FLOW3D-OPD: MULTI-TEACHER ON-POLICY DISTILLATION FOR 3D GEOMETRY GENERATION WITH FLOW-MATCHING DIFFUSION TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent image-to-3D generation models built on flow-matching diffusion Transformers (DiT) can produce high-fidelity meshes, yet their post-training strategy remains largely unexplored. There exist several critical bottlenecks in reinforcement learning: the inherent difficulty of defining comprehensive rewards for 3D geometric quality, and the training instability that arises when jointly optimizing heterogeneous objectives. Inspired by the practicability of on-policy distillation (OPD) in large language models and image generation, we propose **Flow3D-OPD**, a two-stage post-training framework that introduces multi-teacher distillation into 3D geometry generation. In the first stage, we utilize the semi-policy to enhance the foundational capability of the pretrained model and then design an agentic verifier for 3D geometric quality evaluation. Based on the verifier, we cultivate domain-specialized teacher models via direct preference optimization (DPO). In the second stage, we consolidate heterogeneous expertise into a unified student model through on-policy distillation with a mixed divergence optimization strategy, which enhances training stability in joint optimization. Without relying on elaborate modifications, our straightforward yet effective design achieves consistent improvements across various geometric dimensions and surpasses all teacher models in the overall metric. Extensive experiments demonstrate that our approach provides an effective paradigm for reinforcement learning and on-policy distillation in 3D generation.

1 INTRODUCTION

Recent 3D geometry generation pipelines Li et al. (2025); Zhao et al. (2025); Lai et al. (2026; 2025) employ a mixture architecture where a diffusion Transformer (DiT) Peebles & Xie (2023) generates latent representations, and then the subsequent variational autoencoder (VAE) and isosurface extraction algorithm are utilized to decode the representation into explicit 3D meshes, as shown in Fig. 1. The pipeline produces high-quality 3D assets from the image, enabling a wide range of applications in virtual reality, digital twins, and embodied AI Mildenhall et al. (2021); Poole et al. (2022).

Despite this rapid advancement, post-training approaches through reinforcement learning (RL) remain absent from the 3D generation literature, which has long been employed in large language models Ouyang et al. (2022); Rafailov et al. (2023) and image generation Black et al. (2024); Liu et al. (2025). The fundamental obstacle lies in the ambiguity of quality evaluation for 3D geometry. Unlike text-to-image generation where verifiers have been rapidly developed and proven effective Huang et al. (2023); Ghosh et al. (2023), precise quality assessment for 3D meshes requires multi-view rendering and geometric analysis, which relies on specific multi-modal reasoning. Existing approaches Lai et al. (2026); Li et al. (2025) utilize ULIP or Uni3D models Xue et al. (2023); Zhou et al. (2024) to calculate global similarity between 2D images and 3D points, yet such a score cannot recognize fine-grained geometric quality such as orientation alignment, pose accuracy, and textural consistency. Moreover, jointly optimizing these heterogeneous objectives via diverse rewards will encounter unstable training where updates favoring one dimension will degrade others, which hinders consistent improvement in various dimensions.

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.

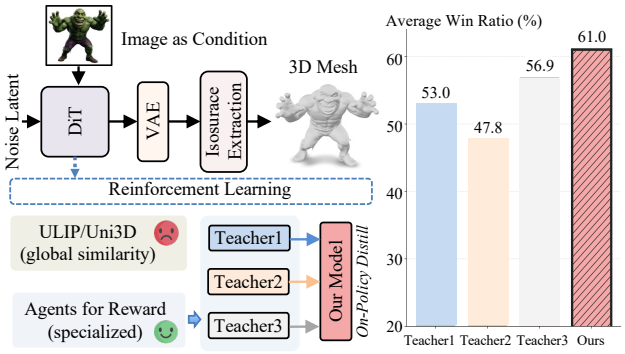

Figure 1: Our framework focuses on the on-policy distillation of the DiT module in the 3D generation pipeline. Our Flow3D-OPD enables to achieve a higher average win ratio than three specialized teachers.

However, recent image generation works introduce the on-policy distillation (OPD) strategy to improve model performance across multiple evaluation dimensions simultaneously. DiffusionOPD Li et al. (2026) and Flow-OPD Fang et al. (2026) prove that multi-teacher OPD can effectively unify various specialized capabilities into a single model by replacing sparse scalar rewards with dense velocity-field supervision along the denoising trajectory in the student model. This distillation paradigm could achieve consistent improvement across the multiple evaluation metrics in image generation. However, applying multi-teacher on-policy distillation to 3D geometry generation remains unexplored due to the difficulties in training domain-specialized teachers with 3D-specific rewards and consolidating these heterogeneous capabilities stably.

In this paper, we propose Flow3D-OPD, which innovatively introduces multi-teacher on-policy distillation into 3D geometry generation. Our approach follows a two-stage pipeline to overcome the above challenges. In the first stage, we train three teacher models targeting distinct evaluation dimensions. Specifically, we employ a semi-policy preference optimization strategy to enhance the base model's capability for fine-tuning, then decompose the quality evaluation into three different dimensions. Subsequently, an agentic verifier based on VLMs is designed for pairwise mesh comparison, which conducts multi-round structured reasoning with tool invocation and case memory mechanisms. Based on this verifier, we train the dedicated teacher for each quality dimension via direct preference optimization (DPO). In the second stage, we distill all three frozen teachers into a unified student model through on-policy distillation with mixed divergence optimization. In detail, the student model generates its own denoising trajectory, and the corresponding teacher model provides step-level guidance along the mixed task-routing. To alleviate distillation instability, we accumulate the divergence in a mini-batch for a single backward pass to ensure balanced multi-dimensional optimization. As shown on the right side of Fig. 1, our student model successfully surpasses all teacher models on the overall win-ratio metric. In summary, our contributions are three-fold as follows:

- We propose the Flow3D-OPD approach, which is, to the best of our knowledge, the first to introduce on-policy distillation in 3D generation through a two-stage training pipeline.
- We design an agentic reward system to enable DPO training of domain-specific teachers, followed by a multi-teacher distillation strategy with mixed divergence optimization that consolidates their capabilities into a unified student model.
- Extensive experiments demonstrate that our method consistently improves generation quality across all dimensions and achieves superior performance than previous works.

2 Related Work

3D Generation Models The field of 3D geometry generation has undergone several paradigm shifts. Primitive explorations leverage GANs Wu et al. (2016), CLIP-guided synthesis Sanghi et al. (2022), and Transformer-based approaches Yan et al. (2022); Yin et al. (2025) to produce category-specific shapes. The subsequent works Ho et al. (2020); Rombach et al. (2022); Poole et al. (2022); Lin et al. (2023) are then introduced to enable text-to-3D and image-to-3D generation by leveraging pretrained 2D diffusion priors, though these methods require expensive per-instance optimization. A more recent line of work trains diffusion models natively on 3D latent spaces, achieving substantial quality improvements. Representative methods such as TripoSG Li et al. (2025), Hunyuan3D Zhao et al. (2025); Lai et al. (2025), and LATTICE Lai et al. (2026) employ flow-matching on structured latent spaces to generate high-fidelity meshes. These systems usually follow a two-stage pipeline

where the flow-matching generator produces latent point embeddings conditioned on the input image, followed by a variational autoencoder that decodes into signed distance fields for mesh extraction via differentiable marching cubes. Our work builds upon this architecture, focusing on the post-training alignment strategy that aims to improve generation quality beyond what pretraining alone can achieve.

