grammatical correction. The use of generative AI systems as components of the research methodology is described explicitly in the main paper. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work studies image-conditioned 3D geometry generation. The human evaluations reported in this paper involve experts ranking generated meshes solely with respect to perceptual quality, fidelity, and geometric plausibility; they do not involve judgments about participants or their personal characteristics. The datasets comprise artist-created and AI-generated meshes and reference images, whose use and any release are subject to applicable licenses and intellectual-property requirements. Potential concerns include biases inherited from the training data and automated evaluation, as well as possible misuse of generated 3D assets. We therefore limit our claims to the evaluated data categories and metrics and encourage responsible use of the method and generated assets in accordance with applicable licenses and legal requirements.

REPRODUCIBILITY STATEMENT

The main paper specifies MAM, TDC, the complete DHO objective, its integration with classifier-free guidance, the asynchronous online training framework, the datasets, and the evaluation protocol. The appendix provides proofs of the endpoint behavior propositions, DHO pseudocode, the reward prompt, the ODE and SDE rollout formulations, detailed training and inference configurations, component ablations, and additional evaluations. Together, these materials document the implementation details, and experimental settings needed to reproduce the reported methodology and results.

REFERENCES

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

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, October 2023.

Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G. Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4456–4465, June 2023.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Haoyou Deng, Keyu Yan, Chaojie Mao, Xiang Wang, Yu Liu, Changxin Gao, and Nong Sang. DenseGRPO: From sparse to dense reward for flow matching model alignment. In *International Conference on Learning Representations*, 2026a.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026b.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 79858–79885, 2023.

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances in Neural Information Processing Systems*, volume 35, pp. 31841–31854, 2022.

D. Guo, D. Yang, H. Zhang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022.

Minu Kim, Yongsik Lee, Sehyeok Kang, Jihwan Oh, Song Chong, and Se-Young Yun. Preference alignment with flow matching. In *Advances in Neural Information Processing Systems*, volume 37, pp. 35140–35164, 2024.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663, 2023.

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, Sheng Zhang, Xin Huang, Di Luo, Fan Yang, Fang Yang, Lifu Wang, Sicong Liu, Yixuan Tang, Yulin Cai, Zebin He, Tian Liu, Yuhong Liu, Jie Jiang, Linus, Jingwei Huang, and Chunchao Guo. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025a.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. Lattice: Democratize high-fidelity 3d generation at scale. *arXiv preprint arXiv:2512.03052*, 2025b.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

Haoxuan Li, Ziya Erkoç, Daniele Sirigatti, Vladislav Rosov, Lei Li, Angela Dai, and Matthias Nießner. Triflow: Generating artist-like 3d mesh topology via nearest-vertex vector fields. *arXiv preprint arXiv:2606.20131*, 2026.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025a.

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 diffusion and interactive geometry refiner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5307–5317, June 2025b.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2025c.

Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Feize Wu, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, Shaodong Wang, Xinhua Cheng, and Li Yuan. Uniworld-v2: Reinforce image editing with diffusion negative-aware finetuning and mllm implicit feedback. *arXiv preprint arXiv:2510.16888*, 2025d.

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, June 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40783–40818, 2025a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di ZHANG, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Weizhe Liu, Yunjie Wu, Xiangqian Shu, Guangwei Wang, Xiangyu Xu, Peng Li, Yujie Li, and Hengkai Guo. Dreamcharacter-1: From 3d generative foundation models to product-ready character generation. *arXiv preprint arXiv:2607.07817*, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and qiang liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024.

Yifu Luo, Haoyuan Sun, Xinhao Hu, Penghui Du, Keyu Fan, Bo Li, SiNan Du, Xu Wan, Zhiyu Chen, Bo Xia, Yongzhe Chang, Kai Wu, Kun Gai, Tiantian Zhang, and Xueqian Wang. Principled RL for flow matching emerges from the chunk-level policy optimization. In *International Conference on Machine Learning*, 2026.

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.

Zhiwei Ning, Zhen Zhou, Puhua Jiang, Xintong Han, Gengming Zhang, Jie Yang, Zhonglong Zheng, Yuanjie Zheng, Wei Liu, and Chunchao Guo. Flow3d-opd: Multi-teacher on-policy distillation for 3d geometry generation with flow-matching diffusion transformer. *arXiv preprint arXiv:2609.07137*, 2026.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations*, 2023.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32271–32297, 2023.

Hanxiao Sun, Mingxin Yang, Shuhui Yang, Zebin He, Xintong Han, Hongbo Fu, Chunchao Guo, and Wenhan Luo. Roar-3d: Routing arbitrary views for high-fidelity 3d generation. *arXiv preprint arXiv:2605.21121*, 2026a.

Qi Sun, Kiyohiro Nakayama, Jing Nathan Yan, Qixing Huang, Alexander Rush, Leonidas Guibas, Gordon Wetzstein, Jing Liao, and Guandao Yang. Meshflow: Mesh generation with equivariant flow matching. In *ACM SIGGRAPH Conference*, July 2026b.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2024.

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, June 2024.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8406–8441, 2023.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, yajie bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 170778–170804, 2025b.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21469–21480, June 2025.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14419–14429, June 2026.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023.

Yixian Xu, Kaiyuan Gao, Yuxiang Chen, Yilei Chen, Zecheng Tang, Zihao Liu, Zikai Zhou, Deqing Li, Hao Meng, Kuan Cao, Jiahao Li, Jie Zhang, Liang Peng, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiaoyue Chen, Yan Shu, Yanran Zhang, Yi Wang, Yu Wu, Yujia Wu, Zekai Zhang, Zhendong Wang, Xiao Xu, Kun Yan, and Chenfei Wu. Qwen-image-2.0-rl technical report. *arXiv preprint arXiv:2606.27608*, 2026.

Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1179–1189, 2023.

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27081–27091, 2024.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. In *Advances in Neural Information Processing Systems*, volume 37, pp. 78630–78652, 2024.

Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025.

Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 01–12, 2025.

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*, 42 (4), July 2023.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics*, 43(4), July 2024.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 73969–73982, 2023.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 46766–46782, 2024.

Zhen Zhou, Jian Liu, Biwen Lei, Jing Xu, Haohan Weng, Yiling Zhu, Zhuo Chen, Junfeng Fan, Yunkai Ma, Dazhao Du, Song Guo, Fengshui Jing, and Chunchao Guo. Mesh-pro: Asynchronous advantage-guided ranking preference optimization for artist-style quadrilateral mesh generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34248–34258, June 2026.

APPENDIX CONTENTS

A Methodology **15**

A.1 Proofs for Endpoint Behavior Analysis Propositions 15

A.2 Reward Design 17

A.3 DHO Pseudocode 18

A.4 Rollout Sampling 19

B Experiments **20**

B.1 More Implementation Details 20

B.2 User Study Evaluation Criteria 20

B.3 Component Ablations 20

B.4 More Quantitative Comparisons with Current Methods 24

B.5 More Qualitative Results 25

C Limitations and Future Work **27**

A METHODOLOGY

A.1 PROOFS FOR ENDPOINT BEHAVIOR ANALYSIS PROPOSITIONS

We provide proofs for the endpoint behavior analysis propositions.

