Carve3D: Improving Multi-view Reconstruction Consistency for Diffusion Models with RL Finetuning

Desai Xie[1, 2] Jiahao Li[1, 3] Hao Tan[1] Xin Sun[1] Zhixin Shu[1]
Yi Zhou[1] Sai Bi[1] Sören Pirk[4] Arie E. Kaufman[2]

[1]Adobe Research [2]Stony Brook University [3]TTIC [4]Kiel University

Abstract

Recent advancements in the text-to-3D task leverage finetuned text-to-image diffusion models to generate multi-view images, followed by NeRF reconstruction. Yet, existing supervised finetuned (SFT) diffusion models still suffer from multi-view inconsistency and the resulting NeRF artifacts. Although training longer with SFT improves consistency, it also causes distribution shift, which reduces diversity and realistic details. We argue that the SFT of multi-view diffusion models resembles the instruction finetuning stage of the LLM alignment pipeline and can benefit from RL finetuning (RLFT) methods. Essentially, RLFT methods optimize models beyond their SFT data distribution by using their own outputs, effectively mitigating distribution shift. To this end, we introduce Carve3D, a RLFT method coupled with the Multi-view Reconstruction Consistency (MRC) metric, to improve the consistency of multi-view diffusion models. To compute MRC on a set of multi-view images, we compare them with their corresponding renderings of the reconstructed NeRF at the same viewpoints. We validate the robustness of MRC with extensive experiments conducted under controlled inconsistency levels. We enhance the base RLFT algorithm to stabilize the training process, reduce distribution shift, and identify scaling laws. Through qualitative and quantitative experiments, along with a user study, we demonstrate Carve3D's improved multi-view consistency, the resulting superior NeRF reconstruction quality, and minimal distribution shift compared to longer SFT. Project webpage: https://desaixie.github.io/carve-3d.

1. Introduction

In recent times, notable advancements have been made in the text-to-3D domain, driven by lifting images generated by 2D diffusion models [9, 40, 42, 48, 51, 55] to 3D. Numerous methods [24, 28, 46, 58] have demonstrated that a set of multi-view images is adequate for generating diverse and detailed 3D models, effectively mitigating the Janus (two-face) problem. Thus, ensuring the 3D consistency across these multi-view images is crucial for 3D generation, as inconsistencies can inevitably introduce artifacts, such as broken geometries, blurring, or floaters, in the NeRF reconstruction. However, the lack of an established multi-view consistency metric has led researchers to rely on qualitative inspections, which are both inefficient and unreliable, highlighting the ongoing issues with inconsistency in current methods.

Existing multi-view diffusion models [24, 27, 28, 46, 58] primarily utilize supervised finetuning (SFT) with datasets derived from rendering perfectly consistent multi-view images of 3D assets [12, 13]. While SFT can achieve some degree of multi-view consistency, it presents a dilemma: prolonged SFT enhances this consistency but also induces a distribution shift that diminishes diversity and realism of the results [24]. Such dilemma has been observed in the research of large language models (LLMs). While SFT changes the output distribution of pre-trained LLMs from text completion to answering instructions, such distribution shift to the instruction dataset also introduces hallucination [44], preventing longer SFT. InstructGPT [36], the paper behind ChatGPT 3.5 [35], introduces RL finetuning (RLFT) to further align the SFT model without causing additional distribution shift. Drawing an analogy between instruction-finetuned LLMs and multi-view diffusion models, RLFT emerges as an essential step following the SFT stage. By adopting RLFT, we aim to enhance the consistency of multi-view diffusion models without introducing the biases from a SFT dataset (Figure 1).

We introduce Carve3D, a RLFT framework paired with our Multi-view Reconstruction Consistency (MRC) metric, to improve the consistency of multi-view diffusion models (Figure 2). MRC compares the output multi-view images from a diffusion model, which serve as inputs for NeRF reconstruction, with images rendered from the NeRF at identical camera viewpoints. We use the sparse-view Large Re-

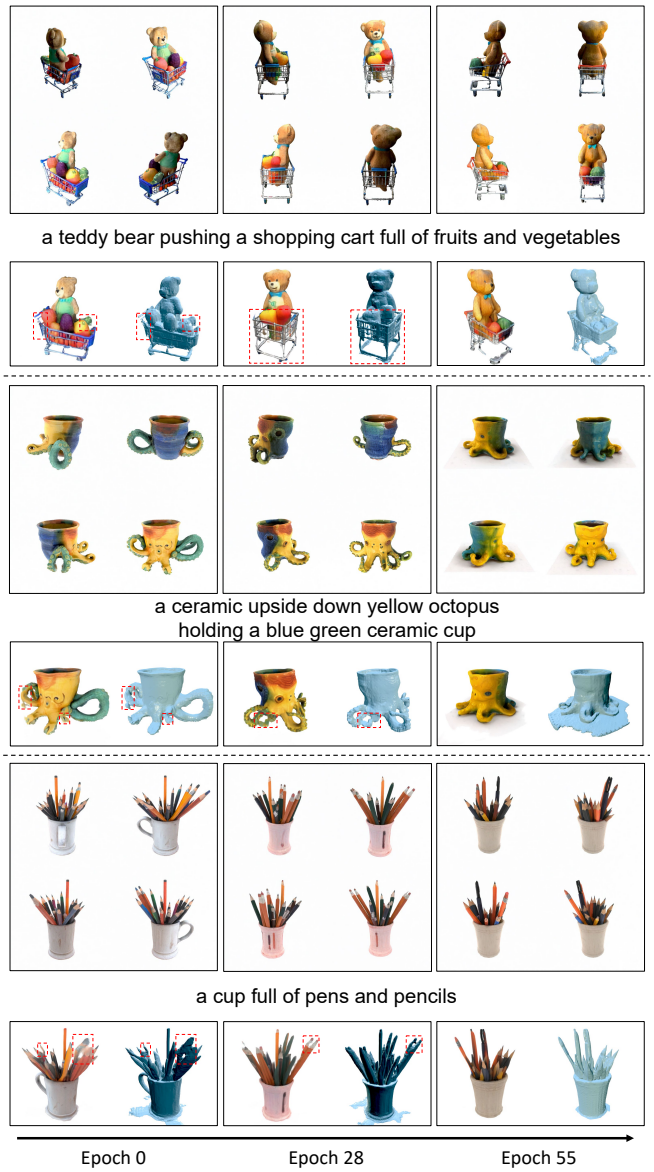

Figure 1. Carve3D steadily improves the 3D consistency of the multi-view diffusion model and the resulting quality of the NeRF and the mesh, without sacrificing its diversity or realism. Here we show results of the finetuned multi-view diffusion model over three epochs on three testing prompts (three blocks separated by dotted line), including the generated multi-view images (top), the reconstructed NeRF and extracted mesh (bottom) and the prompt (middle). The inconsistencies in the multi-view images, e.g. the facing direction of the shopping cart, the position of the octopus arms, and the position of the pencils, lead to artifacts in the NeRF and the mesh, highlighted by the red boxes.

construction Model (LRM) [17, 24] to achieve fast, feed-forward NeRF reconstruction from a few multi-view images. To quantify image similarity, we adopt LPIPS [56] as it is more effective and robust for MRC. We further normalize LPIPS with respect to the bounding boxes of foreground objects to prevent trivial reward hacking through size reduction of the foreground object. To validate the reliability of MRC, we conduct extensive experiments with controlled inconsistency levels; starting from a set of perfectly consistent multi-view images rendered from a 3D asset [12], we manually introduce distortion to one of the views to create inconsistency. Our MRC metric provides robust evaluation of consistency of multi-view images, offers a valuable tool for assessing current multi-view generation methods and guiding future developments in the field.

With MRC, we employ RLFT for multi-view diffusion models. In the RLFT process, we use a set of curated, creative text prompts to repeatedly generate diverse multi-view images with random initial noises and use their MRC reward to update the diffusion model (Figure 2). Such diversity- and quality-preserving finetuning cannot be achieved with SFT, as it is infeasibly expensive to create a dataset of diverse ground-truth multi-view images for these prompts. We make the following improvements to the RLFT algorithm [5]. In addressing the common issue of training instability in RL, we opt for a purely on-policy policy gradient algorithm [53], diverging from the widely adopted, partially on-policy PPO [45] algorithm. We incorporate KL divergence regularization [15, 36] to maintain proximity to the base model and prevent distribution shift. Moreover, we scale up the amount of compute to achieve optimal rewards by applying the scaling laws for diffusion model RLFT, identified from extensive experiments – a topic that has not yet been extensively covered in existing studies [5, 15].

Through quantitative and qualitative experiments, as well as human evaluation, we demonstrate that Carve3D: (1) achieves improved multi-view consistency and NeRF reconstruction quality over the base multi-view diffusion model, Instant3D-10K [24], as well as Instant3D-20K and Instant3D-100K, which utilize more SFT steps, and (2) maintains similar prompt alignment, diversity, and realistic details from the base Instant3D-10k, preventing the degradation in Instant3D-20k and Instant3D-100k. We extend our consistency evaluation to additional multi-view diffusion models using MRC, revealing the universal presence of multi-view inconsistency when relying solely on SFT. Our work is the first application of RLFT to the text-to-3D task, especially with diffusion models at the SDXL [39] scale using a 2.6B-parameter UNet. We hope this work will bolster the research on RLFT for alignment in the computer vision community.

