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

Multi-view diffusion models, obtained by applying Supervised Finetuning (SFT) to text-to-image diffusion models, have driven recent breakthroughs in text-to-3D research. However, due to the limited size and quality of existing 3D datasets, they still suffer from multi-view inconsistencies and Neural Radiance Field (NeRF) reconstruction artifacts. We argue that multi-view diffusion models can benefit from further Reinforcement Learning Finetuning (RLFT), which allows models to learn from the data generated by themselves and improve beyond their dataset limitations during SFT. To this end, we introduce Carve3D, an improved RLFT algorithm coupled with a novel Multi-view Reconstruction Consistency (MRC) metric, to enhance the consistency of multi-view diffusion models. To measure the MRC metric on a set of multi-view images, we compare them with their corresponding NeRF renderings at the same camera viewpoints. The resulting model, which we denote as Carve3DM, demonstrates superior multi-view consistency and NeRF reconstruction quality than existing models. Our results suggest that pairing SFT with Carve3D's RLFT is essential for developing multi-view-consistent diffusion models, mirroring the standard Large Language Model (LLM) alignment pipeline. Our code, training and testing data, and video results are available at: https://desaixie.github.io/carve-3d.

1. Introduction

Recently, significant progress has been made in generating 3D models from text prompts. Images generated by 2D diffusion models [9, 40, 42, 49, 52, 56] can be lifted to 3D representations. Numerous methods [24, 28, 46, 59] have demonstrated that a set of multi-view images is adequate for generating diverse and detailed 3D models, effectively mitigating the multi-face (Janus) problem. 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 Neural Radiance Field (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.

Existing multi-view diffusion models [24, 27, 28, 46, 59] primarily utilize Supervised Finetuning (SFT) with multi-view datasets derived from 3D datasets [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 towards the 3D dataset that has limited size and quality; thus, the distribution shift diminishes diversity, texture details, and realism of the generated results [24]. Such dilemma has been observed in Large Language Model (LLM) research. While SFT can shift the output distribution of pre-trained LLMs to follow instructions, the bias from the instruction dataset also introduces hallucination [44], preventing longer SFT. InstructGPT [36], the paper behind ChatGPT 3.5 [35], introduces Reinforcement Learning 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 bias from a SFT dataset.

We introduce Carve3D, an enhanced RLFT algorithm paired with a novel Multi-view Reconstruction Consistency (MRC) metric, to improve the consistency of multi-view diffusion models. Figs. 1 and 2 shows the capability and overview of Carve3D.

Our MRC metric compares the output multi-view images from a diffusion model with images rendered from the reconstructed NeRF at identical camera viewpoints. We use the sparse-view Large Reconstruction Model (LRM) [17, 24] to achieve fast, feed-forward NeRF reconstruction from

†This work is done while the author is an intern at Adobe Research.