Reinforcement Learning for Generative Models RL-based post-training has become an indispensable paradigm for further enhancing model capabilities Ouyang et al. (2022), progressing from PPO Schulman et al. (2017) and DPO Rafailov et al. (2023) to GRPO Shao et al. (2024). Its application has also been expanded from large language models to image generation. DDPO Black et al. (2024) first demonstrates policy gradient optimization for diffusion models by treating each denoising step as an action in a Markov decision process. DPOK Fan et al. (2023) introduces Kullback-Leibler (KL)-regularized optimization to prevent reward hacking, and DiffusionDPO Clark et al. (2024) adapts preference optimization to the denoising trajectory. In the flow-matching regime, Flow-GRPO Liu et al. (2025) leverages the SDE interpretation to define proper action distributions amenable to group-relative advantage estimation, while DiffusionNFT Zheng et al. (2025) proposes negative-free training that eliminates the need for a reference model. Despite these advances in 2D generation, RL-based optimization for 3D remains largely unexplored, primarily due to the difficulty of designing reliable multi-dimensional quality judgements for 3D meshes during training.

On-Policy Distillation On-policy distillation (OPD) addresses the distribution shift of offline distillation by requiring the student to generate trajectories from its own current policy under real-time teacher supervision Agarwal et al. (2024); Gu et al. (2024). In the LLM, GKD Agarwal et al. (2024) first formalizes on-policy distillation by training on self-generated sequences to mitigate exposure bias. Subsequent work MiniLLM Gu et al. (2024) adopts reverse KL to encourage the student to concentrate probability mass on high-quality modes, while DistiLLM Ko et al. (2024) proposes skewed KL interpolation to balance mode-seeking and mode-covering behaviors. In visual generation, consistency distillation Song et al. (2023); Luo et al. (2023) first explores distilling multi-step diffusion models into fewer-step students. DiffusionOPD Li et al. (2026) extends OPD to diffusion models by exploiting the Markov structure of the denoising chain, deriving closed-form per-step KL divergence between Gaussian transition kernels. Flow-OPD Fang et al. (2026) further generalizes this to flow-matching architectures with stochastic differential equation (SDE) formulations and time-adaptive weighting, showing that multi-teacher distillation resolves the reward sparsity and gradient interference of joint multi-objective RL. Our work adapts the OPD mechanism on 3D geometry generation, where the inherently heterogeneous nature of quality assessment makes multi-teacher distillation a particularly natural fit.

3 PRELIMINARY

DPO in Diffusion Models. Direct preference optimization (DPO) Rafailov et al. (2023) enables learning from pairwise preferences. For language models, given winning and losing responses x^w, x^l conditioned on prompt c, DPO directly optimizes the policy model p_θ via:

$$\mathcal{L}_{\text{DPO}}^{\text{LLM}} = -\mathbb{E}\left[\log\sigma\left(\beta\log\frac{p_\theta(x^w|c)}{p_{\text{ref}}(x^w|c)} - \beta\log\frac{p_\theta(x^l|c)}{p_{\text{ref}}(x^l|c)}\right)\right], \tag{1}$$

where σ is the sigmoid function, β defines the penalty coefficient, and p_{ref} indicates the reference model. DiffusionDPO Clark et al. (2024) extends this to diffusion models by deriving an equivalent loss over the denoising trajectory. Under the evidence lower bound (ELBO), the log-likelihood ratio decomposes into per-step noise prediction differences. The final loss for flow-matching models is:

$$\begin{aligned}\mathcal{L}_{\text{DPO}}^{\text{FM}}(\theta; x^w, x^l) &= -\mathbb{E}_t\left[\log\sigma\left(-\beta\omega(t)\cdot\Delta(\theta, t, x^w, x^l)\right)\right], \\ \Delta(\theta, t, x^w, x^l) &= \left(\|v_\theta(x_t^w) - v^w\|_2^2 - \|v_{\text{ref}}(x_t^w) - v^w\|_2^2\right) \\ &\quad - \left(\|v_\theta(x_t^l) - v^l\|_2^2 - \|v_{\text{ref}}(x_t^l) - v^l\|_2^2\right),\end{aligned} \tag{2}$$

where x_t^w and x_t^l are noisy intermediates of the winner and loser at timestep t, defined via $x_t^{\{w,l\}} = (1-t)x^{\{w,l\}} + t\epsilon$. $v^w = \epsilon - x^w$ and $v^l = \epsilon - x^l$ are the target velocities. v_θ and v_{ref} are the policy and reference velocity predictions, and $\omega(t)$ is a weighting function Ho et al. (2020).

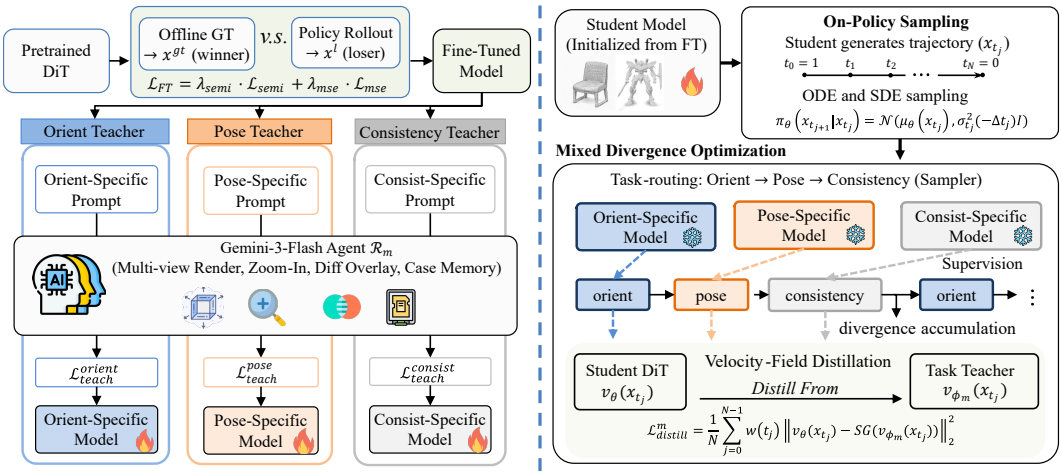

Figure 2: Overview of our Flow3D-OPD framework. Stage 1 (left): the pretrained flow-matching DiT is first fine-tuned with $\mathcal{L}_{\text{FT}}$, then three teachers are trained through DPO with dimension-specific agentic reward. Stage 2 (right): the student model generates its own denoising trajectory via on-policy sampling, and a mixed task-routing cycle through the three frozen teachers with mini-batch divergence accumulation.

On-Policy Distillation. On-policy distillation (OPD) addresses the distribution shift inherent in offline distillation by requiring the student to generate trajectories from its own current policy under real-time teacher supervision Agarwal et al. (2024) . For autoregressive models, this is formulated as minimizing the reverse Kullback-Leibler (KL) divergence:

$$\mathcal{L}_{\text{OPD}} = -\mathbb{E}_{y\sim\pi_\theta}\left[\log \frac{\pi_{\text{teacher}}(y|x)}{\pi_\theta(y|x)}\right] = D_{\text{KL}}(\pi_\theta \| \pi_{\text{teacher}}), \tag{3}$$

where π_θ denotes the student policy, π_{teacher} is the frozen teacher, and y are trajectories sampled on-policy from the student. By training on self-generated distributions, OPD suppresses exposure bias and ensures the student receives supervision at the states it actually visits during inference.

4 Our Method

Our model follows the architecture of Hunyuan3D-2.1 Lai et al. (2025), where the flow-matching DiT generates structured latent $x \in \mathbb{R}^{4096\times 64}$ conditioned on the image features. We employ a two-stage post-training strategy to optimize the pretrained DiT model, as illustrated in Fig. 2.

