Nabla-R2D3: Effective and Efficient 3D Diffusion Alignment with 2D Rewards

Qingming Liu[1,*] **Zhen Liu**[1,*,†] **Dinghuai Zhang**[2] **Kui Jia**[1]
[1]The Chinese University of Hong Kong, Shenzhen [2]Microsoft Research
[*]Equal contribution [†]Corresponding author

Project page: nabla-r2d3.github.io

Figure 1: Our **Nabla-R2D3** can efficiently and robustly finetune 3D-native diffusion models with differentiable reward models learned from human preferences on appearance, geometry and many other attributes.

Abstract

Generating high-quality and photorealistic 3D assets remains a longstanding challenge in 3D vision and computer graphics. Although state-of-the-art generative models, such as diffusion models, have made significant progress in 3D generation, they often fall short of human-designed content due to limited ability to follow instructions, align with human preferences, or produce realistic textures, geometries, and physical attributes. In this paper, we introduce **Nabla-R2D3**, a highly effective and sample-efficient reinforcement learning alignment framework for 3D-native diffusion models using 2D rewards. Built upon the recently proposed Nabla-GFlowNet method, which matches the score function to reward gradients in a principled manner for reward finetuning, our **Nabla-R2D3** enables effective adaptation of 3D diffusion models using only 2D reward signals. Extensive experiments show that, unlike vanilla finetuning baselines which either struggle to converge or suffer from reward hacking, **Nabla-R2D3** consistently achieves higher rewards and reduced prior forgetting within a few finetuning steps.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Recent advances in 3D generative models have enabled non-experts to produce batches of 3D digital assets at low cost for downstream tasks such as gaming, film, and robotics simulation. However, these assets are rarely production-ready and often require post-processing due to low visual fidelity, suboptimal geometry, ethical biases, and poor instruction following (in text-to-3D setups). These issues largely stem from training datasets that diverge from human preferences and 3D constraints.

A common solution is to first obtain a reward model that reflects human preferences and later perform reward finetuning on the generative model—a process commonly known as reinforcement learning from human feedback (RLHF). Originally developed for aligning autoregressive language models, reward finetuning has also been successfully applied to diffusion models, which are among the most popular generative models in the vision domain. It is therefore natural to consider applying similar techniques to 3D diffusion models.

However, extending RLHF to 3D diffusion models is non-trivial due to the lack of high-quality 3D reward models. In 2D settings, reward finetuning typically relies on 2D reward models; by analogy, 3D finetuning would require 3D reward models. Yet collecting diverse and high-quality 3D data remains a longstanding challenge, making it difficult to build reward models that reflect human preferences on attributes such as aesthetics, geometry, instruction following, and physical plausibility.

Inspired by the success of the "lifting from 2D" approach for 3D generation, where one optimizes a 3D shape such that each view is of high likelihood under a pretrained 2D generative model, we propose to similarly finetune 3D-native diffusion models [46, 61, 38, 23, 24, 44] with 2D reward models. Using a 3D-native model, we can sample different camera views and perform the lifting operation in an amortized fashion across training instances, rather than optimizing each object individually. However, sampling from 2D views yields high variance during optimization and can lead to instability and overfitting of 3D-native diffusion models. In light of a state-of-the-art reward finetuning method called Nabla-GFlowNet [25], which proves to be highly robust, efficient and effective on 2D diffusion models, we propose **Nabla-R2D3** (short for **R**eward from **2**D for **D**iffusion Alignment in **3**D via **Nabla**-GFlowNet), which adapts this method to finetune 3D-native diffusion models with 2D rewards. We empirically show that our method produces 3D-native models that are better aligned with human preferences and avoid major artifacts, such as unexpected floaters, commonly produced by prior methods. Furthermore, we demonstrate the effectiveness of different 2D reward models for aligning 3D-native diffusion models for different preferences.

We summarize our major contributions in this paper below:

- To the best of our knowledge, **Nabla-R2D3** is the first method to effectively and robustly align 3D-native diffusion models with human preferences using only 2D reward models.
- We demonstrate several examples of 2D reward models, including appearance-based and geometry-based ones, for aligning 3D-native generative models on different attributes.
- Our extensive experiments show that, compared to the proposed vanilla reward finetuning baselines, our **Nabla-R2D3** can effectively, efficiently and robustly finetune 3D-native generative models from 2D reward models with better preference alignment, better text-object alignment and fewer geometric artifacts.

2 Related Work

Reward Finetuning of Diffusion Models. The earliest attempt at reward finetuning for diffusion models, named DDPO [2], views the denoising process of diffusion models as trajectory sampling in a Markov decision process (MDP), in which each state is a tuple of a noisy image and the corresponding time step; a sampled trajectory starts from a random Gaussian noise at time T and, through the iterative stochastic denoising process, reaches a sample image at time 0. With this MDP defined, DDPO leverages the classical policy gradient method in reinforcement learning (RL) to finetune diffusion models. As reward models are typically learned with neural nets and thus differentiable, DDPO does not effectively leverage the available first-order information in the reward model. To address this issue, methods like ReFL [48] and DRaFT [4] treat each sample trajectory as a deep computational graph and, by assigning reward values to the states, use back-propagation to directly optimize model parameters with respect to reward signals in an end-to-end fashion. While efficient in practice, these

methods lack probabilistic grounding—their training objectives are not designed to approximate the reward-weighted distribution—and thus tend to overfit to the reward model. A parallel path is to adopt stochastic optimal control (SOC) that frames the alignment objective as an optimal control problem. The SOC approach in theory may achieve ideal results, but the proposed methods [42] so far are either ineffective or computationally expensive. Recently, new RL-based diffusion finetuning methods are proposed in the framework of generative flow networks [1, 58, 55, 27, 30, 57, 56, 59, 52] (or GFlowNets in short) that builds generative models on a directed acyclic graph to generate samples according to a reward distribution. The resulting finetuning methods, albeit derived in GFlowNet language, are also deeply rooted in (and in many cases equivalent to) soft Q-learning [9] in reinforcement learning. These new RL methods are constructed from first principles and are shown to be effective and efficient in creating finetuned diffusion models that generate diverse samples.

3D Generation via "Lifting from 2D". Due to the scarcity of 3D data, several works propose generating 3D shapes with only 2D signals. One early work in this direction is Dream Field [14], which initializes a 3D shape and optimizes the shape with CLIP scores on images rendered from randomly sampled camera views of the shape. Since the CLIP model is not a generative model, it is hard, if not impossible, to yield detailed appearance and geometry in generated 3D shapes. Following Dream Field, DreamFusion [31] was proposed to use score distillation sampling (SDS) that replaces CLIP scores with diffusion losses on rendered views. Such an approach is used not only for 3D object synthesis from scratch with 2D models, but also for texture generation on known geometry [36], 3D object synthesis with video generative models [15] and so on. However, the lifting approach is highly unstable and requires extensive hyperparameter tuning [31]. Another issue is the famous Janus problem that implausible 3D objects may be generated even if most of the 2D views are reasonable—demonstrating the issue of the lack of 3D priors. In addition, these lifting-based 3D generation methods take a long time to sample only one single object and are less suitable for downstream tasks due to high computational cost. Our proposed method is free from these issues because it directly works on and infers from 3D-native diffusion models.

Alignment in 3D Generation. Apart from the per-sample alignment with SDS-based "lifting from 2D" approaches [51, 63, 13], probably the methods most relevant to ours are MVReward [43] and Carve3D [47], both of which finetune a multi-view diffusion model with a separate multi-view-based reward model. These methods assume a multi-view representation of 3D objects, a representation that does not guarantee 3D consistency. Moreover, since this representation does not encode ground-truth normal or depth maps, external estimators must be employed to improve geometry using geometric rewards. Another line of 3D alignment is through direct preference optimization (DPO) [35, 63], where a model is finetuned on a preference dataset and without any explicit reward model. While DPO is conceptually simple, alignment with a reward model is generally better [29] and applies to scenarios where we have analytical and/or expert-designed reward models. There is a recent trend of post-training alignment with test-time scaling [16, 28] to filter undesired samples or dynamically adjust sampling strategy during inference. Such a strategy is applied to 3D generation [7], but it is typically costly and do not leverage the gradient information in reward models.