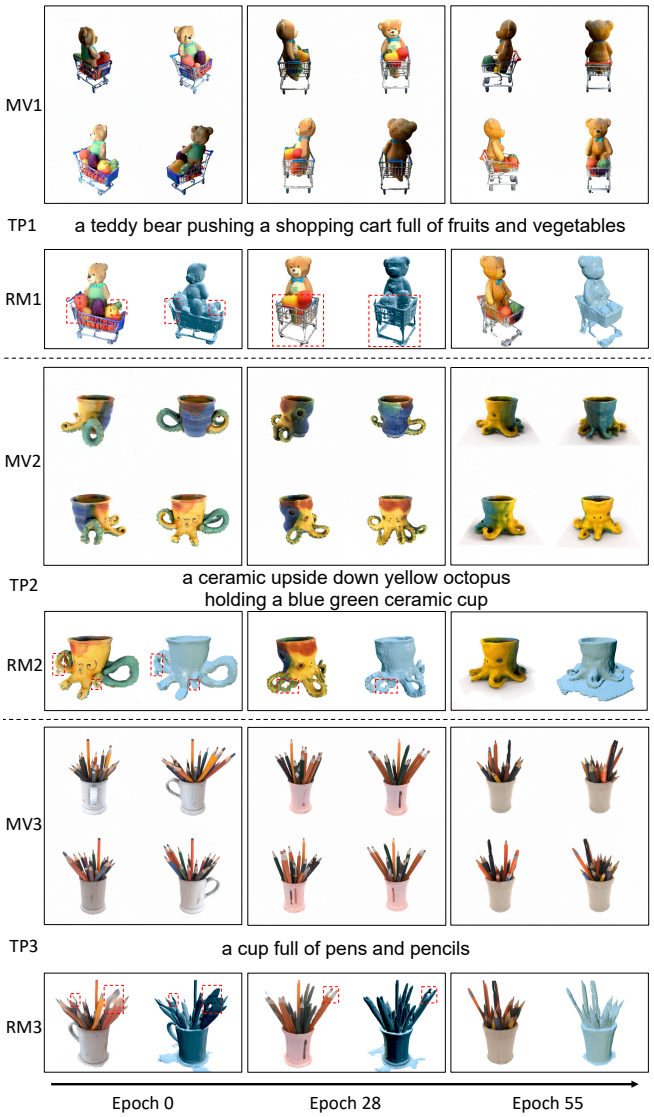

Figure 1. Our Carve3D algorithm steadily improves the 3D consistency of a multi-view diffusion model and the resulting quality of the NeRF and the mesh, without sacrificing its image-prompt alignment, texture details, or realism. Here, we show 3 testing-set results (in 3 rows, numbered as 1-3, separated by dotted lines) from the finetuning process (epoch 0, 28, and 55 in 3 columns). Each row includes the generated multi-view images (denoted as MV), the reconstructed NeRF and extracted mesh (denoted as RM) and the text prompt (denoted as TP). 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 in red).

a few multi-view images. To quantify image similarity, we adopt LPIPS [57] 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 (Fig. 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 training instability issue in RL, we opt for a purely on-policy policy gradient algorithm [54], 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].

By adopting our Carve3D RLFT algorithm on Instant3D-10K [24], a multi-view diffusion model supervised finetuned from SDXL [39], we obtain the Carve3D Model (Carve3DM). Through quantitative, qualitative experiments and a user study, we demonstrate that Carve3DM: (1) achieves improved multi-view consistency and NeRF reconstruction quality over Instant3D-10K, -20K, and -100K models, and (2) maintains similar prompt alignment, diversity, and realistic details as the base Instant3D-10K, preventing the degradation in Instant3D-20K and -100K. Our results indicate that pairing SFT with Carve3D's RLFT is essential for developing multi-view consistent diffusion models. In addition, we extend our MRC evaluation to existing models, revealing the universal presence of multi-view inconsistency when relying solely on SFT. Our work is the first application of RLFT to text-to-3D, especially on a 2.6B-parameter denoising UNet from SDXL [39]. By releasing our code, training and testing data, we hope this work will bolster the RLFT and alignment research in the computer vision community.

2. Related Works

Neural Radiance Field (NeRF) is a neural representation of 3D assets [8, 31, 33]. It infers the direction-dependent radiance at arbitrary volumetric positions with neural models. Many text-to-3D methods rely on it to produce 3D objects.