4.1 Stage 1: Domain-Specialized Teacher Training

Considering the defective capability of the basic model in rolling out high-quality samples for reliable preference learning, we first fine-tune the basic model via semi-policy preference optimization, and then train our teacher models on top of this fine-tuned (FT) model in stage 1.

4.1.1 Fine-Tuning with Semi-Policy Preference Optimization.

In this phase, offline ground-truth 3D meshes serve as the preferred samples, while the current policy generates the rejected samples. Specifically, we replace x^w in preference margin $\Delta(\theta, t, x^w, x^l)$ (Eq. 2) with the ground-truth latent x^{gt} to obtain the semi-policy DPO loss $\mathcal{L}_{\text{semi}}$. This formulation enables the model to perform only a single denoising process to obtain x^l and ensures that the target velocity field is optimal. The denoising trajectory is along the stochastic differential equation (SDE) or ordinary differential equation (ODE) sampling. Meanwhile, we also add the mean squared error (MSE) loss $\mathcal{L}_{\text{semi}}$ to form the overall fine-tuning loss function:

$$\mathcal{L}_{\text{FT}} = \lambda_{\text{semi}}\,\mathcal{L}_{\text{semi}} + \lambda_{\text{mse}}\,\mathbb{E}_t\left[\|v_\theta(x_t) - v^{\text{gt}}\|_2^2\right], \tag{4}$$

where $\lambda_{\text{semi}} = 0.2$, $\lambda_{\text{mse}} = 0.8$. Based on the supervision, we enhance the foundational capability of the basic model and obtain the FT model for subsequent teacher model training.

4.1.2 Agentic Reward Design for Teacher Training.

We design an agentic verifier system $\mathcal{R}_m$ based on Gemini-3-Flash Google DeepMind (2025) for pairwise mesh comparison. Given a 3D mesh pair (M_a, M_b) and the reference image I, our agent performs multi-round structured reasoning:

$$(s_a, s_b) = \mathcal{R}_m(M_a, M_b, I; \mathcal{P}_m, \mathcal{T}, \mathcal{C}), \tag{5}$$

where $\mathcal{P}_m$ is the dimension-specific evaluation prompt ($m \in \mathcal{M} = \{\text{orient}, \text{pose}, \text{consist}\}$), $\mathcal{T}$ denotes the tool set, including zoom-in rendering, viewpoint rotation, and difference-map overlay, which are necessary in geometric quality evaluation. And $\mathcal{C}$ is the case memory which helps the agentic verifier attend to common issues in the corresponding instance category. The agent is allowed up to four rounds of tool invocation before outputting structured preference scores (s_a, s_b). Compared to global scalar metrics like ULIP/Uni3D, our agentic reward provides fine-grained and dimension-specific quality evaluation with multi-view reasoning. Details of prompts and tools are provided in the supplementary material.

Building upon the agentic reward $\mathcal{R}_m$, we train each teacher v_{ϕ_m} via DPO. At each step, the sampling model rolls out two candidate latents (x_a, x_b), which are then decoded to 3D meshes and compared by the agentic reward to produce preference scores (s_a, s_b). Then, we filter out pairs with a small score margin $|s_a - s_b| < \delta$ and form the high-quality winner-loser pair (x^w, x^l). Following Eq. 2, each teacher is optimized via:

$$\mathcal{L}^m_{\text{teach}} = \mathcal{L}^{\text{FM}}_{\text{DPO}}(\theta_m; x^w, x^l), \quad m \in \mathcal{M}. \tag{6}$$

In practice, we observe that the majority of sampled pairs yield score differences below δ, leading to inefficient training under fully online rollout. To address this, we adopt an iterative *rollout-then-train* strategy: the model first generates a batch of candidate pairs over multiple steps and saves the valid ones filtered by δ, then trains on this collected data before the next rollout stage. This strategy improves the efficiency of multi-GPU parallel training. Based on these training objectives, we obtain three domain-specialized teacher models, each excelling at its target quality dimension.

4.2 Stage 2: Multi-Teacher On-Policy Distillation

Given three trained teacher models $\{v_{\phi_m}\}_{m\in\mathcal{M}}$, we then integrate them into a single student v_θ with on-policy distillation and mixed divergence optimization in this stage.

4.2.1 On-Policy Distillation in Flow-Matching.

Following the OPD loss function formulated in Eq. 3, we minimize the reverse KL divergence between the student and teacher transition kernels at each step along the student's on-policy trajectory. Under the SDE discretization (Euler-Maruyama), the one-step transition kernel is Gaussian:

$$\pi_\theta(x_{t_{j+1}}|x_{t_j}) = \mathcal{N}\big(\mu_\theta(x_{t_j}),\ \sigma^2_{t_j}(-\Delta t_j)I\big), \tag{7}$$

where $1 = t_0 > t_1 > \cdots t_N = 0$ define the denoising timesteps and $\Delta t_j = t_{j+1} - t_j < 0$. σ_{t_j} is the per-step diffusion coefficient determined solely by the noise schedule, and the transition mean μ_θ is deterministically parameterized by the student velocity field. The teacher transition kernel π_{ϕ_m} uses the teacher velocity to compute its mean μ_{ϕ_m} but shares the same covariance $\bar{\sigma}^2_{t_j} = \sigma^2_{t_j}(-\Delta t_j)$. Since both kernels share the same isotropic covariance, their reverse KL admits a closed form:

$$\begin{aligned}\text{KL}\big(\pi_\theta \| \pi_{\phi_m}\big) &= \text{KL}\big(\mathcal{N}\big(\mu_\theta,\ \bar{\sigma}^2_{t_j} I\big) \| \mathcal{N}\big(\mu_{\phi_m},\ \bar{\sigma}^2_{t_j} I\big)\big) \\ &= \frac{\|\mu_\theta(x_{t_j}) - \mu_{\phi_m}(x_{t_j})\|_2^2}{2\bar{\sigma}^2_{t_j}}.\end{aligned} \tag{8}$$

The stochastic transition degenerates to a deterministic map under the ODE regime, and distribution matching reduces to direct velocity-field alignment with L2 loss. The reverse KL defines the per-step distillation objective for a single teacher.

4.2.2 MIXED DIVERGENCE OPTIMIZATION

To consolidate the capability of three teacher models into a single student model, we introduce a mixture divergency optimization mechanism. At each training step, the sample carries a data-type label $m \in \{\text{orient}, \text{pose}, \text{consist}\}$ that deterministically routes supervision to the corresponding teacher v_{ϕ_m}, so that its gradient stays free from cross-teacher conflict. The routed OPD loss aggregates per-step velocity-matching divergence along the on-policy trajectory via:

$$\mathcal{L}^{m}_{\text{distill}}(\theta) = \frac{1}{N}\sum_{j=0}^{N-1} w(t_j)\left\|v_\theta(x_{t_j}) - \text{SG}(v_{\phi_m}(x_{t_j}))\right\|_2^2 \tag{9}$$

where the stop-gradient operator $\text{SG}(\cdot)$ detaches the teacher output. The weighting function $w(t_j)$ controls the distillation strength across different denoising timestep t_j, which equals $\frac{\Delta t_j}{2}\left(\frac{\sigma_{t_j}(1-t_j)}{2t_j} + \frac{1}{\sigma_{t_j}}\right)^2$ under SDE sampling and reduces to a constant 1 under ODE sampling. The detailed derivation is provided in the supplementary material.