3 Preliminaries

3.1 Diffusion models and RL-based finetuning

Diffusion models are a powerful class of generative models that generate samples through sequential denoising process. To be specific, it typically starts from time T with a point x_T sampled from a standard Gaussian distribution, and gradually generate cleaner samples through a learned backward process $p(x_{t-1}|x_t)$ until it obtains the final sample x_0. The backward process $p(x_{t-1}|x_t)$ is trained to match a forward process $q(x_t|x_{t-1})$, typically set as a simple Gaussian distribution. For instance, in a popular diffusion model DDPM [11], the corresponding noising process is $q(x_{t+1}|x_t) = \mathcal{N}(\sqrt{\alpha_{t+1}/\alpha_t}x_t, \sqrt{1-\alpha_{t+1}/\alpha_t}I)$ and $q(x_t|x_0) = \mathcal{N}(\sqrt{\alpha_t}x_t, \sqrt{1-\alpha_t}I)$ with a noise schedule $\{\alpha_t\}_t$. To train a DDPM model on a dataset $\mathcal{D}$, we use the score matching loss:

$$\mathbb{E}_{t\sim\text{Uniform}(\{1,\dots,T\}),\epsilon\sim\mathcal{N}(0,I),x_T\sim\mathcal{D}}\ w(t)\|\epsilon_\theta(\sqrt{\alpha_t}x_T + \sqrt{1-\alpha_t}\epsilon, t) - \epsilon\|^2, \tag{1}$$

where $w(t)$ is a weighting scalar, and $\epsilon_\theta(x_t, t)$ is a neural net that predicts the noise vector ϵ from x_t. The stochastic sequential denoising process can be treated as a Markov decision process (MDP) in which (x_t, t) is a state, (x_T, T) is the initial state, $(x_0, 0)$ is the final state, $p(x_t|x_{t+1})$ is the transition function. With such an MDP defined, one can align the underlying diffusion model with any reinforcement learning algorithm [2] and optimize some terminal reward $R(x_0)$.

3.2 Diffusion alignment via gradient-informed RL finetuning

To preserve prior in the pretrained model $p_{\text{base}}(x_t|x_{t+1})$, a typical finetuning objective is to match the "tilted" reward distribution $p_{\text{base}}(x)R^\beta(x)$ where β controls the amount of prior information in the finetuned model. With the MDP defined in Sec. 3.1, it is shown that we may collect on-policy trajectories $\{(x_T, ..., x_0)_k\}_{k=1}^K$ from the finetuned model $p_\theta(x_t|x_{t+1})$ (where K is the batch size) and optimize some RL objective. A recent method called Nabla-GFlowNet shows that we may efficiently and robustly finetune a diffusion model with "score-matching-like" consistency losses:

$$\mathcal{L}_{\text{forward}}(x_{t-1:t}) = \left\|\nabla_{x_{t-1}} \log \tilde{p}_\theta(x_{t-1}|x_t) - \gamma_t\beta\nabla\left[\nabla_{x_{t-1}} \log R(\hat{x}_\theta(x_{t-1}))\right] - g_\phi(x_{t-1})\right\|^2, \quad (2)$$

$$\mathcal{L}_{\text{reverse}}(x_{t-1:t}) = \left\|\nabla_{x_t} \log \tilde{p}_\theta(x_{t-1}|x_t) + \gamma_t\beta\nabla\left[\nabla_{x_t} \log R(\hat{x}_\theta(x_t))\right] + g_\phi(x_t)\right\|^2, \quad (3)$$

and the terminal loss $\mathcal{L}_{\text{terminal}}(x_0) = \|g_\phi(x_0)\|^2$, where $\log \tilde{p}_\theta = \log p_\theta - \log p_{\text{base}}$ represents the log-density ratio between the finetuned and base models., $\hat{x}_\theta(x_t) = (x_t - \sigma_t\epsilon_\theta(x_t))/\alpha_t$ is the expected one-step prediction of x_0 (i.e., $\mathbb{E}[x_0 \mid x_t]$) under the finetuned model, $\epsilon_\theta(x_t)$ is the noise prediction network of the finetuned model (α_t and σ_t denote the signal and noise scales from the forward process). ∇ is the stop-gradient operation, β controls the relative strength of the reward with respect to the prior of the pretrained model and γ_t is the decay factor of the guessed gradient.

The total loss follows (where w_B is a non-negative weighting scalar):

$$\mathcal{L}_{\text{total}} = \mathbb{E}_{x_T\sim\mathcal{N}(0,I),(x_{T-1},...,x_0)\sim p_\theta(x_t|x_{t+1})}\left[\mathcal{L}_{\text{terminal}} + \sum_t(\mathcal{L}_{\text{forward}} + w_B\mathcal{L}_{\text{reverse}})\right]. \quad (4)$$

This loss is originally derived within the framework of GFlowNet—a generative model with the MDP defined on a directed acyclic graph in which the terminal states are sampled with probability proportional to the corresponding rewards. It is shown [25] that this special set of losses is indeed equivalent to a gradient version of the soft Q-learning loss [9] in the reinforcement learning literature.

4 Method

4.1 Efficient and Robust 3D Diffusion Alignment with 2D Rewards

Suppose that we have a 2D reward model $R(x_0)$ that maps some image x_0 to the corresponding reward. We adopt a common assumption in the text-to-3D literature: the 3D reward model can be derived from a 2D reward model via multi-view rendering.

$$\log R_{\text{3D}}(z_0) = \mathop{\mathbb{E}}_{c\sim\mathcal{C}} \log R(h(z_0, c)), \quad (5)$$

where z_0 is a clean 3D shape, $h(z_0, c)$ is the rendering function that maps z_0 to the image with camera pose c sampled from a set of camera poses $\mathcal{C}$. Notice that if $R(\cdot)$ is approximated with the negative diffusion loss with a pretrained 2D diffusion model, the 3D reward is basically the score distillation sampling (SDS) objective in DreamFusion [31] (with slight differences in implementation):

$$\log R_{\text{3D, SDS}}(z_0) = -\mathop{\mathbb{E}}_{\epsilon\sim\mathcal{N}(0,I),t\sim\text{Uniform}(\{1,2,...,T\}),c\sim\mathcal{C},x_t=\alpha_t h(z_0,c)+\sigma_t\epsilon} w_t\|\epsilon_{\text{2D}}(x_t) - \epsilon\|^2. \quad (6)$$

where ϵ_{2D} is the ϵ-prediction network of the 2D diffusion model and w_t is some weight factor.

Based on the above assumption, we derive the following ∇-DB losses for finetuning 3D diffusion models on 2D reward models (with the terminal loss staying the same as $L_{\text{terminal}}(z_0) = \|g_\phi(z_0)\|^2$):

$$L_{\text{forward}}(z_{t-1:t}) = \mathop{\mathbb{E}}_{c\sim\mathcal{C}}\left\|\nabla_{z_{t-1}} \log \tilde{p}_\theta(z_{t-1}|z_t) - \gamma_t\beta\nabla\left[\nabla_{z_{t-1}} \log R(h(\hat{z}_\theta(z_{t-1}), c))\right] - g_\phi(z_{t-1})\right\|^2, \quad (7)$$

$$L_{\text{reverse}}(z_{t-1:t}) = \mathop{\mathbb{E}}_{c\sim\mathcal{C}}\left\|\nabla_{z_t} \log \tilde{p}_\theta(z_{t-1}|z_t) + \gamma_t\beta\nabla\left[\nabla_{z_t} \log R(h(\hat{z}_\theta(z_t), c))\right] + g_\phi(z_t)\right\|^2. \quad (8)$$

Empirically, we find that the following approximate loss alone works well:

$$L_{\text{approx}}(z_{t-1:t}) = \mathop{\mathbb{E}}_{c\sim\mathcal{C}}\left\|\nabla_{z_{t-1}} \log \tilde{p}_\theta(z_{t-1}|z_t) - \gamma_t\beta\nabla\left[\nabla_{z_{t-1}} \log R(h(\hat{z}_\theta(z_{t-1}), c))\right]\right\|^2, \quad (9)$$

with which we assume that our educated guess of the "reward gradient" is accurate and therefore there is no need to learn the correction term g_ϕ with the reverse-direction loss anymore. We use this simple loss throughout the rest of the paper.

4.2 Practical 2D Reward Models for 3D Native Diffusion Models

2D rewards from appearance. It has been shown that RGB appearance in 2D views alone can support high-quality 3D generation, as demonstrated by lifting-from-2D methods [31]. We follow recent methods in SDS-based reward-guided 3D generation and consider the following reward models for finetuning 3D-native diffusion models: 1) *Aesthetic Score* [17], trained on the LAION-Aesthetic dataset [17] to measure the aesthetics of images, and 2) *HPSv2* [45], trained on HPDv2 dataset [45] to measure general human preferences over image quality and text-image alignment.

