NerfDiff: Single-image View Synthesis with NeRF-guided Distillation from 3D-aware Diffusion
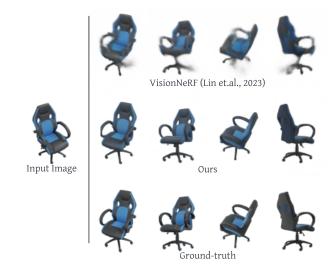
Jiatao Gu[1]   **Alex Trevithick**[2]   **Kai-En Lin**[2]   **Josh Susskind**[1]   **Christian Theobalt**[3]
Lingjie Liu[3][4]   **Ravi Ramamoorthi**[2]

Abstract

Novel view synthesis from a single image requires inferring occluded regions of objects and scenes whilst simultaneously maintaining semantic and physical consistency with the input. Existing approaches condition neural radiance fields (NeRF) on local image features, projecting points to the input image plane, and aggregating 2D features to perform volume rendering. However, under severe occlusion, this projection fails to resolve uncertainty, resulting in blurry renderings that lack details. In this work, we propose NerfDiff, which addresses this issue by distilling the knowledge of a 3D-aware conditional diffusion model (CDM) into NeRF through synthesizing and refining a set of virtual views at test-time. We further propose a novel NeRF-guided distillation algorithm that simultaneously generates 3D consistent virtual views from the CDM samples, and finetunes the NeRF based on the improved virtual views. Our approach significantly outperforms existing NeRF-based and geometry-free approaches on challenging datasets including ShapeNet, ABO, and Clevr3D. Please see the supplementary website (https://jiataogu.me/nerfdiff) for video results.

1. Introduction

Novel view synthesis is a core component of computer graphics and vision applications, including virtual and augmented reality, immersive photography, and the creation of digital replicas. Given a few input views of an object or a scene, one seeks to synthesize new views from other viewing

Figure 1. Renderings from our method in comparison to the SoTA VisionNeRF (Lin et al., 2023). Note how our method can predict sharp renderings despite large occlusion, whereas VisionNeRF cannot handle this uncertainty and shows implausible blurring.

directions. This problem is challenging since novel views must account for occlusions and unseen regions. This problem has a long history, going back to early work in image-based rendering (IBR) (Chen & Williams, 1993; Gortler et al., 1996; Levoy & Hanrahan, 1996; McMillan & Bishop, 1995). However, IBR methods can only produce suboptimal results and are often scene-specific. Recently, neural radiance fields (NeRF) (Mildenhall et al., 2020) have shown high-quality novel view synthesis results, but NeRF requires tens or hundreds of images for overfitting a scene and has no generalization ability to infer new scenes.

This work focuses on novel view synthesis from a single image. In attempts to do so, generalizable NeRF models (Trevithick & Yang, 2021; Yu et al., 2021; Lin et al., 2023) have been proposed, whereby the NeRF representation is conditioned by the projection of 3D points and gathering of corresponding image features. These approaches produce good results, especially for cameras near the input. How-

[1]Apple [2]University of California, San Diego [3]Max Planck Institute for Informatics, Germany [4]University of Pennsylvania. Correspondence to: Jiatao Gu <jiatao@apple.com>, Alex Trevithick <atrevithick@ucsd.edu>, Kai-En Lin <k2lin@ucsd.edu>, Lingjie Liu <lingjie.liu@seas.upenn.edu>.

ever, when the target views are far from the input, these approaches yield blurry results. The uncertainty of large unseen regions in novel views cannot be resolved by projection to the input image. A distinct line of work addresses the uncertainty issue in single-image view synthesis by leveraging 2D generative models to predict novel views conditioned on the input view (Rombach et al., 2021c; Watson et al., 2022). However, these approaches are only able to synthesize partially 3D-consistent images.

In this paper, we propose NerfDiff, a *training-finetuning* framework for synthesizing multi-view consistent and high-quality images given single-view input. Concretely, at the training stage, we jointly train a camera-space triplane-based NeRF together with a 3D-aware conditional diffusion model (CDM) on a collection of scenes. We initialize the NeRF representation given the input image at the finetuning stage. Then, we finetune the parameters from a set of virtual images predicted by the CDM conditioned on the NeRF-rendered outputs. We found that a naive finetuning strategy of optimizing the NeRF parameters directly using the CDM outputs would lead to subpar renderings, as the CDM outputs tend to be multi-view inconsistent. Therefore, we propose *NeRF-guided distillation*, which updates the NeRF representation and guides the multi-view diffusion process in an alternating fashion. In this way, the uncertainty in single-image view synthesis can be resolved by filling in unseen information from CDM; in the meantime, NeRF can guide CDM for multi-view consistent diffusion. An illustration of the proposed pipeline is shown in Figure 2.

We evaluate our approach on three challenging benchmarks. Our results indicate that the proposed NerfDiff significantly outperforms all the existing baselines, achieving high-quality generation with multi-view consistency. See supplementary materials for video results. We summarize the main contributions as follows:

- We develop a novel framework – NerfDiff which jointly learns an image-conditioned NeRF and a CDM, and at test time finetunes the learned NeRF using a multi-view consistent diffusion process (§ 4.3,§ 4.4).
- We introduce an efficient image-conditioned NeRF representation based on camera-aligned triplanes, which is the core component enabling efficient rendering and finetuning from the CDM (§ 4.1).
- We propose a 3D-aware CDM, which integrates volume rendering into 2D diffusion models, facilitating generalization to novel views (§ 4.2).

2. Related Work

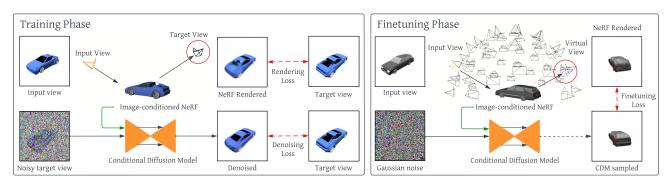### 2.1. Diffusion models for 3D generation

Diffusion-based generative models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019) have re-

cently become state-of-the-art on image synthesis (Dhariwal & Nichol, 2021) and have shown remarkable results in handling highly under-constrained tasks such as text-to-image (Ramesh et al., 2022; Rombach et al., 2021a; Saharia et al., 2022) and text-to-video generation (Ho et al., 2022).

More recently, diffusion models have also been proven effective in 3D generation tasks. On the one hand, many methods propose to directly apply diffusion in 3D space, including point clouds (Nichol et al., 2022), voxels (Müller et al., 2022a) and fields (Zhuang et al., 2023). Some of them learn diffusion in a latent space derived from 3D space (Bautista et al., 2022; Shue et al., 2022). However, a clear limitation of these approaches is that it requires 3D ground truth for learning the diffusion process, which is hard to acquire in the real environment. On the other hand, several works propose to learn 3D representations from diffusion in 2D space. For example, Li et al. (2022) learns the geometry based on a 2-view diffusion model; Anciukevicius et al. (2022) designs an architecture that generates and renders an intermediate 3D representation for each diffusion step. Concurrently related to our method, a series of work (Poole et al., 2022; Wang et al., 2022; Lin et al., 2022; Zhou & Tulsiani, 2022; Deng et al., 2022) have proposed score distillation that learns 3D representation directly from pre-trained 2D diffusion models.

2.2. Single-view Novel View Synthesis

Methods beyond Neural Fields   Most initial attempts at single-view 3D reconstruction relied on ground truth training data to estimate the geometry of objects. These methods typically mapped an image to its depth or directly to a 3D shape (Eigen et al., 2014; Saxena et al., 2009; Fan et al., 2017; Tatarchenko et al., 2017; Tulsiani et al., 2017). Some methods (Kato et al., 2018; Yan et al., 2016; Loper & Black, 2014) provide 3D reconstruction estimates without ground truth 3D supervision using differentiable renderers; however, these methods were limited to reconstructing only the geometry, not the appearance. Recently, other methods have allowed the rendering of novel views without regard for multiview consistency. For example, ENR (Dupont et al., 2020) utilizes convolutions with a projection to decode 3D voxel features to RGB. Targeting more complex scenes, SynSin (Wiles et al., 2020) makes use of a differentiable point cloud renderer and an inpainter to extrapolate to unseen areas. InfiniteNature (Liu et al., 2021) utilizes estimated depth to iteratively inpaint novel views along a camera trajectory. Other works, such as GeoFree (Rombach et al., 2021b) and Pixelsynth (Rockwell et al., 2021), utilize an autoregressive prior to inferring unseen areas of the scene. Finally, light-field-based methods like Sajjadi et al. (2022) and Suhail et al. (2022) condition transformers on features from input images and query rays to directly output colors or directly invert into a latent space (Sitzmann et al., 2021).