While task-routing lets each mini-batch distill the velocity field of its own teacher, a single update driven by one dimension alone destabilizes learning. Therefore, we mix the training data so that every M mini-batch cycle over all geometric dimensions, where step k is routed to a single teacher m_k and computes only its divergence $\mathcal{L}^{m_k}_{\text{distill}}(\theta)$. Rather than updating after each step, we accumulate the per-step divergences across the full cycle and perform a single optimizer update on their average:

$$\mathcal{L}^{\text{total}}_{\text{distill}} = \frac{1}{M}\sum_{k=1}^{M} \mathcal{L}^{m_k}_{\text{distill}}(\theta). \tag{10}$$

In this way, task-routing isolates each teacher's supervision at the step level, while divergence accumulation aggregates them at the update level, substantially mitigating the distillation instability of multi-objective optimization.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Training Settings. We build upon Hunyuan3D-2.1 Lai et al. (2025), a 3B-parameter DiT architecture for 3D latent representation generation. Three domain-specialized teachers are refined from the fine-tuned model via on-policy DPO with learning rate 1×10^{-6}, SDE sampling with noise level $a = 0.3$, and a minimum confidence margin of $\delta = 0.167$ across 2×8 H20 GPUs. During the second stage, the student is also initialized from the fine-tuned model and distilled by randomly sampling states along the 50-step denoising trajectory. To handle the imbalanced sizes of the three sub-datasets, we employ dataset sampling with replacement across data types. At inference, we apply classifier-free guidance with a default scale of 3.0.

Evaluation Metrics. We adopt two complementary evaluation protocols. For multi-dimensional quality assessment, we employ pairwise judgement to report the win rates across orientation, pose, and consistency specific datasets. For overall generation quality evaluation, we report ULIP Xue et al. (2023) and Uni3D Zhou et al. (2024) scores to measure 3D–text and 3D–image cosine similarities in LATTICE-Bench Lai et al. (2026). We also provide qualitative visualizations of generated 3D meshes for intuitive comparison.

5.2 MAIN RESULTS

Multi-Dimensional Quality Evaluation Table 1 demonstrates the effectiveness of our method across multiple quality dimensions. We compare the meshes generated by each model (denoted as model B) against those from the pretrained baseline Hunyuan3D-2.1 Lai et al. (2025) (denoted as model A) with our agentic verifier for pairwise evaluation. "A wins" indicates the ratio that the baseline model performs better, while "B wins" indicates that the evaluated model is superior. Compared with the fine-tuned (FT) model, each teacher achieves strong performance on its target dimension but degrades on others. For instance, the orient teacher achieves the lowest A wins (2.7%) on orientation evaluation but causes consistency to drop significantly (30.8% A wins). This phenomenon

Table 1: Pairwise evaluation across three quality dimensions. Model A is the pretrained baseline; Model B is the evaluated model. A/B wins (%) denote the percentage of cases each is judged superior. Best results are in **bold**, second best are underlined.

Method	Orient-Eval		Pose-Eval		Consistency-Eval		Overall	
	A wins ↓	B wins ↑	A wins ↓	B wins ↑	A wins ↓	B wins ↑	A wins ↓	B wins ↑
FT (semi-policy)	5.5	11.2	0.5	85.1	12.4	79.4	5.9	54.9
Orient-Teacher	**2.7**	**22.7**	0.8	89.0	30.8	53.2	10.3	53.0
Pose-Teacher	10.1	14.3	**0.0**	**93.7**	41.7	41.7	16.0	47.9
Consistency-Teacher	6.5	15.9	1.0	83.5	**8.0**	**82.0**	5.2	56.9
DPO	13.4	7.2	19.1	71.8	13.0	77.0	15.1	48.3
FT + DPO	5.9	14.4	0.9	91.8	14.0	76.0	6.6	57.2
OPD	10.9	10.9	**0.0**	93.4	15.0	79.0	8.6	57.3
FT + OPD (Ours)	5.1	18.8	**0.0**	93.5	10.0	81.0	**4.9**	**61.0**

confirms that the three evaluated dimensions are inherently heterogeneous, and agentic-based DPO can yield substantial improvements on specific target objectives.

We further present our Flow3D-OPD result and compare with different training strategies at the bottom of the table. Directly applying DPO on the initial model without fine-tuning leads to degraded performance due to insufficient basic capability. Applying OPD without the semi-policy stage yields moderate improvements across all dimensions but falls short of the full pipeline. In contrast, our Flow3D-OPD achieves the best overall balance: 18.8%, 93.5%, and 81% wins ratio on orient, pose, and consistency, respectively. Finally, our method consistently outperforms all teachers on the overall metrics with 61.0% wins ratio.

To validate the generalization and effectiveness of our model, we additionally conduct two complementary assessments on Table 2. First, we employ GPT-5.5 OpenAI (2026) as an independent evaluator, which is a different model from the Gemini-3-Flash used in the teacher training reward. Our Flow3D-OPD model also achieves the highest B wins (51.4%) under GPT-5.5, confirming that the observed improvements are not artifacts of evaluator-specific optimization. Moreover, we conduct a blind user study where human annotators rank all methods by their generation quality from best to worst without knowing model identities. Our method also achieves the highest ranking score, further validating that the improvements reflect genuine 3D quality gains perceived by human evaluators.

Table 2: Generalization and effectiveness validation on the GPT-5.5 model and user study.

Method	GPT-5.5		User study
	A wins ↓	B wins ↑	
Orient-Teacher	13.4	47.3	3.68
Pose-Teacher	15.1	43.6	3.61
Consistency-Teacher	11.8	50.9	3.02
DPO	20.4	41.5	3.16
FT + OPD	14.1	44.7	3.62
Flow3D-OPD (Ours)	**9.9**	**51.4**	**3.91**

Comparison with Previous Methods Table 3 compares our method with previous 3D generation approaches on previous 3D–text and 3D–image similarity benchmarks Lai et al. (2026), including Craftsman 1.5 Li et al. (2024), Michelangelo Zhao et al. (2023), Trellis2/512 Xiang et al. (2026), Direct3D-S2 Wu et al. (2026), Hi3DGen Ye et al. (2025), and Hunyuan3D 2.0 Zhao et al. (2025). We evaluate the global similarity with ULIP Xue et al. (2023) and Uni3D Zhou et al. (2024) metrics. Results demonstrate that our Flow3D-OPD achieves consistent gains over the baseline Hunyuan3D-2.1 Lai et al. (2025) across all four metrics and outperforms all compared previous methods.

Table 3: Comparison with previous works on 3D–text (T) and 3D–image (I) similarity calculated by ULIP/Uni3D models. The best results are in **bold**.

Method	ULIP-T	ULIP-I	Uni3D-T	Uni3D-I
Craftsman 1.5	0.074	0.129	0.233	0.299
Michelangelo	0.075	0.115	0.206	0.261
Direct3D-s2	0.074	0.122	0.247	0.314
Hi3DGen	0.076	0.131	0.237	0.299
Trellis2	0.080	0.129	0.241	0.307
HY3D 2.0	0.077	0.130	0.245	0.315
HY3D 2.1	0.081	0.133	0.246	0.314
Flow3D-OPD (Ours)	**0.083**	**0.138**	**0.252**	**0.321**

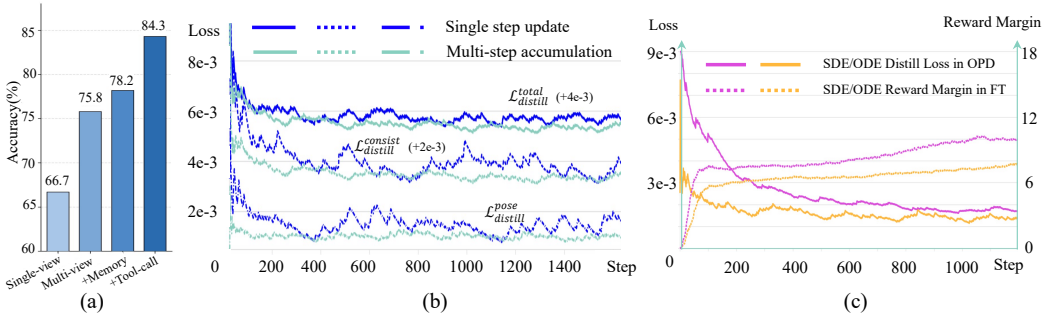

Figure 3: Ablation analysis on our proposed method. (a) Accuracy of the agentic verifier with various settings. (b) Distillation loss under different update strategies. (c) Reward margin and distillation loss with SDE or ODE Sampling.