2D rewards from geometry. In many cases, multiview RGB information can still fall short in creating fine-grained and 3D-consistent geometric details due to the lack of sufficient 3D priors or regularization. Inspired by the recent progress in single-view depth and normal estimation, we propose to explicitly encourage the consistency between rendered RGB images and the 3D geometric predictions inferred from those images. In our experiments, we employ state-of-the-art normal map estimators and take as the reward the inner product between rendered normal maps (with approximate volumetric rendering for Gaussian splatting and NeRF representations) and the normal maps predicted from rendered RGB images:

$$R_{\text{normal}}(z) = \mathop{\mathbb{E}}_{c\sim\mathcal{C}}\left[\left\langle h_{\text{normal}}(z,c), f_{\text{normal}}\big(h_{\text{RGB}}(z,c)\big)\right\rangle\right] \tag{10}$$

where $h_{\text{normal}}(z,c)$ and $h_{\text{RGB}}(z,c)$ denote the rendered normal map and rendered RGB images, respectively, from some 3D representation z at camera pose c in some camera pose set $\mathcal{C}$ and f_{normal} is some pretrained normal map estimator, for which we use the one-step normal map prediction model in StableNormal [50] in our experiments.

Inspire by [53] which proposes to use depth-normal consistency to improve the geometric quality of reconstruction, we also propose to use the Depth-Normal Consistency (DNC) reward:

$$R_{\text{DNC}}(z) = \mathop{\mathbb{E}}_{c\sim\mathcal{C}}\left[\left\langle h_{\text{normal}}(z,c), T\big(h_{\text{depth}}(z,c)\big)\right\rangle\right] \tag{11}$$

where the pseudo normal map $T\big(h_{\text{depth}}(z,c)\big)$ can be computed by taking finite difference of 3D coordinates computed from the rendered depth map.

5 Experiments

5.1 Base model, baseline, metrics and prompt dataset

Base models. We consider two state-of-the-art and open-sourced base models: DiffSplat [18], available in two variants finetuned from PixArt-Σ [3] and StableDiffusion-v1.5 [37] (SD1.5 for short), and GaussianCube [54]. We use 20-step and 50-step first-order SDE-DPM-Solver [26] for DiffSplat-Pixart-Σ and GaussianCube, respectively, and 20-step DDIM [39] for DiffSplat-StableDiffusion1.5. Unless otherwise specified, we use the DiffSplat-PixArt-Σ [3] for experiments.

Baseline methods. With the 3D reward defined as an expectation of 2D rewards, other alignment methods can be similarly applied once we use stochastic samples to compute 3D rewards. Specifically, we consider these baseline methods: 1) DDPO [2], which finetunes diffusion models with the vanilla policy gradient method, 2) ReFL [48], which directly optimizes $R(\hat{z}_\theta(z_t))$ with a truncated computational graph $z_{t+1} \to z_t \to \hat{z}_\theta(z_t)$ and a randomly sampled t, and 3) DRaFT [4] , which directly optimizes $R(z_0)$ with a truncated computational graph $z_K \to z_{K-1} \to \ldots \to z_0$. For ReFL, we sample t from 15 to 19; for DRaFT, we use $K = 1$.

Evaluation metric. Following alignment studies in 2D domains [25, 60, 6], we consider three metrics: 1) average reward value, 2) multi-view FID score [10, 8] for measuring prior preservation and 3) multi-view CLIP similarity score [34] for measuring text-object alignment. The multi-view metrics for any given text prompt are computed by first computing metrics on each view and then taking the average over all views. Similarly, we compute metrics for each prompt and average them to obtain the final multi-view metrics. We use 60 unseen random prompts during the finetuning process for evaluation. For each prompt, we sample a batch of 3D assets (of size 32) to compute the metrics.

Prompt dataset. We use the prompt sets in G-Objaverse [32], a high-quality subset of the large 3D object dataset Objaverse [5]. For experiments on geometry rewards, we filter out the prompts for which the base models yield very low reward values.

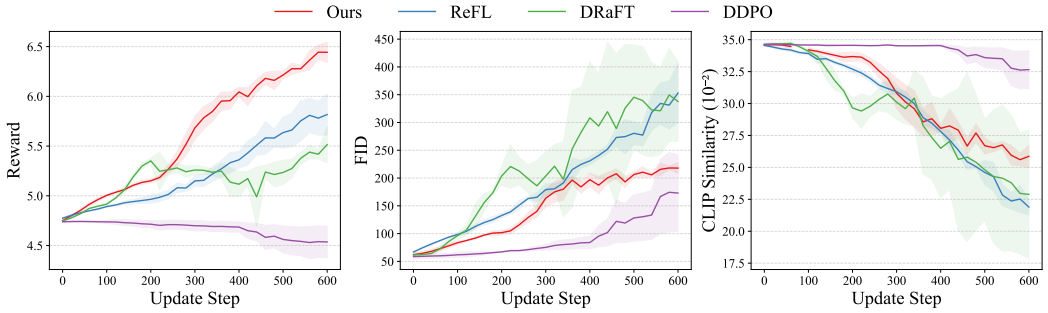

Figure 2: Convergence curves of metrics for different finetuning methods on Aesthetic Score. Our method achieves faster finetuning with better prior preservation and text-object alignment than ReFL and DRaFT. In addition, our method produces results with significantly lower variance in FID and CLIP similarity.

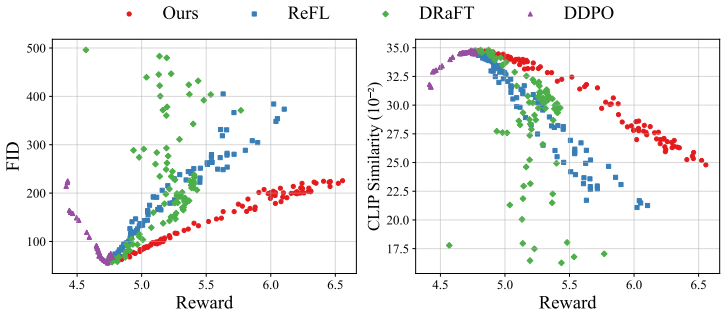

Figure 3: Trade-offs among reward maximization, prior preservation, and text-object alignment for various reward finetuning methods on Aesthetic Score experiments. Each data point represents evaluation results of model checkpoints saved at every 20 finetuning iterations. Models with higher rewards, higher CLIP similarity scores, and lower FID scores are considered superior. Our **Nabla-R2D3** shows better trade-offs between reward-improvement, and other metrics and consistently outperforms the baselines.

Implementation Details. We follow [25] and regularize the gradient updates: $\mathcal{L}_{\text{reg}} = \lambda \left\| \epsilon_\theta \left(x_t \right) - \epsilon_{\theta^\dagger} \left(x_t \right) \right\|^2$, where $\theta^\dagger$ is the diffusion model parameters in the previous update step and λ is a positive scalar. We set λ to $3e3$, $5e3$, $1e4$ for Aesthetic Score, HPSv2 and Geometry Reward respectively. During training, we sub-sample 40% of the transitions from each collected trajectory. For HPSv2 and Aesthetic Score experiments, we set the reward temperature β to $2e6$ and $1e7$, respectively; for geometry rewards, we set β to 1e6. To sample camera views c, we first sample four orthogonal views (front, left, back and right) with randomly sampled elevation $\pm 20°$ and then apply azimuthal perturbations by adding random offsets within a predefined range $\pm 60°$. We use a learning rate of 10^{-4} for ReFL, DDPO, DRaFT and **Nabla-R2D3**. The rest of the implementation details are elaborated in the appendix.

5.2 Results

General experiments. In Tab. 1, we show the metrics (average over 3 random runs) of models finetuned on different reward models with different finetuning methods. Our **Nabla-R2D3** is shown to be capable of achieving the best reward value at the fastest speed, and at the same time preserving the prior from the base model plus text-object alignment. We show in Fig. 2 the evolution of different metrics (both the mean value and the standard deviation) for different methods. As higher rewards inevitably lead to worse prior preservation and text-object alignment, we illustrate which method achieves the best trade-off by presenting the Pareto frontiers of all methods. Specifically, we plot the results from different checkpoints saved at various fine-tuning iterations of each independent run (with different random seeds) in Fig. 3. Furthermore, in Fig. 4, 5 and 6, we visualize the generated assets with the corresponding reward values from different finetuning methods and demonstrate that our method can qualitatively yield better alignment for various reward models meaningful for 3D generation. To illustrate that our method leads to more robust finetuning, we show in Fig. 7 the assets generated using the same random seeds with models finetuned with **Nabla-R2D3** and DRaFT. The shape with the DRaFT-finetuned model exhibits the severe Janus problem, where rendered multi-view

Figure 4: Qualitative comparison of 3D assets produced by models finetuned with different methods on Aesthetic Score. We show for each method on the left the average reward of the visualized assets. For fair comparison, we pick the model checkpoints that generate the highest rewards but without significant overfitting patterns.

images display inconsistent or contradictory characteristics from different viewpoints, while the one from **Nabla-R2D3**-finetuned model does not.

Table 1: Quantitative comparison between our method and the baselines on different reward models. Since it would be possible to trade FID with reward, we further present what the rewards are if FIDs are similar (with early stopping) in Tab. 5.