Figure 2. NerfDiff incorporates a training and finetuning pipeline. We first learn the single-image NeRF and 2D CDM, which are conditioned on the single-image NeRF renderings (left). We use the learned network parameters at test time to predict an initial NeRF representation for finetuning. The NeRF-guided denoised images from the frozen CDM then supervise the NeRF in turn (right).

Methods based on Neural Fields Many methods propose to use neural fields (e.g., neural radiance field (NeRF, Mildenhall et al., 2020)) for this task. For example, Sin-NeRF (Xu et al., 2022) renders novel views near the input image using pseudo geometry. On the other hand, Sitzmann et al. (2019a); Rematas et al. (2021); Jang & Agapito (2021); Müller et al. (2022b) incorporate global latent codes and apply test-time tuning to refine these codes. This process is very similar to inversion into the latent space of 3D NeRF-based GANs (Gu et al., 2021; Chan et al., 2021; Bautista et al., 2022; Cai et al., 2022). Note that it requires (estimated) camera poses at test time, which hinders high-quality results. Furthermore, the global bottleneck hinders capturing fine details, and due to the optimization of the input view, such methods also cannot handle occlusion appropriately. Finally, *image-conditioned* methods (e.g., pixelNeRF (Yu et al., 2021) and VisionNeRF (Lin et al., 2023)) directly utilize local image features to condition NeRF and are the most relevant to our method. Note that, like pixelNeRF, our method can perform view synthesis without pose annotation.

3. Background

3.1. Image-conditioned NeRF

Neural radiance fields (NeRF, Mildenhall et al., 2020) have been proven remarkably effective for novel view synthesis. NeRF defines an implicit function $f_\theta : (\boldsymbol{x}, \boldsymbol{d}) \to (\boldsymbol{c}, \sigma)$ given a spatial location $\boldsymbol{x} \in \mathbb{R}^3$ and ray direction $\boldsymbol{d} \in \mathbb{S}^2$, where θ are the learnable parameters, $\boldsymbol{c}$ and σ are the color and density, respectively. To render a posed image $\boldsymbol{I}$, we march a camera ray through each pixel $\boldsymbol{r}(t) = \boldsymbol{x}_o + t\boldsymbol{d}$ (where $\boldsymbol{x}_o$ is the camera origin) and calculate its color via an approximation of the volume rendering integral:

$$\boldsymbol{I}_\theta(\boldsymbol{r}) = \int_{t_{\mathrm{n}}}^{t_{\mathrm{f}}} \omega(t) \cdot \boldsymbol{c}_\theta(\boldsymbol{r}(t), \boldsymbol{d}) \, \mathrm{d}t, \qquad (1)$$
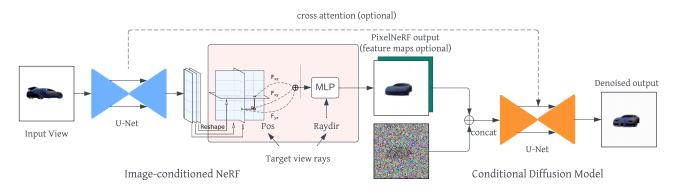
where $\omega(t) = e^{-\int_{t_{\mathrm{n}}}^{t} \sigma_\theta(\boldsymbol{r}(s)) \, \mathrm{d}s} \sigma_\theta(\boldsymbol{r}(t))$, t_{n} and t_{f} are the near and far bounds of the ray, respectively. When multi-view images are available, θ can be easily optimized with the standard MSE loss:

$$\mathcal{L}_\theta^{\mathrm{NeRF}} = \mathbb{E}_{\boldsymbol{I} \sim \mathrm{data}, \boldsymbol{r} \sim \mathcal{R}(\boldsymbol{I})} \|\boldsymbol{I}_\theta(\boldsymbol{r}) - \boldsymbol{I}(\boldsymbol{r})\|_2^2, \qquad (2)$$

where $\mathcal{R}(\boldsymbol{I})$ is the set of rays that composes $\boldsymbol{I}$. To capture high-frequency details, NeRF encodes $\boldsymbol{x}$ and $\boldsymbol{d}$ with sinusoidal positional functions $\xi_{\mathrm{pos}}(\boldsymbol{x}), \xi_{\mathrm{pos}}(\boldsymbol{d})$. Recently, studies have shown that encoding functions with local structures like triplanes (Chan et al., 2021) achieves significantly faster inference speed without quality loss.

The training of NeRF, i.e., the optimization of Eq. (2), requires tens or hundreds of images along with their camera parameters to provide sufficient multi-view constraints. However, in reality, such multi-view data is not easily accessible. Therefore, this work focuses on recovering neural radiance fields from a single image without knowing its absolute camera pose. As this problem is under-constrained, it requires 3D inductive biases learned from a large set of scenes similar to the target scene. Following this philosophy, pixel-aligned NeRFs (Yu et al., 2021; Lin et al., 2023) encode 3D information with local 2D image features so that the learned representations can generalize to unseen scenes after being trained on a large number of scenes.

PixelNeRF Take PixelNeRF (Yu et al., 2021) as an example. Given an input image $\boldsymbol{I}^s$, PixelNeRF first extracts a feature volume $W = e_\psi(\boldsymbol{I}^s)$ where e_ψ is a learnable image encoder. Then, for any 3D point $\boldsymbol{x} \in \mathbb{R}^3$ in the input *camera space*, its corresponding image features are obtained by projecting $\boldsymbol{x}$ onto the image plane as $\mathcal{P}(\boldsymbol{x}) \in [-1, 1]^2$ with known intrinsic matrix, and then bilinearly interpolating the feature volume as $\xi_W(\boldsymbol{x}) = W(\mathcal{P}(\boldsymbol{x}))$. The image features will be combined with the position $\boldsymbol{x}$ and view direction $\boldsymbol{d}$ to infer the color and density. Next, similar to NeRF, the

Figure 3. Details of the architecture of the single-image NeRF for NerfDiff. Using a UNet, we first map an input image to a camera-aligned triplane-based NeRF representation. This triplane efficiently conditions volume rendering from a targeted view, resulting in an initial rendering. This rendering conditions the diffusion process so the CDM can consistently denoise at that target pose.

color of a camera ray is calculated via volume rendering (Eq. 1). Such a model is trained over a collection of scenes, and for each scene, at least two views are needed to form the training pairs $(\boldsymbol{I}^s, \boldsymbol{I})$ for reconstruction:

$$\mathcal{L}_{\theta,\psi}^{\mathrm{IC}} = \mathbb{E}_{(\boldsymbol{I}^s, \boldsymbol{I}) \sim \mathrm{data}, \boldsymbol{r} \sim \mathcal{R}(\boldsymbol{I})} \| \boldsymbol{I}_{\theta,W}(\boldsymbol{r}) - \boldsymbol{I}(\boldsymbol{r}) \|_2^2, \quad (3)$$

where $\boldsymbol{I}_{\theta,W}$ is the volume rendered image.

Challenges However, existing single-image NeRF approaches fail to produce high-fidelity rendering results, especially when severe occlusions exist. This is because single-image view synthesis is an under-constrained problem, as the synthesized occluded regions can exhibit multiple possibilities. Therefore, MSE loss (Eq. (3)) forces single-image NeRF to regress to mean pixel values across all possible solutions, yielding inaccurate and blurry predictions.

