2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# Fine-tuning the Diffusion Model and Distilling Informative Priors for Sparse-view 3D Reconstruction

Jiadong Tang[1,2], Yu Gao[1,2], Tianji Jiang[1,2], Yi Yang[*,1,2], Mengyin Fu[1,2]

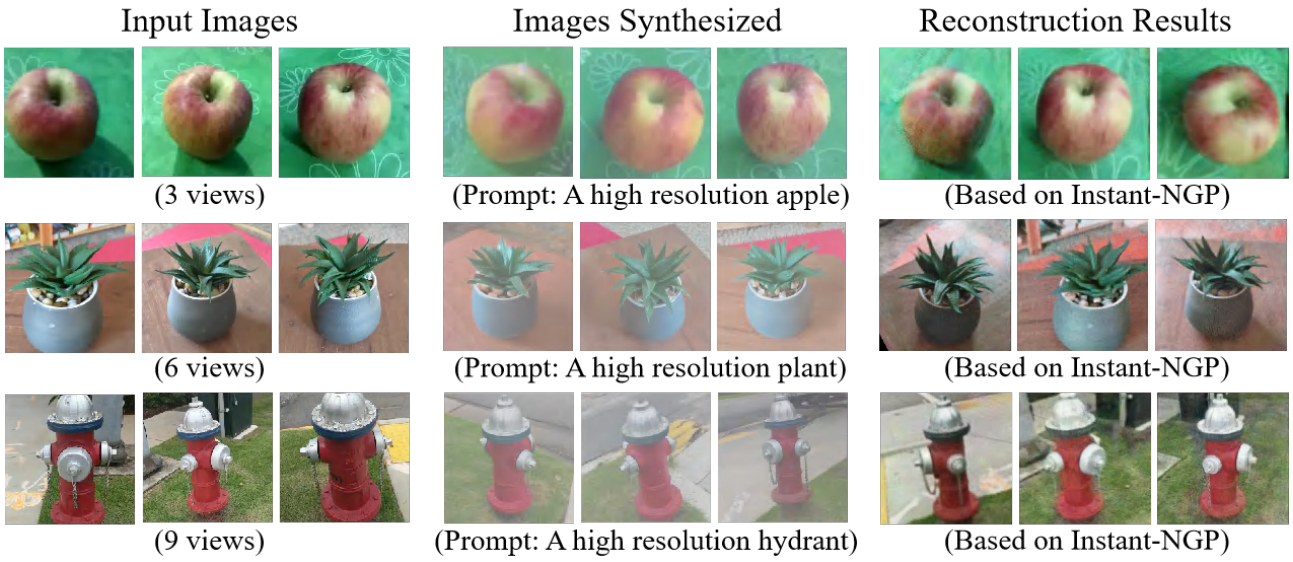


Fig. 1: **Some sparse-view reconstruction results of our method.** Given a few input images with poses and a simple text prompt, our approach is able to generate 3D-aware images under novel views with a fine-tuned diffusion model. Then we distill informative priors from the diffusion model to optimize a 3D model(e.g. Instant-NGP).

***Abstract*— 3D reconstruction methods such as Neural Radiance Fields (NeRFs) are capable of optimizing high-quality 3D representation from images. However, NeRF is limited by the requirement for a large number of multi-view images, making its application to real-world scenarios challenging. In this work, we propose a method that can reconstruct real-world scenes from a few input images and a simple text prompt. Specifically, we fine-tune a pretrained diffusion model to constrain its powerful priors to the visual inputs and generate 3D-aware images, leveraging the coarse renderings obtained from input images as the image condition, along with the text prompt as the text condition. Our fine-tuning method saves a significant amount of training time and GPU memory usage while also generating credible results. Moreover, to enable our method to have self-evaluation capabilities, we design a semantic switch to filter out generated images that do not match real scenes, ensuring that only informative priors from the fine-tuned diffusion model are distilled into the 3D model. The semantic switch we designed can be used as a plug-in and improve performance by 13%. We perform our approach on a real-world dataset and demonstrate competitive results compared to existing sparse-view 3D reconstruction methods. Please see our project page for more visualizations and code: https://bityia.github.io/FDfusion.**

*This work was partly supported by National Key R&D Program of China (2022YFC2603600) and National Natural Science Foundation of China (Grant No. NSFC 62233002)

[1]School of Automation, Beijing Institute of Technology, Beijing, China
[2]National Key Lab of Autonomous Intelligent Unmanned Systems, Beijing Institute of Technology, Beijing, China
*Corresponding author: Y. Yang Email: yang_yi@bit.edu.cn

## I. INTRODUCTION

Visual 3D reconstruction occupies a pivotal position in fields such as digital twin [1]–[3] and robotics [4], [5]. Recently, the emergence of Neural Radiance Field (NeRF) [6] has sparked enthusiasm in the community for 3D reconstruction. NeRF demonstrates remarkable potential in 3D representation and depth estimation. However, high-quality NeRF relies on a large number of input views. In real-world applications, obtaining substantial multi-view images within most scenes is challenging, making it difficult for NeRF to reconstruct the real scene accurately.

To overcome NeRF's dependency on multiple views, some work utilize regularization methods to regularize geometric shape and eliminate floating artifacts, such as frequency regularization [7], semantic consistency regularization [8], geometric and appearance regularization [9], and perceptual regularization [10]. These approaches can improve the quality of reconstruction results. However, when the render viewpoint vastly differs from the input viewpoint, especially in unseen regions, the result shows substantial degradation.

With the powerful image generation capabilities, diffusion models [11] can be used to generate images of unseen areas. Although the generated images struggle to guarantee 3D consistency, introducing diffusion models into the 3D reconstruction pipeline can help optimize 3D models. Some works [12]–[16] are capable of conducting image synthesis

using as few as one input image and generating a 3D model, yet they are limited in reliability, potentially resulting in significant differences between the generated 3D model and the real scene. Other work [17]–[20] attempt to train a diffusion model on extensive datasets to learn priors for generating novel view images and distill them to a plausible 3D representation. This also leads to high training costs that are generally unaffordable. Moreover, existing works tend to neglect the impact of the quality of generated images on the diffusion distillation.