While text-to-image diffusion models are trained on 5

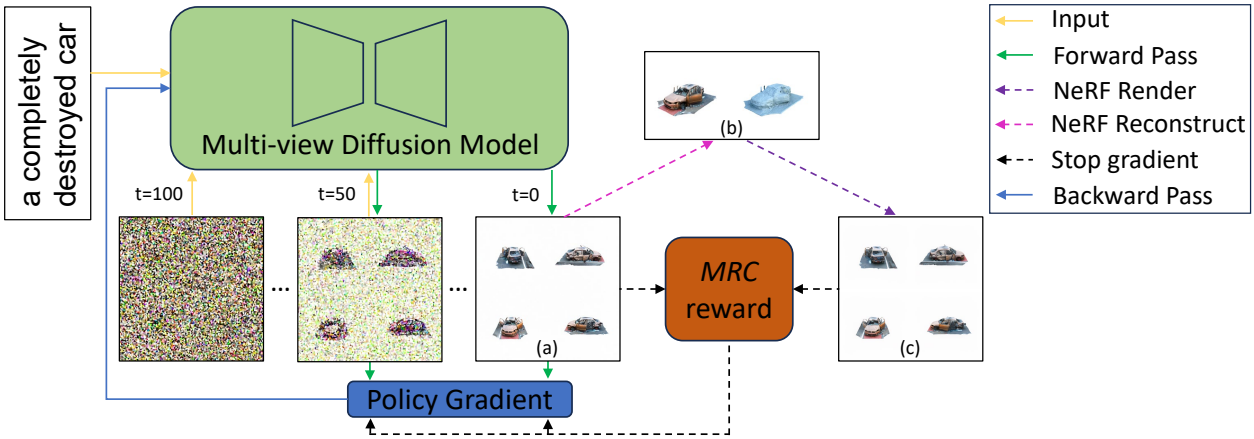

Figure 2. Overview of Carve3D. Given a prompt sampled from our curated prompt set and a initial noisy image, we iteratively denoise the image using the UNet. The final, clean image contains four multi-view images tiled in a 2-by-2 grid. MRC reward is computed by comparing (a) the generated multi-view 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 UNet's predictions, accumulated over all denoising timesteps. By using only a set of training text prompts, our RLFT algorithm finetunes the diffusion model by evaluating its own generated outputs, without relying on ground truth multi-view images.

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, 49, 52, 56] 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, 59] 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 Supervised Finetuning to convergence [24]. By adopting Reinforcement Learning Finetuning (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. While numerous methods have attained notable multi-view consistency by supervised finetuning 2D diffusion models [24, 27, 28, 46, 59], their evaluation has been empirical, lacking explicit metrics. An approach known as 3D consistency scoring [53] measures the consistency of output views by optimizing a NeRF trained on these views. Our MRC metric improves it with sparse-view reconstruction and comparing on all input views.

See Appendices F.1 and F.2 for related works on 3D Generation with 2D Diffusion Models and RLFT of Large Language Models and Diffusion Models.

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

A 3D model represented by Neural Radiance Field (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, 53] – 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 Appendix Fig. 7, 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 Appendix Fig. 7), they will produce a NeRF with broken ge-

ometry 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 Fig. 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 Appendix Listing 1.

Fast Sparse-view Reconstruction We conduct NeRF reconstruction with sparse-view Large Reconstruction Model (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 Appendix Fig. 7.

Image Distance Metric In Sec. 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 [57]. LPIPS exhibits smoother value changes with respect to the consistency of multi-view images compared to PSNR, SSIM, L1, and L2, as shown in Appendix Fig. 14. 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, 59] 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 Fig. 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 Appendix Fig. 9.

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.

Appendix Fig. 7 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. Appendix Fig. 8 presents the quantitative curve of the same experiment. MRC indeed shows a monotonically increasing pattern as the views become more inconsistent. As shown in Appendix Fig. 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 Sec. 3, we proposed a fast and reliable multi-view consistency metric named MRC, and in this section we describe how it can be used to finetune a multi-view diffusion model. We propose an Reinforcement Learning Finetuning (RLFT) algorithm for enhancing the consistency of 2D multi-view diffusion models, using the negative MRC as the reward function (Fig. 2). Building upon DDPO [5],

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

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:

$$s_t = (c, t, x_t), \tag{1}$$

$$a_t = x_{t-1}, \tag{2}$$

$$\pi(a_t|s_t) = p_\theta(x_{t-1}|c, t, x_t), \tag{3}$$

$$r(s_t, a_t) = \begin{cases} r(x_0, c) & \text{if } t = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

$$r(x_0, c) = -\text{MRC}(x_0) \tag{5}$$

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 (Appendix Listing 1) 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 [54] and Proximal Policy Optimization (PPO) [45]. DDPO_{SF} is based on the vanilla policy gradient algorithm, REINFORCE [54], also known as the Score Function (SF) of diffusion models. 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 (Appendix Eq. (9)), which are rewards normalized to have zero mean and unit variance, instead of directly using the rewards. By using the advantage term (A_r in Appendix Eq. (9)) in place of the reward, the DDPO_{SF} policy gradient function is:

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

where the expectation is taken over data generated by the policy π_θ with the parameters θ. The log probability $\log p_\theta(x_{t-1}|c, t, x_t)$ can be easily obtained since the policy is an isotropic Gaussian distribution when using the DDIM sampler [5, 48]. The DDPO_{IS} function (Appendix Eq. (10)) has an additional importance sampling term than Eq. (6).