3.2. Geometry-free View Synthesis

To account for the uncertainty challenge, a distinct line of research explicitly models view prediction $p(\boldsymbol{I}|\boldsymbol{I}^s)$ with 2D generative models, like Dupont et al. (2020); Rombach et al. (2021b); Sajjadi et al. (2022) and more recently conditional diffusion models (3DiM, Watson et al., 2022). Take 3DiM as an example. It learns a conditional noise predictor $\boldsymbol{\epsilon}_\phi$ that de-noises Gaussian-noised target images conditioning on the input view Moreover, the corresponding camera poses. Such a model can be optimized with a denoising loss:

$$\mathcal{L}_\phi^{\mathrm{DM}} = \mathbb{E}_{(\boldsymbol{I}^s, \boldsymbol{I}) \sim \mathrm{data}, \boldsymbol{\epsilon}, t} \| \boldsymbol{\epsilon}_\phi (\boldsymbol{Z}_t, \boldsymbol{I}^s) - \boldsymbol{\epsilon} \|_2^2 \quad (4)$$

where $\boldsymbol{Z}_t = \alpha_t \boldsymbol{I} + \sigma_t \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1), \alpha_t^2 + \sigma_t^2 = 1$ is the noised target for $\boldsymbol{I}$. As shown in Song & Ermon (2019), the denoiser provides an approximation for the score function of the distribution $\boldsymbol{\epsilon}_\phi(\boldsymbol{Z}_t, \boldsymbol{I}^s) \approx -\sigma_t \nabla_{\boldsymbol{Z}_t} \log p_\phi(\boldsymbol{Z}_t | \boldsymbol{I}^s)$. At test time, the learned score $\boldsymbol{\epsilon}_\phi$ is applied iteratively and refines noise images to synthesize novel views.

Challenges Geometry-free models typically suffer from the "alignment problem" where the input view conditioning and target views are not pixel-wise aligned, leading to inferior generalization when applying standard UNet-based diffusion models. Watson et al. (2022) attempted to alleviate this issue by using cross-attention to gather information from the input view. However, this requires models with large capacities, and even with this modification, it still needs more generalizability for complex scenes and out-of-distribution cameras. Moreover, since denoising is conducted in 2D for each view independently rather than in 3D, the synthesized novel views of CDMs in the sampling stage tend to be multi-view inconsistent.

4. NerfDiff

To achieve the best of both worlds, in this paper, we present a *training-finetuning* two-stage approach, dubbed as *NerfDiff*, to incorporate the power of diffusion models into image-conditioned NeRFs for single-image view synthesis. We illustrate the pipelines of the proposed two stages in Figure 2. In the following, we first introduce NerfDiff, which consists of an improved single-image NeRF based on local triplanes (§ 4.1) and a 3D-aware CDM built on top of the single-image NeRF outputs (§ 4.2). An overview of the proposed model is presented in Figure 3. These two components are optimized jointly on the same training set (§ 4.3). At test time, we adopt a second-stage finetuning. Furthermore, to mitigate the inconsistency issue brought by CDM sampling, we present the *NeRF-Guided Distillation* (NGD) algorithm to improve the finetuning performance (§ 4.4).

4.1. Single-image NeRF with Local Triplanes

NerfDiff is built upon an efficient camera-aligned triplane extracted directly from an input image to condition the NeRF. As mentioned in § 3.1, most existing single-view

models (Yu et al., 2021; Lin et al., 2023) query the extracted features via image plane projection: $\mathcal{P} : \mathbb{R}^3 \to [-1, 1]^2$. One issue with this operation is that the depth information of a 3D point is not contained in its extracted features; that is, all points along the same camera ray project to the same location on the 2D image and thus have the same features. Therefore, to differentiate the points along the same camera ray, existing methods need to concatenate the image features with the positional encoding of the global spatial location $\xi_{\text{pos}}(\boldsymbol{x})$ as the representation of point x. However, this 3D representation is not efficient. Thus it needs a deep MLP network to fuse the image features with spatial information for inferring the color and density of x, which slows down the rendering process. Inspired by Chan et al. (2021), we propose an efficient 3D representation that re-shapes the image feature W into a *camera-aligned* triplane: $\{W_{xy}, W_{xz}, W_{yz}\}$[1]. Then, each 3D point receives a unique feature vector by bilinear interpolation within three planes:

$$\xi_W(\boldsymbol{x}) = W_{xy}(\tilde{\boldsymbol{x}}_{xy}) + W_{xz}(\tilde{\boldsymbol{x}}_{xz}) + W_{yz}(\tilde{\boldsymbol{x}}_{yz}), \quad (5)$$

where $\tilde{\boldsymbol{x}} = \left[\mathcal{P}(\boldsymbol{x}), 2 \cdot \frac{\boldsymbol{x}_z - t_{\text{n}}}{t_{\text{f}} - t_{\text{n}}} - 1 \right] \in [-1, 1]^3$, and $t_{\text{n}}, t_{\text{f}}$ are the near and far bounds of the input camera (Eq. (1)). As this representation is expressive in the sense that it can allocate depth information in the xz, yz planes, no additional positional encoding $\xi_{\text{pos}}(\boldsymbol{x})$ is needed to augment the representation, and the deep MLP network can be replaced with a shallow MLP network. This not only makes high-resolution image rendering efficient (§ 4.2) but also enables fast NeRF finetuning, which will be elaborated in § 4.4. Furthermore, modeling triplanes in the camera space of the input image has the following benefits: (1) The triplane can naturally preserve the local image features, same as pixelNeRF (Yu et al., 2021); (2) we do not need to assume a global coordinate system, and global camera poses, which is different from existing triplane-based methods (Chan et al., 2021; Chen et al., 2022; Bautista et al., 2022).

Note that, for the image encoder, we adopt a UNet architecture (Ronneberger et al., 2015; Nichol & Dhariwal, 2021) rather than a pre-trained ResNet (He et al., 2016) used in Yu et al. (2021). Thanks to the U-connection and self-attention blocks, the output layer feature $W = e_\psi(I)$ contains both the local and global information that is essential for predicting occluded views, which works similarly to the feature extractors in Lin et al. (2023). See Figure 3 for details.

4.2. 3D-aware Conditional Diffusion Models

While single-image NeRF produces multi-view consistent images, the outputs tend to be blurry due to the uncertainty issue (§ 3.1). To address the uncertainty issue, we model a 3D-aware CDM as the second part of NerfDiff, which

---

[1]The xy plane is aligned with the input image, while the xz and yz planes are orthogonal to the xy plane and each other.

Algorithm 1 Finetuning with NeRF-guided distillation.

Input: NeRF (MLP f_θ, triplanes W), CDM $\boldsymbol{\epsilon}_\phi$, input $\boldsymbol{I}^s, \gamma, N, B$
1 **Initialize** $I^\pi = I^\pi_{\theta,W}, \boldsymbol{\epsilon}^\pi = \boldsymbol{\epsilon}, \pi \in \Pi, \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$
   for $t = t_{\max} \ldots t_{\min}$ **do**
2   **for** $\pi \in \Pi$ **do**
3       $\boldsymbol{Z}^\pi = \alpha_t \boldsymbol{I}^\pi + \sigma_t \boldsymbol{\epsilon}^\pi;$
      $\boldsymbol{\epsilon}^\pi = \boldsymbol{\epsilon}_\phi(\boldsymbol{Z}^\pi, \boldsymbol{I}^s) + \gamma \sigma_t / \alpha_t \cdot (\boldsymbol{I}^\pi - \boldsymbol{I}^\pi_{\theta,W})$
      $\boldsymbol{I}^\pi = (\boldsymbol{Z}^\pi - \sigma_t \boldsymbol{\epsilon}^\pi)/\alpha_t$
4   **for** $n = 1 \ldots N$ **do**
5   **for** $b = 1 \ldots B$ **do**
6   Sample a view $\pi \sim \Pi$ and sample a ray $\boldsymbol{r}$ from π;
7   Update θ, W with $\nabla_{\theta,W} \frac{1}{B} \sum_{\pi,\boldsymbol{r}} \|\boldsymbol{I}^\pi_{\theta,W}(\boldsymbol{r}) - \boldsymbol{I}^\pi(\boldsymbol{r})\|_2^2$

8 **return** θ, W

---

resolves uncertainty through a generative process. Specifically, the CDM is learned to iteratively refine the rendering of single-image NeRF to match the target views.