Method	Aesthetic Score			HPSv2			Normal Estimator		
	Reward↑	FID↓	CLIP-Sim↑(10^{-2})	Reward↑(10^{-2})	FID↓	CLIP-Sim↑(10^{-2})	Reward↑(10^{-2})	FID↓	CLIP-Sim ↑(10^{-2})
Base Model	4.72	55.26	34.58	22.86	55.26	34.58	89.48	55.26	34.58
ReFL	5.82	352.97	21.89	24.92	274.44	32.40	90.60	112.06	33.99
DDPO	4.54	**172.95**	**32.64**	22.23	**69.59**	34.35	89.45	**63.93**	**34.56**
DRaFT	5.51	337.77	22.89	**32.65**	224.64	33.99	91.11	296.23	28.36
Ours	**6.44**	217.89	25.86	27.85	131.38	**35.35**	**92.03**	104.45	34.18

Figure 8: Finetuning results on two base models, DiffSplat-SD1.5 [18] and GaussianCube [54]. Our method generalizes well to models beyond DiffSpliat-PixArt-Σ.

Different 3D-native generative models. We experiment with different base models, including DiffSplat-SD1.5 [18] and GaussianCube [54], to show that our method is universally applicable. Our

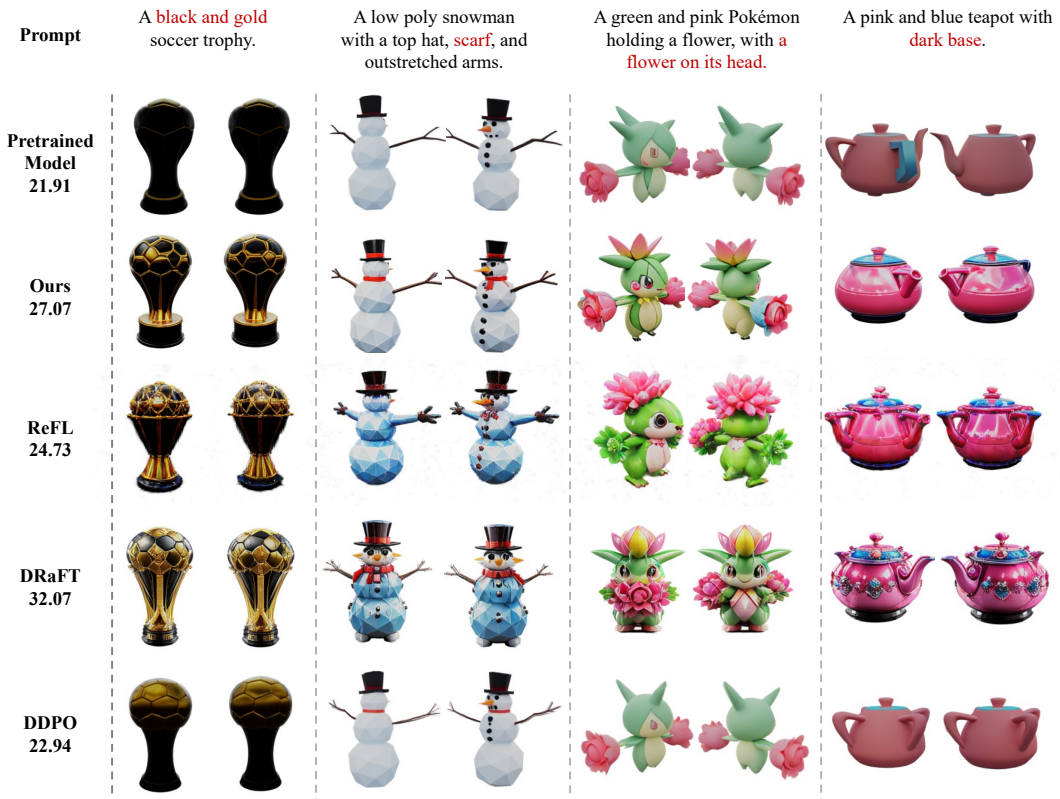

Figure 5: Quantitative comparison on HPSv2 [45]. For each object we present the front and back views. Prompts highlighted in red indicate unsuccessful instruction following by the base models. We further show the severe Janus problem of DRaFT in Fig. 7.

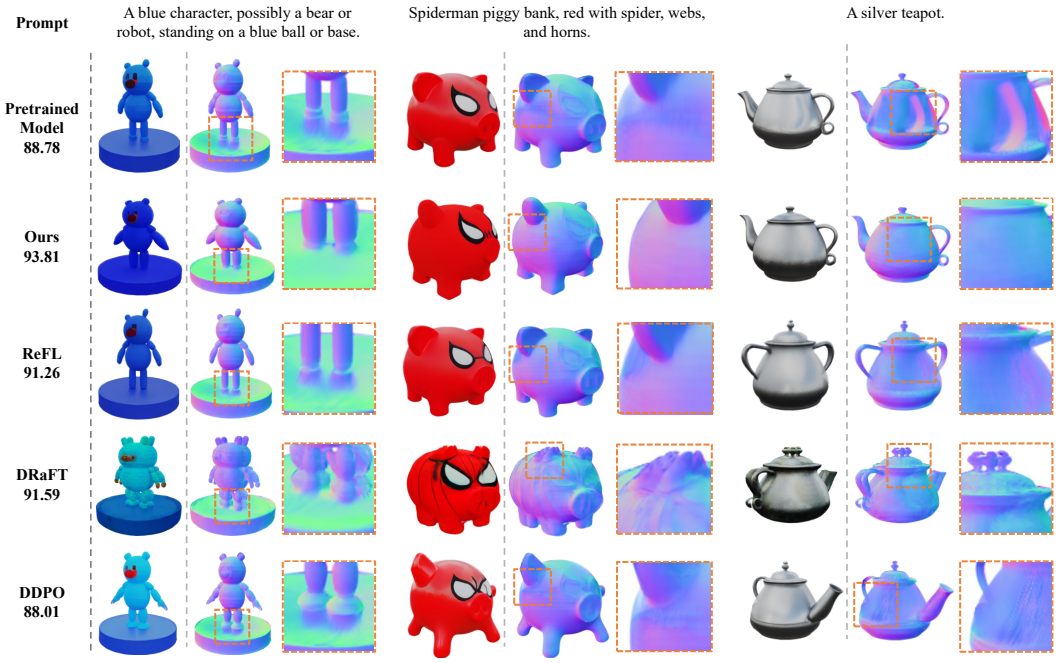

Figure 6: Quantitative comparison of different finetuning methods on the normal estimator reward (Eqn. 10). Left: front-view RGB image. Middle: front-view rendered normal map. Right: Zoomed-in details.

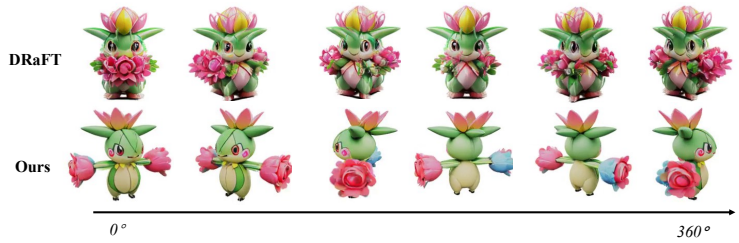

Figure 7: 360° visualization of 3D shapes generated by models finetuned with **Nabla-R2D3** and DRaFT. DRaFT-finetuned model is prone to overfitting and suffers from the Janus problem.

Table 2: Comparison of finetuning methods on different base models with the Aesthetic Score reward.

Method	DiffSplat-SD1.5			GaussianCube		
	Reward ↑	**FID**↓	**CLIP-Sim** ↑(10^{-2})	**Reward** ↑	**FID** ↓	**CLIP-Sim** ↑ (10^{-2})
Base Model	4.81	67.76	34.66	4.40	64.22	28.60
ReFL	6.08	396.79	20.66	5.50	296.46	17.31
DDPO	4.80	**54.79**	**34.61**	4.06	230.48	18.90
DRaFT	**6.40**	395.84	19.08	**6.04**	352.16	15.52
Ours	6.02	154.61	30.95	5.92	**234.96**	**22.57**

results (Fig. 8 and Tab. 2) show that our model consistently outperforms other finetuning methods and delivers desirable 3D assets.

Comparison with 2D-SDS-based lifting alignment method. We compare in Fig. 10 our method with DreamReward [51], a method that incorporates reward gradients into 2D-lifting-based 3D generation. Our method produces more visually-desirable shapes not only because 2D-lifting methods are less robust at synthesizing 3D shapes [31, 44] but also because the 3D-native generative model provides more 3D prior.

Table 3: Comparison with 3D-SDS.