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 Large Language Model (LLM) Reinforcement Learning from Human Feedback (RLHF) literature [2, 3, 36, 51, 55].

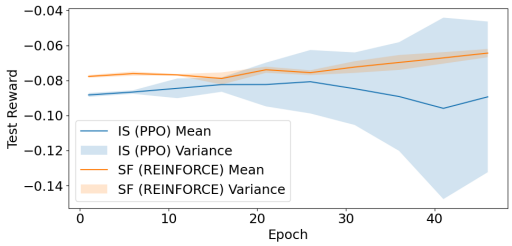

Figure 3. Comparing the IS and the SF versions of Carve3D reward curves on the testing set over 4 different random seeds. The IS version produces reward curves with high variance, including two runs that fails. In contrast, all runs of the SF version stably produces reward curves with low variance.

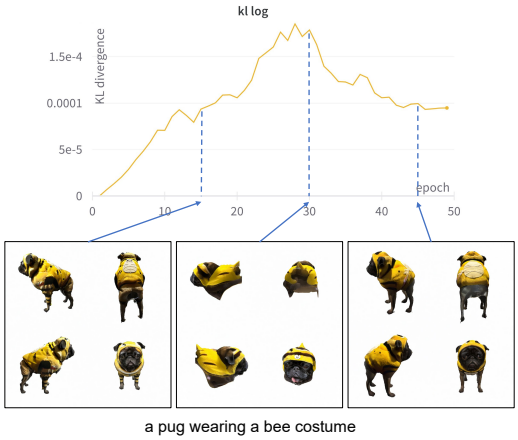

Figure 4. We observe qualitative correlation between the KL value and the prompt alignment degradation. Despite being distant in the finetuning process, epoch 15 and epoch 45, which have lower KL divergence to the base model, generates prompts that align better with the prompts. On the other hand, epoch 30, which has much higher KL divergence from the base model, generates results with broken identity, i.e. the body of the pug is missing.

4.2. Improvements over DDPO

While RLFT with the default DDPO_{IS} and our 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, 60] and traditional RL [14]. With the default DDPO_{IS}, our training process is evidently unstable, as shown in Fig. 3. 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 (Appendix Eq. (10)). As shown in Fig. 3, 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, 51, 55], we choose REINFORCE [54] (DDPO_{SF}) over PPO [45] (DDPO_{IS}) for its superior training stability. We provide two hypotheses behind our surprising finding in Appendix B.5, including the difficulty of the task reward function and the size of the model being finetuned. The favored use of REINFORCE [54] 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 Kullback–Leibler (KL) divergence between the log probabilities of the outputs from the base and the finetuned models. 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 (Fig. 4), suggesting us to adopt KL divergence regularization. We detail our KL divergence computation in Appendix B.4 and Appendix Eq. (11).

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:

$$A_{\text{KL}}(x_0, c) = \frac{\text{KL}(x_0|c, x_T) - \mu_{\text{KL}}(c)}{\sigma_{\text{KL}}(c)}. \tag{7}$$

Our advantage terms now consist of both the normalized reward (Appendix Eq. (9)) and the normalized KL divergence (Eq. (7)). Our final policy gradient function, used in our experiments, is a combination of Eqs. (6) and (7)

$$\hat{g}_{\text{SF,KL}} = \mathbb{E}\left[\sum_{t=0}^{T} \nabla_\theta \log p_\theta(x_{t-1}|c, t, x_t) \cdot (\alpha A_r(x_0, c) - \beta A_{\text{KL}}(x_0, c))\right] \tag{8}$$

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 Reinforcement Learning (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.