To address the issues mentioned above, we propose a method to enhance the sparse-view 3D reconstruction pipeline by fine-tuning a pretrained diffusion model to generate 3D-aware images and distilling informative priors to optimize a 3D model. First, we obtain 3D-aware coarse renderings of novel views based on a feature extractor and the input images with poses. Second, We add an image condition to the pretrained diffusion model as an additional condition channel. Then we utilize LoRA [21] to fine-tune the diffusion model with the coarse renderings as the image condition and a simple text prompt as the text condition, leveraging the strong priors to generate photorealistic images that match the real scene from the novel viewpoint. Furthermore, we optimize the diffusion distillation process by designing a semantic switch, which calculate the similarity between the generated images and the input images, thereby filtering out those generated images that greatly differ the real scene and improving the quality of 3D model.

We demonstrate our approach on a real-world dataset. Fig.1 shows that our method performs well in scenarios of varying complexity. The main contributions of our work are the following:

- We propose a method of fine-tuning the pretrained diffusion model as a cost-effective approach for 3D-aware image synthesis. Incorporated with LoRA and image condition, we can fine-tune the diffusion model to constrain its strong priors to the visual inputs.
- We design a semantic switch to distill informative priors from the fine-tuned diffusion model, enabling our method to have self-evaluation capabilities. The semantic switch can select the generated images that most closely match the real scenes for diffusion distillation, thereby obtaining a high-quality 3D model.

## II. RELATED WORK

### A. Sparse-view NeRF

Reducing the reliance of Neural Radiance Fields on the number of input views is crucial for broadening the application of NeRF in real-world scenarios. Many existing methods focus on regularization constraints to enable NeRF to learn the correct geometric and appearance. FreeNeRF [7] mitigates the occurrence of "floaters" during training by gradually unlocking the high-frequency components in positional encoding and penalizing the density field close to the camera. RegNeRF [9] assumes the world to be segmented and smooth, and therefore proposes a depth smoothness loss to regularize the geometry and appearance of patches rendered from unseen regions. PixelNeRF [22] utilizes image features from the input views, obtaining features for the target view through projection and bilinear interpolation, enabling 3D reconstruction from sparse-view. DietNeRF [8]'s strategy is leveraging a pretrained Vision Transformer to predict the semantic features of both the input view and the novel view, ensuring their consistency. SparseNeRF [23] obtains depth priors from pre-trained depth models or depth sensors, employing depth ranking regularization and spatial continuity regularization to provide precise geometric constraints. VipNeRF [24] introduces a binary visibility map for each pixel, and use this map as supervisory information for each pair of input views to constrain the training of NeRF.

Although these methods can correctly constrain the geometric shape, reducing floaters and artifacts during NeRF optimization, they still tend to underperform in unobserved regions when the number of viewpoints is extremely limited.

### B. Diffusion model in 3D reconstruction

Due to the outstanding generative capabilities, diffusion models emerge as a powerful method to provide additional priors for 3D reconstruction. DreamFusion [25] proposes Score Distillation Sampling (SDS), enabling text-to-3D generation through a 2D text-to-image diffusion model [26], but the generated results face issues such as oversaturation and lack of detail. Zero-1-to-3 [12] fine-tunes a pre-trained diffusion model to allow it to learn a mechanism for controlling camera viewpoint transformations, but its application is limited to single objects without backgrounds. DiffusioNeRF [27] utilizes a Denoising Diffusion Model(DDM) to learn the gradients of logarithms of the RGBD patch distribution, which guides the update of the color and density fields during the training process, ensuring that the generated RGBD patch aligns more consistently with the distribution learned by the DDM. SparseFusion [17] employs an epipolar feature transformer and a view-conditioned diffusion model to generate 3D-aware images, followed by diffusion distillation for extracting 3D modes. However, similar to DiffusioNeRF, SparseFusion exhibits poor generalizability to in-the-wild data. The most recent work, ReconFusion [20], achieves remarkable performance by training a diffusion model on enormous amount of real-world datasets, while the training cost is also extremely high.

## III. METHOD

Our method consists of a fine-tuned diffusion model for 3D-aware novel view images synthesis, coupled with an informative distillation process based on the diffusion model for 3D reconstruction. We introduces the details of fine-tuning the diffusion model(Section III-A), and describes how we distill informative priors from the diffusion model for sparse-view 3D reconstruction(Section III-B).

### A. 3D-aware Novel View Synthesis

The pretrained text-to-image diffusion model takes a single image or random noise as input and uses a text prompt as