2. Related Works

2.1. 3D Generation with 2D Diffusion Models

NeRF is a neural representation of 3D assets [8, 31, 33]. It infers the direction-dependent radiance at an arbitrary

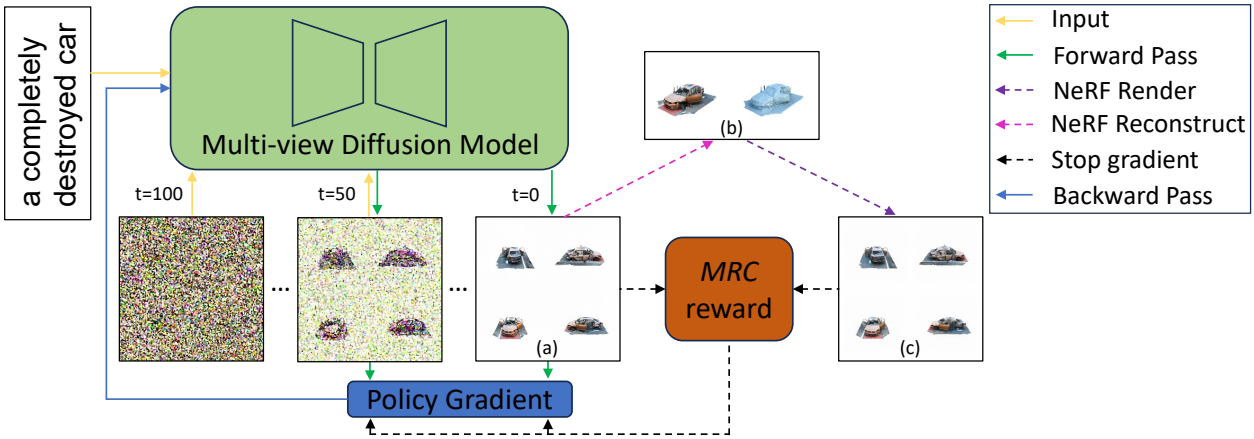

Figure 2. Overview of Carve3D. Given a prompt sampled from the curated prompt set, we run the denoising process to obtain the final denoised image, which contains four multi-view images tiled in a 2-by-2 grid. MRC reward is computed by comparing (a) the generated multi-view images with (c) the corresponding multi-view images rendered at the same camera viewpoints from (b) the reconstructed NeRF. Then, we train the model with policy gradient loss function, where the loss is derived from the reward and log probabilities of the model's predicted noise, accumulated across all denoising timesteps. Using only a set of prompts, this RLFT process finetunes the diffusion model with its own outputs, without relying on ground truth multi-view images.

volumetric position with neural models. Many text-to-3D methods rely on NeRF to produce 3D objects.

While text-to-image diffusion models are trained on 5 billion data [43], the largest public 3D dataset only contains 10 million 3D assets [12, 13] with little text annotation. This gap in the diversity and quality of 3D data has restricted the quality of current 3D diffusion models and their ability in handling complex prompts [19, 34]. To circumvent this limitation, another line of work focuses on lifting 2D images to 3D, thus leveraging the remarkable semantic understanding and high-quality generation capabilities of 2D diffusion models [39, 42]. These methods [9, 40, 48, 51, 55] typically employ 2D diffusion models to provide supervision at the novel views for optimizing 3D objects represented as NeRF or by 3D Gaussian Splatting [21]. Building on this concept, multiple works [24, 26–28, 46, 58] have proposed generating multi-view images using a finetuned 2D diffusion model, providing a more comprehensive visual prior and preventing the multi-face (Janus) problem. However, as the finetuning datasets of multi-view images are rendered from the same 3D dataset [12, 13], the limited quality and diversity remains a challenge, preventing running SFT to convergence [24]. By adopting RLFT, we do not depend on ground truth multi-view images and thus optimize the model beyond the distribution of their SFT dataset.

A key challenge in utilizing multi-view images is achieving 3D consistency, ensuring that the geometry and appearance of an object is uniform across different views. While numerous methods have attained notable multi-view consistency by supervised finetuning 2D diffusion models [24, 27, 28, 46, 58], their evaluation has been empirical, lacking explicit metrics. An approach known as 3D consistency scoring [52] measures the consistency of output views by optimizing a NeRF trained on these views. However, this method requires dense input view sampling for cross-validation, making it unsuitable for evaluating sparse views. To overcome this limitation, we propose the MRC metric. MRC evaluates multi-view consistency by comparing input images with renderings of the generated 3D objects from corresponding views. We employ LPIPS, a perceptual image distance metric [56], to quantify the image differences. Additionally, we validate the reliability of MRC by conducting extensive experiments under controlled inconsistency levels. The robustness of MRC allows us to conduct RLFT on multi-view diffusion models, which significantly enhances their multi-view consistency.

3D models can be derived from either single or multi-view images by optimizing the SDS distillation loss [40, 51]. However, the optimization process is notably time-consuming, requiring multiple hours to generate a single 3D asset. In contrast, LRM [17], trained on the extensive 3D dataset Objaverse [12], can efficiently reconstruct NeRF models from a single image in a feed-forward manner. In this work, we focus exclusively on text-to-3D using feed-forward sparse-view NeRF reconstruction, specifically employing sparse-view LRM [24]. This choice is driven by its significantly faster performance compared to SDS-based optimization methods and its superior quality relative to feed-forward text-to-3D diffusion models [19, 34]. We choose Instant3D [24] as our base multi-view diffusion model, owing to its light-weight SFT that preserves the strong semantic understanding and high-quality image generation capabilities of SDXL [39], similar to the instruction finetuning stage in InstructGPT [36].

2.2. RLFT of LLMs and Diffusion Models

RL has been widely used to finetune large pre-trained models in NLP [2, 3, 22, 36] and CV [5, 11, 15, 38, 41, 57], due to its advantage over SFT. SFT directly fits a model to the distribution of the SFT dataset containing inputs and ground-truth target data, which unavoidably causes some degree of distribution shift [44]. On the contrary, based on an objective function and a dataset containing only inputs, RLFT optimizes a model beyond the limitation of a SFT dataset by using its own outputs and effectively mitigates distribution shift [7].

RLFT of LLMs LLMs like GPT-3 [6] are pre-trained on the next-word prediction task on an internet-scale corpus. While the autoregressive pre-training is a powerful self-supervised objective that allows LLMs to extract substantial knowledge from the internet-scale unlabeled dataset, pre-trained LLMs can only perform the corresponding text completion task. The pre-training lacks an objective that allows LLMs to respond to text prompts. In InstructGPT [36], the paper behind ChatGPT 3.5, a two-stage finetuning solution is proposed to align GPT-3 to answer instructions according to human preferences. In the first stage, InstructGPT employs SFT with a small dataset of hand-crafted prompt-answer pairs. While SFT changes the model's output distribution from text completion to answering instructions, it also introduces hallucination [44]. This is because the output distribution drifts too much towards the instruction-following dataset, and the model tries to imitate the behavior in the data and always provide plausible answers even when the model is uncertain about the answer [44]. To address this issue, InstructGPT opts for a light-weight SFT stage and relies on RLFT in the second stage, using a human-preference reward model. This approach provides general alignment to human values and causes minimal hallucination [44], because RLFT does not rely on a potentially biased dataset containing fixed ground-truth answers, but instead learns the general concept of human-preference through the reward model. The success of InstructGPT [36] and its analogy to the distribution shift problem in multi-view SFT [24] motivate us to pursue RLFT for 2D diffusion models.

RLFT of Diffusion Models Witnessing the success of RLFT methods in LLMs [2, 3, 22, 36], recently, a few RLFT algorithms have been proposed for text-to-image diffusion models. RWR [23] is the first work to bring the human feedback reward finetuning idea to diffusion models. While RWR only finetunes stable diffusion [42] via a single log probability of the entire denoising process, multi-step RLFT can be facilitated by treating the denoising process as a multi-step MDP, as demonstrated in DDPO [5] and DPOK [15]. Our RLFT is based on DDPO [5], while our KL-divergence regularization is similar to DPOK [15] and InstructGPT [36]. Furthermore, RWR, DDPO, and DPOK all finetune SD-1.5 [42], while we finetune a much larger diffusion model based on SDXL. We also study training stability, a notorious challenge in both traditional RL and RLFT [7, 59], and scaling laws [20] for RLFT.

3. Multi-view Reconstruction Consistency

In this section, we propose the Multi-view Reconstruction Consistency (MRC) metric, for quantitative and robust evaluation of the consistency of multi-view images, which we define to be *the degree of geometry and appearance uniformity of an object across the views*.

3.1. Evaluate Consistency via NeRF Reconstruction

NeRF [31] is widely adopted as the 3D representation for learning text-to-3D tasks. A 3D model represented by NeRF can be reconstructed from the view images of the object and their corresponding camera poses. The quality of a NeRF notably depends on the consistency of the provided images images [31, 52] – inconsistent views lead to artifacts in the NeRF, which includes floaters, blurring, and broken geometry. To address this challenge, we introduce a metric for assessing the consistency among multiple views.

The intuition behind MRC comes from the relationship between multi-view consistency and the reconstructed NeRF. As shown in Figure 3, when the multi-view images are consistent, they can produce a well reconstructed NeRF, preserving almost all the visual cues from the input images; therefore, the views rendered from the NeRF at the same camera viewpoints will look the same as the original views; conversely, when the multi-view images are inconsistent (e.g., intentionally introduced inconsistency in Figure 3), they will produce a NeRF with broken geometry and floater artifacts; thus, the NeRF rendered views will look different from the original views. Building upon this observation, we propose the MRC metric, defined as the image distances between the original multi-view images and the views of the reconstructed NeRF rendered at the same viewpoints, as illustrated in Figure 2.

3.2. Implementation

We formulate the implementation of MRC as three parts: fast sparse-view NeRF reconstruction, measuring image distance between the input images and the rendered images, and a normalization technique for the image distance. The pseudo code for our MRC implementation is shown in Listing 1 in Appendix.