Method	Reward ↑	FID ↓	CLIP-Sim ↑
3D-SDS ($\eta = 3$)	5.38	194.98	0.31
3D-SDS ($\eta = 1$)	5.27	**97.99**	**0.34**
Ours (200 steps)	5.29	114.45	0.32
Ours (600 steps)	**6.24**	205.16	0.26

Comparison with native 3D prior guided SDS baseline. To underscore the benefit of inference with a finetuned 3D-native diffusion model compared to the SDS-based sampling approaches, we experiment with SDS on the DiffSplat base model with 3D diffusion loss plus the multi-view 2D reward model of Aesthetic Score: $\nabla L = \epsilon_{3D}(z_t, t) - \epsilon - \eta \nabla_{z_0} \log R_{3D}(z_0)$ with reward strength η. We observe (Fig. 9) that the SDS approach indeed yields worse appearance and geometry, even with better 3D prior from the base 3D-native diffusion model and increasing the strength η does not improve results (Tab. 3). Moreover, running the 3D-SDS inference takes around 5 minutes, wherear inference with the finetuned model takes only around 8 seconds.

Figure 9: Comparison with 3D-SDS baselines on Aesthetic Score. We show two opposite views of the presented assets.

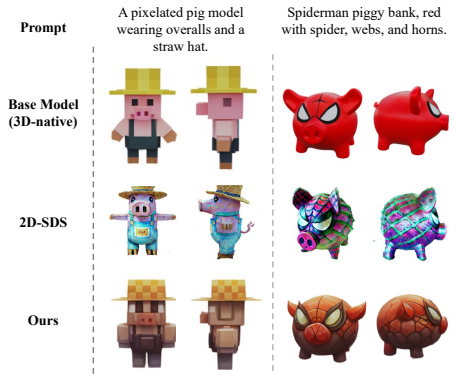

Figure 10: Comparison with 2D-SDS-based alignment method [51].

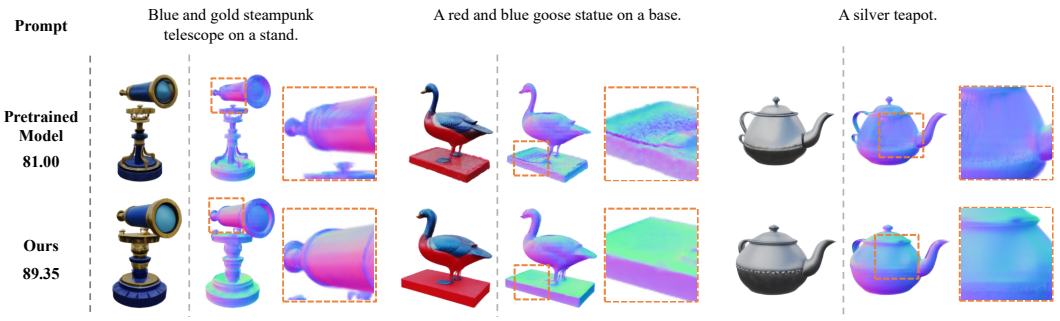

Figure 11: Qualitative comparison with pretrained model on DNC reward.

Results on DNC reward. We show the results of finetuning with different methods on DNC reward in Tab. 4. Our method achieves the best reward improvement without incurring much loss in FID and Clip-similarity. In Fig. 11, we visualize the geometry from the pretrained model and the model finetuned on the DNC reward. The results demonstrate that the DNC reward, independent of any priors from external normal estimators, can improve the sample geometry quality of 3D-native diffusion models. We further show the error map for the base model and the model finetuned with our method in Fig. 13.

Table 4: Comparison between the base model and the one finetuned with different methods on Depth-Normal Consistency.

Method	Reward↑ (10^{-2})	FID↓	CLIP-Sim↑ (10^{-2})
Base Model	87.81	55.26	34.58
ReFL	89.15	158.19	32.74
DDPO	87.99	**69.12**	34.48
DRaFT	88.13	74.82	**34.52**
Ours	**89.53**	99.01	34.08

Visualization of the evolution process. We visualize the evolution (every 50 update steps) of our method in the Aesthetic Score experiments (Fig. 12). We use the same seed, prompt, and initial noise, and we render the generated assets from the identical camera pose.

Figure 12: Visual evolution of the generated object with the same random seed during the finetuning process.

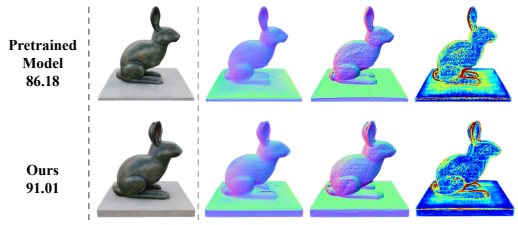

Figure 13: Results on DNC reward. From left to right: rendered RGB image, rendered normal map, depth-induced normal map, and the corresponding error map.

6 Discussions

Reward finetuning vs. test-time scaling. One may incorporate reward signals during inference using extra computational resources, a strategy known as test-time scaling [52]. Reward finetuning can be treated as an amortized process such that one does not have to pay the typically high cost of reward evaluation. Moreover, reward finetuning benefits from implicit regularization during network training, particularly in the case of LoRA finetuning [12].

Limitations. Our finetuning method suffers from the same issues as the lifting-from-2D approaches: no supervision for shape inner structures. Our method requires expensive gradient computations during the forward pass, underscoring the need for improved numerical algorithms and architectural designs for finetuning [33, 21]. Furthermore, our method focuses solely on parameter-level alignment and does not explore prompt-based alignment strategies [52].

7 Conclusion

We propose an efficient alignment method, dubbed **Nabla-R2D3**, for finetuning 3D-native generative methods with 2D differentiable rewards in a manner that avoids overfitting issues commonly seen in lifting-from-2D approaches. We demonstrate that **Nabla-R2D3** outperforms baseline methods across both appearance- and geometry-based reward models, as well as across different base architectures. The development of better alignment methods for diffusion models, including this work, contributes toward constructing virtual worlds aligned with human values.

Acknowledgements

This project is supported by Key-Area Research and Development Program of Guangdong Province, China under Grant 2024B0101040004.

References

[1] Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. GFlowNet foundations. *Journal of Machine Learning Research*, 2023. 3

[2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024. 2, 3, 5

[3] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-Σ: Weak-to-Strong Training of Diffusion Transformer for 4K Text-to-Image Generation. In *ECCV*, 2024. 5, 18

[4] Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *ICLR*, 2024. 2, 5

[5] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 5

[6] Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *ICLR*, 2025. 5, 17

[7] Yuan Dong, Qi Zuo, Xiaodong Gu, Weihao Yuan, Zhengyi Zhao, Zilong Dong, Liefeng Bo, and Qixing Huang. Gpld3d: Latent diffusion of 3d shape generative models by enforcing geometric and physical priors. In *CVPR*, 2024. 3

[8] 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 *NeurIPS*, 2022. 5

[9] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *ICLR*, 2017. 3, 4

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 5

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 10, 18

[13] Savva Victorovich Ignatyev, Nina Konovalova, Daniil Selikhanovych, Oleg Voynov, Nikolay Patakin, Ilya Olkov, Dmitry Senushkin, Alexey Artemov, Anton Konushin, Alexander Filippov, Peter Wonka, and Evgeny Burnaev. A3d: Does diffusion dream about 3d alignment? In *ICLR*, 2025. 3

[14] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, 2022. 3

[15] Hyeonho Jeong, Jinho Chang, Geon Yeong Park, and Jong Chul Ye. Dreammotion: Space-time self-similar score distillation for zero-shot video editing. In *ECCV*, 2024. 3

[16] Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *ICLR*, 2025. 3

[17] LAION. Laion aesthetic score predictor. https://laion.ai/blog/laion-aesthetics/, 2024. Accessed: 2024-09-27. 5

[18] Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable 3d gaussian splat generation. In *ICLR*, 2025. 5, 7, 18

[19] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 17

[20] 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. 17

[21] Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In *ICLR*, 2024. 10

[22] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 17

[23] Zhen Liu, Yao Feng, Michael J. Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. In *ICLR*, 2023. 2

[24] Zhen Liu, Yao Feng, Yuliang Xiu, Weiyang Liu, Liam Paull, Michael J. Black, and Bernhard Schölkopf. Ghost on the shell: An expressive representation of general 3d shapes. In *ICLR*, 2024. 2

[25] Zhen Liu, Tim Z. Xiao, Weiyang Liu, Yoshua Bengio, and Dinghuai Zhang. Efficient diversity-preserving diffusion alignment via gradient-informed gflownets. In *ICLR*, 2025. 2, 4, 5, 6, 17

[26] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*, 2022. 5

[27] George Ma, Emmanuel Bengio, Yoshua Bengio, and Dinghuai Zhang. Baking symmetry into gflownets. *arXiv preprint arXiv:2406.05426*, 2024. 3