Table 4: Ablation on different distillation strategies during OPD. Best results are in **bold**.

Distill Strategy	A-wins ↓	B-wins↑	Uni3D-I
Single-step + Seq.	7.1	59.8	0.319
Single-step + MTR	6.6	58.5	0.320
Multi-step + Seq.	6.1	60.4	0.319
Multi-step + MTR	**4.9**	**61.0**	**0.321**

Table 5: Ablation on SDE/ODE sampling during FT and OPD stages. Best results are in **bold**.

Stage	Strategy	A-wins ↓	B-wins↑	Uni3D-T	Uni3D-I
FT	ODE	6.4	51.7	0.245	0.314
	SDE	5.9	54.9	0.248	0.315
OPD	ODE	**4.9**	**61.0**	**0.252**	**0.321**
	SDE	7.3	54.8	0.251	0.319

5.3 Ablation Studies

Effect of the Agentic Verifier We first validate the accuracy of the agentic verifier against human expert annotations. As shown in Fig. 3 (a), the single-view verifier attains only 66.7% accuracy, which rises to 75.8% when multi-view inputs are provided. Further incorporating the case-memory and tool-call mechanisms improves the consistency to 84.3%. The monotonic improvement across components demonstrates the necessity of the complete pipeline in our reward model design.

Effect of Distillation Strategy We compare different distillation strategies during the OPD stage in Table 4. Single step and multiple step define the model update frequency, while 'Seq.' and 'MTR' represent the sequential sampling and mixed task-routing sampling. The results suggest that our configuration achieves the best performance across all metrics, while mixed task-routing sampling with multiple step accumulation is better than sequential sampling. Moreover, the loss curves in Fig. 3 (b) shows the training stability in our approach. Our strategy produces more stable distillation loss curves than the single-step backward variant, and also converges to a lower loss value across both total and per-task objectives.

Comparison on the SDE and ODE Sampling We compare the effectiveness of SDE and ODE sampling mechanisms across the FT and OPD stages in Table 5. In the FT stage, SDE sampling outperforms ODE across both global and specialized metrics. We attribute this to the additional stochasticity that generates more diverse on-policy losers and produces higher reward margins between winner and loser pairs, as demonstrated by the dashed curves in Fig. 3 (c). However, in the OPD stage, the deterministic ODE formulation achieves better results with faster loss convergence than SDE, as illustrated in the bottom of Table 5 and Fig. 3 (c). This is because the student model can align more precisely with the teacher's velocity field at each step without the interference introduced by the stochastic diffusion term.

5.4 Qualitative Visualization

Fig. 7 presents qualitative comparisons across various approaches or training strategies: Hi3DGen, Trellis2, Hunyuan3D-2.1, OPD-only (distillation without the FT stage), and our full pipeline Flow3D-OPD model. Previous models exhibit limited capability in generating fine-grained details, such as the geometric structures of buildings and the propellers of aircraft models. Our baseline model, Hunyuan3D-2.1, produces geometrically plausible but misaligned or structurally inconsistent meshes. OPD-only model yields moderate improvements yet lacks the stable foundation for

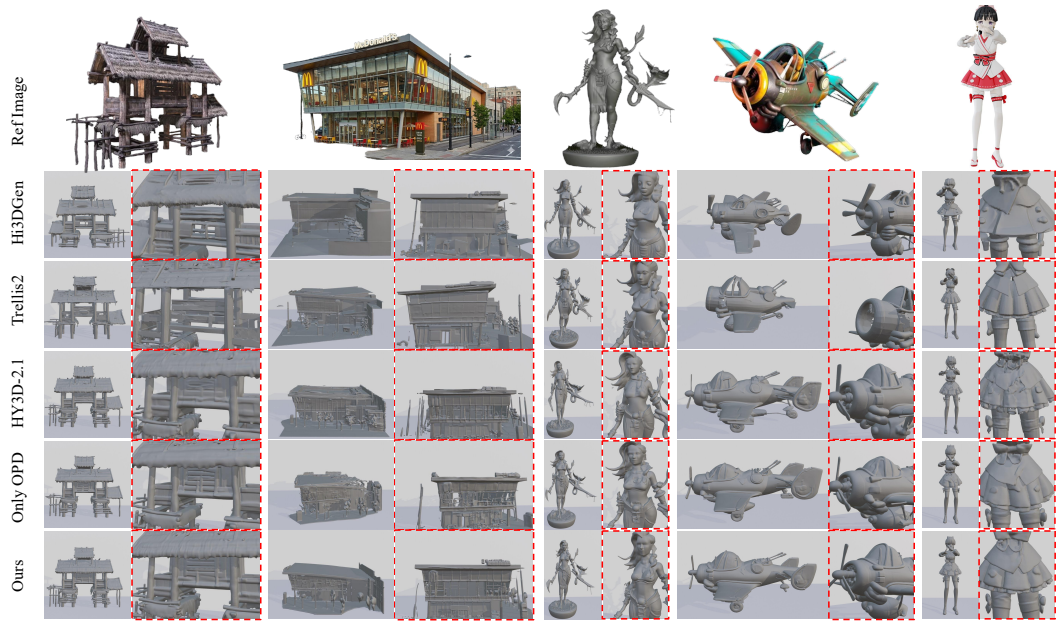

Figure 4: Qualitative comparison on 3D meshes generated with various approaches or training strategies. Our model produces 3D meshes with accurate and consistent structure.

consistent high-quality generation. Our full model produces meshes with more coherent and precise structure, proving the validity of basic capability scaling and multi-teacher on-policy distillation.

Fig. 5 further compares our distilled student model against the three individual teacher models on their respective areas of expertise. In the first row of orientation validation, the pose teacher and consistency teacher fail to align the mesh to the canonical front-facing direction, which further introduces asymmetric artifacts in the head region, while our student maintains correct orientation. In the second row of pose validation, the orient teacher and consistency teacher produce visible deformations at the elbows of the people, whereas our student preserves a faithful articulated pose though it is unseen in the reference image. In the third row, our student exhibits richer detail in the surface texture compared to the non-specialized teachers. These results confirm that the student model unifies the strengths of all three teachers while avoiding their individual weaknesses in other dimensions.