Fast Sparse-view Reconstruction We conduct NeRF reconstruction with sparse-view Large Reconstruction Model

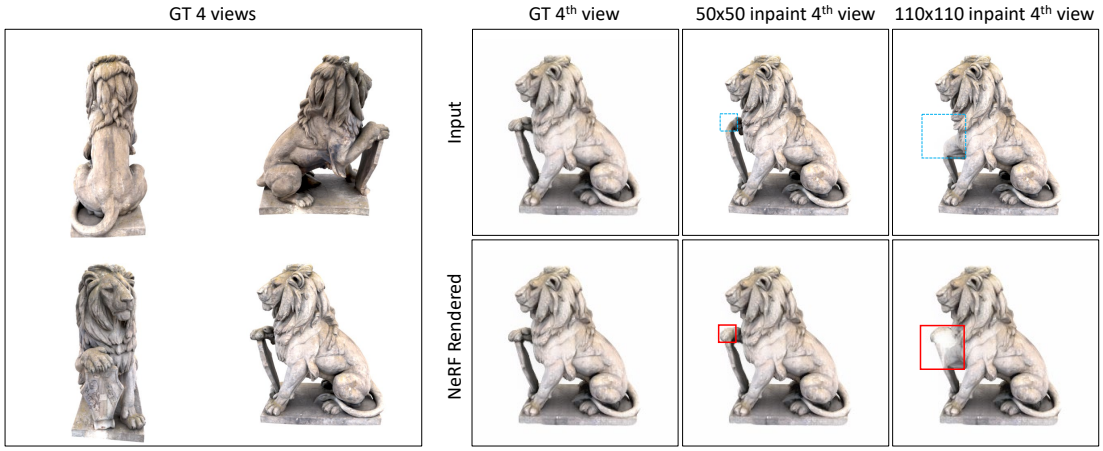

Figure 3. Qualitative correlation between MRC and inconsistency with increasing intensity, introduced by inpainting with increasing mask sizes. Left: the four ground truth views. Right: the 4th view is inpainted with increasing area sizes, i.e. 0×0, 50×50 and 110×110 pixels. The top row is the image after inpainting and the bottom row is the image rendered from the NeRF reconstructed with the top inpainted 4th view and the other 3 original views. We mark the inpainting area with blue and red boxes. Since the lion's right paw in the inpainted 4th views look different from the other three original views, its geometry is broken in the NeRF rendered views. This difference is captured in MRC's image dissimilarity metric.

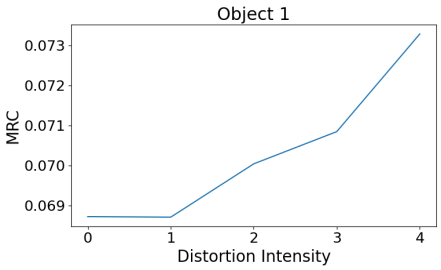

Figure 4. Quantitative correlation between MRC and inconsistency with increasing intensity, for the object shown in Figure 3. As inconsistency intensity rises, MRC also monotonically increases.

(LRM) proposed in [17, 24]. Different from dense view NeRF reconstruction [8, 31, 33], sparse-view LRM reconstructs a NeRF with only 4-6 view images. Also, with its feed-forward reconstruction, it can achieve a speed two orders of magnitude faster than previous optimization-based reconstruction methods. MRC leverages all multi-view images for both NeRF reconstruction and 3D consistency evaluation. Although the NeRF is reconstructed based on the visual prior of the input multi-views images, the rendering from the same views still exhibits notable differences if there is inconsistency inside the input, as shown in Figure 3.

Image Distance Metric In Section 3.1, the consistency problem is reduced from 3D to a 2D image dissimilarity problem. To measure the image dissimilarity between the input views and their corresponding NeRF rendered views, we utilize the perceptual image distance metric, LPIPS [56]. LPIPS exhibits smoother value changes with respect to the consistency of multi-view images compared to PSNR, SSIM, L1, and L2, as shown in Figure 14 in Appendix). Such smoothness is derived from the non-pixel-aligned computation in LPIPS, as opposed to the other image distance metrics that are more pixel-aligned. Also, the smoothness is a crucial aspect for MRC to serve as the reward function in RLFT, because non-smooth, high-variance reward functions makes the RLFT training more challenging.

Bounding-box Normalization Current multi-view diffusion models [24, 28, 46, 58] target single object generation with background. Consequently, if computing LPIPS on the entire image, trivially reducing the object's relative size (as illustrated in Appendix Figure 9's car example) can exploit MRC, as the majority of images will be the white background. Therefore, we propose normalizing our metric with respect to the object's size. Specifically, we identify the smallest square bounding box of the foreground object in the input view image. Then we crop both the input images and the rendered images with that bounding box, resize them to a fixed resolution, and evaluate the LPIPS. This normalization effectively prevents the reward hacking of MRC by diminishing foreground object sizes, as shown in Figure 9 in Appendix.

3.3. Metric Experiment

The two key objectives for introducing the MRC metric are (1) to assess the consistency of any multi-view generative model and (2) to enable RLFT for improving the

consistency of multi-view diffusion models. Thus, the proposed consistency metric should ideally present two respective properties: (1) MRC should monotonically increase as inconsistency increases; (2) the MRC vs. inconsistency curve should be smooth.

To validate the effectiveness and robustness of MRC, i.e. whether it satisfies the two properties, we conduct evaluation on sets of multi-view images with controlled level of inconsistency. Starting from a set of perfectly-consistent ground truth views rendered from a 3D asset from Objaverse [12], we manually introduce inconsistency to one image. We select a portion of this image and inpaint it with an image-to-image diffusion model[1]. Therefore, we get different levels of distortion on one image, determined by the size of the inpainting area, that corresponds to different levels of inconsistency of the set of images.

Figure 3 shows the qualitative result on one object of our MRC metric experiment. With increased inpainting area size, the NeRF rendered view also shows larger image difference, which is then captured by MRC's image distance metric, LPIPS. Figure 4 presents the quantitative curve of the same experiment. MRC indeed shows a monotonically increasing pattern as the views become more inconsistent. As shown in Figure 14, MRC constantly exhibits monotonically increasing pattern, and it is also smoother than the other MRC variants using PSNR, SSIM, L1, and L2. For metric experiments on other distortion types, see Appendix D.

4. RLFT for Multi-view Consistency

In the previous section, we proposed a fast and reliable multi-view consistency metric, and in this section we describe how it can be used to finetune a multi-view diffusion model. We propose RLFT for enhancing the consistency of 2D multi-view diffusion models, using the negative MRC as the reward function (Figure 2). Building upon DDPO [5], we opt for its pure on-policy policy gradient algorithm over the default partially on-policy version for substantially improved training stability. To maintain proximity to the base model, we incorporate KL divergence regularization similar to [15, 36]. In addition, we scale up the RLFT to achieve higher rewards by studying the scaling laws [20] of diffusion model RLFT through extensive experiments.

4.1. Preliminaries on DDPO

Markov Decision Process To use RL for finetuning, we need to formulate the task as a Markov Decision Process (MDP). In a MDP, an agent interacts with the environment at discrete timesteps; at each timestep t, the agent is at a state s_t, takes an action a_t according to its policy $\pi(a_t|s_t)$, receives a reward r_t, and transitions to the next state s_{t+1}. Following denoising diffusion policy optimization (DDPO) [5], the denoising process of a diffusion model is formulated as a multi-step MDP:

$$\begin{aligned} s_t &= (c, t, x_t), \\ a_t &= x_{t-1}, \\ \pi(a_t|s_t) &= p_\theta(x_{t-1}|c, t, x_t), \\ r(s_t, a_t) &= \begin{cases} r(x_0, c) & \text{if } t = 0, \\ 0 & \text{otherwise,} \end{cases} \\ r(x_0, c) &= -\text{MRC}(x_0) \end{aligned}$$

where each denoising step is a timestep, c is the context, i.e. the text prompt, x_t is the image being denoised at step t, p_θ is the diffusion model being finetuned, x_T is the initial noisy image, x_0 is the fully denoised image, and $r(x_0, c)$ is the negative MRC (Listing 1 in Appendix) computed on the fully denoised image.

Policy Gradient In order to optimize the model with respect to the reward function, a family of RL algorithms, known as policy gradient methods, are commonly adopted, such as REINFORCE [53] and Proximal Policy Optimization (PPO) [45]. DDPO_{SF} is based on the vanilla policy gradient algorithm, REINFORCE [53], also known as the Score Function (SF). On the other hand, DDPO_{IS} builds upon PPO [45] and conducts multiple optimization steps per round of data using an importance sampling (IS) estimator and importance weight clipping.

As a common practice to reduce the variance of the policy gradients [32], DDPO [5] uses the advantages, which are rewards normalized to have zero mean and unit variance, instead of directly using the rewards. Specifically, the mean and standard deviation statistics of the rewards are tracked for each prompt c:

$$A_r(x_0, c) = \frac{r(x_0, c) - \mu_r(c)}{\sigma_r(c)} \tag{1}$$

DDPO's normalizing advantage replaces the value model that is more widely adopted in PPO-based RLHF methods [36, 50, 54]. This is similar to the recent work [25], which shows that the value model creates unnecessary computation cost that can be replaced with a simpler advantage formulation.

By using the advantage term (Equation (1)) in place of the reward, the DDPO_{SF} policy gradient function is:

$$\hat{g}_{\text{SF}} = \mathbb{E}\left[\sum_{t=0}^{T} \nabla_\theta \log p_\theta(x_{t-1}|c, t, x_t) A_r(x_0, c)\right] \tag{2}$$

where the expectation is taken over data generated by the policy π_θ with the parameters θ. The log probability

[1] We use Adobe Photoshop's Generative Fill [1] without text prompt to add inpainting distortion, which is based on a diffusion model.

