ExtraNeRF: Visibility-Aware View Extrapolation of Neural Radiance Fields with Diffusion Models

Meng-Li Shih[1] Wei-Chiu Ma[1,2] Lorenzo Boyice[3] Aleksander Holynski[3,4] Forrester Cole[3]
Brian Curless[1,3] Janne Kontkanen[3]
[1] University of Washington [2] Cornell University [3] Google Research [4] UC Berkeley

BaseNeRF (View 1) ExtraNeRF (View 1) BaseNeRF (View 2) ExtraNeRF (View 2)

Figure 1. **BaseNeRF vs ExtraNeRF:** We train a BaseNeRF model and our ExtraNeRF model on six input views and render the scene from extrapolated viewpoints. Using our visibility-aware, diffusion-guided inpainting and enhancement modules, we are able to synthesize sharp content in disoccluded regions, whereas the BaseNeRF suffers from blurry results (see the red boxes, green boxes, and the close-up insets).

Abstract

We propose ExtraNeRF, a novel method for extrapolating the range of views handled by a Neural Radiance Field (NeRF). Our main idea is to leverage NeRFs to model scene-specific, fine-grained details, while capitalizing on diffusion models to extrapolate beyond our observed data. A key ingredient is to track visibility to determine what portions of the scene have not been observed, and focus on reconstructing those regions consistently with diffusion models. Our primary contributions include a visibility-aware diffusion-based inpainting module that is fine-tuned on the input imagery, yielding an initial NeRF with moderate quality (often blurry) inpainted regions, followed by a second diffusion model trained on the input imagery to consistently enhance, notably sharpen, the inpainted imagery from the first pass. We demonstrate high-quality results, extrapolating beyond a small number of (typically six or fewer) input views, effectively outpainting the NeRF as well as inpainting newly disoccluded regions inside the original viewing volume. We compare with related work both quantitatively and qualitatively and show significant gains over prior art.

1. Introduction

Reconstructing a scene from photographs is an important and long-standing problem in computer vision. Recent advances, following the introduction of Neural Radiance Fields (NeRF) [29] have led to an explosion of progress. Nevertheless, a limitation of NeRF in its base form is that it is far better at interpolating than extrapolating, and requires dense views for the interpolation. But what if you want to take just a few views, a practical constraint in a live capture setting, and extrapolate beyond them to enable a bit more freedom in viewing the scene? While there has been significant progress in scene-level sparse NeRF reconstruction, the progress on NeRF-based view extrapolation is primarily limited to object-centric scenarios. Advances in generative techniques, particularly diffusion models, have demonstrated unforeseen capabilities to synthesize previously unseen imagery. This presents an opportunity to expand the operating range of NeRF more broadly to view extrapolation.

Our core strategy employs neural radiance fields (NeRF [29]) to capture scene-specific, fine-grained details and utilizes 2D diffusion models [40] to extend the scene beyond the limits of observed data. A straightforward fusion of these technologies initially results in NeRF-rendered images that appear blurry and detail-deficient. This is primarily due to the discord between 2D diffusion priors when applied to a 3D scene from varying perspectives, particularly evident in scene-level view extrapolation where intricate details (such as leaves and branches) are significantly diminished.

To address these challenges, we develope a multi-stage

process (see Fig. 2) that includes: (1) employing a specialized visibility module to identify all 3D content which is visible from the observed data; (2) utilizing a visibility-aware inpainting module, which is tailored for each scene, to imagine and add plausible 3D content into NeRF for view extrapolation and ensure the content from observed data remains unaltered; and (3) enriching view-consistent details in hallucinated content using a carefully designed diffusion enhancement model.

Through this novel pipeline, we demonstrate high quality view extrapolation from a small number of input views, filling in the newly revealed areas outside the original view volume (see Fig. 1). Our qualitative and quantitative evaluation show significant gains over previous work.