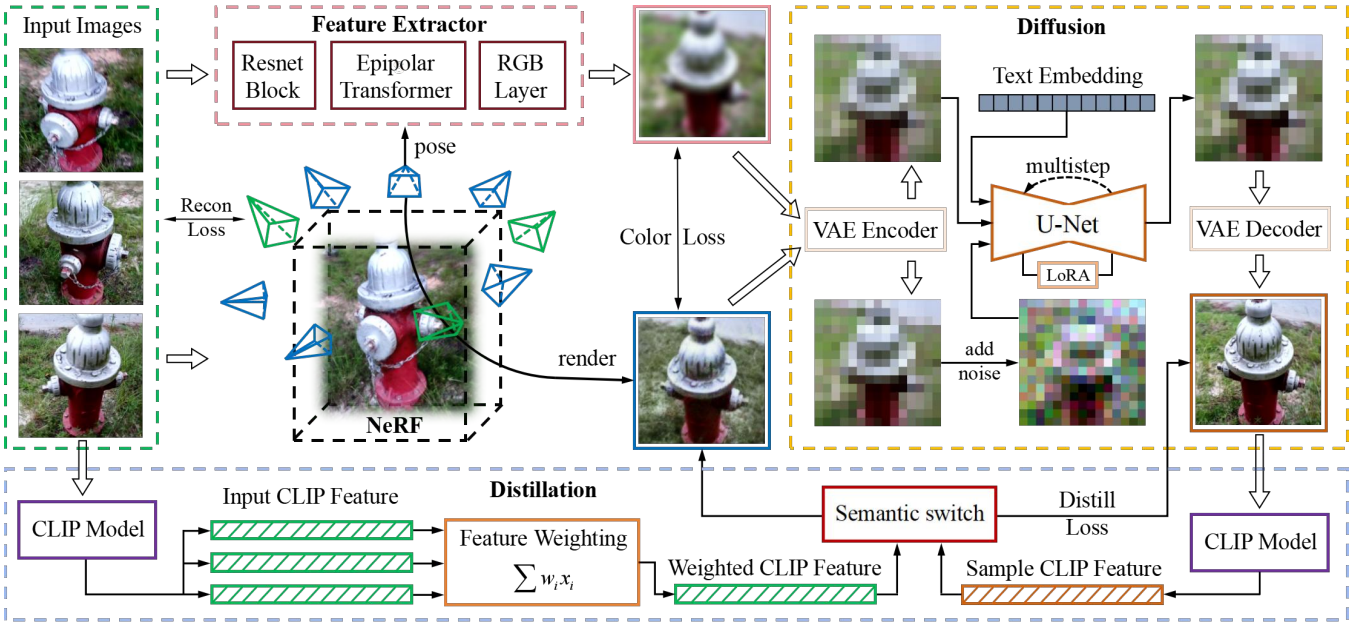


Fig. 2: **The 3D reconstruction pipeline of our method.** Our goal is to optimize a 3D model represented by NeRF, given sparse input views. First, we minimize a reconstruction loss $\mathcal{L}_{\text{Recon}}$ between the rendered images and the real images under the input viewpoints. Then, given input images and a sampled novel view, we use a feature extractor to obtain a coarse RGB rendering(denoted as EFT image). We minimize a color loss $\mathcal{L}_{\text{Color}}$ between the image rendered by NeRF and the EFT image under the sampled novel viewpoint. We take the rendered image as the input for the diffusion model, along with the EFT image as the image condition and text prompt as text condition, then we can have a generated image. Finally, we compute the CLIP features of the generated image and input images. When their similarity exceeds a threshold, we turn on the semantic switch and minimize a distillation loss $\mathcal{L}_{\text{Distill}}$ between the rendered image and the generated image.

the condition. This makes it challenging to maintain 3D consistency between the generated images and the input image, relying on sufficiently precise text prompts. Consequently, we divide the novel view synthesis into two stages: first, we use a feature extrator to generate a coarse RGB rendering of the target view. This coarse RGB rendering, along with the corresponding text prompt, is then fed into the fine-tuned diffusion model to generate 3D-aware image.

*1) Epipolar Feature Transformer:* Referring to SparseFusion's approach, we utilize a ResNet [28] and an Epipolar Feature Transformer(EFT) [17] to extract features from input posed images, and take the output of the RGB layer as a corase rendering under the target view $\pi$. During the training process, Epipolar Feature Transformer optimizes a network $f_\phi$ by minimizing the photometric error between the rendered images and the ground truth images. The loss function for training the EFT is:

$$\mathcal{L}_{\text{EFT}}(\phi) = \sum_{r \in R(\pi)} \| f(r, C, \Pi) - G(r) \|^2 \qquad (1)$$

where $r$ is a sampled ray from the ray sets $R(\pi)$ under view $\pi$. $C$ is the set of training images while $\Pi$ is the set of their corresponding poses, and $G(r)$ is the ground truth pixel value of ray $r$.

The implementation of the EFT ensures 3D awareness and enables our method to support an arbitrary number of input views. The obtained RGB coarse rendering will serve as the image condition of the diffusion model, guiding the denoising process.

*2) Fine-tuning a Diffusion Model:* Training a diffusion model from scratch for novel view synthesis is computationally expensive. Therefore, we fine-tune the diffusion model with a strategy called Low-Rank Adaptation(LoRA) [21] for 3D-aware image synthesis. Large pretrained diffusion models like Stable Diffusion possess a vast amount of parameters, enabling them to learn and store extensive information during training and generate photorealistic images. LoRA freezes the weights of pretrained models(denoted $\theta_0$) and injects trainable layers(denoted as $\theta_*$) into transformer blocks and convolution blocks. This approach significantly reduces the number of trainable parameters and the GPU memory requirements(see Table II). By utilizing LoRA for fine-tuning, diffusion models can generate images more consistent with the distribution of the datasets while still retaining the broad knowledge and generative capabilities acquired during their original training. For convenience, in the following text, we will also use "LoRA" to represent the trainable layers injected during the fine-tuning process.

Considering the challenge of generating images that match real scenes using only text prompts as conditions, we augment the pretrained diffusion model with a image condition. Specifically, we use the coarse RGB rendering obtained from the feature extractor as the image condition during denoising. In order to fuse the input image of diffusion model

with the image condition, we initialize a convolution layer $conv\_in_2$ identical to the UNet's $conv\_in$ layer. The $conv\_in$ layer is used to process the input image while $conv\_in_2$ layer is used to process the condition image. The output of two convolution layers are aggregated before being passed through the subsequent modules of the UNet. In addition, we set a simple text prompt $p$ for each scene, "a high resolution *", "*" represents the category of objects contained in this scene. As shown in Fig.3, with image condition, the diffusion model can generate images that are more closely match the real scenes. The loss function for fine-tuning the diffusion is:

$$\mathcal{L}_{\text{Diff}}(\theta_*) = \mathbb{E}_{z,\epsilon,t,y,p} \left\| \epsilon - \epsilon_\theta(z_t, t, z_y, p) \right\|^2 \tag{2}$$

where $\theta = \theta_0 + \theta^*$, $\theta$ represents the weight of diffusion model injected with LoRA. $t \in \{1, 2, ..., T\}$ is the denosing timesteps, $\epsilon \sim \mathcal{N}(0, 1)$, $z_t = \alpha_t z + \sigma_t \epsilon$ is the noisy latent at timestep $t$.