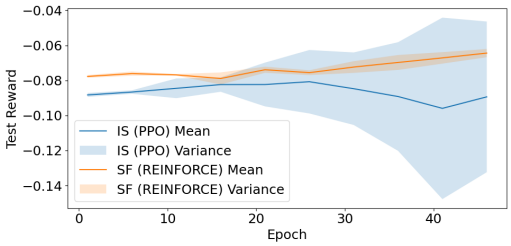

Figure 5. Reward curves on the testing set with 4 different random seeds for IS and SF versions, where negative MRC is used as the reward. The IS version produces reward curves with high variance, including two runs that fail to improve the reward and collapse. In contrast, the SF version stably produces reward curves with low variance.

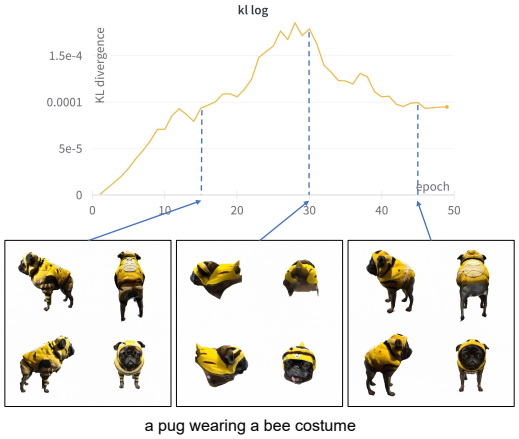

Figure 6. When we only plot KL divergence without incorporating KL regularization, we observe qualitative correlation between the KL value and the prompt alignment degradation. Despite being distant in the finetuning process, epoch 15 and epoch 45, which have lower KL divergence to the base model, generates prompts better aligned with the prompts. On the other hand, epoch 30, which has much higher KL divergence to the base model, generates results with broken identity, i.e. the body of the pug is missing.

$\log p_\theta(x_{t-1}|c,t,x_t)$ can be easily obtained since the policy is an isotropic Gaussian distribution when using the DDIM sampler [5, 47]. The DDPO_{IS} (Equation (6) in Appendix) function has an additional importance sampling term than Equation (2).

Black *et al*. [5] choose DDPO_{IS} as the default policy gradient function, because it exhibits better sample efficiency than DDPO_{SF} (Fig. 4 of [5]). Such choice is consistent with the use of PPO [45] in LLM RLHF literature [2, 3, 36, 50, 54].

4.2. Improvements over DDPO

While RLFT using the default DDPO_{IS} coupled with MRC can enhance the 3D consistency of multi-view diffusion models, it still faces challenges regarding training stability, the shift of output distributions, and an unclear training scale setting to achieve optimal rewards with minimal distribution shift. To address these issues, we propose three improvements over DDPO [5] in this section. Given the universal nature of these challenges in RLFT, our enhancements may offer broader applicability across various tasks.

4.2.1 Pure On-policy Training

Training stability is a major challenge in both RLFT [7, 59] and traditional RL [14]. With the default DDPO_{IS}, our training process is evidently unstable, as shown in Figure 5. Training experiments with different random seeds or a slight change of hyperparameters can lead to different reward curves and qualitative results. This complicates the training result evaluation as we cannot distinguish meaningful improvement or deterioration from the variance introduced by random seed.

We argue that such high variance is derived from the multi-step update in DDPO_{IS} [5], originally proposed in PPO [45]. While it theoretically allows for better sample efficiency similar to off-policy methods [45], it also causes the training to be more unstable and the reward curves to be more variant, because it uses data collected with the older policy to update the newer policy. Due to the undesirable consequences of training instability, we adopt the pure on-policy variant DDPO_{SF}, discarding the multi-step update from PPO (Equation (6) in Appendix). As shown in Figure 5, DDPO_{SF} significantly improves the training stability of our RLFT, while maintaining a comparable sample efficiency as the default DDPO_{IS}.

Diverging from DDPO [5] and most LLM RLHF literature [2, 3, 36, 50, 54], we choose REINFORCE [53] (DDPO_{SF}) over PPO [45] (DDPO_{IS}) for its superior training stability. We provide two hypotheses behind our surprising finding in Appendix B.3, including the difficulty of the task reward function and the size of the model being finetuned. The favored use of REINFORCE [53] over PPO [45] could apply to broader scenarios that meet these two conditions. We leave the verification of our hypotheses as future work.

4.2.2 KL Divergence Regularization

In RLFT methods, distribution shift (also known as reward overoptimization) can lead to low-quality results, such as cartoon-like, less realistic style [5] or oversaturated colors and unnatural shape [15], despite achieving high rewards. In our case, we observe this as degradation of diversity, texture details and prompt alignment after prolonged RLFT with the MRC reward. Previous methods [15, 36] suggest mitigating reward overoptimization by incorporating a penalty on the KL divergence between the log probabilities of the outputs from the base and the finetuned mod-

els. In our case, the base model is Instant3D-10K [24] without any additional finetuning. By plotting the KL divergence values during finetuning, we also find that KL divergence correlates to the reward overoptimization problem (Figure 6), suggesting us to adopt KL divergence regularization.

Following the widely adopted implementation in LLM RLHF [36, 54], we incorporate KL penalty into the reward function. Subtraction of the log probabilities is commonly used to approximate the full KL divergence [50, 54]:

$$\begin{aligned}&\mathrm{KL}\left(\log p_\theta(x_0|c,T,x_T)||\log p_{\theta_{\text{base}}}(x_0|c,T,x_T)\right)\\&=\sum_{t=0}^{T}\frac{\log p_\theta(x_{t-1}|c,t,x_t)-\log p_{\theta_{\text{base}}}(x_{t-1}|c,t,x_t)}{T+1}\end{aligned}\tag{3}$$

where $p_{\theta_{\text{base}}}$ is the base model. We will denote this approximated KL divergence term as $\mathrm{KL}(x_0|c,x_T)$ for clarity in presentation.

KL divergence values starts at 0 and unavoidably increases as finetuning proceeds, making it hard to determine an optimal coefficient for the penalty term. To enable a steady KL divergence regularization throughout the finetuning process, we propose to normalize the KL divergence penalty term. This normalization ensures that the gradient consistently favors low-KL-divergence, high-reward samples, even in the early stages when KL divergence is still low compared to the later stages. We extend DDPO's [5] per prompt stat tracking to also track the mean and standard deviation statistics of the KL divergence term in order to to normalize it:

$$A_{\mathrm{KL}}(x_0,c)=\frac{\mathrm{KL}(x_0|c,x_T)-\mu_{\mathrm{KL}}(c)}{\sigma_{\mathrm{KL}}(c)}.\tag{4}$$

Our advantage terms now consist of both the normalized reward and the normalized KL divergence. Our final policy gradient function, used in our experiments, is a combination of Equations (2) and (4)

$$\begin{aligned}\hat{g}_{\text{SF,KL}}=\mathbb{E}\Bigg[&\sum_{t=0}^{T}\nabla_\theta\log p_\theta(x_{t-1}|c,t,x_t)\\&\cdot\left(\alpha A_r(x_0,c)-\beta A_{\mathrm{KL}}(x_0,c)\right)\Bigg]\end{aligned}\tag{5}$$

where α and β are the coefficients for the reward advantage and the KL advantage, respectively.

4.2.3 Scaling Laws for RLFT

The training of RL is highly sensitive to the chosen scale setting [14], impacting various results, including the final converged reward. Through the scaling laws identified from extensive experiments, we scale up the amount of compute (equivalent to scaling up the batch size in our case) to achieve the optimal reward. Although our scaling experiments are only conducted with the multi-view consistency task, our insights into the scaling laws of diffusion model RLFT are general and can be beneficial in broader scenarios.

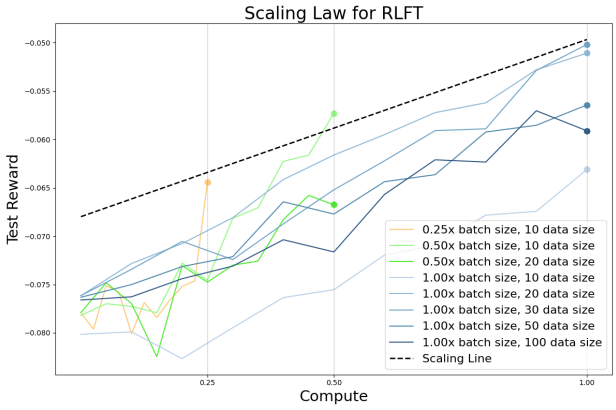

Figure 7. Scaling law for Carve3D's diffusion model RLFT algorithm. When we scale up the amount of compute for RLFT, the model improves its reward smoothly under the optimal data size. The amount of compute scales linearly with respect to the batch size. The reward curves also become more stable (less variant) with a larger batch size. The reward curves are reported up to epoch 50.

Compute and Batch Size The reward curves from our experiments demonstrate a positive scaling law of the model's reward at epoch 50 with respect to the amount of compute (Fig. 7); the scaled up compute brings smooth improvement to the model's reward, under the optimal data sizes at each batch size. Note that the amount of compute scales directly with respect to the batch size.

Data Size The model's reward does not directly scale with the data size but there exists a more complicated relationship between them. As shown in Figure 7, the optimal data size at each batch size grows as the batch size get larger, indicating that both factors need to be scaled up in tandem; after the optimal data size, naively continuing to scale up the data size actually reduces the model's reward. Surprisingly, even when trained on a prompt set as small as a size of 10, the model still shows generalization to the testing prompts. We choose data size of 30 with batch size 768 in our final experiments as it achieves the highest reward in our analysis.