Proposition 1. *Let $x_0 \sim p_0$ be a base noise sample that deterministically maps to a negative sample x^l under the pre-trained reference policy $v_{\theta_{ref}}$. As the policy v_θ converges to the optimal dynamic homing field $v^*(x_t, t, c) = \frac{x^{w*} - x_t}{1-t}$ via training, the generative mapping $F_\theta : x_0 \mapsto x_1$ shifts its terminal endpoint such that $F_{\theta^*}(x_0) = x^{w*}$.*

Proof. After DHO training, we use ordinary differential equation (ODE) sampling for inference. The generation process is an Initial Value Problem (IVP) governed by the ODE:

$$\frac{dx_t}{dt} = v^*(x_t, t, c) = \frac{x^{w*} - x_t}{1-t}, \tag{14}$$

with initial condition $x(0) = x_0$. This is a first-order linear ordinary differential equation, which can be rewritten as:

$$\frac{dx_t}{dt} + \frac{1}{1-t}x_t = \frac{x^{w*}}{1-t} \tag{15}$$

We solve this ODE by multiplying both sides by the integrating factor $\mu(t) = \exp\left(\int \frac{1}{1-t}dt\right) = \frac{1}{1-t}$:

$$\begin{aligned} \frac{1}{1-t}\frac{dx_t}{dt} + \frac{1}{(1-t)^2}x_t &= \frac{x^{w*}}{(1-t)^2} \\ \implies \frac{d}{dt}\left(\frac{x_t}{1-t}\right) &= \frac{x^{w*}}{(1-t)^2} \end{aligned} \tag{16}$$

Integrating both sides from 0 to t gives

$$\begin{aligned}\frac{x_t}{1-t} - \frac{x_0}{1} &= \int_0^t \frac{x^{w*}}{(1-s)^2} ds \\ &= x^{w*} \left[\frac{1}{1-s}\right]_0^t \\ &= x^{w*} \left(\frac{1}{1-t} - 1\right)\end{aligned} \tag{17}$$

Multiplying the entire equation by $(1-t)$ isolates x_t:

$$x_t - x_0(1-t) = x^{w*}\left(1-(1-t)\right) = tx^{w*} \tag{18}$$

Rearranging for x_t yields the linear interpolation path toward the assigned target:

$$x_t = tx^{w*} + (1-t)x_0 \tag{19}$$

Taking the limit as $t \to 1$ gives

$$\lim_{t\to 1} x_t = 1 \cdot x^{w*} + 0 \cdot x_0 = x^{w*} \tag{20}$$

This completes the proof. By minimizing the dynamic homing objective, the base noise x_0 that originally formulated the negative mode x^l is successfully rewired to map to the positive mode x^{w*}.

Proposition 2. *For a fixed x^l and its assigned x^{w*}, assume $v_\theta^{init}(x,t) = \frac{x^l - x}{1-t}$ for $t \in [0,1)$. Let the selected policy snapshots satisfy the idealized velocity update $v_\theta^{\tau+1}(x,t) = v_\theta^\tau(x,t) + \alpha\big(\frac{x^{w*}-x}{1-t} - v_\theta^\tau(x,t)\big)$ pointwise over (x,t), where $\tau \geq 0$ indexes successive policy snapshots rather than necessarily every raw SGD step and $\alpha \in (0,1)$ is an effective contraction factor between consecutive snapshots. Then the terminal generation is $x_1(\tau) = \big(1-(1-\alpha)^\tau\big)x^{w*} + (1-\alpha)^\tau x^l$, and hence $\lim_{\tau\to\infty} x_1(\tau) = x^{w*}$.*

Proof. Let the initial policy toward the negative sample be

$$v_\theta^{\text{init}}(x,t) = \frac{x^l - x}{1-t}. \tag{21}$$

For the fixed conditional target x^{w*}, the time-weighted negative objective $\mathcal{L}_{\text{neg}}$ and the corresponding flow-matching regression share the minimizer $v^* = \frac{x^{w*}-x}{1-t}$. We analyze selected policy snapshots under the idealized assumption that their velocity values contract pointwise toward this minimizer by an effective factor α. $\alpha \in (0,1)$ is an effective contraction factor between consecutive snapshots.

$$v_\theta^{\tau+1}(x,t) = v_\theta^\tau(x,t) + \alpha\big(\frac{x^{w*}-x}{1-t} - v_\theta^\tau(x,t)\big). \tag{22}$$

This recurrence is a field-space model between the selected snapshots, not an assertion that every raw Stochastic Gradient Descent (SGD) step follows the same update in the parameter space of a finite neural network. Subtracting $\frac{x^{w*}-x}{1-t}$ from both sides isolates the tracking residual $E^\tau(x,t) = v_\theta^\tau(x,t) - \frac{x^{w*}-x}{1-t}$:

$$E^{\tau+1}(x,t) = E^\tau(x,t) - \alpha\, E^\tau(x,t) = (1-\alpha)E^\tau(x,t). \tag{23}$$

Unrolling this geometric recurrence from $\tau = 0$ gives

$$E^\tau = (1-\alpha)^\tau E^0, \tag{24}$$

hence

$$v_\theta^\tau(x,t) = \frac{x^{w*}-x}{1-t} + (1-\alpha)^\tau \left(v_\theta^{\text{init}}(x,t) - \frac{x^{w*}-x}{1-t}\right). \tag{25}$$

We write $\beta_\tau = (1-\alpha)^\tau$. Since $\alpha \in (0,1)$, we have $\beta_\tau \in (0,1]$ with $\beta_0 = 1$ and $\beta_{\tau\to\infty} \to 0$. After the initial velocity $v_\theta^{\text{init}}(x,t) = \frac{x^l - x}{1-t}$ is substituted into (25), the two terms share the common gain $\frac{1}{1-t}$ and combine:

$$v_\theta^\tau(x,t) = \frac{(1-\beta_\tau)(x^{w*} - x) + \beta_\tau(x^l - x)}{1-t} = \frac{m_\tau - x}{1-t}, \tag{26}$$

where $m_\tau := (1-\beta_\tau)\, x^{w*} + \beta_\tau\, x^l$. The iterate is thus again a well-defined homing field, now targeting the interpolated anchor m_τ rather than x^{w*}. For each fixed τ, the generative IVP $\frac{dx_t(\tau)}{dt} = \frac{m_\tau - x_t(\tau)}{1-t}$ with $x_0(\tau) = x_0$ has the form solved in Proposition 1 with target m_τ, yielding

$$x_t(\tau) = t\, m_\tau + (1-t)\, x_0, \tag{27}$$

whose terminal step is $x_1(\tau) = m_\tau$. At $\tau = 0$, $\beta_0 = 1$ gives $x_1(0) = x^l$, recovering the negative generation of the initial policy. Since $|1-\alpha| < 1$ and $\beta_{\tau\to\infty} \to 0$, we have

$$\lim_{\tau\to\infty} x_1(\tau) = \lim_{\tau\to\infty} m_\tau = x^{w*}. \tag{28}$$

This is an idealized field-space analysis: it describes how the terminal endpoint of a particular negative generation evolves when selected policy snapshots exhibit pointwise contraction toward a fixed conditional homing field. It is not intended to characterize the parameter-space dynamics of a finite neural network or to require the recurrence to hold after every SGD step. The index τ may instead enumerate selected snapshots at which this effective contraction model is used to describe the observed field evolution.