Fig. 3: **Comparison before and after finetune.** The fine-tuned diffusion model is capable of generating photographic images that match real scenes.

### B. 3D Reconstruction via Informative Diffusion Distillation

NeRF can optimize 3D-consistent models, but floater artifacts may occur when the input viewpoints are sparse. The fine-tuned diffusion model can generate photorealistic novel view images conditioned on the coarse RGB renderings and the text prompts. Thus, we propose a 3D reconstruction pipeline(see Fig.2) via diffusion distillation to keep NeRF's performance under sparse-views.

*1) Reconstruction Loss:* The essence of NeRF is to learn and reconstruct spatial information by utilizing cross light constraints generated from multi-view images [29]. Given a set of posed images, NeRF optimizes a randomly initialized 3D model by minimizing the pixel error between the rendered images and the ground truth images. The loss function for reconstruction is:

$$\mathcal{L}_{\text{Recon}}(\psi) = \sum_{r \in R(\pi_0)} \left\| g(r) - G(r) \right\|^2 \tag{3}$$

where $r$ is a sampled ray from the ray sets $R(\pi_0)$ under input view $\pi_0$. $\psi$ represents the parameters of the NeRF $g$. $G(r)$ is the ground truth pixel value of ray $r$.

*2) Color Loss:* The reconstruction loss ensures that NeRF renders high-quality images from the input views. However, for novel views unobserved in the inputs, the performance of NeRF will greatly degenerate. To enable NeRF to rapidly converge to the correct geometric shape, we incorporate color loss during the early stage of training. We sample a novel view $\pi'$ and obtain a rendered image $x(\pi')$ using the neural radiance field $g$. In the meanwhile, we utilize the trained feature extractor to obtain a coarse RGB rendering $x_c(\pi')$ based on the input images and the sampled novel viewpoint. Then we minimize the color loss between the rendered image $x$ and the coarse RGB rendering $x_c$, and the loss function is:

$$\mathcal{L}_{\text{Color}}(\psi) = \sum_{\pi' \in \Pi'} \left\| x_c(\pi') - x(\pi') \right\|^2 \tag{4}$$

where $\psi$ represents the parameters of the NeRF $g$, and $\Pi'$ is the set of novel views we want to sample.

*3) Informative Distillation:* Reconstruction loss and color loss can help us regularize the approximate geometric and appearance of the scene, but obtaining high-quality 3D models and realistic renderings is beyond their capabilities. Consequently, we distill the 2D priors from the fine-tuned diffusion model to further optimize the NeRF. Specifically, we add Gaussian noise $\epsilon$ to the latent variable $z_t$ of the rendered image $x$, and the noisy latent is used as the input of the diffusion model. The text embedding, obtained by passing the text prompt through a text encoder, along with the coarse RGB rendering $x_c$ in the latent space, serves as the text condition and the image condition for the diffusion model. We perform k-step denoising through DDIM sampling to obtain a latent sample $z_0$. $z_0$ is decoded into pixel space as a generated image $\hat{x}_0$, which is used to supervise the optimization of NeRF.

Despite the fact that the fine-tuned diffusion model is able to produce 3D-aware images, it is inevitable that some generated images will still significantly differ from the actual scenes. If these generated images are also involved in the diffusion distillation process, they could negatively affect the final reconstruction results. To mitigate the above situation, we incorporate a self-evaluation mechanism into the diffusion distillation process. We utilize the CLIP model [30] to calculate the CLIP feature for each input image, and then perform a weighted summation to yield the CLIP feature for the input image set, denoted as $F_{CLIP}(X)$. Meanwhile, we calculate the CLIP feature for the generated images $\hat{x}_0$, denoted as $F_{CLIP}(\hat{x}_0)$. We calculate the cosine similarity between $F_{CLIP}(X)$ and $F_{CLIP}(\hat{x}_0)$. If it exceeds the threshold $\gamma$, then we compute the loss between the rendered image $x$ and the generated image $\hat{x}_0$. The loss function for distillation is:

$$\mathcal{L}_{\text{Distill}}(\psi) = \mathbb{E}_{\pi,\epsilon,t} \left\| \hat{x}_0^* - x \right\|^2 \tag{5}$$

where $\psi$ represents the parameters of the NeRF and $t$ is the denosing timestep. $\hat{x}_0^*$ represents those $\hat{x}_0$ satisfy the $cos\_sim(F_{CLIP}(C), F_{CLIP}(\hat{x}_0)) > \gamma$.

## IV. EXPERIMENTS

We demonstrate the performance of our approach for sparse-view 3D reconstruction on a challenging real world dataset CO3Dv2(Sec.IV-B). Additionally, we conduct two ablation studies on the process of diffusion model fine-tuning and informative distillation to validate the effectiveness of our method.(Sec.IV-C).