Training Epochs With the pure on-policy DDPO_{SF} (REINFORCE [53]), the model steadily and smoothly improves its rewards throughout the finetuning process, meaning that more training epochs constantly lead to higher

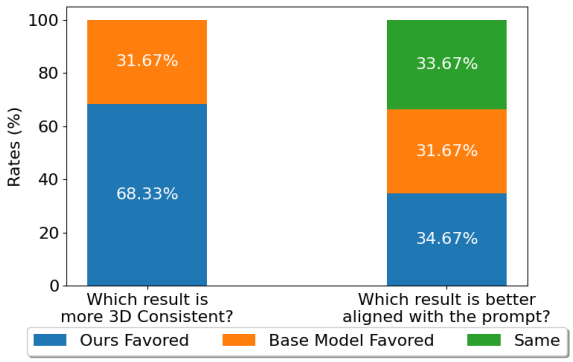

Figure 8. We conducted a user study with 20 randomly selected testing prompts and the corresponding outputs from both the base and fine-tuned model. 15 participants took part in the study, with a majority favoring the 3D consistency of our fine-tuned model. Opinions are evenly split on which has better prompt alignment.

reward. However, from our qualitative results, we also observe worse distribution shift, e.g. the degradation of prompt alignment and texture details, as training epoch increases. Due to the correlation between KL divergence and the quality degradation (Figure 6), we stop the fine-tuning early when a predefined KL divergence threshold is reached. This threshold is empirically chosen based on qualitative results. For fair comparisons, we report the reward curves up to epoch 50 in Figure 7. See Appendix B.1 for the definition of epoch in RLFT, which is different from its definition in supervised learning.

5. Experiments

In this section, knowing that Carve3D's RLFT steadily improves the MRC reward on the training set (Figure 5), we aim to answer the following questions:

1. Does Carve3D's improved MRC generalize to the testing set?
2. Qualitatively, is Carve3D more consistent than the base model? And comparing to longer SFT?
3. Qualitatively, does Carve3D sacrifice the diversity, prompt alignment, and texture details of the base model?

We quantitatively and qualitatively compare Carve3D with three versions of Instant3D [24], with 10K, 20K, and 100K SFT steps respectively. Instant3D-10K is both the default model in [24] and also our base model. For fair qualitative comparisons, results from each model are generated from the the same initial noise. Since 20K and 100K versions of Instant3D and Carve3D are all finetuned from Instant3D-10K, their output tend to represent the same object when given the same initial noise (e.g. Figures 1 and 6).

	Avg MRC on Testing Set ↓
MVDream	0.1222
Instant3D-10k (Base)	0.0892
Instant3D-20k	0.0795
Instant3D-100k	0.0685
Carve3D (Ours)	**0.0606**

Table 1. Carve3D (RLFT with MRC reward on Instant3D-10k), achieves substantially better MRC than baselines, which corresponds to better multi-view 3D consistency. We evaluate these text-to-multiview diffusion models on the DreamFusion testing set, containing 414 text prompts. We generate 4 outputs for each prompt, and the average MRC is computed over the 1656 outputs. For each output, we use the same randomly sampled initial noise for all models to ensure the comparison is fair.

5.1. Comparison with Base Model and Longer SFT

Quantitative Comparison and Generalization As shown in Table 1, when evaluated on the testing set, Carve3D achieves substantially improved MRC over the base model. More SFT steps indeed provides better multi-view consistency and achieves better MRC, with Instant3D's 100K version performing the best and 10K version performing the worst. However, Carve3D still outperforms even the most consistent 100K version of Instant3D by a notable gap. This suggests that the explicit multi-view consistency objective in MRC, paired with our RLFT algorithm, can improve the model's consistency more efficiently than SFT. Furthermore, our RLFT provides generalization of the improved multi-view consistency, although only finetuned on 30 prompts. Such generalization, also observed in [5, 36], is likely derived from the strong knowledge from the base model.

Multi-view Consistency and NeRF Artifacts Figure 12 shows the improved multi-view consistency and the resulting NeRF reconstruction quality. While the multi-view images generated by the base model may be inconsistent, causing artifacts such as floaters and broken geometry, Carve3D can fix such inconsistency in the multi-view images and produce NeRF with clean geometry, free of artifacts. In Figure 11, Carve3D continues to show superior multi-view consistency and reduced NeRF artifacts, but such improvement is less and further less obvious when compared to the 20K and 100K version of Instant3D [24], which aligns with the qualitative results in Table 1.

Prompt Alignment and Texture Details By virtue of our RLFT with KL-divergence regularization (Section 4.2), which prevents distribution shift, and curated low-reward prompts, which describes complex objects (Appendix C.1), Carve3D preserves the prompt alignment and the texture details of the base model, as shown in Figure 12. On the

other hand, longer SFT causes additional distribution shift in Instant3D [24] from the base SDXL [39] towards the SFT training set [12]. As shown in Figure 11, Instant3D-20K and Instant3D-100K exhibits degradation in diversity, realism, and level of detailed textures. This quality degradation with longer SFT is also observed in [24].

Diversity As shown in Figure 13, Carve3D can preserve the diversity of the base model. This owes to our RLFT process, which repeatedly samples different initial noises for the diffusion model to generate diverse results (Figure 2).

5.2. Evaluating Existing Methods

At the time of writing, MVDream [46] is the only text-to-multi-view diffusion model with released code and model weights. As shown in Table 1 and Fig. 11, its outputs have notably worse multi-view consistency, realism, and level of details than all three Instant3D variants and Carve3D. This supports our claim in Section 1 that current multi-view diffusion models that solely rely on SFT still suffer from the inconsistency problem and could benefit from RL finetuning.

5.3. User Study

In addition to the quantitative and qualitative comparisons in Section 5.1, we conducted a user study to further understand the qualitative results of Carve3D's RLFT when perceived by human. To run the study we *randomly* selected 20 unseen testing prompts. For each text prompt, we generated a pair of data from both the base and the finetuned models with the same initial noise. Then, we provided both the tiled 4-view image and the turntable video of the reconstructed NeRF to participants and asked them the following two questions: (1) Which result is more 3D-consistent? and (2) Which result is better aligned with the prompt? As shown in Figure 8, 68.33% of participants believe that Carve3D's generated results are more 3D consistent than that of the base model [24]. Given that the multi-view consistency in the base model has already been much improved with SFT [2], the nearly 37% gain in human preference introduced by Carve3D on *randomly* selected testing prompts is impressive. Furthermore, the Carve3D finetuned model exhibits similar prompt alignment, as participants' votes are evenly distributed among ours, base model, and "same". The preservation of alignment can be attributed to the KL divergence regularization (Section 4.2) as well as early stopping the RLFT according to KL divergence (Section 4.2.3).

[2]Please see https://jiahao.ai/instant3d/ for base model's 3D consistency

6. Conclusion

In this paper, we propose Carve3D, an RL finetuning method to improve the reconstruction consistency of 2D diffusion models. The reward of Carve3D relies on MRC, a novel metric that measures the reconstruction consistency by comparing input multi-view images with the renderings of the reconstructed NeRF at the same viewpoints. The effectiveness and robustness of MRC are also validated by showing its correlation with intentional distortions. Lastly, we conduct experiments and a user study to show that Carve3D significantly improves the reconstruction consistency of multi-view images and the resulting quality of the NeRF. These enhancements are achieved without sacrificing the prompt alignment, texture details, or prompt alignment of the base model.

Our MRC metric can serve as a valuable tool for evaluating any multi-view generative methods and guiding future developments in the field. Although we only demonstrate our RLFT with MRC on one multi-view diffusion model [24], it can be directly adapted to other text-to-multi-view diffusion models; such adaptation only requires tuning a few hyperparameters related to the scaling laws for diffusion model RLFT (Section 4.2.3). Our surprising finding behind the choice of REINFORCE [53] over PPO [45] for better training stability could also be applied in broader RLFT scenarios.

As AI models grow more powerful, it becomes more important to evaluate and improve their safety and reduce their bias. RLFT has been widely used for LLM alignment as it allows models to be finetuned with hard-to-specify objectives and its results are generalizable without undermining the base model's knowledge. As the first work to use RLFT for text-to-3D and on diffusion models at the SDXL scale, we hope Carve3D can inspire more alignment research in the computer vision community.

References

[1] Adobe. Adobe firefly. https://www.adobe.com/sensei/generative-ai/firefly.html, 2023. Accessed: 2023-11-15. 6

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. 4, 7, 1

[3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol

Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. 4, 7, 1

[4] Kevin Black. ddpo-pytorch. https://github.com/kvablack/ddpo-pytorch, 2023. Accessed: 2023-11-17. 2

[5] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2023. 2, 4, 6, 7, 8, 9, 1

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 4

[7] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Bıyık, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023. 4, 7

[8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 5

[9] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting, 2023. 1, 3

[10] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 1

[11] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards, 2023. 4

[12] 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, 2022. 1, 2, 3, 6, 10

[13] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects, 2023. 1, 3

[14] Theresa Eimer, Marius Lindauer, and Roberta Raileanu. Hyperparameters in reinforcement learning and how to tune them, 2023. 7, 8

[15] 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, 2023. 2, 4, 6, 7

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[17] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d, 2023. 2, 3, 5

[18] 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. *arXiv preprint arXiv:2106.09685*, 2021. 2

[19] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions, 2023. 3

[20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. 4, 6

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

[22] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback, 2023. 4

[23] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback, 2023. 4

[24] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model, 2023. 1, 2, 3, 4, 5, 8, 9, 10

[25] Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models, 2023. 6

[26] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization, 2023. 3

[27] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023. 1, 3

[28] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Learning to generate multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 1, 3, 5

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[30] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module, 2023. 4

[31] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 4, 5

[32] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016. 6

[33] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2, 5

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

[35] OpenAI. Chatgpt. https://chat.openai.com/, 2023. Accessed: 2023-11-15. 1, 2

[36] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 1, 2, 3, 4, 6, 7, 8, 9

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 2

[38] André Susano Pinto, Alexander Kolesnikov, Yuge Shi, Lucas Beyer, and Xiaohua Zhai. Tuning computer vision models with task rewards, 2023. 4