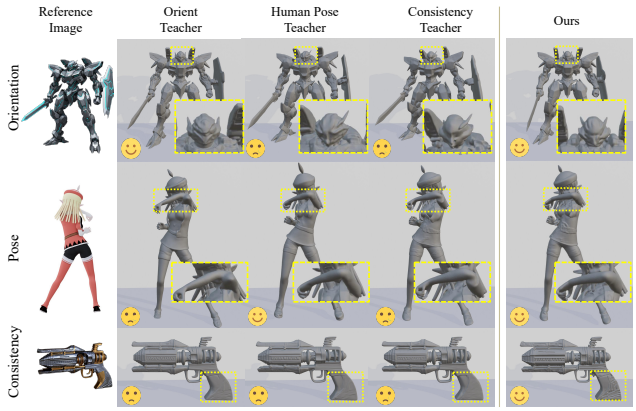

Figure 5: Visualization comparison between our model and individual teachers on orientation (top), pose (middle), and consistency (bottom) validation dimensions.

6 Conclusion

We present Flow3D-OPD, a two-stage post-training framework that first introduces multi-teacher on-policy distillation into 3D geometry generation. In the first stage, we design an agentic reward system for fine-grained geometric evaluation and train domain-specialized teachers via DPO across three quality dimensions. In the second stage, multiple teacher distillation with mixed divergence optimization enables the student to absorb heterogeneous expertise. Extensive experiments demonstrate that Flow3D-OPD achieves consistent improvements across all geometric dimensions and surpasses teachers on the averaged metric.

AI USE STATEMENT

In this work, we used generative AI tools to help polish the content of the paper and correct grammatical errors. All generated content was carefully checked by us to ensure that no hallucinations or errors were introduced.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

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.

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. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, et al. Flow-opd: On-policy distillation for flow matching models. *arXiv preprint arXiv:2605.08063*, 2026.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Google DeepMind. Gemini 3 flash. *Google AI Blog*, 2025.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

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

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

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

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19982–19992, 2026.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. Diffusionopd: A unified perspective of on-policy distillation in diffusion models. *arXiv preprint arXiv:2605.15055*, 2026.

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*, 2024.

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

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.

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

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

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. *Communications of the ACM*, 65(1):99–106, 2021.

OpenAI. Gpt-5.5. *OpenAI Blog*, 2026.

Long Ouyang, Jeffrey 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. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4172–4182, 2023.

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

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

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.

Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18582–18592, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, Yao Yao, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *Advances in Neural Information Processing Systems*, 38:170778–170804, 2026.

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

Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6229–6239, 2022.

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.

Fukun Yin, Xin Chen, Chi Zhang, Biao Jiang, Zibo Zhao, Wen Liu, Gang Yu, and Tao Chen. Shapegpt: 3d shape generation with a unified multi-modal language model. *IEEE Transactions on Multimedia*, 27:4107–4120, 2025.

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*, 36:73969–73982, 2023.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, 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. *arXiv preprint arXiv:2509.16117*, 2025.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations*, volume 2024, pp. 46766–46782, 2024.

SUPPLEMENTARY MATERIAL

.1 DETAILS OF THE TRAINING PIPELINE

.1.1 HYPERPARAMETER CONFIGURATION

Table 6 summarizes the hyperparameter configurations for all training stages. All stages are trained with 5K steps on 2×8 H20(90G) GPUS. The pipeline begins with semi-policy fine-tuning (stage 1a) on the offline ground-truth winners and on-policy generated losers. The resulting fine-tuned model serves as the initialization for both the domain-specialized teachers (stage 1b) and the final student (stage 2). Each teacher is trained via on-policy DPO with Gemini-3-Flash pairwise reward using an iterative rollout-then-train strategy to handle the low valid-pair ratio. In stage 2, the three frozen teachers are distilled into the student through on-policy trajectory sampling with mixed divergence optimization across the three data types.

Table 6: Hyperparameter configurations across all training stages.

Hyperparameter	Stage 1a: Semi-Policy FT	Stage 1b: On-Policy DPO	Stage 2: Multi-Teacher OPD
Base / Initialization	Hunyuan3D-2.1 (3B DiT)	Fine-tuned model (1a)	Fine-tuned model (1a)
DPO β	1000	1000	–
λ_{dpo} / λ_{distill}	0.2	1.0	1.0
DPO type	Semi-policy	On-policy (fully online)	–
Rollout sampler	SDE (a=0.3)	SDE (a=0.3)	ODE (deterministic)
Rollout NFE	50	50	50
Candidates per step	1 (loser only)	2 (pairs)	1
Confidence margin δ	–	0.167	–
Run mode	End-to-end	Iterative rollout/train(400/100)	End-to-end

.1.2 DATASET STATISTICS

Table 7 reports the dataset sizes and the description for each quality dimension. Specifically, in the training stage, we curate three separate preference datasets, one per dimension: an orientation set targeting canonical axis alignment, a pose set covering diverse articulated human configurations, and a consistency set emphasizing multi-view structural and textural coherence. In the evaluation stage, to isolate the model's generation ability along each aspect, we construct three dedicated test sets that mirror the three training dimensions.

Table 7: Dataset statistics across the three quality dimensions.

Dimension	Train Samples	Eval Samples
Orientation	3,661	133
Pose	4,083	111
Consistency	1,063	100

.2 AGENTIC VERIFIER DETAILS

In this section, we introduce the details of the agentic verifier used to construct preference pairs during DPO training, depicted in Fig. 6. The verifier is built on the Gemini-3-Flash proprietary model and follows a structured, tool-augmented reasoning process to compare two candidate meshes generated from the same reference image. Below we describe its system prompt, tool call, and case-memory mechanism.

.2.1 SYSTEM PROMPT

The reward agent receives a structured system prompt that defines its role, input context, dimension-specific instructions, and metrics. The key components are shown as follows.

Role and Input. The agent acts as a strict 3D geometry review agent that compares two candidate meshes (model *A* vs. model *B*) generated from the same reference image. It receives: (1) the reference image, (2) multi-view rendered images of two meshes.

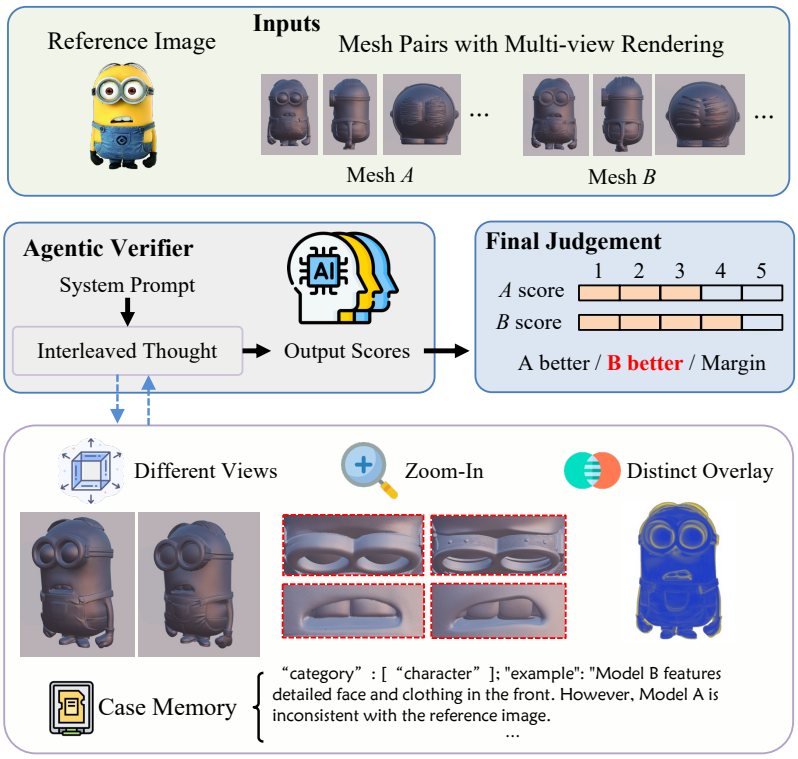

Figure 6: The detailed pipeline of our agentic verifier.