2. Related Work

View synthesis: Given a set of posed images, the goal of view synthesis is to simulate how a scene would look like from novel viewpoints [8, 21, 52]. The problem is traditionally formulated as an image-based rendering task [12, 73], and impressive results can be achieved by blending pixel colors across views based on depth maps [7] or by compositing images using proxy geometry [19]. Recently, with the help of deep neural networks [26, 37, 38, 57, 72], the results have been further improved. Together with carefully curated scene representations [15, 45, 56, 62], researchers have been able to synthesize novel views even from a single image [39, 55, 60]. Similar to the these recent efforts, our work seeks to extrapolate beyond what is visible and predict the content that is occluded in all images. However, instead of relying on deep nets to learn the geometric relationships and hallucinate the content in a purely data-driven fashion, we bake the 3D inductive biases (*e.g.*, visibility) into the pipeline to ground the generation process. This allows us to generate high-quality, realistic and coherent scene content.

Neural radiance fields (NeRF): NeRF [29] has revolutionized the field with its simplicity and extraordinary performance [5, 11, 22, 23, 50, 64, 69, 70]. However, existing NeRF-based models tend to be under-constrained, leading to the following limitations: first, they require *dense* observations of the scene; and second, their performance degrades significantly when extrapolating rather than interpolating. To alleviate these issues, researchers have proposed regularizing the underlying scene representation by data-driven statistics [16, 33, 63] or geometry constraints [54]. While these approaches greatly reduce the required number of input images, they still assume that the input views have a wide coverage of the scene. The task thus still falls under the view interpolation setup. In this paper, we focus on a common yet extremely challenging setup in live capture setting where we only have access to a few images with small baselines. We show that by carefully integrating generative models with NeRF, we can effectively expand the operating range of NeRF and produce high-quality renderings.

Diffusion models: Diffusion models [13, 40, 48, 49] have drawn wide attention across the vision community due to their capacity and scalability. They have demonstrated remarkable performance on a plethora of 2D tasks such as image inpainting [28, 42], deblurring [20, 61], and have enabled high-quality, diverse image generation [40, 43]. By combining with neural rendering [29], the learned diffusion priors can be further lifted to 3D to enable applications such as text-to-3D [24, 35, 51, 59] or single-/multi-image 3D generation [25, 27, 36, 44, 46, 47, 53]. Similar to these works, we also leverage diffusion models to synthesize novel views and fuse the generation results back to 3D. However, rather than focusing on object-centric setup, we study how to model the 3D content of the scene. Furthermore, we explicitly track the visibility across views, which allows us to produce both realistic and consistent 3D reconstructions. Concurrently with our work, Sargent *et al*. [44] also attempt to extrapolate 3D scenes. While their focus is primarily on generating content beyond the visible image boundaries, our approach predicts both disoccluded regions and areas that are not observed.

3. Preliminaries

Neural radiance fields: A neural radiance field (NeRF [29]) is an implicit scene representation. At its core lies a continuous function $f_\theta : \mathbb{R}^3 \times \mathbb{R}^2 \mapsto \mathbb{R}^+ \times \mathbb{R}^3$, parameterized by a neural network, that maps a 3D point $\mathbf{x} \in \mathbb{R}^3$ and a view direction $\mathbf{d} \in \mathbb{R}^2$ to a volume density $\sigma \in \mathbb{R}^+$ and an RGB radiance $\mathbf{c} \in \mathbb{R}^3$. A NeRF can be rendered into a 2-d image as follows. For each pixel, we cast a ray $\mathbf{r}(s) = \mathbf{o} + s\mathbf{d}$ from the camera center $\mathbf{o}$ through the pixel center in direction $\mathbf{d}$, and sample a set of 3D points along the ray and query their radiance and density. Then we aggregate the samples and obtain the color of the pixel via volume rendering:

$$\mathbf{C}(\mathbf{r}) = \sum_{i=1}^{N_\mathbf{r}} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i. \tag{1}$$

Here, $\delta_i = s_{i+1} - s_i$ is the distance between adjacent samples, and $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ represents the accumulated transmittance along the ray till s_i. Intuitively, one can think of T_i as *visibility*, since it is the probability that the ray travels to s_i *without* hitting any other particle.