Compared to existing geometry-free methods (Watson et al., 2022), we avoid the "alignment" problem by applying single-image NeRF to render the target-view images as the conditioning to CDM rather than using the input-view image as conditioning. As shown in Figure 3, we adopt the standard conditional UNet architecture (Nichol & Dhariwal, 2021) where the noisy image is concatenated with the rendered image. Similar to Watson et al. (2022), we can also employ cross-attention blocks between the CDM UNet and the encoder UNet to strengthen the conditioning. Note that the efficiency of the triplane rendering (see Table 3) allows the NeRF to be trained in tandem with the CDM, which would take far too long otherwise.

4.3. Training Phase

Given a collection of scenes where each scene has at least two views, we train end-to-end by combining Equations (3) and (4): $\mathcal{L}^{\text{Train}}_{\theta,\psi,\phi} = \lambda_{\text{IC}} \mathcal{L}^{\text{IC}}_{\theta,\psi} + \lambda_{\text{DM}} \mathcal{L}^{\text{DM}}_\phi$ to optimize the single-image NeRF and CDM jointly. Note that the training of single-image NeRF receives supervision from both the photometric error $\mathcal{L}^{\text{IC}}_{\theta,\psi}$ and the CDM denoising loss $\mathcal{L}^{\text{DM}}_\phi$. See Figure 2 (left) for details.

4.4. Fine-tuning Phase

While the 3D-aware CDM resolves the uncertainty issue in the single-image NeRF and thus makes the synthesized images sharper, it compromises multi-view consistency, as the 2D diffusion process is independently applied to each novel view. To synthesize multi-view consistent and high-quality results, we propose a novel finetuning strategy at test time to *distill* the CDM's knowledge.

As shown in Figure 2 (right), given an input view $\boldsymbol{I}^s$ of an unseen scene, we generate a set of "virtual views" with the trained single-image NeRF and 3D-aware CDM. Then

	ShapeNet Cars				ShapeNet Chairs				Amazon-Berkeley Objects			
	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
LFN (Sitzmann et al., 2021)*	22.42	0.89	–	–	22.26	0.90	–	–	–	–	–	–
3DiM (Watson et al., 2022)*	21.01	0.57	–	**8.99**	17.05	0.53	–	6.57	–	–	–	–
SRN (Sitzmann et al., 2019a)	22.25	0.88	0.129	41.21	22.89	0.89	0.104	26.51	–	–	–	–
PixelNeRF (Yu et al., 2021)	23.17	0.89	0.146	59.24	23.72	0.90	0.128	38.49	–	–	–	–
CodeNeRF (Jang & Agapito, 2021)	22.73	0.89	0.128	–	23.39	0.87	0.166	–	–	–	–	–
FE-NVS (Guo et al., 2022)	22.83	0.91	0.099	–	23.21	0.92	0.077	–	–	–	–	–
VisionNeRF (Lin et al., 2023)	22.88	0.90	0.084	21.31	24.48	0.92	0.077	10.05	28.61	0.93	0.095	33.38
NerfDiff-B (Ours)	23.51	**0.92**	0.082	18.09	24.79	0.94	**0.056**	5.65	32.81	0.96	0.057	7.77
w/o NGD	23.81	**0.92**	0.093	42.37	24.77	0.93	0.068	15.72	32.07	0.95	0.063	18.01
NerfDiff-L (Ours)	23.76	**0.92**	**0.076**	15.49	**24.95**	**0.94**	**0.056**	**5.34**	**32.84**	**0.97**	**0.042**	**6.31**
w/o NGD	**23.95**	**0.92**	0.092	43.26	24.80	0.93	0.070	15.50	32.00	0.96	0.061	17.73

Table 1. Comparisons on ShapeNet Cars & Chairs and ABO datasets across baselines. * indicates geometry-free model. The results of the baselines except VisionNeRF (Lin et al., 2023) are copied from the official papers. – denotes the results are unavailable.

we finetune the triplane parameters $W = e_\psi(\boldsymbol{I}^s)$ and MLP parameters θ (pink box in Fig. 3) with the generated virtual views. Here, we treat W as learnable parameters. It performs best when virtual views cover the region of interest.

NeRF Guided Distillation   A naive optimization strategy is the same as that in Mildenhall et al. (2020) (Eq. (2)), i.e., replacing the targets with the "virtual views" sampled from the CDM. We found that this naive optimization typically leads to noisy results with severe floating artifacts, as the inconsistent CDM predictions cause conflicts in optimizing the NeRF model. Instead, we propose *NeRF Guided Distillation (NGD)* that alternates between NeRF distillation and diffusion sampling. Inspired by classifier guidance (Dhariwal & Nichol, 2021), we incorporate 3D consistency into multi-view diffusion by considering the joint distribution for each virtual view $\boldsymbol{I}$:

$$p_\phi(\boldsymbol{Z}_t, \boldsymbol{I}_{\theta,W}|\boldsymbol{I}^s) = p_\phi(\boldsymbol{Z}_t|\boldsymbol{I}^s) \cdot p(\boldsymbol{I}_{\theta,W}|\boldsymbol{Z}_t, \boldsymbol{I}^s)$$
$$\propto p_\phi(\boldsymbol{Z}_t|\boldsymbol{I}^s) \cdot e^{-\frac{\gamma}{2}\|\boldsymbol{I}_t - \boldsymbol{I}_{\theta,W}\|_2^2}, \quad (6)$$

where $\boldsymbol{I}_t = (\boldsymbol{Z}_t - \sigma_t\boldsymbol{\epsilon}_\phi(\boldsymbol{Z}_t, \boldsymbol{I}^s))/\alpha_t$ is the predicted target image at the intermediate timestep. The second term introduces multi-view constraints from a given NeRF. Therefore, the goal is to find NeRF parameters (θ, W) that maximize Eq. (6) while sampling the most likely virtual views $(\boldsymbol{Z}_t)$ from the joint distribution. In practice, we adopt an iterative-based updating rule at each diffusion step t. For generating virtual views with the CDM, we follow the modified diffusion score derived from Eq. (6):

$$\tilde{\boldsymbol{\epsilon}}_\phi(\boldsymbol{Z}_t, \boldsymbol{I}^s) = \boldsymbol{\epsilon}_\phi(\boldsymbol{Z}_t, \boldsymbol{I}^s) + \gamma\frac{\sigma_t}{\alpha_t}(\boldsymbol{I}_t - \boldsymbol{I}_{\theta,W}), \quad (7)$$

where $\tilde{\boldsymbol{\epsilon}}_\phi$ will be used in regular DDIM sampling (Song et al., 2020) [2]. Note that for $\gamma = \alpha_t^2/\sigma_t^2$ (SNR), following

---

[2] We consider $\partial \boldsymbol{I}_t/\partial \boldsymbol{Z}_t \approx 1/\alpha_t$ to avoid backpropagation through the UNet, similar to DreamFusion (Poole et al., 2022).

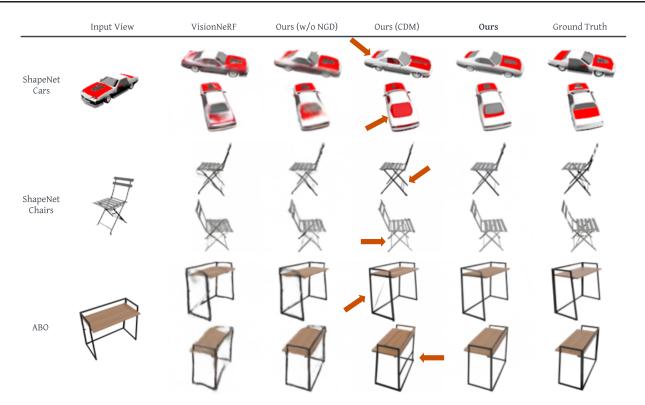| Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
VisionNeRF (Lin et al., 2023)	**35.94**	**0.97**	0.065	11.18
NerfDiff-B (Ours)	34.81	**0.97**	**0.040**	**6.76**

Table 2. Quantative results on Clevr3D

the modified score Eq. (7) is equivalent to replacing the denoised images with the NeRF rendering. For distilling NeRF, we directly maximize the log-likelihood of this joint distribution w.r.t. the NeRF parameters, which is equivalent to minimizing the MSE loss between the denoised images $\boldsymbol{I}_t$ and the NeRF renderings $\boldsymbol{I}_{\theta,W}$ across all virtual views:

$$\mathcal{L}^{\text{FT}}_{\theta,W} = \mathbb{E}_{\pi\sim\Pi, \boldsymbol{r}\sim\mathcal{R}(\boldsymbol{I}_t^\pi)}\|\boldsymbol{I}^\pi_{\theta,W}(\boldsymbol{r}) - \boldsymbol{I}^\pi_t(\boldsymbol{r})\|_2^2, \quad (8)$$

where Π is a prior distribution on the relative camera poses to the input and $\boldsymbol{I}^\pi_t, \boldsymbol{I}^\pi_{\theta,W}$ are the corresponding images at the relative camera π. Note that to reduce computation, we sample the rays $\boldsymbol{r}$ with batch size B from all views, supervise only the corresponding pixels, and finetune for N steps. The algorithm details are shown in Algorithm 1.