A.2 Reward Design

We provide the reward model Gemini with the reference image and six canonical-view renders of one generated mesh. Following common VLM-as-a-judge practice, the prompt specifies the evaluator role, input evidence, assessment criteria, and output format. We use the following prompt template:

> **Role.** You are a strict 3D asset reviewer. Given a reference image and six canonical-view renders of a generated mesh, evaluate the visible geometric evidence. Inspect all views before assigning a score.
>
> **Reconstruction fidelity.** Compare the generated mesh against the reference. For silhouette proportions, check whether the mesh introduces unsupported parts that alter the outer contour, loses or breaks visible contour-defining parts, or poorly reconstructs salient regions such as the face, hands, or prominent accessories. For surface appearance, penalize image textures, color boundaries, or material patterns that are incorrectly converted into physical protrusions, grooves, or other geometry.
>
> **Geometric plausibility.** Check whether structures that should be symmetric remain balanced. Penalize isolated structures with no support from the reference or the object's geometry. Across views, verify that detail density is coherent: unseen or occluded regions should contain neither implausibly elaborate additions nor conspicuously sparse details relative to corresponding visible regions.
>
> **Orientation alignment.** Determine whether the mesh is aligned with the canonical world axes rather than inheriting the diagonal tilt, perspective skew, or camera pose of the reference image. A dominant object axis should appear horizontal or vertical in canonical views, and bilaterally symmetric objects should appear macroscopically balanced in the front view.
>
> **Output.** Briefly justify each of the three dimensions above. First, assign each dimension an individual score from 1 to 5, where a higher score indicates better generation quality. Then, summarize the scores using a weighting ratio of Reconstruction Fidelity : Geometric Plausibility : Orientation Alignment $= 3 : 2 : 1$, and return a single integer reward from 1 (severe defects) to 5 (high fidelity, plausible geometry, and canonical alignment). Output the final reward in the form `<answer>[1--5]</answer>`.

As illustrated in Figure 7, the reference image and six canonical views jointly provide evidence for evaluating fidelity, plausibility and orientation. Guided by the prompt template, the reward model first assesses reconstruction fidelity, geometric plausibility, and orientation alignment, and then aggregates the three assessments to produce an integer reward. The example demonstrates how the final reward is grounded in explicit observations of the generated geometry.

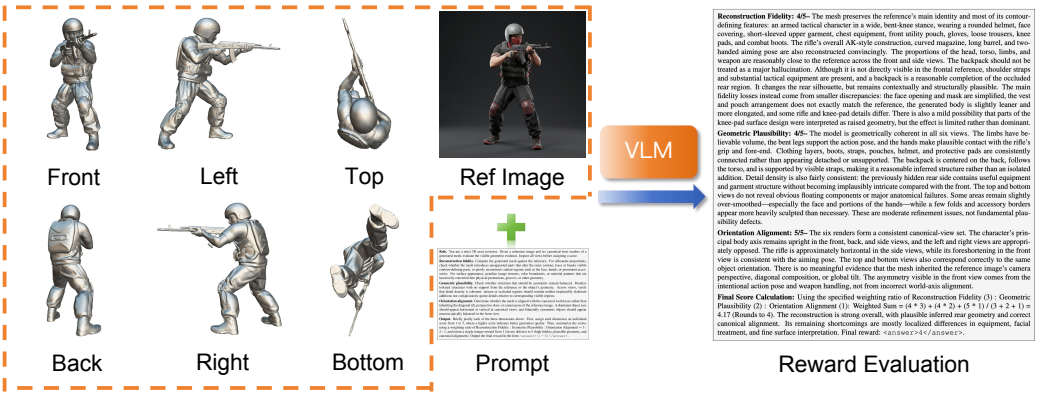

Figure 7: Illustration of the VLM-based reward evaluation process. The reward model receives a reference image, six canonical-view renderings of the generated mesh, and an evaluation prompt template, then produces criterion-wise assessments and a final scalar reward.

Algorithm 1 Dynamic Homing Optimization

Require: Policy v_θ, reference policy $v_{\theta_{\text{ref}}}$, reward model $r(\cdot)$, CFG scale $s \geq 1$, weights $\eta \in (0,1)$ and $\beta > 0$, learning rate γ

Ensure: Optimized policy v_θ

repeat

$\hat{v}_\theta(x,t,c) \leftarrow v_\theta(x,t,c_\varnothing) + s\big(v_\theta(x,t,c) - v_\theta(x,t,c_\varnothing)\big)$

$\hat{v}_{\theta_{\text{ref}}}(x,t,c) \leftarrow v_{\theta_{\text{ref}}}(x,t,c_\varnothing) + s\big(v_{\theta_{\text{ref}}}(x,t,c) - v_{\theta_{\text{ref}}}(x,t,c_\varnothing)\big)$

Roll out G samples $\{x_i\}_{i=1}^{G}$ for prompt c using $\hat{v}_\theta$ and evaluate rewards $r_i \leftarrow r(x_i,c)$

Compute group-normalized advantages $A_i \leftarrow (r_i - \bar{r})/s_r$

Form balanced sets $\{(x_i^w, \omega_i^w)\}_{i=1}^{B}$ and $\{x_i^l\}_{i=1}^{B}$ for prompt c

Sample times $t_i \sim \mathcal{U}(0,1)$

Minimum-Cost Attractive Matching

Build $C_{i,j} \leftarrow \|x_i^l - x_j^w\|_2^2$

Solve $\boldsymbol{\pi}^* \leftarrow \arg\min_{\boldsymbol{\pi}\in\Pi_B} \langle \mathbf{C}, \boldsymbol{\pi} \rangle$

Assign $(x_i^{w*}, \omega_i^{w*}) \leftarrow (x_{\pi^*(i)}^w, \omega_{\pi^*(i)}^w)$

Time-Aware Dynamic Correction

for $i \leftarrow 1$ to B **do**

$x_{t,i}^w \leftarrow (1-t_i)x_{0,i}^w + t_i x_i^w$

$x_{t,i}^l \leftarrow (1-t_i)x_{0,i}^l + t_i x_i^l$

$\tilde{v}_i \leftarrow (x_i^{w*} - x_{0,i}^l) - t_i(x_i^l - x_{0,i}^l)$

end for

Policy Evaluation and Update

Positive samples use identity self-matching, $x_i^{w*} = x_i^w$.