The volume rendering operation is generic and can be adapted to render other properties of the scene, such as geometry (*i.e*. depth) or visibility (see Sec. 4.3). For instance, by replacing the color radiance $\mathbf{c}_i$ in Eq. 1 with distance s_i,

we can compute the expected termination depth:

$$\mathbf{D}(\mathbf{r}) = \sum_{i=1}^{N_\mathbf{r}} T_i(1 - \exp(-\sigma_i\delta_i))s_i. \quad (2)$$

The neural radiance field f_θ is learned on a *per-scene basis*. Given a *dense* set of images, the parameters θ can be learned by minimizing the discrepancy between target pixel colors $\mathbf{C}^{\text{target}}(\mathbf{r})$ and the colors rendered by corresponding rays $\mathbf{C}(\mathbf{r})$, *i.e.* $L^{\text{rgb}} = \sum_\mathbf{r} \|\mathbf{C}^{\text{target}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2$. If depth information is available, or can be computed with methods such as multi-view stereo, one can additionally adopt geometric supervision: $L^{\text{depth}} = \sum_\mathbf{r} \|\mathbf{D}^{\text{target}}(\mathbf{r}) - \mathbf{D}(\mathbf{r})\|_2^2$. As we will show in the later sections, explicitly regularizing the geometry of the underlying 3D scene is critical when only a few input images are available. It can also enable better extrapolation to unseen, disoccluded regions.

Diffusion models: Diffusion [13, 40, 48, 49] has emerged as a powerful approach for generative image synthesis. Diffusion models rely on the learned denoising module $\Psi(x_t, t, l)$ that takes a noisy input image x_t and possible extra conditioning signals (*e.g.*, text prompts l, timestep t), and predicts the noise ϵ. By iteratively predicting the noise and subtracting it from the data, the model gradually converts the original noisy data x_t to a target sample of interest x. To train such a model, various levels of Gaussian noise ϵ are added to original clean data points, and the denoiser Ψ is tasked with predicting the noise:

$$L = \mathbb{E}_{x,t,\epsilon}\|\epsilon_\Psi(x_t, t, l) - \epsilon\|_2^2 \quad (3)$$

Since training diffusion models from scratch is often costly and requires large amount of data, researchers typically fine-tune pre-trained models for specific tasks using Eq. 3 with a smaller, domain-specific dataset. This fine-tuning can be achieved either by directly adjusting the weights of the denoiser Ψ [14, 41] or by introducing additional parameters such as learnable embeddings [10].

4. Method

Given a sparse set of images of the scene, our goal is not only to synthesize photo-realistic results between the input views, but also generate high-quality view extrapolations with inpainted disocclusions.

In this section, we first briefly review the basic building blocks of our approach. Next, we explain each component in more detail. Finally, we discuss how we fine-tune our diffusion models and other design choices.

4.1. Extrapolating Neural Radiance Fields

We create a NeRF capable of view extrapolation in three steps (see Fig. 2):

1. **Training the BaseNeRF**: We follow a standard process to train a NeRF on a sparse set of input images.
2. **Diffusion-guided inpainting**: We iteratively optimize NeRF with virtual views and the original inputs. Each virtual view is rendered from the NeRF and then inpainted using our diffusion model. Then the NeRF can be supervised with this virtual image, backpropagating the newly inpainted regions to the NeRF. Through this iterative process, we construct a consistent neural radiance field that extends beyond the original input images.
3. **Diffusion guided enhancement**: We find that the previous iterative optimization tends to introduce blur and color drift in the inpainted regions. In the final stage, we use a fine-tuned diffusion model to increase sharpness and improve color consistency in these regions.

We now describe each component in more detail.

Training the BaseNeRF: Given a sparse set of images $\{I_i\}_{i=1}^n$ and their associated camera poses $\{\Pi_i\}_{i=1}^n$, we first train a BaseNeRF (see Sec. 3). Due to the lack of dense multi-view images for effective regularization of the underlying 3D space, we utilize the method proposed in [54] to compute dense depth maps $\{D_i\}_{i=1}^n$ for each input image for geometric supervision. To further reduce "floater" artifacts (spuriously reconstructed bits of content in empty regions of the volume), we incorporate distortion loss [1] and hash decay loss [2] and apply gradient scaling [34] to regularize the learning procedure.

Diffusion-guided Inpainting: Once we have the BaseNeRF, the next step is to augment it such that it can handle extrapolated viewpoints.