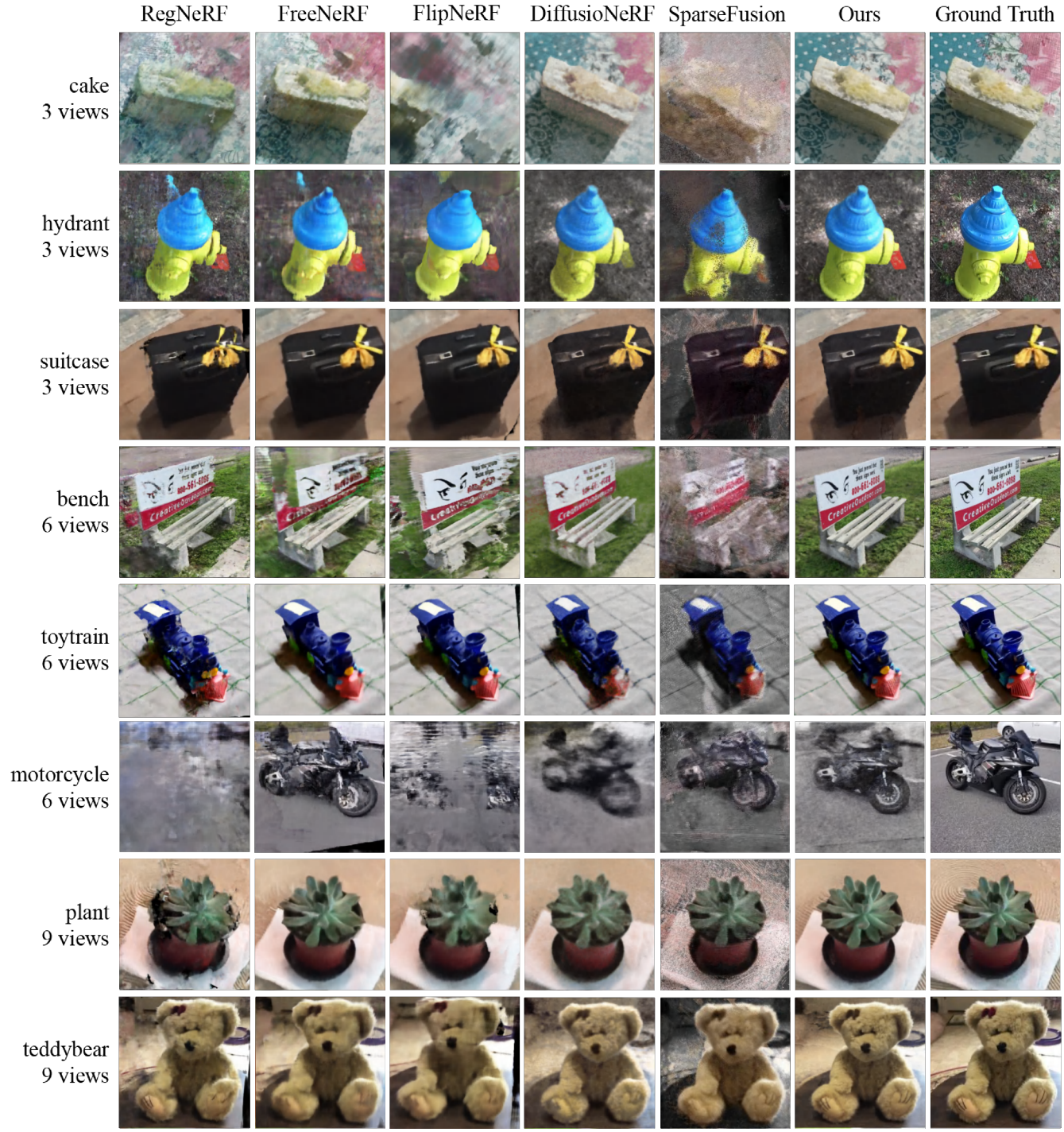


Fig. 4: **Qualitative results of sparse-view 3D reconstruction on real-world dataset.** We show reconstruction results for a cake, hydrant, and suitcase with 3 views; a bench and toytrain with 6 views; and a plant and teddybear with 9 views. Corresponding metrics can be found in Table I.

### A. *Experiment Setup*

**Dataset.** We perform experiments on a multi-view real-world dataset CO3Dv2 [31], which contains 51 object categories and 1.5 million camera-annotated frames. To evaluate performance in real-world scenarios, we do not employ the foreground masks provided by the dataset in our experiments. Furthermore, to standardize the input for the diffusion model, we crop and scale all images to the size of 256x256 based on their bounding boxes. Considering the substantial computational effort required to optimize NeRFs across all scenes, we perform our experiments on 8 representative categories by evaluating 5 scenes per category.

**Baselines.** We compare our approach against several state-

TABLE I: **Quantitative results of sparse-view 3D reconstruction comparing our method with baseline methods.**