[39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2, 3, 10, 1, 4

[40] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 1, 3, 2

[41] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2023. 4

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 3, 4

[43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 3

[44] John Schulman. Rl and truthfulness: Towards truthgpt. YouTube, 2023. 1, 4

[45] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. 2, 6, 7, 10, 1

[46] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 1, 3, 5, 10

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 7

[48] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 1, 3

[49] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 2

[50] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020. 6, 7, 8, 1, 2

[51] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation, 2022. 1, 3

[52] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models, 2022. 3, 4

[53] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. 2, 6, 7, 8, 10, 1

[54] Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, Zhongzhu Zhou, Michael Wyatt, Molly Smith, Lev Kurilenko, Heyang Qin, Masahiro Tanaka, Shuai Che, Shuaiwen Leon Song, and Yuxiong He. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales, 2023. 6, 7, 8, 1, 2

[55] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arxiv:2310.08529*, 2023. 1, 3

[56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2, 3, 5

[57] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese,

Stefano Ermon, Caiming Xiong, and Ran Xu. Hive: Harnessing human feedback for instructional visual editing. 2023. 4

[58] Minda Zhao, Chaoyi Zhao, Xinyue Liang, Lincheng Li, Zeng Zhao, Zhipeng Hu, Changjie Fan, and Xin Yu. Efficientdreamer: High-fidelity and robust 3d creation via orthogonal-view diffusion prior, 2023. 1, 3, 5

[59] Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. Secrets of rlhf in large language models part i: Ppo, 2023. 4, 7

A. Appendix Summary

In the appendix, we cover additional details of DDPO and RLFT (Appendix B, the training data details (Appendix C.1), training details (Appendix C.2), additional MRC metric experiments (Appendix D), ablation studies (Appendix E), and future work (Appendix F).

B. Additional Details of DDPO and RLFT

B.1. Definitions

Following [5, 36, 45, 54], an *epoch* is defined as one round of data collection (*sampling*), which may consists multiple PPO update steps (*training*), as discussed in Equation (6) and Sec. 4.2. This definition of "epoch" is different from the meaning in supervised learning which usually refers to go through all data once. Since we opt for using pure on-policy training (vanilla policy gradient), as discussed in Section 4.2, we only do one training step per sampling step, and thus our sampling batch size and training batch size are equal.

B.2. DDPO_{IS} Policy Gradient Function

By using the advantage term (Equation (1)) in place of the reward, the DDPO_{IS} policy gradient function is:

$$\hat{g}_{\text{IS}} = \mathbb{E}\left[\sum_{t=0}^{T} \frac{p_\theta(x_{t-1}|c,t,x_t)}{p_{\theta_{\text{old}}}(x_{t-1}|c,t,x_t)} \cdot \nabla_\theta \log p_\theta(x_{t-1}|c,t,x_t) A_r(x_0,c)\right] \quad (6)$$

where the expectation is taken over data generated by the policy $\pi_{\theta_{\text{old}}}$ with the parameters θ_{old}.

B.3. Hypotheses on Stability and Sample Efficiency

Diverging from DDPO [5] and most LLM RLHF literature [2, 3, 36, 50, 54], we choose REINFORCE [53] (DDPO_{SF}) over PPO [45] (DDPO_{IS}) for its superior training stability. We provide two hypotheses behind our surprising finding.

(1) Training stability is more vital than sample efficiency when the task reward function is more challenging. When a reward function is more variant with respect to the model's output, it becomes more difficult for the model to discover the pattern of high-reward outputs and to improve its rewards. The high-variance prompt alignment reward curves in Fig. 5 of DDPO [5] indicates the challenging nature of the prompt alignment task as opposed to the smooth reward curves for the aesthetics and compressibility tasks in Fig. 4 of DDPO [5].

(2) The RLFT sample efficiency is less important for a large model which requires less finetuning steps, as demonstrated in studies of LLM instruction finetuning [10]. Similarly, our RLFT on a 2.6B-parameter UNet from SDXL [39] only takes 55 epochs, as opposed to DDPO's [5] RLFT on a 860M-parameter UNet from SD 1.4 [42] using 200 epochs. Therefore, the potential sample efficiency gain provided by the multi-step update of PPO [45] gets outweighted by the training stability provided by REINFORCE [53].

The favorableness of REINFORCE [53] could apply to broader scenarios that fits these two conditions. We leave the verification of our hypotheses as future work.

C. Implementation Details

C.1. Training Data

An advantage of RLFT over SFT is that, we can manually create a high-quality text prompts training set, while creating a dataset of diverse ground truth multi-view images for these high-quality text prompts is prohibitively expensive for SFT. By relying on samples generated by the model itself to compute the reward and the loss, RLFT can optimize a model beyond the limitation of a dataset and preserves the diversity and the style of the base model. In Carve3D, our training prompts preparation process involves two strategies.

Training Data Curation Instead of randomly sampling prompts from a dataset, we employ a data curation strategy where prompts with lowest rewards are selected. Specifically, we run inference of the base model on a prompt dataset, generating four results per prompt, compute the MRC rewards for each result, and sort the prompts according to their average reward. This is derived from observation that, for certain prompts, the model generates nearly optimal outputs with rewards close to the rewards of ground truth views of a 3D asset [12] (Figure 4). Thus, the curated lowest-reward prompts have substantial room for 3D consistency improvement and prevent learning stagnation. This approach not only brings more efficient training but

```
def compute_mrc(ori_views, ori_cam_poses, lrm, lpips, resize_res):
    nerf = lrm(ori_views, ori_cam_poses)
    nerf_views = nerf.render(ori_cam_poses)
    square_bbox = compute_square_bbox(ori_views) # bounding box coordinates for each view
    x_min, y_min, x_max, y_max = square_bbox
    ori_views_bbox = [resize(o[:, y_min:y_max + 1, x_min:x_max + 1], resize_res) for o in
    ↪  ori_views]
    nerf_views_bbox = [resize(n[:, y_min:y_max + 1, x_min:x_max + 1], resize_res) for n in
    ↪  nerf_views]
    mrc = lpips(ori_views_bbox, nerf_views_bbox).mean()
    return mrc
```

Listing 1. Pseudo code for our MRC implementation. ori_views and ori_cam_poses are the multi-view images to be evaluated and their camera poses. lrm is the sparse-view LRM [17, 24]. lpips the the LPIPS [56] metric. resize_res is a fixed resolution to which we resize the bounding box patches.

also provides a more generalized improvement in 3D consistency to the testing set.

Creating New Training Prompt Set The prompt dataset from DreamFusion [40], which contains 414 prompts and is commonly used as testing set. To employ the DreamFusion prompt set also as our testing set, we create a new prompt dataset with ChatGPT4 [35]. Following our training data curation strategy, we first sort the DreamFusion [40] prompts according to their rewards attained by the base Instant3D [24] model. We provide the sorted prompt set to ChatGPT4, and ask it to summarize the characteristics of the low-reward prompts by looking at the low-reward, median-reward, and high-reward prompts. ChatGPT4 summarizes low-reward prompts to possess properties of "complex and creative". We then ask it to generate 100 low-reward prompts that are both complex and creative, and another 100 low-reward prompts that are "complex but not too creative". For each set, again, we sort the prompts according to their rewards, and select those with the lowest rewards to be our training prompt set. Our best results are obtained with the "complex but not too creative" set.

C.2. Training Details

All of our RL finetuning experiments are run on 6 AWS EC2 P4de nodes with 8 NVIDIA A100-SXM4-80GB GPUs, a total of 48 GPUs. We use batch size of 768, which is 2x compared to that of DDPO. One experiment takes 16.5 hours to reach 55 epochs. The number of finetuning epochs is determined by our KL-divergence early-stopping mechanism, which we empirically choose to be $3.2e-4$ according to the level of reward overoptimization shown on qualitative results.

We use minibatches of size 8 during sampling and 4 during training due to the limited GPU memory. The total batch size of 768 is evenly distributed among each GPU, so that the per GPU batch size is 16. The model samples two minibatches of size 8 on all GPUs to reach the total batch size. Similarly, the model accumulates gradients over four minibatches of size 4 on all GPUs, before synchronizing the gradients and performing an optimizer step. We use a per prompt stat tracker with windows of size 76, so that it roughly tracks all the rewards per prompt ever 3 epochs. This is much larger than DDPO's default tracking window of size 32 for better training stability. The coefficients for the advantage terms in Equation (5) are $\alpha = 1$ and $\beta = 0.2$.

The rest of our RL finetuning setup follows DDPO [4, 5]. We use the AdamW [29] optimizer with a fixed learning rate $3e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$ and a weight decay of $1e-4$. The high learning rate is paired with Low Rank Adaptation (LoRA) [18] finetuning with rank 4, which significantly reduces the memory and computation requirements for finetuning. We freeze all networks in the base model and set their precision to fp16, and only finetune the LoRA weights of the unet under fp32 via mixed precision training.