$\mathcal{L}_{\text{pos}} \leftarrow \frac{1}{B}\sum_i \omega_i^w \|\hat{v}_\theta(x_{t,i}^w, t_i, c) - (x_i^w - x_{0,i}^w)\|_2^2$

$\mathcal{L}_{\text{neg}} \leftarrow \frac{1}{B}\sum_i \omega_i^{w*} \|(1-t_i)\hat{v}_\theta(x_{t,i}^l, t_i, c) - \tilde{v}_i\|_2^2$

$\mathcal{L}_{\text{reg}} \leftarrow \frac{1}{2B}\sum_i \sum_{x_t \in \{x_{t,i}^w, x_{t,i}^l\}} \|\hat{v}_\theta(x_t,t_i,c) - \hat{v}_{\theta_{\text{ref}}}(x_t,t_i,c)\|_2^2$

$\theta \leftarrow \theta - \gamma\nabla_\theta\big(\eta\mathcal{L}_{\text{neg}} + (1-\eta)\mathcal{L}_{\text{pos}} + \beta\mathcal{L}_{\text{reg}}\big)$

until convergence

A.3 DHO Pseudocode

Algorithm 1 summarizes the DHO optimization procedure. For each prompt, rollout samples are first ranked using group-normalized advantages. Positive- and negative-advantage samples form the positive and negative sets, respectively, and the larger set is subsampled to obtain balanced sets for one-to-one assignment. The retained positive samples are weighted by their normalized positive advantages, while each negative sample later inherits the weight of its assigned target. CFG is applied to both rollout generation and policy updates, with $s = 1$ corresponding to the CFG-free

setting. Note that Algorithm 1 only presents the core DHO update, while Flow3D-Pro executes sample rollout and policy optimization asynchronously.

MAM computes distances in latent feature space between the two sets and solves a minimum-cost assignment using the Hungarian algorithm, yielding a distinct positive target for every negative sample. After sampling the flow time and constructing forward-process interpolants, TDC converts each assignment into a time-aware regression target that accounts for the remaining displacement to the positive endpoint. The positive branch retains the standard flow-matching target through self-assignment. Finally, DHO combines the advantage-weighted positive and negative losses with velocity-space regularization toward the frozen reference policy, and updates the policy using the resulting objective.

A.4 Rollout Sampling

Because DHO reconstructs policy-update interpolants from initial noises and rollout endpoints rather than stored sampling trajectories, its data collection is compatible with both ordinary differential equation (ODE) and stochastic differential equation (SDE) samplers. For each reference image, either sampler generates a group of G meshes for reward evaluation.

ODE sampling. DHO adopts the noise-to-data convention

$$x_t = (1-t)x_0 + tx_1, \qquad x_0 \sim \mathcal{N}(0, \mathbf{I}), \qquad dx_t = v_\theta(x_t, t, c)\, dt, \tag{29}$$

where t increases from 0 (noise) to 1 (data). For each reference image c, we independently draw G initial noises $\{x_{0,i}\}_{i=1}^{G}$, where $x_{0,i} \sim \mathcal{N}(0, \mathbf{I})$, and integrate the deterministic flow ODE from $t = 0$ to $t = 1$ using the same condition c. Since the ODE is deterministic given its initial state, diversity within the rollout group arises from the independent initial noises. The resulting endpoints $\{x_{1,i}\}_{i=1}^{G}$ are decoded into G meshes.

SDE sampling. Inspired by Flow-GRPO Liu et al. (2025a), we convert the deterministic flow ODE into a stochastic process for rollout exploration. Flow-GRPO uses a data-to-noise time variable τ and performs generation using the reverse-time SDE

$$dz_\tau = \left[\widetilde{v}_\theta(z_\tau, \tau, c) - \frac{\widetilde{\sigma}_\tau^2}{2}\nabla_{z_\tau} \log p_\tau(z_\tau \mid c)\right] d\tau + \widetilde{\sigma}_\tau\, dW_\tau, \qquad \widetilde{\sigma}_\tau = \kappa\sqrt{\frac{\tau}{1-\tau}}, \tag{30}$$

where κ is a scalar hyper-parameter that controls the noise level. Under the change of variables

$$t = 1 - \tau, \qquad x_t = z_{1-t}, \qquad v_\theta(x_t, t, c) = -\widetilde{v}_\theta(z_\tau, \tau, c), \tag{31}$$

the equivalent noise-to-data SDE used by DHO becomes

$$dx_t = \left[v_\theta(x_t, t, c) + \frac{\sigma_t^2}{2}\nabla_{x_t} \log p_t(x_t \mid c)\right] dt + \sigma_t\, dW_t, \qquad \sigma_t = \kappa\sqrt{\frac{1-t}{t}}. \tag{32}$$

For the rectified-flow path, the score can be recovered from the velocity as

$$\nabla_{x_t} \log p_t(x_t \mid c) = \frac{t v_\theta(x_t, t, c) - x_t}{1-t}. \tag{33}$$

Substituting it into Eq. (32) gives

$$dx_t = \left[v_\theta(x_t, t, c) + \frac{\sigma_t^2}{2(1-t)}\left(t v_\theta(x_t, t, c) - x_t\right)\right] dt + \sigma_t\, dW_t. \tag{34}$$

Using Euler–Maruyama discretization, we obtain the rollout update:

$$\begin{aligned} x_{t+\Delta t} &= x_t + \left[v_\theta(x_t, t, c) + \frac{\sigma_t^2}{2(1-t)}\left(t\, v_\theta(x_t, t, c) - x_t\right)\right]\Delta t + \sigma_t\sqrt{\Delta t}\,\epsilon, \\ &= x_t + \left[v_\theta(x_t, t, c) + \frac{\kappa^2}{2t}\left(t\, v_\theta(x_t, t, c) - x_t\right)\right]\Delta t + \kappa\sqrt{\frac{1-t}{t}}\sqrt{\Delta t}\,\epsilon, \end{aligned} \tag{35}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. We set $\kappa = 0.2$ for SDE rollout experiments.