Relationship to SDS   Our method shares similarities with the recently proposed *score distillation sampling* (SDS, Poole et al., 2022). Although SDS also *distills* the diffusion models into 3D, there is a fundamental difference. In SDS, a random-scaled noise is injected into NeRF's output from a random angle. The noised image is then denoised by a 2D diffusion model to provide supervision. In contrast, our method initializes a set of virtual views and uses NeRF to guide the diffusion process of each view (and alternatingly refines NeRF based on this diffusion). As a result, our pipeline completes the full diffusion trajectory for every view, following a naturally decreasing noise schedule. In Appendix E, we show additional comparisons and potential reasons SDS is practically worse than our method.

Figure 4. A qualitative comparison of our approach versus baselines in single-image view synthesis on multiple datasets. Compared to 3D methods like VisionNeRF (Lin et al., 2023) and Ours(w/o NGD), our proposed NerfDiff synthesizes significantly sharper results behind occlusions. Compared to Ours (CDM), our full model showcases its built-in multi-view consistency. The red arrows display the CDM's inability to synthesize consistently across views.

5. Experiments

5.1. Experimental Settings

Datasets We evaluate NerfDiff on three benchmarks – SRN-ShapeNet (Sitzmann et al., 2019a), Amazon-Berkeley Objects (ABO, Collins et al., 2022) and Clevr3D (Stelzner et al., 2021) – for testing novel view synthesis under single-category, category-agnostic, and multi-object settings, respectively. SRN-ShapeNet includes two categories: *Cars* and *Chairs*. Dataset details are given in Appendix A.
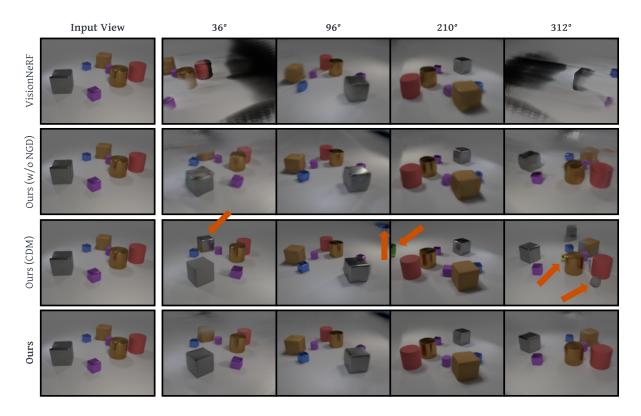
Baselines We choose the pixel-aligned method Vision-NeRF (Lin et al., 2023) as the main baseline for comparison considering its state-of-the-art performance in single-image view synthesis. We additionally evaluate our proposed single-image NeRF without the fine-tuning stage (denoted as "Ours (w/o NGD)"), Furthermore, show qualitative results from the CDM prediction without NeRF guidance (denoted as "Ours (CDM)"). Besides, we include publicly-available results for other methods such as SRNs (Sitzmann et al., 2019a), CodeNeRF (Jang & Agapito, 2021), FE-NVS (Guo et al., 2022), and geometry-free approaches LFN (Sitzmann et al., 2021) and 3DiM (Watson et al., 2022).

Evaluation Metrics We evaluate our model and the baselines by comparing the generated images and target views given a single image, and the relative target camera poses as input. We report four standard metrics: PSNR, SSIM, LPIPS (Zhang et al., 2018), and FID (Heusel et al., 2017). PSNR measures the mean-squared error per pixel, while SSIM measures the structural similarity; LPIPS is a deep metric that reflects the perceptual similarity between images. Finally, FID measures the similarity between the distribution of the rendered and ground truth images of all test scenes. Note that generative frameworks–due to their multimodal nature–generally perform poorly with respect to PSNR, which prioritizes proximity to the mean pixel values.

5.2. Main Results

We show results with two variant sizes (NerfDiff-B:∼ 400M parameters, NerfDiff-L:∼ 1B parameters). Details of the implementation specifics are given in Appendix B.

Quantitative evaluation Tables 1 and 2 show the quantitative comparisons of our proposed models to the SoTA geometry-free and single-view NeRF methods on all three datasets. The quantitative scores of the baselines are copied

Figure 5. A qualitative comparison on Clevr3d (Stelzner et al., 2021) which consists of images from cameras rotated 120 degrees about the *z*-axis. We showcase generalization to OOD cameras in this figure. As can be seen, VisionNeRF gets a degenerate result, while NerfDiff provides sharper renderings with fewer artifacts.

from the official papers if available. Our proposed NerfDiff (with and without NGD finetuning) significantly outperform all baselines in PSNR and SSIM, displaying the models' ability to synthesize accurate pixel-level details with the local triplane representations. Additionally, in LPIPS, our proposed NerfDiff is better than all previous approaches indicating its ability to create perceptually correct completions behind occlusions. Finally, about FID, our method outperforms all single-view NeRF methods, only having worse scores than 3DiM on ShapeNet-Cars as it is purely 2D. Note that, as mentioned in the original paper (Watson et al., 2022), 3DiM cannot generalize well to the out-of-the-distribution testing cameras of ShapetNet-Chairs, thus performing poorly. In contrast, with the 3D-aware CDM, our approach can easily handle unseen viewpoints. In addition, the proposed NGD finetuning, while slightly hurting PSNR in some cases, significantly improves the sharpness of the results, thus resulting in better FID and LPIPS scores. Besides, scaling up further yields higher perceptual quality.

Qualitative evaluation Figure 4 displays the qualitative comparison of our approach to the main baseline, Vision-NeRF (Lin et al., 2023), and two ablated models. Our method produces much more detailed results than the

Method	Image encoding	Rendering
PixelNeRF (Yu et al., 2021)	0.007s	1.639s
VisionNeRF (Lin et al., 2023)	0.015s	0.678s
NerfDiff-B	0.024s	0.018s
NerfDiff-L	0.031s	0.018s

Table 3. Comparison of encoding and rendering speed on ShapeNet Cars dataset between models.

ablated model single-image NeRF and VisionNeRF on ShapeNet and ABO. Due to their reliance on projected image features, these methods cannot handle uncertainty behind occlusion and thus regress mean pixel values, resulting in blurry renderings. The CDM results are worse aligned and inconsistent across views, as demonstrated by the red arrows in Figure 4. Figure 5 shows additional qualitative results on Clevr3D. Our method again shows consistent and high-quality renderings. At the same time, VisionNeRF overfits the camera distribution and fails to synthesize viewpoints close to the input (see Appendix B for more details). The CDM results are again inconsistent with objects appearing and disappearing. **Please refer to the supplementary materials for uncurated and extensive video results.**

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
No Fine-tuning	23.81	0.915	0.093	42.37
Fine-tuning				
Direct distillation	23.46	0.911	0.105	35.88
w. SDS (Poole et al., 2022)	20.32	0.882	0.106	28.06
w. NGD (Ours)	**23.51**	**0.917**	**0.082**	**18.09**

Table 4. Ablation on fine-tuning strategy on ShapeNet-Cars.

# virtual views	5	10	25	50	100	w/o NGD
PSNR↑	22.86	23.16	23.34	23.51	**23.55**	23.81
SSIM↑	0.901	0.913	0.915	**0.917**	0.916	0.915
LPIPS↓	0.095	0.085	**0.083**	**0.083**	0.087	0.093
FID↓	27.41	**16.04**	17.14	18.09	19.13	42.37

Table 5. Comparison on the number of virtual views used for fine-tuning on ShapeNet Cars.