Orientation Prompt

```
You are a review agent focused on canonical orientation. The
generated 3D model must be aligned to the standard world axes, while
baking in the diagonal tilt, skewed perspective, or dynamic camera
angle of the 2D reference is strictly prohibited. For example, an
object with a dominant axis (e.g., a trumpet, a standing animal)
must have that axis horizontally or vertically aligned in the
orthographic views, not slanted. A bilaterally symmetric object
(e.g., a forward-facing human) must appear macroscopically symmetric
in the front view. Any retained diagonal slant or perspective baking
is a fatal defect and must be scored ``Bad''.
```

Pose Prompt

```
You are a review agent focused on human pose and body structure.
Evaluate, in descending priority: Macro limb structure -- exactly two
arms and two legs, each a single continuous limb, with no forking,
and with limbs/head smoothly connected to the torso (no floating or
severed parts). Pose accuracy -- limb angles, bend directions, and
global posture must match the reference; penalize mirrored poses,
collapsed poses, and residual T-pose. Overall shape and accessories
-- clothing, hats, and attachments should be consistent with the
reference.
```

Consistency Prompt

```
You are a senior 3D geometry review agent specialized in geometric
consistency. Your task is to evaluate how faithfully two 3D models
reproduce the geometric or textural structures visible in the
```

```
reference image (Image-to-3D consistency). You evaluate whether
the 3D model accurately reconstructs what is shown in the reference
image | silhouette match, structural element count, proportions,
and geometric details. You do NOT evaluate orientation alignment or
artistic style | only consistency with the reference.
```

Dimension-Specific Instructions. Based on the shared system prompt, we design a dedicated instruction for each of the three judge dimensions. In the orientation dimension, the agent enforces canonical axis alignment and treats perspective baking as a fatal defect. In the human-pose dimension, the agent prioritizes correct limb counts and connectivity, then joint angles and overall posture relative to the reference. In the consistency dimension, the agent checks that structures and texture inferred from one viewpoint remain plausible and coherent with the reference image. The three prompts are summarized below.

.2.2 Case Memory Mechanism

The agent maintains a case memory of previously judged hard examples to improve judgment accuracy. Before evaluating a new pair, the system retrieves similar past cases (based on object category) and includes them as few-shot examples in the context. This mechanism addresses the variance inherent in VLM-based evaluation by anchoring judgments to established precedents.

.2.3 Toolcall Definition

The agent has access to three tools during evaluation, described below.

New views render. This tool renders additional viewpoints of both models for closer inspection. The input obtains a list of three viewpoints, each specifying a `yaw` (0=Front, 90=Left, 180=Back, 270=Right), a `pitch` (0=eye-level, $\pm$30=tilted, $\pm$90 = top/bottom), and an optional render normal `flag`. It returns the rendered images at the specified viewpoints along with their corresponding difference maps.

Zoom in. This tool renders zoomed close-up views for fine-detail inspection. The input of the toolcall function contains a list of one to four zoom viewpoints, each specifying a `yaw`, a `pitch`, a `label`, and an optional `fov_deg` that controls the zoom level ($\sim$50° for standard, $\sim$30° for moderate zoom, $\sim$15° for strong zoom, and $\sim$8° for extreme close-up), together with a reason field describing what detail to inspect; it returns the zoomed rendered images at the requested viewpoint and zoom level.

Difference overlay. This tool outputs a heatmap that helps the agent localize discrepancies between the two meshes. It takes an `image_index` argument identifying the pair of rendered images to compare, and returns a difference heatmap of the two meshes rendered from the same viewpoint.

.3 Derivation of OPD in Flow-Matching

We provide the complete derivation of reverse-KL on-policy distillation (OPD) objective in Eq. 9. Let π_θ and π_{ϕ_m} denote the student and teacher policies. The denoising trajectory of the student policy is defined as $x_{0:N} = (x_{t_0}, \ldots, x_{t_N})$, where $1 = t_0 > t_1 > \cdots > t_N = 0$. Considering the denoising process is Markovian, we decompose the whole OPD objective into stepwise KL along the student rollout:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x_{0:N}\sim\pi_\theta}\left[\sum_{j=0}^{N-1} \mathrm{KL}\big(\pi_\theta(\cdot \mid x_{t_j}) \,\big\|\, \pi_{\phi_m}(\cdot \mid x_{t_j})\big)\right], \tag{11}$$

where $\pi_\theta(\cdot \mid x_{t_j})$ and $\pi_{\phi_m}(\cdot \mid x_{t_j})$ are the student and teacher one-step transition kernels at the same state x_{t_j}.

For flow-matching denoising with SDE sampling, we discretize the reverse time by Euler–Maruyama with step $\Delta t_j := t_{j+1} - t_j < 0$. Let $\sigma_t = a\sqrt{t/(1-t)}$ be the SDE diffusion coefficient

with global noise level a, and define $v_\theta(x_{t_j})$ and $v_{\phi_m}(x_{t_j})$ for the student and teacher velocity fields in t_j. The student SDE step is:

$$\begin{aligned} x_{t_{j+1}} &= x_{t_j} + \Big[v_\theta + \tfrac{\sigma_{t_j}^2}{2t_j}\big(x_{t_j} + (1-t_j)\, v_\theta\big)\Big]\Delta t_j + \sigma_{t_j}\sqrt{-\Delta t_j}\,\varepsilon_j \\ &= \Big(1 + \tfrac{\sigma_{t_j}^2}{2t_j}\Delta t_j\Big)x_{t_j} + \Big(1 + \tfrac{\sigma_{t_j}^2(1-t_j)}{2t_j}\Big)v_\theta\,\Delta t_j + \sigma_{t_j}\sqrt{-\Delta t_j}\,\varepsilon_j \end{aligned} \tag{12}$$

where $\varepsilon_j \sim \mathcal{N}(0, I_d)$, and $v_\theta := v_\theta(x_{t_j})$. Collecting the deterministic drift as the transition mean and the noise term as the covariance, the one-step kernel is the isotropic Gaussian:

$$\pi_\theta(\cdot \mid x_{t_j}) = \mathcal{N}\big(\mu_\theta(x_{t_j}),\ \bar\sigma_{t_j}^2 I_d\big), \quad \bar\sigma_{t_j}^2 := \sigma_{t_j}^2(-\Delta t_j), \tag{13}$$

where the mean is defined as:

$$\mu_\theta(x_{t_j}) = \Big(1 + \tfrac{\sigma_{t_j}^2}{2t_j}\Delta t_j\Big)x_{t_j} + \Big(1 + \tfrac{\sigma_{t_j}^2(1-t_j)}{2t_j}\Big)v_\theta\,\Delta t_j. \tag{14}$$

The transition kernel in the teacher policy π_{ϕ_m} is defined in a similar formulation:

$$\begin{aligned} \pi_{\phi_m}(\cdot \mid x_{t_j}) &= \mathcal{N}\big(\mu_{\phi_m}(x_{t_j}),\ \bar\sigma_{t_j}^2 I_d\big), \\ \mu_{\phi_m}(x_{t_j}) &= \Big(1 + \tfrac{\sigma_{t_j}^2}{2t_j}\Delta t_j\Big)x_{t_j} + \Big(1 + \tfrac{\sigma_{t_j}^2(1-t_j)}{2t_j}\Big)v_{\phi_m}\,\Delta t_j. \end{aligned} \tag{15}$$