| Category | Method | PSNR ↑ | | | SSIM ↑ | | | LPIPS ↓ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 3 views | 6views | 9views | 3 views | 6views | 9views | 3 views | 6views | 9views |
| Bench | RegNeRF | 13.8917 | 15.8855 | 16.9222 | 0.5055 | 0.5446 | 0.5591 | 0.5889 | 0.5623 | 0.5389 |
| | FreeNeRF | 13.8772 | 16.7826 | 18.4504 | 0.4215 | 0.4711 | 0.5248 | 0.5754 | 0.5217 | 0.5028 |
| | FlipNeRF | 13.9868 | 15.7599 | 17.4873 | 0.3916 | 0.4268 | 0.4716 | 0.5834 | 0.5643 | 0.5365 |
| | DiffusioNeRF | 15.0229 | 17.8902 | 19.5836 | 0.4355 | 0.4612 | 0.5141 | 0.5829 | 0.5613 | 0.5213 |
| | SparseFusion | 13.3005 | 15.6744 | 17.7384 | 0.5134 | 0.5558 | 0.5780 | 0.6905 | 0.6626 | 0.6180 |
| | Ours | 17.4638 | 18.9967 | 20.1751 | 0.4742 | 0.5462 | 0.5693 | 0.5869 | 0.5602 | 0.5190 |
| Cake | RegNeRF | 13.6127 | 17.9262 | 21.1080 | 0.6149 | 0.7004 | 0.7779 | 0.5374 | 0.4759 | 0.4354 |
| | FreeNeRF | 16.7085 | 17.4639 | 20.4103 | 0.5467 | 0.5717 | 0.6379 | 0.4728 | 0.4596 | 0.4447 |
| | FlipNeRF | 15.4359 | 19.2940 | 22.1179 | 0.5433 | 0.6399 | 0.7209 | 0.4956 | 0.4412 | 0.4043 |
| | DiffusioNeRF | 18.9417 | 20.2887 | 22.0839 | 0.5935 | 0.6353 | 0.6547 | 0.4658 | 0.4314 | 0.4105 |
| | SparseFusion | 14.8113 | 18.4691 | 19.0189 | 0.4871 | 0.6741 | 0.7229 | 0.6118 | 0.5531 | 0.5318 |
| | Ours | 19.9437 | 23.1534 | 24.7163 | 0.5943 | 0.6830 | 0.7681 | 0.5756 | 0.5223 | 0.4418 |
| Hydrant | RegNeRF | 12.0897 | 16.0621 | 17.9589 | 0.4295 | 0.5049 | 0.5985 | 0.6664 | 0.5357 | 0.5019 |
| | FreeNeRF | 11.1392 | 17.3728 | 18.3820 | 0.1942 | 0.4554 | 0.4792 | 0.6734 | 0.5517 | 0.5214 |
| | FlipNeRF | 13.0239 | 15.7213 | 17.2307 | 0.2420 | 0.3607 | 0.4439 | 0.6435 | 0.6051 | 0.5592 |
| | DiffusioNeRF | 16.2308 | 17.5758 | 18.2781 | 0.3376 | 0.3676 | 0.3842 | 0.6003 | 0.5942 | 0.5627 |
| | SparseFusion | 14.5405 | 15.7581 | 17.2892 | 0.3033 | 0.3620 | 0.3725 | 0.5939 | 0.5736 | 0.5425 |
| | Ours | 16.3130 | 18.3053 | 18.8614 | 0.3785 | 0.4691 | 0.4952 | 0.5978 | 0.5341 | 0.5109 |
| Motorcycle | RegNeRF | 12.6152 | 13.9168 | 16.0053 | 0.3663 | 0.4658 | 0.5287 | 0.6216 | 0.5907 | 0.5756 |
| | FreeNeRF | 11.2851 | 15.8947 | 17.1524 | 0.2115 | 0.3608 | 0.4411 | 0.6354 | 0.5561 | 0.5384 |
| | FlipNeRF | 12.2329 | 13.2103 | 13.9805 | 0.2550 | 0.2788 | 0.3027 | 0.6301 | 0.6087 | 0.5999 |
| | DiffusioNeRF | 13.2106 | 15.4030 | 18.6461 | 0.2411 | 0.3185 | 0.3814 | 0.7395 | 0.6330 | 0.6115 |
| | SparseFusion | 10.7963 | 13.7981 | 14.5926 | 0.1805 | 0.2483 | 0.2890 | 0.7213 | 0.6402 | 0.6170 |
| | Ours | 15.3376 | 16.7334 | 17.2701 | 0.3022 | 0.3638 | 0.3902 | 0.6407 | 0.5866 | 0.5632 |
| Plant | RegNeRF | 13.4566 | 14.8046 | 17.7919 | 0.6041 | 0.6453 | 0.7074 | 0.5569 | 0.5333 | 0.4993 |
| | FreeNeRF | 11.6493 | 14.2664 | 18.1031 | 0.4095 | 0.5514 | 0.6618 | 0.5692 | 0.5288 | 0.4953 |
| | FlipNeRF | 12.1361 | 13.8066 | 16.1049 | 0.5172 | 0.5849 | 0.6187 | 0.5479 | 0.5152 | 0.5110 |
| | DiffusioNeRF | 14.2206 | 17.4263 | 20.3853 | 0.6175 | 0.6539 | 0.6962 | 0.5137 | 0.4631 | 0.4232 |
| | SparseFusion | 10.7947 | 15.3898 | 17.4946 | 0.3105 | 0.4984 | 0.5625 | 0.7263 | 0.6065 | 0.5457 |
| | Ours | 17.2601 | 20.0152 | 20.8491 | 0.6095 | 0.6699 | 0.7049 | 0.5507 | 0.5115 | 0.4908 |
| Suitcase | RegNeRF | 16.7187 | 20.4675 | 21.1256 | 0.5553 | 0.6766 | 0.7247 | 0.6065 | 0.5467 | 0.5302 |
| | FreeNeRF | 18.8328 | 22.0083 | 22.6406 | 0.4517 | 0.6031 | 0.6381 | 0.5619 | 0.5369 | 0.5143 |
| | FlipNeRF | 18.0378 | 21.3206 | 22.2341 | 0.4719 | 0.5649 | 0.6251 | 0.5898 | 0.5478 | 0.5209 |
| | DiffusioNeRF | 17.8890 | 19.9514 | 21.8474 | 0.5128 | 0.5506 | 0.6125 | 0.5589 | 0.5129 | 0.4834 |
| | SparseFusion | 16.6920 | 18.0983 | 19.9869 | 0.3841 | 0.4177 | 0.5148 | 0.8674 | 0.7820 | 0.7184 |
| | Ours | 18.8709 | 21.5305 | 22.4276 | 0.5244 | 0.5751 | 0.6425 | 0.5961 | 0.5204 | 0.5087 |
| Teddybear | RegNeRF | 11.5974 | 17.9680 | 19.6752 | 0.5303 | 0.7378 | 0.7913 | 0.5544 | 0.4536 | 0.4175 |
| | FreeNeRF | 14.8848 | 18.6939 | 20.4244 | 0.5642 | 0.6955 | 0.7289 | 0.4663 | 0.4238 | 0.4182 |
| | FlipNeRF | 14.7060 | 17.2316 | 19.5608 | 0.5515 | 0.6732 | 0.7301 | 0.4750 | 0.4473 | 0.4208 |
| | DiffusioNeRF | 16.7526 | 19.1670 | 21.2461 | 0.6106 | 0.6591 | 0.6914 | 0.4589 | 0.3785 | 0.3496 |
| | SparseFusion | 13.9567 | 15.9479 | 17.4357 | 0.4560 | 0.5616 | 0.6057 | 0.7168 | 0.6183 | 0.5492 |
| | Ours | 16.4539 | 20.2717 | 21.7512 | 0.5689 | 0.6683 | 0.7341 | 0.4964 | 0.4219 | 0.4060 |
| Toytrain | RegNeRF | 12.9777 | 14.9275 | 16.5095 | 0.4031 | 0.4433 | 0.5194 | 0.7719 | 0.6951 | 0.6078 |
| | FreeNeRF | 13.1675 | 15.6723 | 19.1807 | 0.4682 | 0.5392 | 0.6058 | 0.7251 | 0.6430 | 0.5979 |
| | FlipNeRF | 14.0896 | 15.7112 | 18.6238 | 0.4373 | 0.4912 | 0.6133 | 0.6603 | 0.6398 | 0.5901 |
| | DiffusioNeRF | 15.4209 | 17.8208 | 20.3995 | 0.5064 | 0.5294 | 0.5543 | 0.5651 | 0.5160 | 0.4796 |
| | SparseFusion | 12.9703 | 15.3407 | 16.9265 | 0.3865 | 0.4576 | 0.4908 | 0.8154 | 0.7209 | 0.6670 |
| | Ours | 15.6856 | 18.6760 | 19.6327 | 0.4720 | 0.5471 | 0.6164 | 0.5987 | 0.5143 | 0.4722 |