5.3. Ablation Studies

We provide ablations on the Shapenet-Cars dataset to validate our model's key design choices, making our ablation results directly comparable to the ShapeNet Cars results in Figure 4. In Table 4, we compare our model without any CDM-based finetuning and various CDM-based finetuning strategies. As seen in the results, finetuning with a CDM will improve the unconditional FID. However, only our NGD sampling will yield the state-of-the-art conditional SSIM and LPIPS. For details of the sampling baselines Direct Distillation and SDS compared to our NGD, please see Appendix E. Figure 6 also provides a qualitative comparison. Next, in Table 5, we also provide ablations on the number of virtual views for finetuning. With too few (e.g. 5) virtual views, the NeRF overfits the denoised images resulting in subpar renderings. We find that 50 virtual views provide a good tradeoff between efficiency and performance.

6. Discussion

Limitations Our proposed method has two main limitations. Firstly, we require at least two views of a scene at training time. Secondly, our finetuning process is expensive in time, limiting application in real-time domains. Future work may address these issues.

Future work For future research, it is also possible to investigate our proposed NGD to improve the fidelity of text-to-3D pipelines (Jain et al., 2022; Poole et al., 2022; Lin et al., 2022). Additionally, more complex datasets such as the Waymo Open Dataset (Sun et al., 2020) may be explored, leaving the challenging task of occlusion-handling to large-scale pretrained 2D diffusion models as we do in this paper. Also, incorporating our finetuning strategy in the context of 3D GANs (Gu et al., 2021; Chan et al., 2021) may

improve inversion performance and 3D editing capabilities as well. Finally, it may also be interesting to figure out how to properly incorporate priors such as symmetry.

7. Conclusion

We introduced NerfDiff, a generative framework for single-image view synthesis which distills a 3D-aware CDM to a triplane-based image-conditioned NeRF. We further introduced NeRF-guided distillation to sample multiple views from the CDM while simultaneously improving the NeRF renderings. Our method achieved the state-of-the-art results on multiple challenging benchmarks.

Acknowledgements

References

Anciukevicius, T., Xu, Z., Fisher, M., Henderson, P., Bilen, H., Mitra, N. J., and Guerrero, P. RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation. *arXiv*, 2022.

Bautista, M. A., Guo, P., Abnar, S., Talbott, W., Toshev, A., Chen, Z., Dinh, L., Zhai, S., Goh, H., Ulbricht, D., Dehghan, A., and Susskind, J. Gaudi: A neural architect for immersive 3d scene generation. *arXiv*, 2022.

Cai, S., Obukhov, A., Dai, D., and Van Gool, L. Pix2nerf: Unsupervised conditional π-gan for single image to neural radiance fields translation. *arXiv preprint arXiv:2202.13162*, 2022.

Chan, E. R., Lin, C. Z., Chan, M. A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., et al. Efficient geometry-aware 3d generative adversarial networks. *arXiv preprint arXiv:2112.07945*, 2021.

Chen, A., Xu, Z., Geiger, A., Yu, J., and Su, H. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022.

Chen, S. and Williams, L. View Interpolation for Image Synthesis. In *SIGGRAPH 93*, pp. 279–288, 1993.

Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Yago Vicente, T. F., Dideriksen, T.,

Arora, H., Guillaumin, M., and Malik, J. Abo: Dataset and benchmarks for real-world 3d object understanding. *CVPR*, 2022.

Deng, C., Jiang, C., Qi, C. R., Yan, X., Zhou, Y., Guibas, L., Anguelov, D., et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. *arXiv preprint arXiv:2212.03267*, 2022.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Dupont, E., Martin, M. B., Colburn, A., Sankar, A., Susskind, J., and Shan, Q. Equivariant neural rendering. In *International Conference on Machine Learning*, pp. 2761–2770. PMLR, 2020.

Eigen, D., Puhrsch, C., and Fergus, R. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. *NIPS'14*, pp. 1–9, 2014.

Fan, H., Su, H., and Guibas, L. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 605–613, 2017.

Gortler, S., Grzeszczuk, R., Szeliski, R., and Cohen, M. The Lumigraph. In *SIGGRAPH 96*, pp. 43–54, 1996.

Gu, J., Liu, L., Wang, P., and Theobalt, C. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

Guo, P., Bautista, M. A., Colburn, A., Yang, L., Ulbricht, D., Susskind, J. M., and Shan, Q. Fast and explicit neural view synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3791–3800, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Jain, A., Mildenhall, B., Barron, J. T., Abbeel, P., and Poole, B. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 867–876, 2022.

Jang, W. and Agapito, L. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12949–12958, 2021.

Kato, H., Ushiku, Y., and Harada, T. Neural 3D Mesh Renderer. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3907–3916, 2018.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2015.

Levoy, M. and Hanrahan, P. Light Field Rendering. In *SIGGRAPH 96*, pp. 31–42, 1996.

Li, G., Zheng, H., Wang, C., Li, C., Zheng, C., and Tao, D. 3ddesigner: Towards photorealistic 3d object generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2211.14108*, 2022.

Lin, C.-H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.-Y., and Lin, T.-Y. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022.

Lin, K.-E., Yen-Chen, L., Lai, W.-S., Lin, T.-Y., Shih, Y.-C., and Ramamoorthi, R. Vision transformer for nerf-based view synthesis from a single input image. In *WACV*, 2023.

Liu, A., Tucker, R., Jampani, V., Makadia, A., Snavely, N., and Kanazawa, A. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14458–14467, 2021.

Loper, M. M. and Black, M. J. OpenDR: An approximate differentiable renderer. In *Computer Vision – ECCV 2014*, volume 8695 of *Lecture Notes in Computer Science*, pp. 154–169. Springer International Publishing, 2014.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

McMillan, L. and Bishop, G. Plenoptic Modeling: An Image-Based Rendering System. In *SIGGRAPH 95*, pp. 39–46, 1995.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pp. 405–421. Springer, 2020.

Müller, N., Siddiqui, Y., Porzi, L., Bulò, S. R., Kontschieder, P., and Nießner, M. Diffrf: Rendering-guided 3d radiance field diffusion. *arXiv preprint arXiv:2212.01206*, 2022a.

Müller, N., Simonelli, A., Porzi, L., Bulò, S. R., Nießner, M., and Kontschieder, P. Autorf: Learning 3d object radiance fields from single view observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3971–3980, 2022b.

Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., and Chen, M. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

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

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Rematas, K., Martin-Brualla, R., and Ferrari, V. ShaRF: Shape-conditioned Radiance Fields from a Single View. *ICML*, 2021.

Rockwell, C., Fouhey, D. F., and Johnson, J. Pixelsynth: Generating a 3d-consistent experience from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14104–14113, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2021a.

Rombach, R., Esser, P., and Ommer, B. Geometry-Free View Synthesis: Transformers and no 3D Priors. *arXiv:2104.07652*, 2021b.

Rombach, R., Esser, P., and Ommer, B. Geometry-free view synthesis: Transformers and no 3d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14356–14366, 2021c.

Ronneberger, O., Fischer, P., and Brox, T. U-Net : Convolutional Networks for Biomedical Image Segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Sajjadi, M. S. M., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lucic, M., Duckworth, D., Dosovitskiy, A., Uszkoreit, J., Funkhouser, T., and Tagliasacchi, A. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. *CVPR*, 2022. URL https://srt-paper.github.io/.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Saxena, A., Sun, M. S. M., and Ng, A. Y. a. Learning 3-D Scene Structure from a Single Still Image. *IEEE PAMI*, 31(5):824–840, 2009. ISSN 1550-5499. doi: 10.1109/ICCV.2007.4408828.

Shue, J. R., Chan, E. R., Po, R., Ankner, Z., Wu, J., and Wetzstein, G. 3d neural field generation using triplane diffusion. *arXiv preprint arXiv:2211.16677*, 2022.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. *Advances in Neural Information Processing Systems*, pp. 1119–1130, 2019a.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. *Advances in Neural Information Processing Systems*, pp. 1119–1130, 2019b.