B EXPERIMENTS

B.1 MORE IMPLEMENTATION DETAILS

Training and inference configurations for our method. Stage I is based on the Hunyuan3D 2.1 Yang et al. (2025) architecture. SFT uses a learning rate of 5×10^{-6} and is trained for 5K steps on 32 H20 GPUs with a per-GPU batch size of 4. DHO is trained for 5K optimization steps with AdamW at a learning rate of 2.5×10^{-6}. Training is performed in bfloat16 with a per-GPU batch size of 2. We set the homing coefficient η to 0.1, the regularization coefficient β to 1.0, the rollout group size G to 8, and the CFG scale to 3.0. For asynchronous training, we allocate $N = 28$ GPUs to rollout and $M = 4$ GPUs to policy updates, with $\lambda_{\min} = 4$, $\lambda_{\max} = 8$, and $K = 16$. Although DHO supports both ODE- and SDE-based rollouts, in practice, we favor SDE sampling for greater rollout diversity. After DHO training, Stage I inference uses 50 ODE sampling steps. The output voxel representation of Stage I has a resolution of 518. Stage II is based on LATTICE Lai et al. (2025b) and uses a shape VAE to decode structure-conditioned latents into an implicit geometric field for mesh extraction. Meshes are extracted at a decoding resolution of 1536. It uses 5 ODE sampling steps and a CFG scale of 5.0, and remains frozen for geometric refinement during inference.

RL objective comparison settings. In the RL objective comparison experiments, for a fair comparison, all RL variants share the Stage I architecture, SFT initialization, reward model, and GenMesh-Test evaluation set, and their outputs are processed by the same frozen Stage II refiner. Our implementations of GRPO-style, DPO-style, and NFT-style methods follow Flow-GRPO Liu et al. (2025a), Flow-DPO Liu et al. (2025b); Wu et al. (2025a), and DiffusionNFT Zheng et al. (2026), respectively. The positive and negative samples in DPO-style are paired using MAM, and $\beta_{\mathrm{DPO-style}}$ is set to 5000.0. For GRPO-style, we reparameterize Flow-GRPO's SDE from its original data-to-noise time convention into the equivalent noise-to-data formulation adopted in this work. We set the noise-level $\kappa_{\mathrm{GRPO-style}}$ to 0.2, matching the setting used in DHO. $\beta_{\mathrm{NFT-style}}$ is set to 1.0. All methods use asynchronous online training, with GPU configurations held constant across methods for both training and inference. Training is performed in bfloat16 with a per-GPU batch size of 2 and a rollout group size of $G = 8$. Inference uses 50 ODE sampling steps. All RL objectives are trained for 5K optimization steps using AdamW with a learning rate of 2.5×10^{-6}.

B.2 USER STUDY EVALUATION CRITERIA

For relative preference scoring, technical art experts are asked to sort all generated meshes from highest to lowest. The main considerations for the User Study are fidelity and plausibility. Fidelity mainly reflects how recognizably the generated mesh conveys the identity and overall proportions of the reference image, while plausibility mainly reflects whether the resulting shape reads as a coherent and well-formed 3D object in its own right. Experts make this judgment holistically from their professional experience rather than following a prescribed checklist. All evaluations are conducted in a blind manner: method identities are anonymized and the presentation order is randomized for every sample. Each reference image is independently evaluated by three experts, who rank the outputs of all methods from best to worst. The resulting ranking scores are averaged across the three experts and then across all reference images. Human experts rank the generated meshes of U methods from best (U - 1) to worst (0), revealing relative preference relationships.

B.3 COMPONENT ABLATIONS

This section provides additional quantitative and qualitative evaluations of the key components of DHO. For the component studies, we use the same Stage I architecture, training pipeline, reward model, and evaluation protocol, varying only the factor under study. Since each User Study produces a relative ranking of the variants within its own comparison group, these scores should be interpreted within each table or figure rather than across different ablations. We conduct a series of ablation studies on Stage I and evaluate the resulting models on GenMesh-Test.

Target assignment. We first isolate the target-selection mechanism in MAM. All training and evaluation settings are held fixed while MAM is replaced with random pairing, nearest-neighbor matching, or soft attraction. Adapted from the positive-sample attractive component of Drifting

Models Deng et al. (2026b), soft attraction uses a normalized similarity-weighted aggregation over all retained positive samples to produce a soft multi-target direction for each negative sample. This comparison examines whether minimum-cost one-to-one assignment provides a more effective homing target than unconstrained or soft diffuse alternatives.

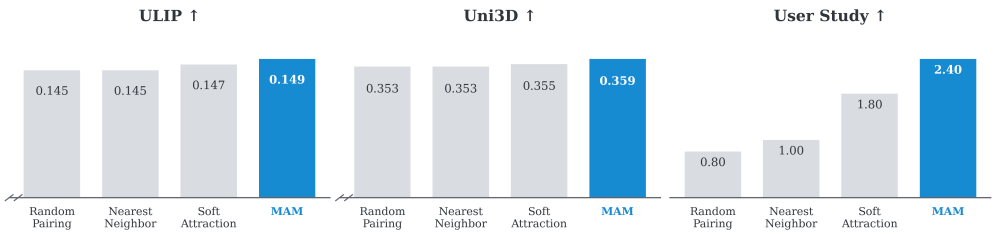

Figure 8: Effect of the target-assignment strategy in DHO.

Figure 8 shows that MAM obtains the highest scores on all three metrics, outperforming soft attraction as well as random and nearest-neighbor pairing. Soft attraction aggregates multiple positive samples and can retain a diffuse multi-target direction near equilibrium, whereas MAM assigns each negative sample a single positive target under a one-to-one constraint. These results support the use of deterministic, minimum-cost one-to-one assignment over unconstrained or soft multi-target guidance.

Sample selection. We evaluate whether prioritizing higher-advantage positive samples in MAM benefits DHO. When positives outnumber negatives, we compare retaining the B highest-advantage positives with randomly selecting B positives to form balanced sets for MAM.

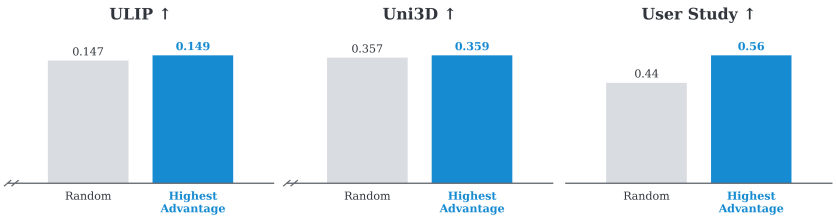

Figure 9: Effect of positive-sample selection in MAM.

As shown in Figure 9, highest-advantage selection performs better than random selection, with User Study increasing from 0.44 to 0.56. The results support prioritizing higher-advantage positive samples as homing targets in MAM.

Time-aware correction. We next study the temporal weighting in TDC by comparing the unweighted objective, $w(t) = 1$, with the proposed $w(t) = (1-t)^2$. The latter cancels the $1/(1-t)$ factor in the homing velocity when the loss is rewritten, yielding a numerically stable regression target over the full time horizon. All other components, including MAM and advantage weighting, are unchanged.

Table 3: Effect of time-aware weighting in TDC.