Our base text-to-multiview diffusion model setup follows Instant3D [24], which uses the same architecure as SDXL [39]. It produces images of resolution 1024x1024, which contains four images of resolution 512x512 tiled in a 2-by-2 fashion. Instant3D requires 100 denoising steps during inference, doubling the required computation than the default 50 steps for SDXL. It uses Classifier Free Guidance [16] with a scale of 5.0

Our code is mainly based on DDPO's [5] official implementation, the ddpo-pytorch [4] Github repository, which uses Hugginface diffusers [49] and PyTorch [37] libraries. Our KL divergence regularization implementation is inspired by the codebases of DeepSpeedChat [54], TRL [50], and DPOK [15]. We thank the authors of these repositories for releasing the high-quality implementations and promoting open-sourced research. We are going to release the code for computing MRC and our improved DDPO implementation. However, due to the fact that Sparse View LRM and

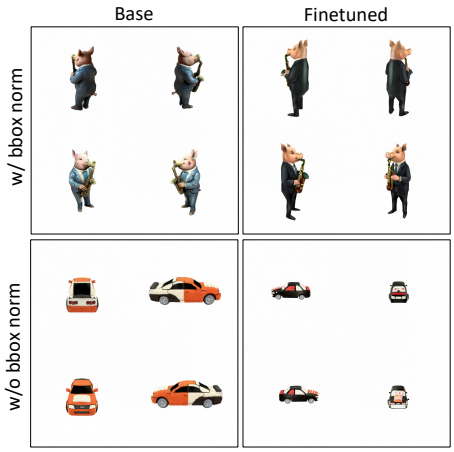

Figure 9. Alation study on the boundingbox normalization of LPIPS for MRC evaluation. Top: with normalization, the 3D objects keep similar size after finetuning. Bottom: without normalization, the sizes of 3D objects are greatly reduced after RL finetuning.

Instant3D do not plan to release their code, we have to leave these as empty, abstract functions in our released code.

D. Additional MRC Metric Experiments

Distortion Types Here, we show the full results for the metric experiments for the inpainting distortion (Figure 14) discussed in Section 3.3 and Figs. 3 and 4. We also conduct metric experiments with other distortions types: azimuth rotation (Figure 15, and elevation rotation (Figure 16). In azimuth and elevation rotation, for one out of the four views, we rotate the object with an azimuth or elevation rotation by 3.6 or 4 degrees, before rendering that view, and also use the original camera extrinsic matrix as the input to Sparse View LRM. The quantitative results matches our expectations, where MRC (LPIPS) monotonically decreases as we intentionally add more distortion.

LPIPS vs. Other Image Similarity Metrics Here, we compare substituting LPIPS with L1, L2, PSNR, and SSIM in the metric experiments on all distortion types. In the inpainiting distortion experiments (Figure 14), which is the most representative of diffusion model's inconsistencies, LPIPS is more linear than other pixel level image metrics. In azimuth and elevation distortion experiments (Figures 15 and 16), all image metrics shows monotonically decreasing pattern, while pixel-level image metrics are more linear. This is expected as the distortion is pixel-aligned and more structured.

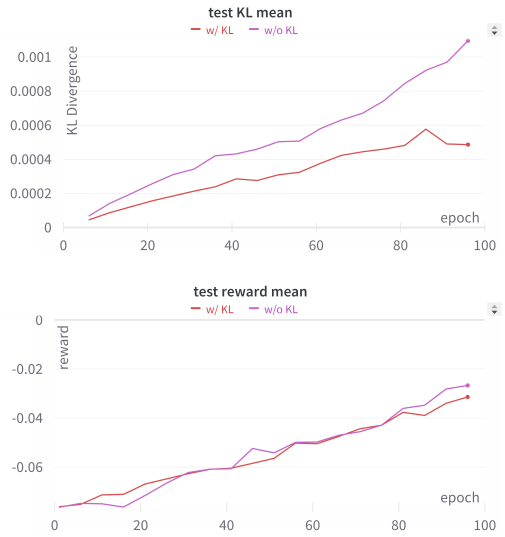

Figure 10. Ablation study on KL divergence regularization. Top: KL Divergence between the base model and the finetuned model on testing set. Bottom: mean MRC reward on testing set. Our KL divergence regularization does not sacrifice the model's efficiency on improving the reward. Without KL divergence regularization, the finetuned model's KL divergence to the base model on the training and the testing set grows faster, which results in degraded object identity and reduced texture details.

E. Ablation Studies

Bounding Box Normalization As shown in Figure 9, when the bounding box normalization is removed from MRC, the model would trivially increase the reward by reducing the size of the foreground object on the white background. This would lead to the model generating images containing only the white background, after longer finetuning. With bounding box normalization, the model would learn the harder task of improving the reconstruction consistency of the multiview images.

KL Divergence Regularization As shown in Figure 10 Our KL divergence regularization does not sacrifice the model's efficiency on improving its reward. Without KL divergence regularization, the KL divergence grows much faster. As discussed in Section 4.2, this leads to degraded object identity and loss of texture details.

F. Limitations and Future Work

Carve3D is limited by the reconstruction quality of Sparse View LRM [17, 24]. Because its reconstruction is not perfect, this leads to non-zero MRC metric on GT views as shown in Figures 14 to 16. Due to this limitation of Sparse View LRM, Carve3D RL finetuned model can produce less high-frequency details than the base model in or-

der to lower the image distance to the NeRF rendered views. This might be solved by using a future sparse view reconstructor that can preserve more details or training a dedicated model for computing MRC.

Further increasing data size and batch size to improve generalization of the improved reconstruction consistency is possible. However, in this work, we are limited by the high computation cost of SDXL [39], Instant3D's 100 denoising steps, and the high number of samples needed in DDPO. A few concurrent works could address this challenge. It is possible to substantially reduce the computation cost by switching to Consistency Models for one/few-step inference (e.g., LCM-LoRA [30]). In addition, we can also switch from DDPO to direct backpropagation of reward (e.g. Align-Prop [41], and DRaFT [11]) to reduce the number of samples needed. We leave these extensions as future work.

Figure 11. Qualitative comparison of MVDream, Instant3D with 10k, 20k, and 100k SFT steps, and Carve3D (five columns) on four prompts (four blocks separated by dotted line). In each block, we show their generated multi-view images in the 2-by-2 grid (top), reconstructed NeRF and extracted mesh (bottom) when given the prompt (middle). When compared to the base Instant3D-10K: Carve3D maintains the detailed texture and provides improved multi-view consistency and higher quality NeRF; in contrast, the models with prolonged SFT of 20K and 100K steps exhibit worse level of details and realism, while only providing slightly improved consistency.

Figure 12. Qualitative comparison of Instant3D (the base model) and Carve3D (the model finetuned from Instant3D) on 12 prompts (in 12 blocks separated by dotted line). In each block, we show the their generated multi-view images in the 2-by-2 grid (top), the reconstructed NeRF and the extracted mesh (bottom) when given the prompt (middle). We draw red boxes on the NeRF and the extracted mesh to highlight the artifacts in the NeRF and the mesh, resulting from the inconsistencies in the multi-view images. Carve3D maintains the detailed texture and provides improved multi-view consistency and higher quality NeRF than the base Instant3D.

Figure 13. Diverse results from original Instant3D (left) and our Carve3D (right) on 4 prompts (in 4 blocks separated by the dotted line). In each block, we show the their generated multi-view images in the 2-by-2 grid (top), the reconstructed NeRF and the extracted mesh (bottom) when given the prompt (middle). Our RLFT does not compromise the diversity of the base Instant3D model, while improving the consistency.

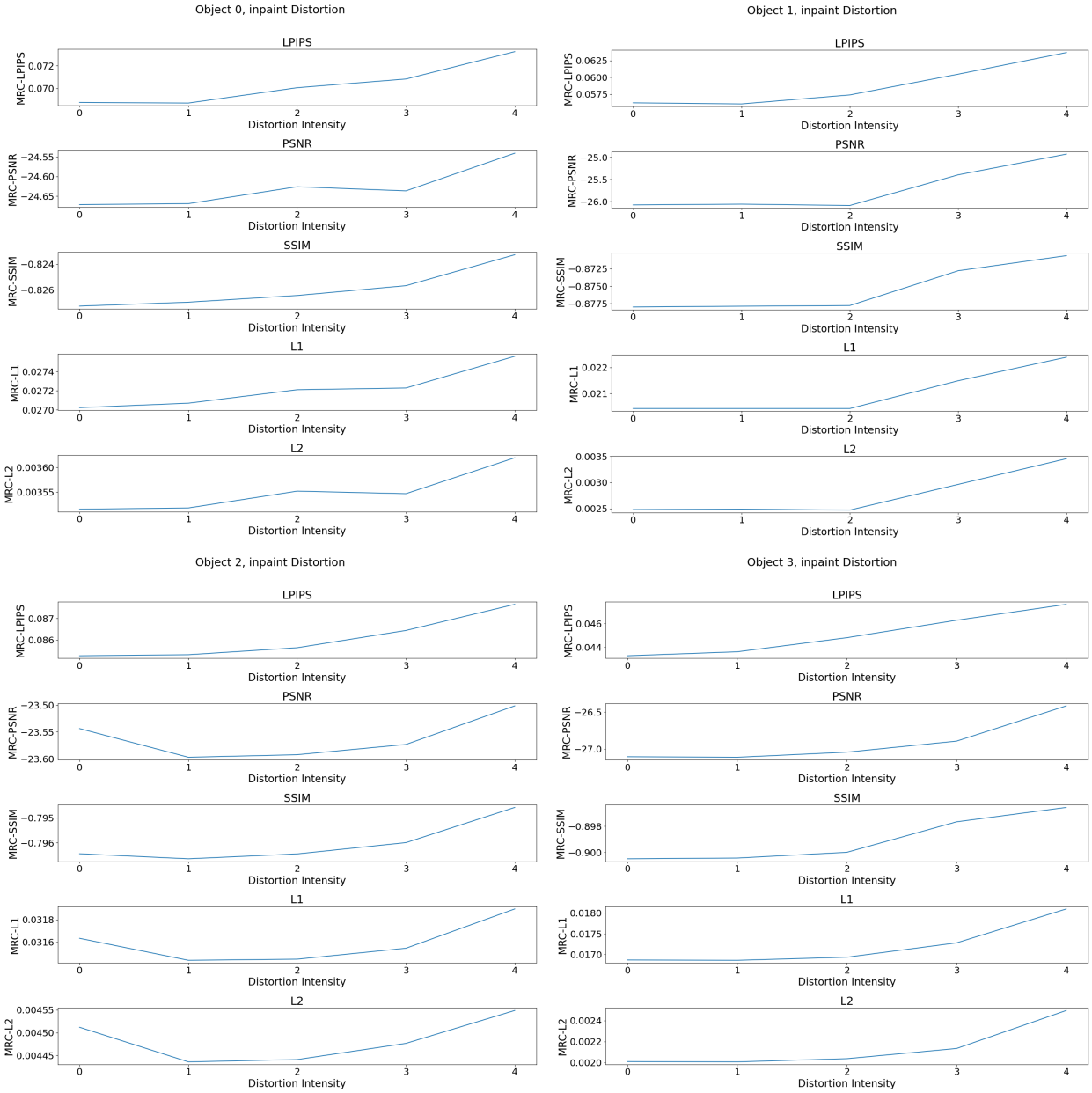

Figure 14. Quantitative correlation between five variants of MRC (our default LPIPS, as well as PSNR, SSIM, L1, and L2) and inconsistency introduced by inpaint distortion with increasing intensity on four objects. We take negative of the similarity metrics (PSNR and SSIM) for easy comparisons to the distance metrics (LPIPS, L1, and L2). LPIPS constantly exhibits monotonically increasing pattern with respect to the increased inconsistency, while other image metrics do not.