Sitzmann, V., Rezchikov, S., Freeman, W. T., Tenenbaum, J. B., and Durand, F. Light Field Networks : Neural Scene Representations with Single-Evaluation Rendering. *arXiv:2106.02634*, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Stelzner, K., Kersting, K., and Kosiorek, A. R. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148*, 2021.

Suhail, M., Esteves, C., Sigal, L., and Makadia, A. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pp. 156–174. Springer, 2022.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

Tatarchenko, M., Dosovitskiy, A., and Brox, T. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs. *IEEE International Conference on Computer Vision*, pp. 2088–2096, 2017.

Trevithick, A. and Yang, B. GRF: Learning a General Radiance Field for 3D Representation and Rendering. *International Conference on Computer Vision*, 2021.

Tulsiani, S., Zhou, T., Efros, A. A., and Malik, J. Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2626–2634, 2017.

Wang, H., Du, X., Li, J., Yeh, R. A., and Shakhnarovich, G. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *arXiv preprint arXiv:2212.00774*, 2022.

Watson, D., Chan, W., Martin-Brualla, R., Ho, J., Tagliasacchi, A., and Norouzi, M. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022.

Wiles, O., Gkioxari, G., Szeliski, R., and Johnson, J. SynSin: End-to-end View Synthesis from a Single Image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Xu, D., Jiang, Y., Wang, P., Fan, Z., Shi, H., and Wang, Z. Sinnerf: Training neural radiance fields on complex scenes from a single image. *arXiv preprint arXiv:2204.00928*, 2022.

Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision. *Advances in Neural Information Processing Systems*, pp. 1696–1704, 2016.

Yu, A., Ye, V., Tancik, M., and Kanazawa, A. pixelNeRF: Neural Radiance Fields from One or Few Images. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Zhou, Z. and Tulsiani, S. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. *arXiv preprint arXiv:2212.00792*, 2022.

Zhuang, P., Abnar, S., Gu, J., Schwing, A., Susskind, J. M., and Bautista, M. Á. Diffusion probabilistic fields. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ik91mY-2GN.

Appendix

A. Datasets

ShapeNet ShapeNet-Cars and -Chairs (Sitzmann et al., 2019a) are standard for few-shot view synthesis benchmarking. We use the data hosted by pixelNeRF (Yu et al., 2021), which can be downloaded from GitHub (`https://github.com/sxyu/pixel-nerf`). The chairs dataset consists of 6591 scenes, and the cars dataset has 3514 scenes, both with a predefined train/val/test split. Each training scene contains 50 posed images taken from random points on a sphere. Each testing scene contains 250 posed images taken on an Archimedean spiral along the sphere. All scenes share intrinsic, and images are rendered at a resolution of $(128, 128)$. At testing, we choose one pose as input (index 64) and keep this camera input constant for all scenes.

Amazon Berkeley Objects (ABO) We also consider the ABO dataset (Collins et al., 2022) from `https://amazon-berkeley-objects.s3.amazonaws.com/index.html` under the title "ABO 3D Renderings." We randomly sampled a custom split. The dataset thus consists of 6743 training scenes, 396 validation scenes, and 794 testing scenes. Each scene consists of 30 images of an object rendered onto a white background in a physically-based manner. The objects are drawn from 64 different object categories, providing an extensive evaluation of the generalization capabilities of various models. The images have resolution $(256, 256)$, and we crop and adjust the intrinsics so that the models are fed with images of size $(128, 128)$. The cameras are not uniformly distributed, but all point at the object. For testing, we use an input camera index of 0.

Clevr3D We consider the Clevr3D dataset provided in (Stelzner et al., 2021) for multi-object/scene level learning, which can be downloaded from the github `https://github.com/stelzner/obsurf`. We define a custom split in which there are 70000 training scenes and 1000 held-out testing scenes. Each scene consists of 3 posed images at a resolution of $(120, 160)$ with the camera pointing at the origin. These images are rendered at 120 degree rotations about the z-axis with varying distances from the origin. We select one input image (index 0) at testing and render the other two views.

B. Implementation Details

B.1. Architecture and Hyperaparameters

For all datasets, we learn NerfDiff based on the U-Net architecture adopted from ADM (Dhariwal & Nichol, 2021) with two sets of configurations (-B: base $\sim$ 400M parameters,

-L: large $\sim$ 1B parameters). More specifically, we set the model dimension $d = 192$ with 2 residual blocks per resolution for the base architecture and $d = 256$ with 3 residual blocks per resolution for the large architecture. All other hyperparameters follow the default setting as ADM.

Note that the image encoder retains the same architecture and hyperparameters as the CDM outlined above. Similar to (Watson et al., 2022), we incorporate a cross-attention module between the CDM and the image encoder after every attention block to strengthen the conditioning. The last layer output of the image encoder is reshaped to a triplane. As a result, the triplane has the same spatial resolution as the input image, and we set the feature dimension of the triplane as 48. We implement the NeRF module (pink box in Fig. 3) using a 2-layer MLP with a hidden size of 64. For NeRF rendering, we follow Lin et al. (2023) and uniformly sample 64 points along each ray, with 64 additional points by importance sampling. As mentioned in § 4.2, we directly concat the NeRF rendered image with the noised input and send it to the CDM for denoising.

B.2. Training phase

The CDM is trained with cosine noise schedule $\alpha_t = \cos(0.5\pi t)$ based on velocity prediction (Salimans & Ho, 2022). We set $\lambda_{IC} = \lambda_{DM} = 1$, which means that we add the two losses of the two modules without re-weighting. All models are trained using AdamW (Loshchilov & Hutter, 2017) with a learning rate of $2e-5$ and an EMA decaying rate of 0.9999. We train all models with a batch size of 32 images for 500K iterations on 8 A100 GPUs. Training takes $3-4$ days to finish for base models.

Note that for the Clevr3D dataset (Stelzner et al., 2021), we noticed that the models tended to overfit to the input view easily, creating a plane of density orthogonal to the camera axis and thus clearly degenerate geometry. To ameliorate this, we trained input view reconstruction with slightly noisy camera locations (variance 0.3). We found that this fixed the issue for our method, but it still failed for VisionNeRF (Lin et al., 2023), even after increasing the noise.

B.3. Finetuning phase

When finetuning with NGD, we define K virtual views relative to the input image by sampling near the test trajectory. By default, we set $K = 50$ for shapeNet and Clevr3D and 30 for ABO. See Appendix C for specific K per dataset and how to obtain these poses. For the multiview diffusion process, we run 64 DDIM (Song et al., 2020) steps with the CDM for each view, respectively. At every diffusion step, we update the NeRF parameters $N = 64$ steps, with a batch size of $B = 4096$ rays. We use Adam optimizer (Kingma & Ba, 2015) set the learning rates for NeRF MLPs $1e-4$ and the triplane features $5e-2$, respectively. Our empirical

results indicate that a large learning rate on a triplane can boost the finetuning efficiency.

C. Prior Relative Camera Distribution Π

In order to approximate the expectation in Eq. (8), we require a sampling of Π, i.e., a sampling of K 'important' or 'relevant' cameras, which adequately capture the region of interest. For each of the three datasets, we rely on the relative (to the input) camera poses of the testing set for this. The cameras are very different between datasets, requiring a slightly different procedure.

ShapeNet Because the Shapenet testing trajectory is an Archimedean spiral around the object consisting of 251 views, we simply uniformly sample every 5th camera yielding 50 cameras total from which we can approximate Π, thus yielding $K = 50$ cameras to approximate Eq. (8).

Amazon Berkeley Objects (ABO) Because each scene contains only 30 cameras, we use all the relative poses of the testing set ($K = 30$) to approximate Eq. (8).

Clevr3D As Clevr3D contains only three cameras per scene, creating a good sample of Π is slightly more difficult. As we know, all of the Clevr3D cameras are pointing at the origin; we can calculate the relative position of the world origin in camera coordinates by intersecting two of the optical axes of the relative cameras. This will serve as the look-at-point for our virtual cameras. Note that the Clevr cameras are additionally all of a similar height relative to the ground plane of the scene. Thus, we can approximate the up direction in camera coordinates by taking the normal plane containing all three cameras. In order to resolve uncertainty about whether this is the up direction or its negation, we check that the cosine similarity with the camera directions is negative, as they lie in the upper halfspace in world coordinates. Given a camera center in camera coordinates, we can thus create a camera pose. To define these centers, we uniformly sample a circle in the plane containing all three cameras, which approximately goes through each one (the radius is the mean distance from the approximate world origin). We choose $K = 50$ camera centers and create camera poses with the estimated world origin and up direction for the finetuning process. We use these to approximate Eq. (8).