Time Weighting	ULIP ↑	Uni3D ↑	User Study ↑
$w(t) = 1$	0.146	0.354	0.45
$w(t) = (1-t)^2$	**0.149** (+2.05%)	**0.359** (+1.41%)	**0.55**

As described in Table 3, the time-aware objective obtains higher scores than the unweighted variant on all reported metrics: ULIP increases from 0.146 to 0.149, Uni3D from 0.354 to 0.359, and User Study from 0.45 to 0.55. Within this controlled comparison, the result supports the empirical benefit of using the numerically stable TDC regression.

Dynamic correction. The benefit of dynamic correction is assessed by replacing the negative-branch homing velocity $(x^{w*} - x_t^l)/(1-t)$ with the static action $x^{w*} - x_0$, while keeping MAM and other training settings unchanged.

Table 4: Static versus dynamic velocity targets for the negative branch.

Velocity Target	ULIP ↑	Uni3D ↑	User Study ↑
Static	0.145	0.353	0.31
Dynamic	**0.149** (+2.76%)	**0.359** (+1.70%)	**0.69**

Dynamic correction achieves higher scores on all three metrics in Table 4, raising User Study from 0.31 to 0.69. Accounting for the current state and remaining time therefore yields more effective guidance than a static action.

Advantage weighting. We evaluate whether the quality differences among positive samples should affect their contributions to the DHO objective. The weighted variant scales each positive-sample loss by its normalized positive advantage and transfers the same weight to negative samples assigned to that positive target. The unweighted variant treats all retained pairs equally.

Table 5: Ablation of positive-sample advantage weighting in DHO.

Advantage Weighting	ULIP ↑	Uni3D ↑	User Study ↑
Without	0.145	0.354	0.42
With	**0.149** (+2.76%)	**0.359** (+1.41%)	**0.58**

As shown in Table 5, advantage weighting performs better than the unweighted variant. This result suggests that emphasizing higher-advantage positive targets is beneficial in DHO.

Classifier-free guidance. We investigate how the CFG scale affects DHO. The same CFG scale is used consistently for rollout generation, policy updates, and inference. When the scale is set to 1.0, the guided velocity reduces to the conditional velocity, so the unconditional prediction has no effect. When the scale is greater than 1.0, the conditional prediction is extrapolated away from the unconditional prediction to increase conditioning strength.

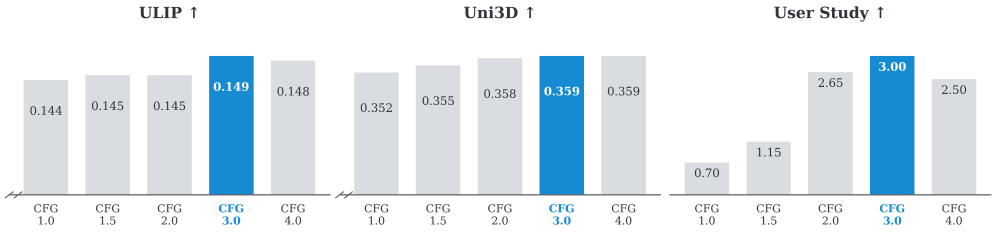

Figure 10: Ablation study on the CFG scale.

As shown in Figure 10, using CFG with $s = 3.0$ performs better than the CFG-free setting ($s = 1.0$) and the other evaluated guidance scales. This result suggests that $s = 3.0$ provides a favorable balance between rollout diversity and generation quality during policy optimization.

Negative homing and reference regularization. Moreover, we evaluate the balance among positive anchoring, negative homing, and reference regularization. We vary the homing coefficient η and the regularization coefficient β while retaining the remaining training configuration.

As shown in Table 6, among the tested settings, $\eta = 0.1$ and $\beta = 1.0$ obtain the highest scores and are therefore used as the default configuration. Since η weights $\mathcal{L}_{\text{neg}}$, setting $\eta = 0.0$ removes the dynamic homing of negative samples, leaving only the positive-sample branch. This variant drops from 0.149/0.359/1.92 to 0.145/0.354/1.13, indicating that the gains do not come from the positive-sample branch alone. However, increasing η to 0.5 reduces performance, suggesting that negative

Table 6: DHO performance with the homing coefficient η and the regularization coefficient β. The upper block fixes β and varies η (where $\eta = 0.0$ disables the dynamic homing branch for negative-samples), while the lower block fixes η and varies β (where $\beta = 0.0$ removes the reference-policy regularization). Each block is an independent User Study comparison group.

η				β			ULIP ↑	Uni3D ↑	User Study ↑
0.0	0.1	0.2	0.5	0.0	1.0	5.0			
✓					✓		0.145	0.354	1.13
	✓				✓		**0.149 (+2.76%)**	**0.359 (+1.41%)**	**1.92**
		✓			✓		0.147 (+1.38%)	0.357 (+0.85%)	1.68
			✓		✓		0.145 (+0.00%)	0.355 (+0.28%)	1.27
	✓			✓			0.140	0.342	0.30
	✓				✓		**0.149 (+6.43%)**	**0.359 (+4.97%)**	**1.70**
	✓					✓	0.146 (+4.29%)	0.358 (+4.68%)	1.00

homing should be appropriately balanced with positive-sample anchoring. Removing the reference-policy regularization ($\beta = 0.0$) yields worse results, where the policy deviates from the pretrained velocity field and tends to produce degenerate geometry. These results suggest that dynamic homing of negative samples supplies the corrective signal driving the improvement, while the reference-policy regularization keeps it within a stable neighborhood of the pretrained policy.

Rollout group size. The rollout group size G determines how many candidate meshes are generated for each reference image before computing group-normalized advantages. We study its effect by varying group sizes while keeping the Stage I architecture and DHO objective unchanged.

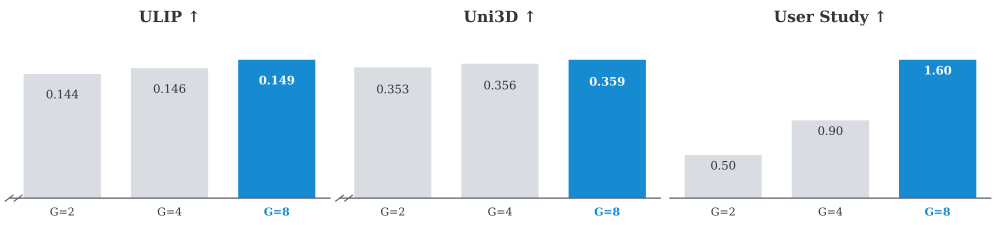

Figure 11: Effect of the rollout group size G on DHO performance.

As shown in Figure 11, performance improves consistently as G increases from 2 to 8, with the User Study score rising from 0.50 to 1.60. This suggests that a larger candidate set enables broader exploration, provides more reliable relative preference signals, and yields a richer set of positive anchors for MAM. These results suggest the potential of larger rollout groups to improve DHO performance.

Rollout sampler. We compare different rollout samplers, including ODE and SDE, to evaluate their influence on DHO performance. Both variants generate $G = 8$ candidate meshes for each reference image, while the SDE sampler additionally injects stochasticity along each sampling trajectory with $\kappa = 0.2$.