To do this, we repeatedly optimize the NeRF over the set of original views and virtual views that extend beyond the original viewing domain. For each virtual view, we render it using the NeRF and then use a diffusion inpainting model Ψ^{inpaint} to predict the unobserved regions.

As our inpainting module Ψ^{inpaint}, we adopt the inpainting variant of latent diffusion from [40], which we further fine-tune on a per-scene basis (see Sec. 4.2). To limit the inpainting to the unobserved regions (*e.g.* areas where NeRF lacks supervision), our diffusion inpainter Ψ^{inpaint} takes three inputs: noisy image, visibility mask, and masked clean image that lacks data in areas to inpaint (see Fig. 3). The visibility masks are computed by checking whether the 3D sample points along the ray at each pixel have been observed in the training images (see Sec. 4.3).

For each virtual view, we also inpaint the depth conditioned on the inpainted color image using a depth completion network (see Sec. 4.3).

Once the image and depth for the virtual view are inpainted, they are used to further supervise the NeRF through

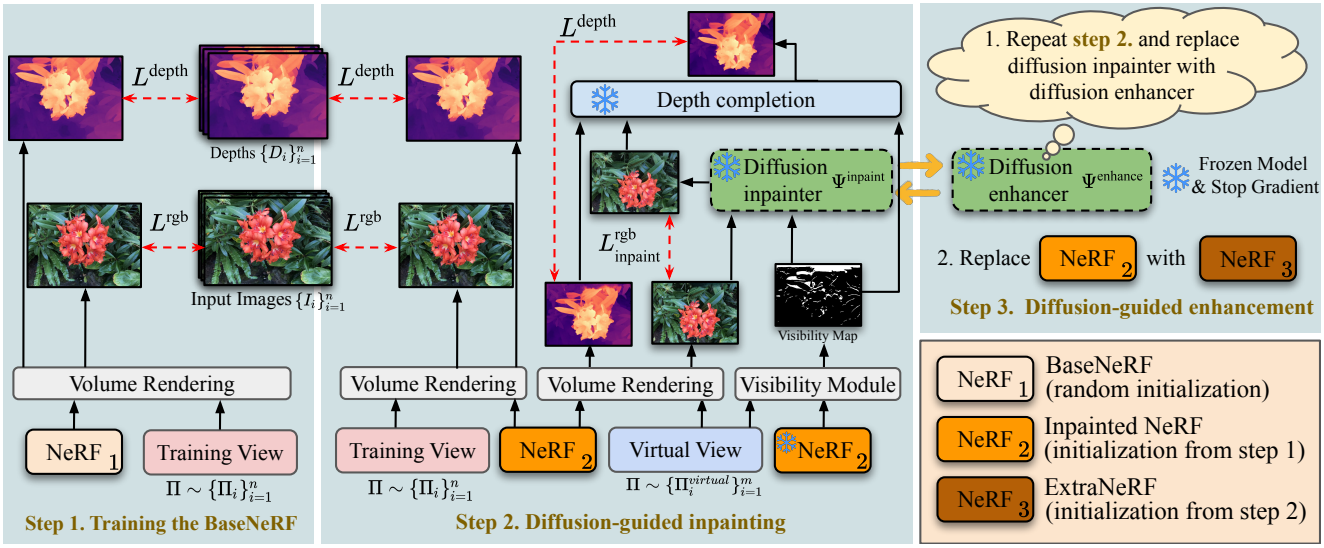

Figure 2. **Overview of our method:** We start from n input images, their camera poses, and depth maps (predicted as described in Sec. 4). In Step 1, we train a BaseNeRF by supervising with this input data. In Step 2, we add supervision from virtual views. We repeatedly inpaint the areas that are unsupervised by the original input views by a diffusion model while continuing to supervise the NeRF with the virtual views. In Step 3, we iterate in similar fashion, but instead of inpainting we apply another diffusion model specifically designed to further improve the detail and color consistency in inpainted regions.

$L_{\text{inpaint}}^{\text{rgb}}$ and L^{depth} respectively (see Fig. 2). $L_{\text{inpaint}}^{\text{rgb}}$ is computed as follows:

$$L_{\text{inapaint}}^{\text{rgb}} = \sum_{\mathbf{r}} w(t) |\mathbf{C}^{\text{inpaint}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})|, \tag{4}$$

where $w(t)$ is a noise-level dependent weighting function, $\mathbf{C}^{\text{inpaint}}$ is the inpainted colors and $\mathbf{C}$ is the rendered image from NeRF. We chose to run small number of diffusion denoising steps on each virtual view at the time (*e.g.* 10), but we repeat the whole process by iterating over the views several times.