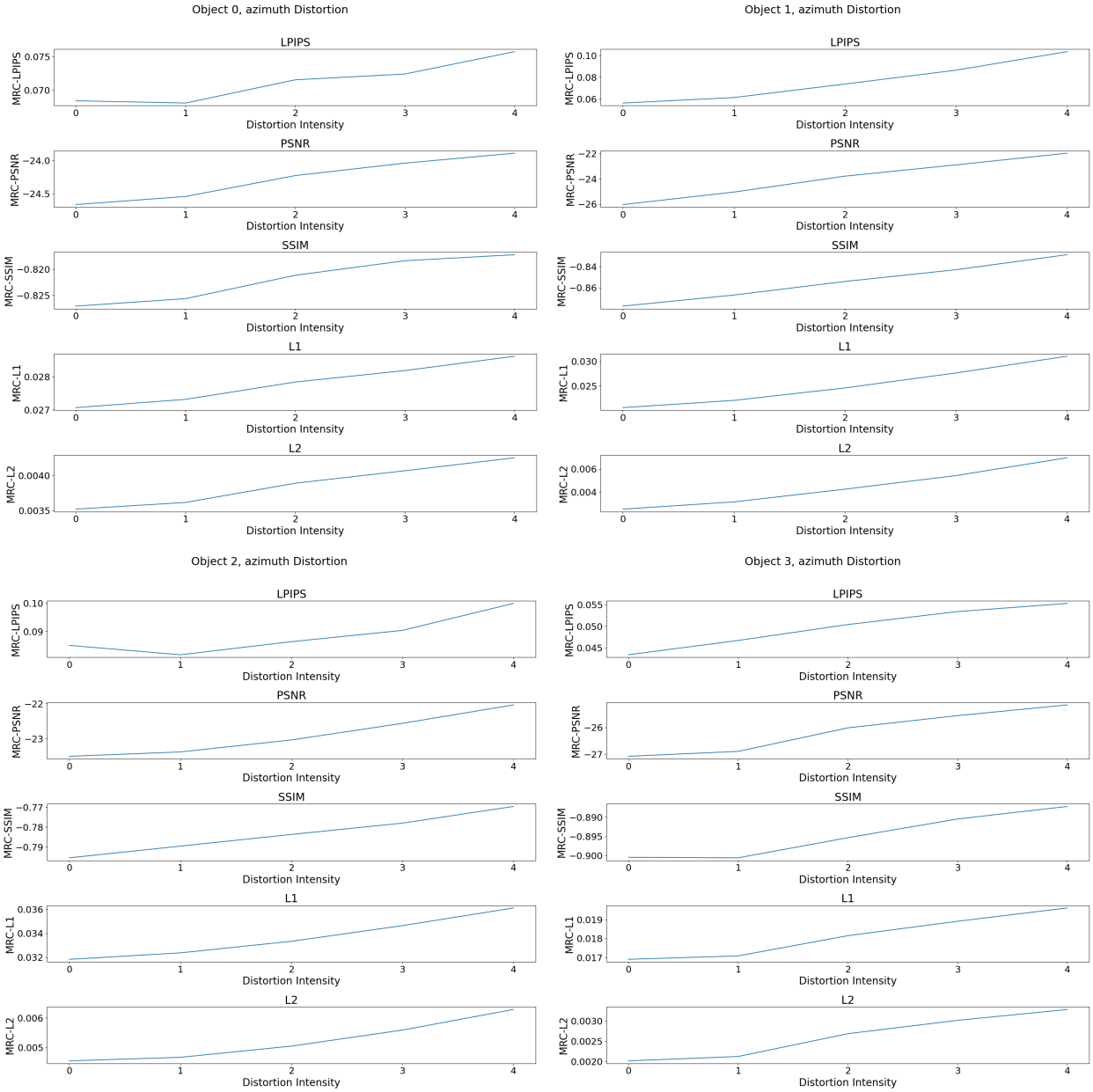

Figure 15. Quantitative correlation between five variants of MRC (our default LPIPS, as well as PSNR, SSIM, L1, and L2) and inconsistency introduced by azimuth rotation distortion with increasing intensity on four objects. We take negative of the similarity metrics (PSNR and SSIM) for easy comparisons to the distance metrics (LPIPS, L1, and L2). All metrics constantly exhibits monotonically, steadily increasing pattern with respect to the increased inconsistency.

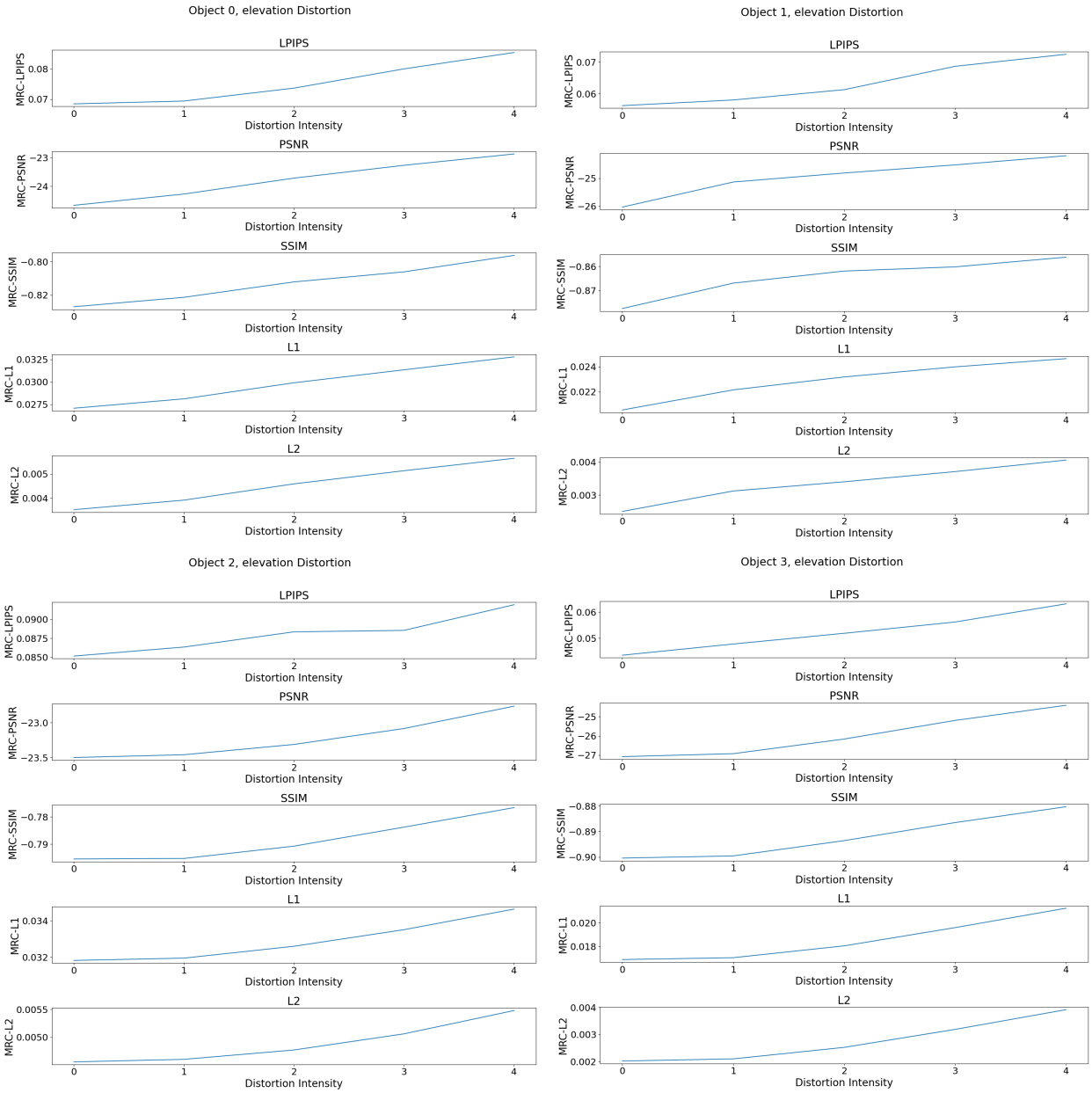

Figure 16. Quantitative correlation between five variants of MRC (our default LPIPS, as well as PSNR, SSIM, L1, and L2) and inconsistency introduced by elevation rotation distortion with increasing intensity on four objects. We take negative of the similarity metrics (PSNR and SSIM) for easy comparisons to the distance metrics (LPIPS, L1, and L2). All metrics constantly exhibits monotonically, steadily increasing pattern with respect to the increased inconsistency.