of-the-art sparse-view NeRF methods, including RegNeRF [9], FreeNeRF [7] and FlipNeRF [32], which are based on regularizations, as well as DiffusioNeRF [27] and SparseFusion [17], which utilize diffusion models.

**Implementation Details.** We utilize Stable Diffusion as our diffusion model. The reconstruction part of our method can be applied to the majority of typical sparse-view NeRF approaches. In our experiments, we utilize Instant NGP [33] to rapidly obtain reconstruction results. For each instance, we optimize Instant NGP for 4,000 steps.

### *B. Reconstruction Results*

We report category-specific quantitative results in Table I. We show qualitative comparisons in Figure 4. For each scene, we load 50 views and randomly sample 3, 6, and 9 input views for reconstruction, then evaluate the remaining views. When the number of input viewpoints is extremely limited, like 3 views, methods based on regularization struggle to get the correct geometric shape of the scene. Although methods based on diffusion models can regularize the geometric shape, they lack detailed appearance. Our method

TABLE II: **Ablation experiments on fine-tuning.** 'Pretrained' indicates whether we initialize the diffusion model's weights from a pretrained text-to-image model. 'Image condition' refers to whether we utilize the coarse rendering obtained from the feature extractor as the image condition for the diffusion model. A LoRA dimension of 0 means not using LoRA, with weights being updated directly across the entire U-Net. Training time refers to the time required for fine-tuning. Without using pre-trained weights and image condition, and setting the LoRA dimension to 0, fine-tuning for 1000 iterations requires 0.98 hours. Loss (n) indicates the loss after fine-tuning for n iterations.

| Pretrain | Image condition | LoRA Dim | Training Param | Training Time | Loss(10) | Loss(100) | Loss(1k) | PSNR(1k) ↑ | SSIM(1k) ↑ | LPIPS(1k) ↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 859M | 100% | 16.7982 | 6.7084 | 1.1356 | 5.7428 | 0.0842 | 0.8976 |
| ✓ | ✓ | 0 | 859M | 100% | 0.7734 | 0.5515 | 0.1747 | 10.5759 | 0.3705 | 0.7333 |
| ✓ | ✓ | 16 | 3M | 47% | 0.6053 | 0.3246 | 0.0190 | 13.2275 | 0.4354 | 0.6074 |
| ✓ | ✓ | 32 | 6M | 48% | 0.4918 | 0.1824 | 0.0142 | 14.5476 | 0.4688 | 0.5820 |
| ✓ | | 32 | 6M | 48% | 0.3845 | 0.0876 | 0.0683 | 12.0524 | 0.4107 | 0.7152 |
| ✓ | ✓ | 64 | 12M | 50% | 0.4572 | 0.1467 | 0.0128 | 14.7031 | 0.4497 | 0.5912 |

can demonstrate reliable reconstruction results with good geometry and appearance.

Differences in the reconstruction metrics across various categories can be attributed to the diversity in the appearances, and textures of the objects contained within them. For example, compared to a cake, a motorcycle possesses more complex geometric shapes and subtle details. Our approach outperforms the compared baselines in PSNR while remaining competitive in SSIM and LPIPS.

### C. Ablation Studies

*1) Ablation experiment on fine-tuning:* In Table II, we ablate three aspects of the fine-tuning process. Utilizing pretrained model weights can help us achieve relatively low loss values in the early stages of training. The role of the image condition is to aid the diffusion model in more accurately modeling the distribution of novel views, thereby generating plausible 3D-aware images that more closely match real scenes. We leverage LoRA to reduce the number of training parameters and accelerate the convergence of the training. To balance training time and final metrics, we chose a LoRA with a dimension of 32 to fine-tune the diffusion model's U-Net, as indicated by the yellow line in the table.

*2) Ablation experiment on informative distillation:* To verify the effect of informative distillation, we present comparisons of our method under different thresholds in Fig.5. Furthermore, we incorporate the semantic switch into SparseFusion, which is similar to our reconstruction pipeline, and obtain metrics under different thresholds. As show in Fig.5, both our method and SparseFusion* significantly outperform indiscriminate distillation (indicated by the dashed line). Our designed semantic switch effectively filters out generated images that deviate from real scenes, thereby improving the quality of the 3D model.

However, when the threshold reaches 0.9, the metrics might decline. This is because only a very small number of generated images have a similarity higher than 0.95 with the input images. In the reconstruction experiments, we set the threshold to 0.9 to achieve optimal informative distillation.

It can be noted that, without incorporating the semantic switch, the metrics of SparseFusion are better than those of our method. We speculate that this may be because SparseFusion uses a diffusion model trained from scratch on CO3D; hence, when tested on the dataset, the distribution of the diffusion model aligns more closely with that of CO3D.

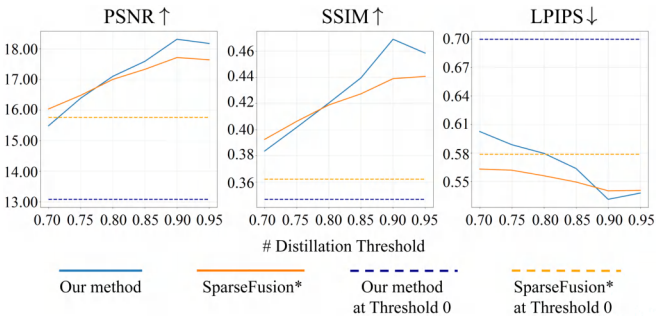


Fig. 5: **Ablation experiments regarding the impact of different thresholds on diffusion distillation.** The value of the threshold is the cosine similarity between the generated image's CLIP feature and the input images' CLIP feature. SparseFusion* refers to SparseFusion with our designed semantic switch incorporated.