Note that while inpainting in multiple views separately could lead to inconsistencies, our iterative approach does converge, because at each virtual view the diffusion process is bootstrapped via the noisy image that is re-estimated from the continuously improving NeRF on every iteration. This is similar to [35], although in our work we opted to run more than one step of diffusion before we move to a new view.

Diffusion-guided enhancement: While the iterative inpainting converges into a consistent result, we have observed that some blurriness and color drift may still occur in the NeRF after the inpainting stage.

To alleviate this, we utilize a diffusion-based enhancement model, Ψ^{enhance}, which has the same architecture as Ψ^{inpaint} but specifically trained for the enhancement (see Sec. 4.2).

Similar to inpainting, we use an iterative approach to update our NeRF. In each training iteration we 1) render the image and compute the visibility mask from the NeRF, 2) create a triplet of input data from the rendered image and visibility mask, and 3) leverage our Ψ^{enhance} model to generate an enhanced image from the triplet. In contrast to the inpainting process, we do not mask out the pixels in the intact rendered image (see Fig. 3). Instead, we want Ψ^{enhance} to enhance detail in these areas. Once the enhanced image is generated, we then complete the depth. Finally, we supervise the NeRF following steps similar to the inpainting stage but replace $L_{\text{inpaint}}^{\text{rgb}}$ with $L_{\text{enhance}}^{\text{rgb}}$. $L_{\text{enhance}}^{\text{rgb}}$ is almost identical to $L_{\text{inpaint}}^{\text{rgb}}$ except that we replace $\mathbf{C}^{\text{inpaint}}$ with $\mathbf{C}^{\text{enhance}}$ (i.e. enhanced colors). As shown in Fig. 7, this process can improve detail and overall image quality.

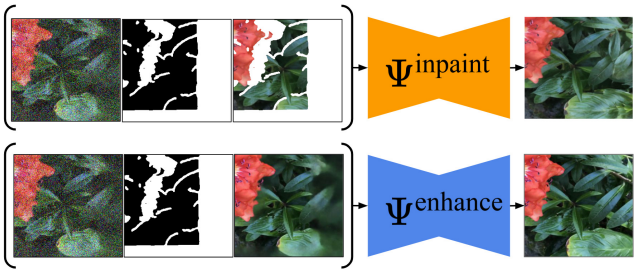

Figure 3. The input triplet of diffusion model consists of noisy-image, mask, and an guidance image. While masked pixels of guidance images of Ψ^{inpaint} are erased, they are preserved as the guidance for Ψ^{enhance}.

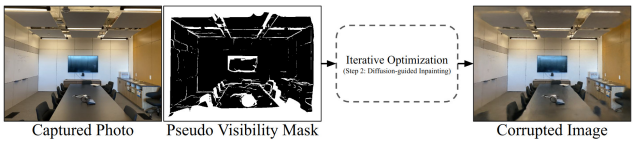

Figure 4. Illustration of data collection for enhancement model. We draw a pseudo visibility mask in a captured photo. Ground-truth supervision in the mask is replaced by inpainting supervision when we iteratively optimize NeRF. The optimization corrupts pixels in the mask when rendered with NeRF. A captured photo along with several corrupted images from different optimization iterations can used to train Ψ^{enhance}

4.2. Fine-tuning the Diffusion Models:

As mentioned earlier, we use the inpainting-variant of the latent diffusion model [40] for both inpainting and enhancement. For the best quality it is essential to fine-tune both Ψ^{inpaint} and Ψ^{enhance} for the scene and their respective tasks using our sparse set of input images $\{I_i\}_{i=1}^{n}$. This is perhaps obvious in the case of Ψ^{enhance} as its task is not exactly inpainting, but more similar to deblurring. However, as shown in Fig 7, scene-specific fine-tuning is also important for the inpainting module.