Because the covariance $\bar\sigma_{t_j}^2 I_d$ is identical, π_θ and π_{ϕ_m} differ only in their means. For two d-dimensional Gaussians with common isotropic covariance $\sum$:

$$\mathrm{KL}\big(\mathcal{N}(\mu_1, \textstyle\sum) \,\|\, \mathcal{N}(\mu_2, \textstyle\sum)\big) = \tfrac{1}{2}(\mu_1 - \mu_2)^T \textstyle\sum^{-1}(\mu_1 - \mu_2). \tag{16}$$

Applying (16) to the kernels (13) and (15) gives the per-step KL as:

$$\mathrm{KL}\big(\pi_\theta(\cdot \mid x_{t_j}) \,\|\, \pi_{\phi_m}(\cdot \mid x_{t_j})\big) = \frac{\|\mu_\theta(x_{t_j}) - \mu_{\phi_m}(x_{t_j})\|_2^2}{2\,\bar\sigma_{t_j}^2}. \tag{17}$$

Subtracting $\mu_{\phi_m}(x_{t_j})$ from $\mu_\theta(x_{t_j})$, the state-dependent terms cancel, and only the velocity-dependent term remains:

$$\begin{aligned} &\mu_\theta(x_{t_j}) - \mu_{\phi_m}(x_{t_j}) = \\ &\quad \Big(1 + \tfrac{\sigma_{t_j}^2(1-t_j)}{2t_j}\Big)\Delta t_j\,\big(v_\theta(x_{t_j}) - v_{\phi_m}(x_{t_j})\big). \end{aligned} \tag{18}$$

Substituting (18) into (17), the per-step KL becomes a weighted squared velocity difference:

$$\begin{aligned} &\mathrm{KL}\big(\pi_\theta \,\|\, \pi_{\phi_m}\big) = w(t_j)\big\|v_\theta(x_{t_j}) - v_{\phi_m}(x_{t_j})\big\|_2^2, \\ &w(t_j) = \frac{(-\Delta t_j)}{2\,\sigma_{t_j}^2}\Big(1 + \tfrac{\sigma_{t_j}^2(1-t_j)}{2t_j}\Big)^2 = \frac{\Delta t_j}{2}\Big(\frac{\sigma_{t_j}(1-t_j)}{2t_j} + \frac{1}{\sigma_{t_j}}\Big)^2. \end{aligned} \tag{19}$$

Based on Eq. 19, we apply the stop-gradient operator $\mathrm{SG}(\cdot)$ to the frozen teacher. The final OPD objective is the time-weighted velocity-matching loss $\mathcal{L}_{\text{distill}}^m(\theta)$, defined as:

$$\mathbb{E}_{x_{0:N}\sim p_{S,\theta}}\left[\sum_{j=0}^{N-1} w(t_j)\,\big\|v_\theta(x_{t_j}) - \mathrm{SG}\big(v_{\phi_m}(x_{t_j})\big)\big\|_2^2\right]. \tag{20}$$

Deterministic (ODE) regime. Setting the noise level $a \to 0$ makes the transition deterministic: for a given x_{t_j}, the student and teacher each induce a single transition target. Distribution matching then reduces to pointwise matching of the predicted velocities, and the reverse-KL objective is replaced by a direct squared L_2 loss, which is formulated as:

$$\mathbb{E}_{x_{0:N}\sim p_{S,\theta}}\left[\frac{1}{N}\sum_{j=0}^{N-1}\big\|v_\theta(x_{t_j}) - \mathrm{SG}\big(v_{\phi_m}(x_{t_j})\big)\big\|_2^2\right], \tag{21}$$

which is the ODE instantiation used in our OPD stage. This closed-form, zero-variance objective is the flow-matching analogue of the per-step KL distillation used for autoregressive models, replacing high-variance policy gradients with direct backpropagation through the student velocity field.

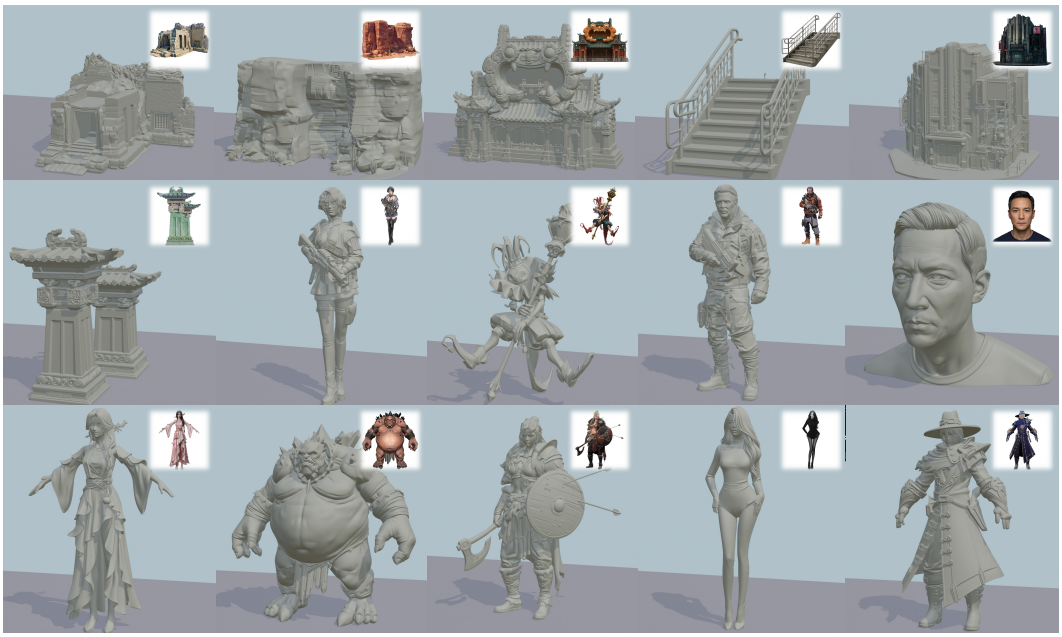

Figure 7: The quality visualization of the refinement in our Flow3D-OPD

Table 8: Per-dimension ablation on the CFG scale w used for the teacher during the OPD stage. Best results are in **bold**.

CFG Scale w	A-wins/B-wins		
	Orient-Eval	Pose-Eval	Consist-Eval
$w = 1.0$	**5.1/18.8**	**0.0/93.5**	**10.0/81.0**
$w = 3.0$	7.1/17.6	1.0/91.9	15.7/77.5
$w = 6.0$	7.1/17.9	0.0/88.7	14.0/75.3

Table 9: Ablation on the CFG scale w during inference after OPD in the student model. Best results are in **bold**.

CFG Scale w	A-wins ↓	B-wins	ULIP-I	Uni3D-I
$w = 1.0$	32.3	38.5	0.133	0.314
$w = 3.0$	**4.9**	**61.0**	**0.138**	**0.321**
$w = 6.0$	5.2	60.6	0.137	0.320

.4 Additional Experiments

.4.1 Impact of the CFG Weights

We further study the effect of the CFG weight w in both the training and the inference stages, as reported in Table 8 and Table 9. During the OPD stage, we find that using the velocity field predicted under the default conditional signal (i.e., $w = 1.0$, without classifier-free amplification) as the distillation target yields the best student, indicating that the raw conditional guidance already provides a sufficiently informative supervision signal. At inference time, the model performs best under the default CFG configuration used in Hunyuan3D-2.1 ($w = 3.0$), which balances conditional fidelity and generation quality.

.4.2 More Quality Visualization

In this section, we provide additional quality visualizations, as shown in Fig. 7. These cases are based on meshes generated by our Flow3D-OPD and further post-processed with LATTICE for high-resolution refinement.