## V. CONCLUSIONS

In this paper, we propose an enhanced sparse-view 3D reconstruction pipeline that involves fine-tuning the pretrained diffusion model and distilling informative priors from the diffusion model. Based on the expriments presented, the following conclusions can be drawn:

(1) Focusing on the challenge of generating 3D-aware images at a low cost, we propose a cost-effective approach to fine-tune the pretrained diffusion model to generate images that match the real scene.

(2) To address the impact of the quality of generated image on diffusion distillation, we design a semantic switch to filter out generated images that deviate from the real scene and improve the quality of the 3D model.

However, our method also has some limitations: the reliance on accurate poses of input views, and the images

generated by the fine-tuned diffusion model may lack in saturation. While our 3D reconstruction is based on NeRF, we believe other image-based 3D reconstruction methods(such as 3DGS [34]) could also benefit from our approach.

## ACKNOWLEDGMENT

This work is partly supported by National Key R&D Program of China (2022YFC2603600) and National Natural Science Foundation of China (Grant No. NSFC 62233002).The authors would like to thank Zhaoxiang Liang, Xihan Wang, Dianyi Yang and all other members of ININ Lab of Beijing Institute of Technology for their contribution to this work.

## REFERENCES

[1] A. To, M. Liu, M. Hazeeq Bin Muhammad Hairul, J. G. Davis, J. S. Lee, H. Hesse, and H. D. Nguyen, "Drone-based ai and 3d reconstruction for digital twin augmentation," in *International conference on human-computer interaction*, pp. 511–529, Springer, 2021.

[2] M. G. Bevilacqua, M. Russo, A. Giordano, and R. Spallone, "3d reconstruction, digital twinning, and virtual reality: Architectural heritage applications," in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 92–96, IEEE, 2022.

[3] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7210–7219, 2021.

[4] I. Zhura, D. Davletshin, N. D. W. Mudalige, A. Fedoseev, R. Peter, and D. Tsetserukou, "Neuroswarm: Multi-agent neural 3d scene reconstruction and segmentation with uav for optimal navigation of quadruped robot," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2525–2530, IEEE, 2023.

[5] M. Turan, Y. Almalioglu, E. P. Ornek, H. Araujo, M. F. Yanik, and M. Sitti, "Magnetic- visual sensor fusion-based dense 3d reconstruction and localization for endoscopic capsule robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1283–1289, 2018.

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, p. 99–106, dec 2021.

[7] J. Yang, M. Pavone, and Y. Wang, "Freenerf: Improving few-shot neural rendering with free frequency regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8254–8263, 2023.

[8] A. Jain, M. Tancik, and P. Abbeel, "Putting nerf on a diet: Semantically consistent few-shot view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5885–5894, October 2021.

[9] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5470–5480, 2022.

[10] J. Y. Zhang, G. Yang, S. Tulsiani, and D. Ramanan, "NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild," in *Conference on Neural Information Processing Systems*, 2021.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[12] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9298–9309, 2023.

[13] R. Shi, H. Chen, Z. Zhang, M. Liu, C. Xu, X. Wei, L. Chen, C. Zeng, and H. Su, "Zero123++: a single image to consistent multi-view diffusion base model," *arXiv preprint arXiv:2310.15110*, 2023.

[14] M. Liu, C. Xu, H. Jin, L. Chen, M. Varma T, Z. Xu, and H. Su, "One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[15] M. Liu, R. Shi, L. Chen, Z. Zhang, C. Xu, X. Wei, H. Chen, C. Zeng, J. Gu, and H. Su, "One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion," *arXiv preprint arXiv:2311.07885*, 2023.

[16] J. Ye, P. Wang, K. Li, Y. Shi, and H. Wang, "Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models," *arXiv preprint arXiv:2310.03020*, 2023.

[17] Z. Zhou and S. Tulsiani, "Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12588–12597, 2023.

[18] E. R. Chan, K. Nagano, M. A. Chan, A. W. Bergman, J. J. Park, A. Levy, M. Aittala, S. De Mello, T. Karras, and G. Wetzstein, "Generative novel view synthesis with 3d-aware diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4217–4229, 2023.

[19] H. Chen, J. Gu, A. Chen, W. Tian, Z. Tu, L. Liu, and H. Su, "Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2416–2425, 2023.

[20] R. Wu, B. Mildenhall, P. Henzler, K. Park, R. Gao, D. Watson, P. P. Srinivasan, D. Verbin, J. T. Barron, B. Poole, *et al.*, "Reconfusion: 3d reconstruction with diffusion priors," *arXiv preprint arXiv:2312.02981*, 2023.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[22] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021.

[23] G. Wang, Z. Chen, C. C. Loy, and Z. Liu, "Sparsenerf: Distilling depth ranking for few-shot novel view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9065–9076, 2023.

[24] N. Somraj and R. Soundararajan, "Vip-nerf: Visibility prior for sparse input neural radiance fields," in *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

[25] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[26] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in neural information processing systems*, vol. 35, pp. 36479–36494, 2022.

[27] J. Wynn and D. Turmukhambetov, "Diffusionerf: Regularizing neural radiance fields with denoising diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4180–4189, 2023.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[29] Y. Gao, L. Su, H. Liang, Y. Yue, Y. Yang, and M. Fu, "Mc-nerf: Muti-camera neural radiance fields for muti-camera image acquisition systems," *arXiv preprint arXiv:2309.07846*, 2023.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[31] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny, "Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10901–10911, 2021.

[32] S. Seo, Y. Chang, and N. Kwak, "Flipnerf: Flipped reflection rays for few-shot novel view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22883–22893, 2023.

[33] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[34] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.