[28] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025. 3

[29] Qinwei Ma, Jingzhe Shi, Can Jin, Jenq-Neng Hwang, Serge Belongie, and Lei Li. Gradient imbalance in direct preference optimization. *arXiv preprint arXiv:2502.20847*, 2025. 3

[30] Ling Pan, Dinghuai Zhang, Moksh Jain, Longbo Huang, and Yoshua Bengio. Stochastic generative flow networks. *UAI*, 2023. 3

[31] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 3, 4, 5, 9

[32] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *CVPR*, 2024. 5

[33] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In *NeurIPS*, 2023. 10

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 5

[35] 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 *NeurIPS*, 2023. 3

[36] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 conference proceedings*, 2023. 3

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 5

[38] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *CVPR*, 2024. 2

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 5

[40] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 17

[41] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 18

[42] Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024. 3

[43] Weitao Wang, Haoran Xu, Yuxiao Yang, Zhifang Liu, Jun Meng, and Haoqian Wang. Mvreward: Better aligning and evaluating multi-view diffusion models with human preferences. In *AAAI*, 2025. 3

[44] 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 *NeurIPS*, 2023. 2, 9

[45] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 5, 8

[46] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 2

[47] Desai Xie, Jiahao Li, Hao Tan, Xin Sun, Zhixin Shu, Yi Zhou, Sai Bi, Sören Pirk, and Arie E Kaufman. Carve3d: Improving multi-view reconstruction consistency for diffusion models with rl finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6369–6379, 2024. 3

[48] 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 *NeurIPS*, 2023. 2, 5

[49] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025. 17

[50] Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics (TOG)*, 2024. 5

[51] Junliang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. Dreamreward: Text-to-3d generation with human preference. In *ECCV*, 2024. 3, 9

[52] Taeyoung Yun, Dinghuai Zhang, Jinkyoo Park, and Ling Pan. Learning to sample effective and diverse prompts for text-to-image generation. In *CVPR*, 2025. 3, 10

[53] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade-gs: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*, 2024. 5

[54] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling. *arXiv preprint arXiv:2403.19655*, 2024. 5, 7, 18

[55] Dinghuai Zhang, Ricky T. Q. Chen, Nikolay Malkin, and Yoshua Bengio. Unifying generative models with GFlowNets and beyond. *arXiv preprint arXiv:2209.02606v2*, 2022. 3

[56] Dinghuai Zhang, Ricky TQ Chen, Cheng-Hao Liu, Aaron Courville, and Yoshua Bengio. Diffusion generative flow samplers: Improving learning signals through partial trajectory optimization. In *ICLR*, 2024. 3

[57] Dinghuai Zhang, Hanjun Dai, Nikolay Malkin, Aaron C Courville, Yoshua Bengio, and Ling Pan. Let the flows tell: Solving graph combinatorial problems with gflownets. In *NeurIPS*, 2023. 3

[58] Dinghuai Zhang, Nikolay Malkin, Zhen Liu, Alexandra Volokhova, Aaron Courville, and Yoshua Bengio. Generative flow networks for discrete probabilistic modeling. In *ICML*, 2022. 3

[59] Dinghuai Zhang, Ling Pan, Ricky T. Q. Chen, Aaron Courville, and Yoshua Bengio. Distributional GFlownets with quantile flows. *Transactions on Machine Learning Research*, 2024. 3

[60] Dinghuai Zhang, Yizhe Zhang, Jiatao Gu, Ruixiang Zhang, Joshua M. Susskind, Navdeep Jaitly, and Shuangfei Zhai. Improving GFlownets for text-to-image diffusion alignment. *Transactions on Machine Learning Research*, 2025. 5

[61] 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 (TOG)*, 2024. 2

[62] Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky TQ Chen. Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*, 2023. 17

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

Appendix

Table of Contents

A Overall algorithm

Algorithm 1 3D-Native Diffusion Alignment with 2D Rewards using **Nabla-R2D3**

Inputs: Pretrained diffusion model with sampling probability $p_{\text{base}}(x_t|x_{t+1})$, 2D reward model $R(\cdot)$
Initialization: Model to finetune with sampling probability $p_\theta(x_t|x_{t+1})$ where $\theta = \theta_{\text{base}}$.
Sample the initial batch of trajectories $\mathcal{D}_{\text{prev}} = \{(x_T, ..., x_0)_i\}_{i=1...N}$ with the current finetuned diffusion model p_θ.
Set $\theta^\dagger \leftarrow \theta$.
while not converged **do**
    Sample a batch of trajectories $\mathcal{D}_{\text{curr}} = \{(z_T, ..., z_0)_i\}_{i=1...N}$ with the finetuned diffusion model.
    Subsample the time steps to train with: the full set $\mathcal{T}_i = \{0, ..., T-1\}$ or the sampled set $\mathcal{T}_i = \text{Sample-N}(\{0, ..., T-1\})$ where Sample-N is some unbiased sampling algorithm to randomly pick N samples.
    Append the last timestep $t = 0$ to the sample trajectory set above.
    Sample a set of camera poses $c_1, ..., c_M$.
    Compute the loss (f is the output of the diffusion denoising network)
    $\frac{1}{M(N+1)} \sum_{j=1}^{M} \sum_{t \in \mathcal{T}_i}$
    $\left\| \nabla_{z_t} \log \tilde{p}_\theta(z_t|z_{t+1}) - w_t \beta \nabla \left[\nabla_{z_t} \log R(h(\hat{z}_\theta(z_t), c_j)) \right] \right\|^2$
    $+ \lambda \| f_\theta(x_{t+1}) - f_{\theta^\dagger}(x_{t+1}) \|^2$.
    Set $\theta^\dagger \leftarrow \theta$.
    Gradient update of θ with the loss function.
    Set $\mathcal{D}_{\text{prev}} \leftarrow \mathcal{D}_{\text{curr}}$.
end while
return finetuned model f_θ.

B Proof of Unbiasedness of Nabla-R2D3

We start with the original Nabla-GFlowNet forward loss (Eqn. 2) for random variable z_t but with the 3D reward specified in Eqn. 5:

$$L = \left\| \nabla_{z_{t-1}} \log \tilde{p}_\theta(z_{t-1}|z_t) - \gamma_t \beta \nabla \left[\nabla_{z_{t-1}} \mathop{\mathbb{E}}_{c\sim\mathcal{C}} \log R(h(\hat{z}_\theta(z_{t-1}), c)) \right] - g_\phi(z_{t-1}) \right\|^2 \tag{12}$$

The corresponding gradient is

$$\begin{aligned}
\nabla_\theta L = & -2 \Big\langle \nabla_\theta \nabla_{z_{t-1}} \log \tilde{p}_\theta(z_{t-1}|z_t), \gamma_t \beta \nabla \left[\nabla_{z_{t-1}} \mathop{\mathbb{E}}_{c\sim\mathcal{C}} \log R(h(\hat{z}_\theta(z_{t-1}), c)) \right] - g_\phi(z_{t-1}) \Big\rangle \\
& \nabla_\theta \left\| \nabla_{z_{t-1}} \log \tilde{p}_\theta(z_{t-1}|z_t) \right\|^2 \\
= & -2 \mathop{\mathbb{E}}_{c\sim\mathcal{C}} \Big\langle \nabla_\theta \nabla_{z_{t-1}} \log \tilde{p}_\theta(z_{t-1}|z_t), \gamma_t \beta \nabla \left[\nabla_{z_{t-1}} \log R(h(\hat{z}_\theta(z_{t-1}), c)) \right] - g_\phi(z_{t-1}) \Big\rangle \\
& \nabla_\theta \left\| \nabla_{z_{t-1}} \log \tilde{p}_\theta(z_{t-1}|z_t) \right\|^2 \\
= & \nabla_\theta L_{\text{forward}}(z_{t-1:t})
\end{aligned} \tag{13}$$

which proves the unbiasedness of the proposed loss in Eqn. 7. The proof for the reverse loss is similar.

C Application to Flow Matching Models

As discussed in prior works [6, 25], the popular generative model of flow matching [19] that samples $x_1 = x(1)$ via $\dot{x} = v(x,t), x_0 = x(0) \sim \mathcal{N}(0, I)$ can be turned into an equivalent diffusion model (but with a non-linear noising process). The sampling (denoising) process of this equivalent diffusion model is:

$$\mathrm{d}X_t = \left(v(X_t, t) + \frac{\sigma(t)^2}{2\beta_t \left(\frac{\dot{\alpha}_t}{\alpha_t} \beta_t - \dot{\beta}_t \right)} \left(v(X_t, t) - \frac{\dot{\alpha}_t}{\alpha_t} X_t \right) \right) \mathrm{d}t + \sigma(t)\, \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(0, I). \tag{14}$$

where $\sigma(t)$ is an arbitrary diffusion term. We may therefore use **Nabla-R2D3** to obtain a finetuned diffusion model (and therefore the corresponding probability flow ODE [40]) from a pretrained flow matching model.