D. Details of Baseline Methods

The baselines are shown in Table. 1, we gathered the error metrics of LFN (Sitzmann et al., 2021), 3DiM (Watson et al., 2022) on the ShapeNet dataset from their respective papers. As for SRN (Sitzmann et al., 2019b), PixelNeRF (Yu et al., 2021), CodeNeRF (Jang & Agapito, 2021), FE-NVS (Guo et al., 2022), and VisionNeRF (Lin et al., 2023), we obtain

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
No CDM Fine-tuning	23.81	0.915	0.093	42.37
CDM architecture				
Concat	20.72	0.874	0.135	56.27
Cross-attention	21.13	0.885	0.123	35.49
3D-aware (Ours)	**23.51**	**0.917**	**0.082**	**18.09**

Table 6. Experiments of showing importance for 3D-aware CDM on ShapeNet-Cars.

the ShapeNet results from the VisionNeRF paper and conduct FID calculation using the renderings provided by the authors of PixelNeRF and VisionNeRF. Moreover, to compare against VisionNeRF on the ABO and Clevr3D datasets, we used its publicly available source code and modified the dataloader accordingly. For training, we use the same hyper-parameter setup denoted by the VisionNeRF paper. Namely, we set the image feature channels to 512. The learning rate of the feature extractor is set to $1e-5$ and MLP to $1e-4$. We keep the same learning rate schedule and apply warm-up and decay, as shown in the original paper. We trained VisionNeRF for 500k steps on the ABO dataset and 250k steps on the Clevr3D dataset since we found it easier to overfit the Clevr3D scenes. Moreover, we also adjusted the batch and ray bundle sizes to fit into the GPU memory.

E. Additional Comparison Details

E.1. Importance of 3D-aware Diffusion

In Table 6, we showed the comparison with different CDM architectures. "Concat" means directly concat the input view with the noisy image, while "Cross-attention" adopts a similar conditioning as X-UNet (Watson et al., 2022). Both did not involve the volume rendering in the encoding time, which can be seen as 3D-unaware. The results showed that, when applying a 3D-aware diffusion model, we can consistently achieve better results and generate more coherent views based on the input.

E.2. Fine-tuning Strategies

In Table 4 and Figure 6, we showed results for multiple sampling methods for finetuning NeRF. Here we give the details of these methods.

v.s. Direct distillation For Direct distillation, we directly sample virtual views from the CDM given the initial pixel-NeRF renderings and use these to finetune the NeRF directly with a standard L2 loss. Note that these renderings are unlikely to be multiview consistent as the denoising process takes place independently for each. Thus, the resultant renderings are inconsistent and incongruous with the input, which is also reflected in the learned NeRF.
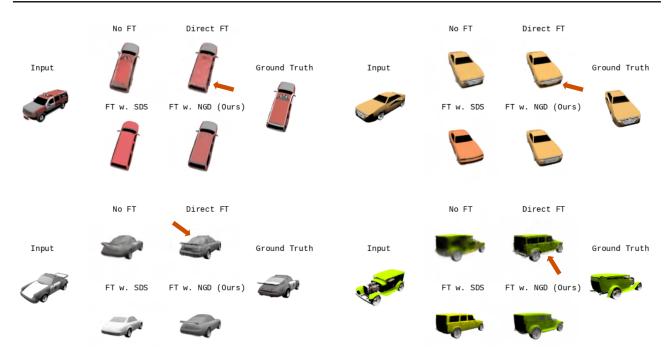
Figure 6. Qualitative examples of ablation studies on fine-tuning strategy. FT refers to finetuning. The red arrow shows floating and noisy artifacts due to learning from inconsistent CDM predictions.

v.s. Score distillation sampling (SDS) We also compared SDS (Poole et al., 2022), where virtual views are continually predicted by adding noise directly to renderings and taking an L2 loss between the NeRF rendering and the resultant denoised images. Here we note three significant differences with SDS, which may result in its poorer performance:

1. Inconsistent noise schedule. In our method, we only sample once per view, continually decreasing the noise with greater NeRF guidance. In contrast, SDS will provide inconsistent gradient updates as random amounts of noise are added to the NeRF renderings and then denoised, yielding blurry results which regress the mean of the supervision.

2. The learned score function of the CDM may be inadequate. That is to say, the modes of the PDF may not reflect sharp images of the dataset, causing poorer results. Our method uses NeRF to guide the process, which avoids directly seeking a mode.

3. Out-of-distribution inputs. At low noise levels, rendering a NeRF with additional noise will not resemble a real image with a similar amount of noise. Thus, the inputs to the denoiser may be out-of-distribution. In contrast, our method uses the CDM to refine the NeRF during sampling, keeping the samples close to the data manifold.

v.s. Stochastic Conditioning As shown in the main paper, naively finetuning the NeRF parameters from the CDM's generation typically leads to noisy results with severe floating artifacts. It is the inconsistent CDM predictions that cause conflicts in learning NeRF. Targeting on this, Watson et al. (2022) proposed "stochastic conditioning" – an autoregressive approach for synthesizing virtual views in a sequence, where for generating a novel view, each diffusion step stochastically conditions on previously generated views. Although this model's dependencies are across the virtual views, the generated images are not guaranteed to be multiview consistent. Moreover, our initial exploration showed that the imperfect autoregressive prediction accumulated errors easily, resulting in degenerated results for long sequences without stable geometry.

F. Additional Qualitative Results

Finally, we provide additional qualitative results for our base single-image models compared with VisionNeRF (Lin et al., 2023) on ShapeNet Cars (Figure 7), Chairs (Figure 8) and ABO dataset (Figures 9 and 10). Images are rendered at a specific viewpoint given a single image input. Please refer to supplementary materials for more video results.

Figure 7. Additional examples of single-image view synthesis on ShapeNet Cars.

| Input View | VisionNeRF | Ours (w/o NGD) | **Ours** | Ground Truth |

Figure 8. Additional examples of single-image view synthesis on ShapeNet Chairs.

| | Input View | VisionNeRF | Ours (w/o NGD) | **Ours** | Ground Truth |

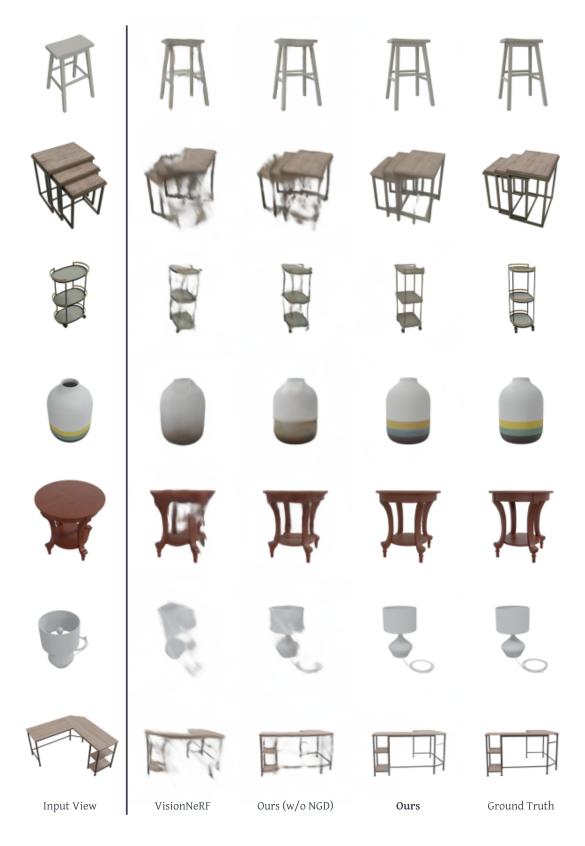*Figure 9.* Additional examples of single-image view synthesis on ABO dataset.

Figure 10. Additional examples of single-image view synthesis on ABO dataset.