Table 7: Comparison of ODE and SDE rollout samplers.

Rollout Sampler	ULIP ↑	Uni3D ↑	User Study ↑
ODE	0.145	0.353	0.33
SDE	**0.149 (+2.76%)**	**0.359 (+1.70%)**	**0.67**

As shown in Table 7, SDE rollout performs better than ODE sampling, especially increasing User Study from 0.33 to 0.67. Compared with deterministic ODE sampling, the additional trajectory-level stochasticity of SDE broadens exploration.

In Figure 12, for negative samples, Δ_{neg} measures how much the current policy reduces the mean squared error (MSE) to the homing target compared with the frozen reference. For positive samples,

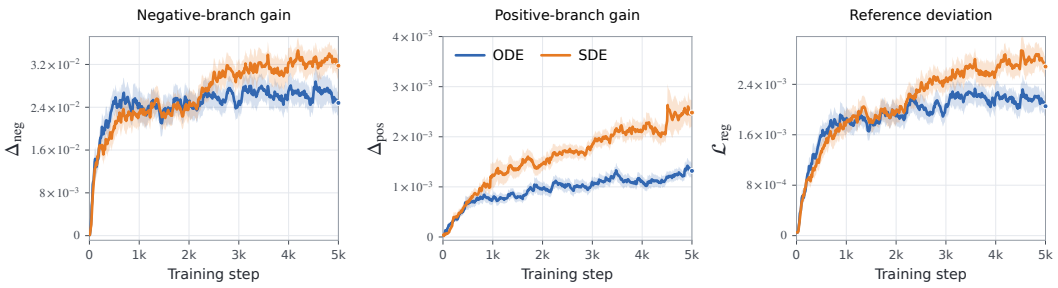

Figure 12: DHO training dynamics with ODE and SDE rollouts. From left to right: negative-branch gain, positive-branch gain, and reference-policy deviation.

Δ_{pos} measures the corresponding reduction for the positive-sample target. Larger values indicate greater improvements in target fitting. $\mathcal{L}_{\text{reg}}$ is the mean squared policy–reference velocity difference across both branches. SDE yields larger gains in both branches.

Reward gain. The contribution of each reward dimension to the geometric quality is further evaluated. Starting from the same SFT initialization, we progressively enable reconstruction fidelity, geometric plausibility, and orientation alignment in the reward prompt, and post-train Stage I with DHO.

Table 8: Impact of the reward dimensions used for DHO post-training, where "–" denotes the SFT model.

Reconstruction Fidelity	Geometric Plausibility	Orientation Alignment	User Study ↑
–	–	–	0.63
✓			1.46
✓	✓		1.85
✓	✓	✓	**2.06**

As presented in Table 8, reconstruction fidelity yields the largest gain, with further improvements from progressively incorporating geometric plausibility and orientation alignment.

Runtime analysis. At inference time, Stage I takes an average of 62 s per sample on an NVIDIA H20 GPU. This runtime includes image loading and preprocessing, conditional feature extraction, 50-step ODE sampling, VAE-based mesh decoding, and mesh conversion.

Stage II geometry refinement takes an average of 81 s per sample on the same GPU using 5-step ODE sampling. This runtime includes image loading and preprocessing, DINO-based conditional feature extraction, input mesh loading and voxelization, flow-matching sampling, VAE-based implicit-field decoding, and mesh conversion.

B.4 More Quantitative Comparisons with Current Methods

Comparison with 3D generators on LATTICE-Bench. We conduct additional experiments with current 3D geometry generation methods on LATTICE-Bench Lai et al. (2025b), using ULIP and Uni3D similarity metrics.

Consistent with the results on GenMesh-Test, Table 9 indicates that, after post-training with DHO, Flow3D-Pro obtains the highest scores among the listed representative image-to-3D mesh generation methods, reaching 0.140 in ULIP and 0.326 in Uni3D. These results demonstrate the role of DHO in improving geometric quality.

Comparison with additional post-training methods. We conduct experiments at Stage I, comparing SFT, rejection fine-tuning (RFT), PFM Kim et al. (2024), and DHO on GenMesh-Test to evaluate DHO against positive-only fine-tuning and auxiliary preference-flow alignment. RFT fine-tunes

Table 9: Quantitative evaluation on the LATTICE-Bench.

Method	ULIP ↑	Uni3D ↑
Michelangelo Zhao et al. (2023)	0.115	0.261
Craftsman 1.5 Li et al. (2025b)	0.129	0.298
Trellis Xiang et al. (2025)	0.126	0.311
Hunyuan3D 2.0 Zhao et al. (2025)	0.130	0.315
Hi3DGen Ye et al. (2025)	0.112	0.299
Direct3D-s2 Wu et al. (2025b)	0.122	0.314
LATTICE Lai et al. (2025b)	0.130	0.315
Flow3D-OPD Ning et al. (2026)	0.138	0.321
Flow3D-Pro	**0.140**	**0.326**

the SFT model only on positive-advantage rollouts, whereas PFM trains an auxiliary latent-space flow from negative to positive samples while keeping the SFT generator frozen. The preference-based variants share the training images and reward model. Base-generator inference settings and mesh decoding are held fixed, with PFM additionally integrating its preference flow.

Table 10: Comparison with additional post-training methods.

Method	ULIP ↑	Uni3D ↑	User Study ↑
SFT	0.144	0.351	0.97
RFT	0.145 (+0.69%)	0.355 (+1.14%)	1.39
PFM Kim et al. (2024)	0.147 (+2.08%)	0.356 (+1.42%)	1.70
DHO	**0.149 (+3.47%)**	**0.359 (+2.28%)**	**1.94**

As shown in Table 10, DHO outperforms both RFT and PFM. These results suggest that directly correcting the original generative field toward matched positive targets improves geometric quality more effectively than positive-only fine-tuning or auxiliary preference-flow alignment.

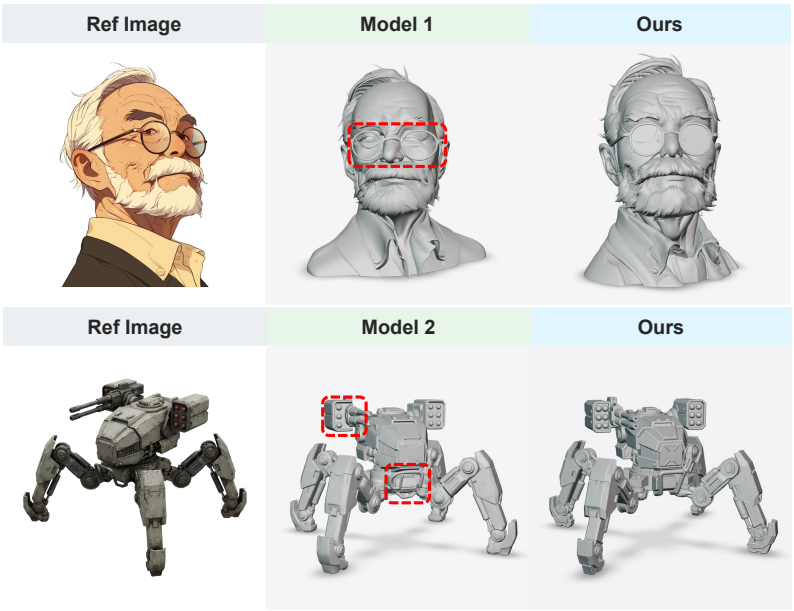

Figure 13: Visual comparison against commercial models. To facilitate visualization, all generated meshes are rotated to a common orientation.