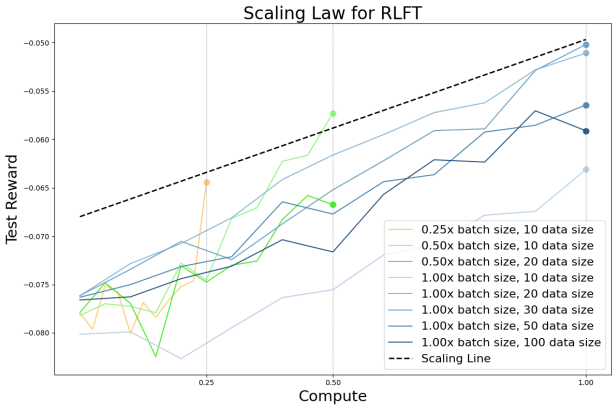

Figure 5. Scaling law for Carve3D's diffusion model RLFT algorithm. When we scale up the amount of compute, the model improves its reward smoothly under the optimal data size. The amount of compute scales linearly with respect to the batch size. The reward curves also become more stable (less variant) 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. 5); 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 Fig. 5, 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 [54]), the model steadily and smoothly improves its rewards throughout the finetuning process, meaning that more training epochs constantly lead to higher 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 (Fig. 4), we stop the finetuning 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 Fig. 5. See Appendix B.1 for the definition of epoch in RLFT, which is different from that in other contexts.

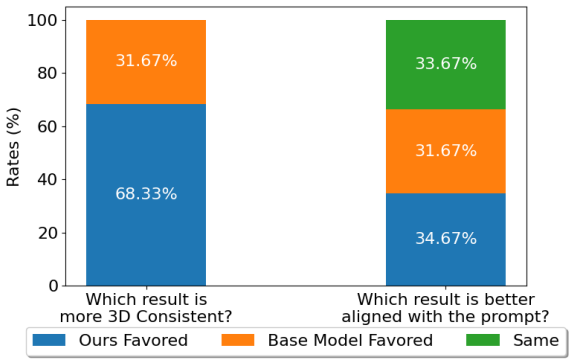

Figure 6. We conducted a user study with 20 randomly selected testing prompts and the corresponding outputs from both the base and fine-tuned model. 15 participants took part in the study, with a majority favoring the 3D consistency of our fine-tuned model. Opinions are evenly split on which has better prompt alignment.

5. Experiments

In this section, we evaluate the Carve3D Model (Carve3DM), obtained by applying the Carve3D Reinforcement Learning Finetuning (RLFT) algorithm on Instant3D [24]. See Appendix C.3 for our experiment settings and Appendix E for our ablation studies.

5.1. Comparison with Base Model and Longer SFT

As shown in Fig. 3, Carve3DM's Multi-view Reconstruction Consistency (MRC) reward steadily improves on the training set. We aim to further answer the following questions:

1. Does Carve3DM's improved MRC generalize to the testing set?
2. Quantitatively, is Carve3DM more consistent than the base model and the model with more Supervised Finetuning (SFT) steps?
3. Qualitatively, does Carve3DM sacrifice the diversity, prompt alignment, and texture details of the base model?

We quantitatively and qualitatively compare Carve3DM with Instant3D models with 10K, 20K, and 100K SFT steps, where Instant3D-10K is the default model in [24].

Quantitative Comparison and Generalization As shown in Tab. 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, i.e. Instant3D's 100K version is better than the 20K version, which is better than the 10K version. However, Carve3DM

	Avg MRC on Testing Set ↓
MVDream	0.1222
Instant3D-10K (Base)	0.0892
Instant3D-20K	0.0795
Instant3D-100K	0.0685
Carve3DM (Ours)	**0.0606**
Zero123++	0.0700
SyncDreamer	0.1018

Table 1. Quantitative comparison of MVDream [46], Instant3D [24] with 10K (the base model), 20K, and 100K SFT steps, Carve3DM (ours, finetuned from Instant3D-10K), Zero123++ [47], and SyncDreamer [28]. We evaluate them by generating 4 outputs for each of the 414 text prompts in the DreamFusion [40] testing set. We let Zero123++ and SyncDreamer to use one of Carve3DM's output multi-view images as their input image conditioning. Carve3DM achieves substantially better MRC than all baselines, indicating its superior multi-view consistency.