To fine-tune these two models Ψ^{inpaint} and Ψ^{enhance}, we devise a process to produce ground truth training data using our input images. The first step is to create visibility masks similar to those that would occur in the virtual views. For each training image, we compute a corresponding pseudo visibility mask by checking if pixels are visible in all other training views. Pixels not visible in one more view are treated as disocclusions, and we fine-tune Ψ^{inpaint} by asking it to inpaint the regions under these masks.

We fine-tune both models using standard diffusion loss (Eq. 3) following the DreamBooth [41] pipeline.

To produce training data for Ψ^{enhance} a further step is needed. To produce corrupted input for enhancement, we optimize a NeRF by intentionally replacing supervision from input viewpoints with inpainting supervision from Ψ^{inpaint} for pixels in the pseudo-disocclusion masks. We find that this adequately simulates the blur and color drift that the Ψ^{enhance} is tasked to reduce. See Fig. 4 for an example.

4.3. Implementation details

Visibility map: The visibility map indicates whether the 3D points corresponding to the pixels of a virtual view are visible in the input images. They might be hidden if they are outside the input view frustums or occluded by a closer object.

It plays a critical role in our system as it helps us determine which areas are unobserved in the original images and require inpainting. As indicated in Sec. 3, the accumulated transmittance from NeRF encodes essential visibility information. This enables us to estimate the visibility of any 3D point w.r.t the input views.

Figure 5. The depth completion model takes a masked depth along with a guidance image as input and completes the depth in the masked region using the guidance of the RGB image.

To compute the visibility map for a single pixel of a virtual view, we first construct a ray through that pixel. For each sampled 3D point along this ray, we then compute the transmittance towards each training view (e.g. another ray march). To aggregate the transmittance values across the input views, we simply select the second largest value. This is based on the rationale that the geometry of a 3D point is only reliable if observed by at least two views (the minimum for triangulation). If a 3D point is seen by only one training view, its estimated depth might be unreliable. Finally, these aggregated transmittance samples are aggregated together to the visibility map pixel by volume rendering, similarly to color values.

Depth completion module: We develop a depth completion module to complete the depth maps for virtual views required by L^{depth} (Fig. 5). The depth completion network takes the inpainted RGB image, visibility mask, and masked depth-map as input, and inpaints depth map in the masked region. The model is based on the pretrained weights of MiDaS-v3 [4] with two additional input channels for the input mask and masked depth-map. The model is fine-tuned with a self-supervised approach on the Places2 dataset [71] (see Suppl. for details).

Hyper-parameters: We fine-tune our inpainting and enhancement models for 500 iterations with a learning rate of 5e-6 for the diffusion U-Net and 4e-5 for the LoRA layer of the text encoder. Our NeRF uses Instant-NGP [32] as the backbone, with scene contraction [1] to handle unbounded scenes. We propose a 3-stage pipeline to train our NeRF. In step 1, we train the BaseNeRF for 5000 iterations using L^{rgb} and L^{depth} with an initial learning rate of 1e-2, gradually decreasing to 3e-4. In step 2, in addition to L^{rgb} and L^{depth}, the NeRF also receives supervision from the inpainting model Ψ^{inpaint} via $L^{\text{rgb}_{\text{inpaint}}}$ using the virtual views for 500 iterations. In Stage 3, we supervise the NeRF for another 500 iterations but replace the inpainting model Ψ^{inpaint} with the enhancement model Ψ^{enhance}.

Time consumption: In our experiments, we used one A100 GPU. Step 1, the bottleneck, involved 6 hrs of training Sparf [54] with early stopping at 30K iterations to create depth maps, and 8 minutes to train the BaseNeRF model. Steps 2 and 3 typically took 2-3 hrs, including 1 hour for data collection and fine-tuning two diffusion models, as well as 1-2 hrs for training the ExtraNeRF model. Our total optimization time is shorter than that of NeRF [29], which can take a day or more when running on a single GPU.

5. Experiments

5.1. Experimental setup

LLFF Datasets: We primarily utilize the LLFF dataset to demonstrate the effectiveness of our method. This dataset offers two settings for the training/test split that we have explored. In the first protocol, our goal is to assess performance in the task of view extrapolation. Therefore, 6 out of 30-40 images, whose viewpoints are closest to the center position, are chosen as the training set, and 8 images, whose viewpoints are farthest from the center position, are chosen as the test set (see Tab. 1). The second protocol follows the standard few-shot view synthesis setup [33] (see Tab. 2).