For the special case of rectified flows [22] with $x_0 \sim \mathcal{N}(0, I)$, the velocity field $v(x,t)$ and the probability flow $p(x,t)$ are related via the following formula ([62], Lemma 1):

$$\nabla \log p(x,t) = -\left[\frac{1}{t} x + \frac{1-t}{t} v(x,t) \right]. \tag{15}$$

The corresponding reverse process of the equivalent diffusion model is therefore [20, 49]:

$$dx = \left[v(x,t) + \frac{\sigma_t^2}{2t} \big(x + (1-t) v(x,t) \big) \right] + \sigma_t dw \tag{16}$$

D Implementation Details

General experiments. We use LoRA [12] parametrization with a rank of 16 on all attention layers plus the final output layer for DiffSplat-Pixart-Σ [3], and a rank of 8 for DiffSplat-SD1.5 [18] and GaussianCube [54]. The CFG scales are set to 7.5, 7.5, and 3.5 for DiffSplat-Pixart-Σ, DiffSplat-SD1.5, and GaussianCube, respectively. All experiments were conducted with either two Nvidia Tesla V100 GPUs or GeForce GTX 3090 GPUs. It takes no more than one day to finetune models with our method and all other baselines.

3D SDS. During 3D SDS optimization, we sample timesteps $t \in [0, 400]$ and train the object for 1000 steps. For each step, the rewards are evaluated from four randomly sampled views. To compute test metrics, we sample 32 objects for each of 30 selected prompts, compute the mean per-prompt metrics and take the metrics averaged over all prompts.

E Comparison with Multi-view Generative Model

To emphasize the importance of 3D native representations for 3D consistency after finetuning, we use our method to finetune MVDream, a multi-view diffusion model, on Aesthetic score and compare the results with those with DiffSplat. The generated multi-view images are passed to Large Multi-view Gaussian Model (LGM) [41] to build reconstructed 3D objects so that they can be directly compared to 3D objects generated by 3D native models. Qualitative comparisons are presented in Fig. 14. The results from MVDream exhibit noticeable artifacts (illustrated in the green boxes in the figure), whereas the objects produced by the finetuned 3D native models do not.

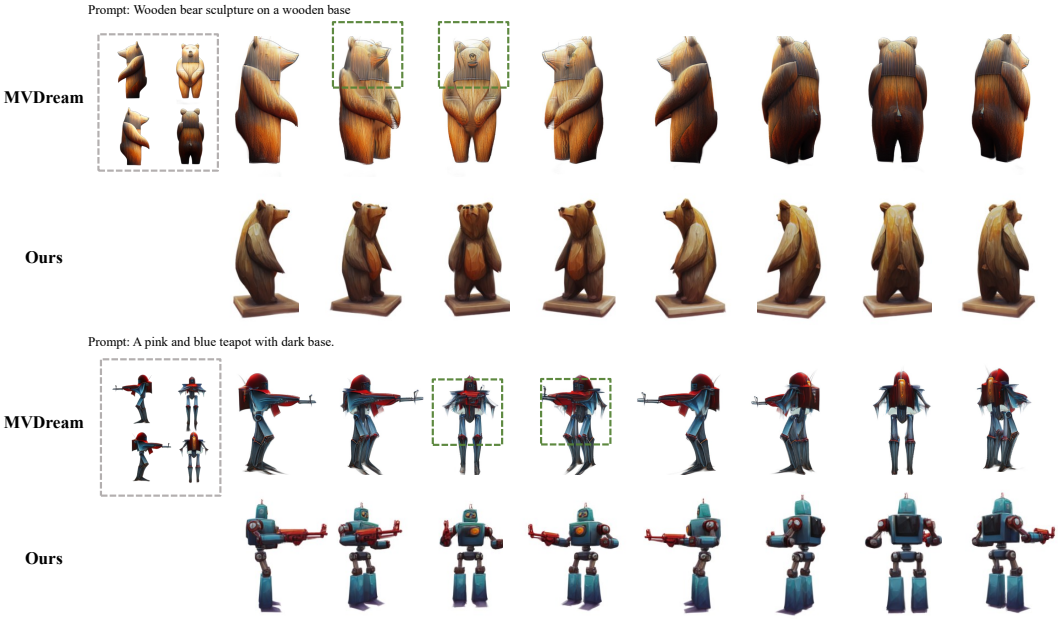

Figure 14: Qualitative comparison with MVDream. For MVDream, we show the multi-view images directly generated by the finetuned model in the gray dashed boxes; all other images for MVDream are rendered from the objects reconstructed by LGM (with the corresponding generated multi-view images).

F More Ablation Studies and Visualization

Effect of different reward temperatures. We experiment with different temperature values $\beta \in \{5e5,\ 1e6,\ 2e6\}$, and observe in Fig. 15 that a higher temperature leads faster convergence at the cost of worse text-object alignment and prior preservation. Since previous experiments (Fig. 2) showed that the variance of our method's metric is minimal, the statistics for the standard deviation (std) are omitted here.

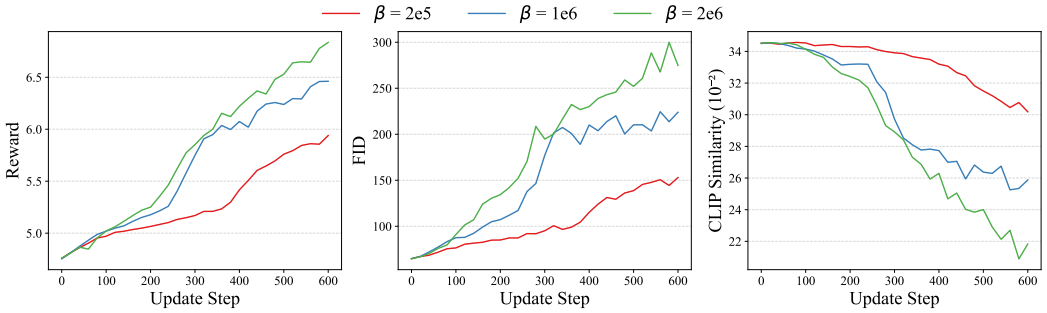

Figure 15: The relationship between the temperature parameter β and Reward, FID scores, and CLIP-Similarity scores. Higher values of β result in faster convergence, but at the cost of worse text-object alignment and diminished prior preservation.

Effect of different learning rates. As illustrated in Fig. 16, higher learning rates lead to faster convergence, but with compromises in prior preservation and text-object alignment.

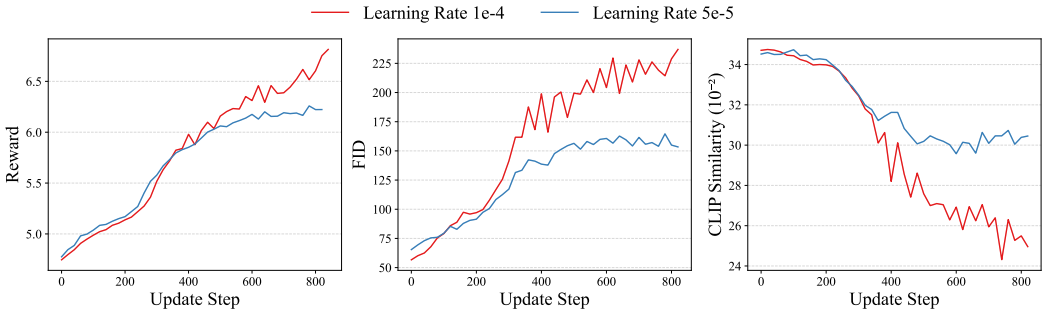

Figure 16: Convergence curves of metrics on different learning rate.

Comparison of metrics given the same FID level on HPSv2 Reward. We show the results on HPSv2 (with DiffSplat-Pixart-Σ) in Tab. 5, where we select checkpoints with roughly the same FID. Specifically, we pick checkpoints for models finetuned with ReFL and DRaFT such that their FIDs are approximately equal to that of our model in Tab. 1. The results clearly show that, given a similar FID level, our method achieves higher reward and CLIP-Sim scores.

Table 5: Results on HPSv2 (with DiffSplat-Pixart-Σ).

Method	**Reward** ↑(10^{-2})	**CLIP-Sim**↑(10^{-2})	**FID** ↓
ReFL	22.18	34.11	127
DRaFT	25.65	34.41	143
Ours	27.85	35.35	131

G More Qualitative Results

360° videos of the generated objects can be found at our project website: https://nabla-R2D3.github.io.

Figure 17: More qualitative results on Aesthetic Score.