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, although our RLFT only uses 30 training prompts, it brings multi-view consistency improvement that generalizes to the testing set containing 415 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 Appendix Fig. 12 shows the improved multi-view consistency and the resulting Neural Radiance Field (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 Appendix Fig. 11, Carve3DM continues to show superior multi-view consistency and reduced NeRF artifacts, but such improvement is less obvious when compared to the 20K and 100K version of Instant3D [24], similar to our quantitative results in Tab. 1.

Prompt Alignment and Texture Details By virtue of our RLFT with KL-divergence regularization (Sec. 4.2.2), which prevents distribution shift, and our curation of training prompt dataset (Appendix C.1), Carve3DM preserves the prompt alignment and the texture details of the base model, as shown in Appendix Fig. 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 Appendix Fig. 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 Appendix Fig. 13, Carve3DM preserves the generation 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 (Fig. 2).

5.2. Comparison with Existing Methods

We further compare Carve3DM with the text-conditioned multi-view diffusion model, MVDream [46], and two image-conditioned models, Zero123++ [47] and SyncDreamer [28]. As shown in Tab. 1 and Appendix Fig. 11, Carve3DM's outputs have notably better multi-view consistency, realism, and level of details than the three baselines.

5.3. User Study

In addition to the quantitative and qualitative comparisons in Sec. 5.1, we conduct a user study to further understand the qualitative results of Carve3DM when perceived by human. As shown in Fig. 6, 68.33% of participants believe that Carve3DM'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, participants find that Carve3DM exhibits similar prompt alignment as Instant3D. The preservation of alignment can be attributed to the KL divergence regularization (Sec. 4.2.2) and early stopping the RLFT regarding KL divergence (Sec. 4.2.3). See Appendix C.4 for more user study details.

6. Conclusion

In this paper, we propose Carve3D, a Reinforcement Learning Finetuning algorithm to improve the reconstruction consistency of 2D multi-view diffusion models. Carve3D relies on MRC, a novel metric that measures the reconstruction consistency by comparing multi-view images with their corresponding NeRF renderings at the same viewpoints. The resulting model, Carve3DM, demonstrates substantially improved multi-view consistency and NeRF quality without sacrificing the prompt alignment, texture details, or prompt alignment of the base model. Our results conclude that pairing SFT with Carve3D's RLFT is essential for developing multi-view-consistent diffusion models.

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

References

[1] Adobe. Adobe firefly. https://www.adobe.com/sensei/generative-ai/firefly.html, 2023. Accessed: 2023-11-15. 4

[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. 5, 6, 1, 4

[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. 5, 6, 1, 4

[4] Kevin Black. ddpo-pytorch. https://github.com/kvablack/ddpo-pytorch, 2023. Accessed: 2023-11-17. 2, 3

[5] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2023. 2, 4, 5, 6, 8, 1, 3

[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. 5

[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. 6, 5

[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, 4

[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, 6

[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, 4, 8

[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. 6

[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, 5, 6, 3, 4

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[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. 1, 4, 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, 4

[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. 5

[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, 5

[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. 5

[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, 6, 7, 8, 5

[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. 1

[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, 4, 8, 7

[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. 6

[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, 3, 4

[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. 5

[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, 4

[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, 4

[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, 5, 6, 8, 4

[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. 3

[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, 8, 1, 4, 6

[40] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 1, 3, 8, 2

[41] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2023. 4, 6

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 3, 5

[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, 5

[45] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. 2, 5, 6, 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, 4, 8, 7

[47] Ruoxi Shi et al. Zero123++: a single image to consistent multi-view diffusion base model, 2023. 8, 3, 7

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 5

[49] 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

[50] 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. 3

[51] 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. 5, 6, 1, 3

[52] 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

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

[54] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. 2, 5, 6, 7, 1

[55] 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. 5, 6, 1, 3

[56] 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

[57] 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, 4, 3

[58] 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

[59] 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, 4

[60] 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. 6, 5