B.5 More Qualitative Results

Commercial model comparison. Figure 13 compares Flow3D-Pro with two closed-source commercial methods, denoted as Model 1 and Model 2. In the shown examples, Model 1 exhibits

a skewed nose and eyeglasses with missing lenses, while Model 2 shows inconsistent left–right launcher layouts and distorted central geometry. Flow3D-Pro better preserves facial symmetry and bilateral component consistency–benefiting from the orientation-alignment and geometric-plausibility rewards, which jointly facilitate learning symmetric structures–while achieving more plausible global shapes and higher reconstruction fidelity.

More generation results. We provide additional visual results to evaluate Flow3D-Pro on diverse reference images. Figure 14 covers a broad range of categories, including human and stylized characters, animals, plants, vehicles, architecture, everyday objects, and hard-surface assets. Each panel displays the reference image in the upper-right corner and the generated mesh in the main view.

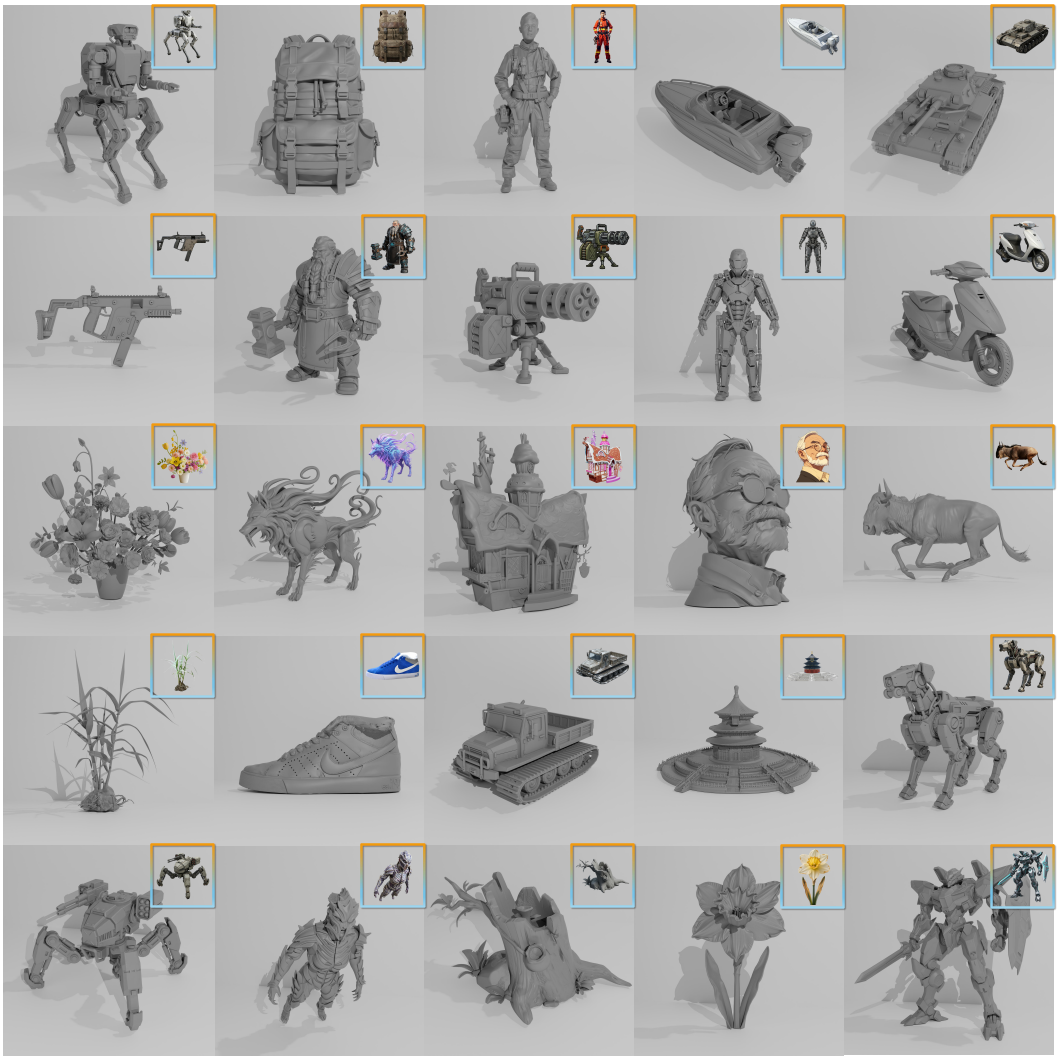

Figure 14: More qualitative results of Flow3D-Pro across diverse object categories and geometric structures.

Across the displayed examples, Flow3D-Pro reconstructs recognizable global silhouettes and category-specific structures while retaining local details such as character garments and facial features, articulated mechanical components, vehicle structures, and foliage. Together, these examples complement the quantitative evaluations and demonstrate Flow3D-Pro's versatility across diverse object categories and geometric structures.

Video demo. The supplementary materials include a video showcasing additional visual results of our method.

C LIMITATIONS AND FUTURE WORK

Designing a reward that comprehensively characterizes the geometric quality of a reference-image-guided 3D mesh remains difficult, and our current dimensions are not fine-grained enough for higher-precision geometric properties to be accurately reflected. We plan to further decompose the reward dimensions and collect feedback from more professional artists to refine the reward design and improve generation quality.

In addition, DHO is applied only to the lifting stage while the refinement stage remains frozen, since the latter is tightly conditioned on the Stage I output and its rollouts therefore exhibit limited diversity for policy optimization. We plan to explore ways of enriching the rollout diversity of the Stage II.

In current 3D generation, constrained by pretrained model capabilities, rollout diversity, and reward-distribution complexity, DHO with positive-sample attraction-guided dynamic homing achieves better mesh post-training quality than representative DPO-, GRPO-, and NFT-style objectives, which steer predicted velocities away from negative-trajectory directions without explicitly specifying a target velocity field toward preferred samples. Since 2D visual generation differs from the 3D setting in pretrained priors and reward distributions, we plan to carry out further exploratory experiments to examine the performance of DHO in 2D visual generation.