Figure 18: More qualitative results on HPSv2.

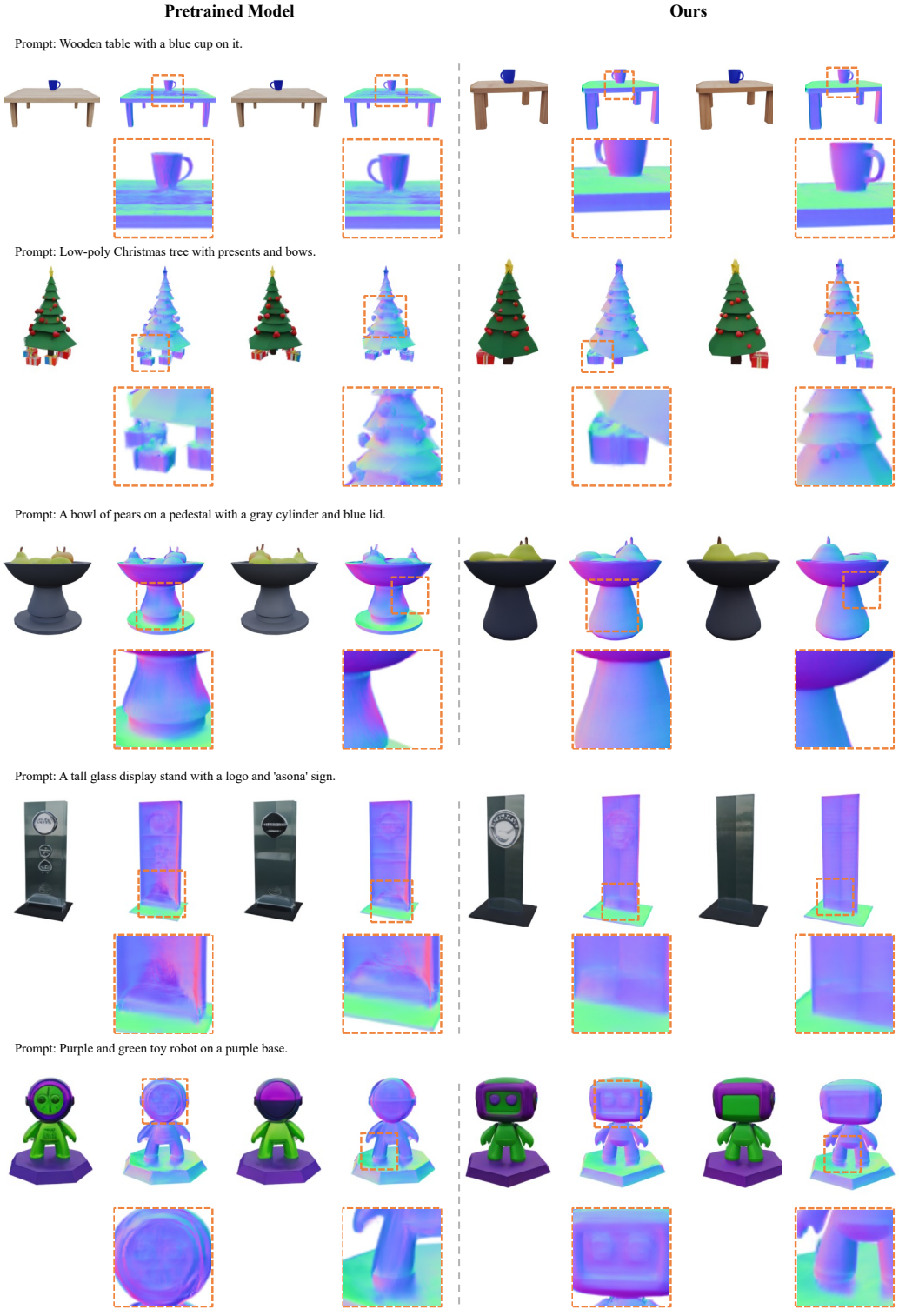

Figure 19: More qualitative results on the normal estimator reward.

H Failure Cases

Similar to alignment for the 2D image domain, our method still suffers from two issues: 1) imperfect rewards and 2) reward hacking. The first one can easily be observed in our normal-estimator-based reward model, which inevitably will hallucinate non-existing geometry based on single-view RGB cues (Fig. 20). If we finetune the target diffusion model for too long, reward hacking becomes apparent as the generated shapes tend to over-optimize the rewards in the way that the natural geometry and semantics are gradually forgotten (Fig. 21).

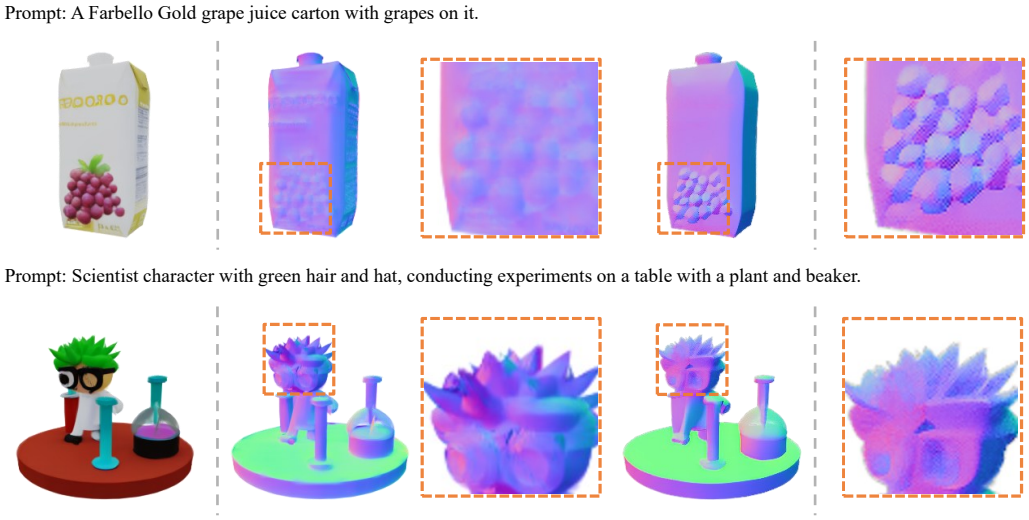

Figure 20: Failure cases of the normal estimator reward. Left: rendered RGB image, Middle: rendered normal map, Right: estimated normal map. The estimator produces wrong normal maps and therefore guides the diffusion model to generate wrong geometry.

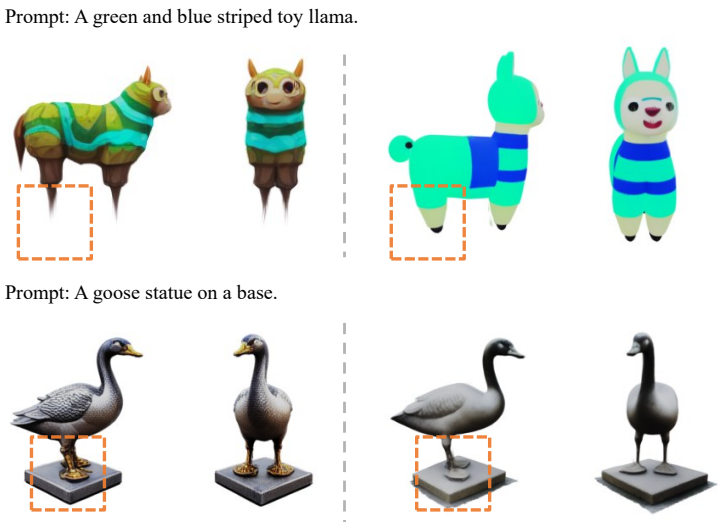

Figure 21: Failure cases of the Aesthetic Score (first row) and HPSv2 (second row). The left column shows results from the finetuned model, while the right column presents results from the pretrained model. In both cases, the feet of animals become unnatural after finetuning.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. Our main contributions are also detailed in Sec. 1. Also see Sec. 5.2 for more experimental evidence.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, please see Sec. 6 for limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide proof to the unbiasedness of the proposed loss that use 2D rewards in Appendix. All other formula are justified in reference papers.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper fully discloses necessary information required to reproduce the main experimental results in the Sec. 5 and Appendix. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released the whole set of code, instructions and data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper provides detailed descriptions of all experimental setups in Sec. 5 and Appendix. D, including prompts, base model, and hyperparameter configurations and evaluation metrics.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Following common practice, we provide standard deviation based on repeated random seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details of the GPU platform used in Appendix. D

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the positive impacts of this work in the conclusion section (Sec.7).

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not applicable for our case.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we credited them in appropriate ways.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not include any crowdsourcing activities or research involving human participants.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not include any crowdsourcing activities or research involving human participants.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper does not use LLMs as important, original, or non-standard components of the core methods.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.