Tanks & Temples Datasets: We also utilize the Tanks & Temples dataset [18] to demonstrate our method's capability to manage more complex scenes in real-world settings. The data processed by NeRF++ [67] serves as our basis. In each scene, we select 3-5 nearby views as the training set and choose another 5-6 views whose viewpoints surround the training viewpoints, as the test set.

Metrics: We adopt the same metrics as [30], since our goals, akin to theirs, involve evaluating the performance of synthesized 3D content. Accordingly, we utilize two sets of metrics: full-reference (FR) and no-reference (NR). For the FR metrics, we exclusively use LPIPS[68], KID [3], and also include PSNR and SSIM [58] for a comprehensive assessment. However, it's noteworthy that PSNR and SSIM are not considered reliable metrics for evaluating generative tasks [6, 9, 44]. For NR metrics, MUSIQ [17] is employed to assess the visual quality of rendered images.

Baselines: We compare our method with six related baselines for which code is available: (1) Sparf [54], one of the state-of-the-art (SOTA) methods for sparse view reconstruction. (2) FreeNeRF [66], another SOTA method for sparse view reconstruction. (3) DiffusioNeRF [63], which employs a patch-wise diffusion model to provide RGB and depth supervision for a NeRF. (4) SPIn-NeRF [31], aimed at inpainting unobserved content behind an object in 3D, given a complete object mask. In our setting, where no object mask exists, we substitute it with a visibility mask, denoted as *SPIn-NeRF. (5) *SinNeRF [65], capable of extrapolating views in 3D from a single image and an accurate depth map. For a fair comparison, we provide RGB supervision from all images in the training set. (6) *SDS [35] loss, widely used in 3D content generation. Here, we substitute the color supervision from the inpainted image with SDS loss.

Table 1. Quantitative comparison of view extrapolation.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓	KID ↓	MUSIQ ↑
Sparf	20.38	0.650	0.324	0.0199	40.32
FreeNeRF	20.16	0.663	0.329	0.0203	39.51
DiffusioNeRF	19.94	0.683	0.296	0.0198	50.03
*SinNeRF	16.86	0.373	0.558	0.0458	31.54
*SPIn-NeRF	20.40	0.672	0.284	0.0156	51.84
*SDS	20.56	0.654	0.338	0.0351	49.35
Ours	**20.76**	**0.688**	**0.269**	**0.0154**	**54.13**

Table 2. Quantitative comparison of few-shot view synthesis [33].

Metrics	DiffusioNeRF	Sparf	FreeNeRF	Ours
PSNR ↑	19.79	20.20	19.63	**21.17**
SSIM ↑	0.568	0.630	0.612	**0.719**
LPIPS ↓	0.338	0.383	0.308	**0.264**

Table 3. Ablation study.

Methods	LPIPS ↓	KID ↓	MUSIQ ↑
BaseNeRF	0.323	0.0220	49.31
w/ pretrained Ψ^{inpaint}	0.291	0.0158	52.90
w/ fine-tuned Ψ^{inpaint}	0.282	0.0155	53.15
w/ fine-tuned Ψ^{inpaint} & Ψ^{enhance}	**0.269**	**0.0154**	**54.13**

5.2. LLFF

Comparison of view extrapolation: In Table 1, our method surpasses related works across various metrics, showcasing our approach's superior ability to inpaint unseen regions in view extrapolation tasks. Furthermore, Figure 6 presents a qualitative comparison, highlighting the distinctions between our method and competing approaches.

While Sparf and FreeNeRF demonstrate proficiency in estimating geometry and appearance for regions captured by input viewpoints, they fall short in generating meaningful content for view extrapolation scenarios. DiffusioNeRF, sharing our utilization of a diffusion prior to enhance NeRF quality, is limited by its patch-based model's narrow receptive field, preventing the synthesis of coherent content. Our diffusion model, in contrast, processes the entire image to generate meaningful and consistent content. *SPIn-NeRF, employing perceptual loss to address inconsistencies in su-

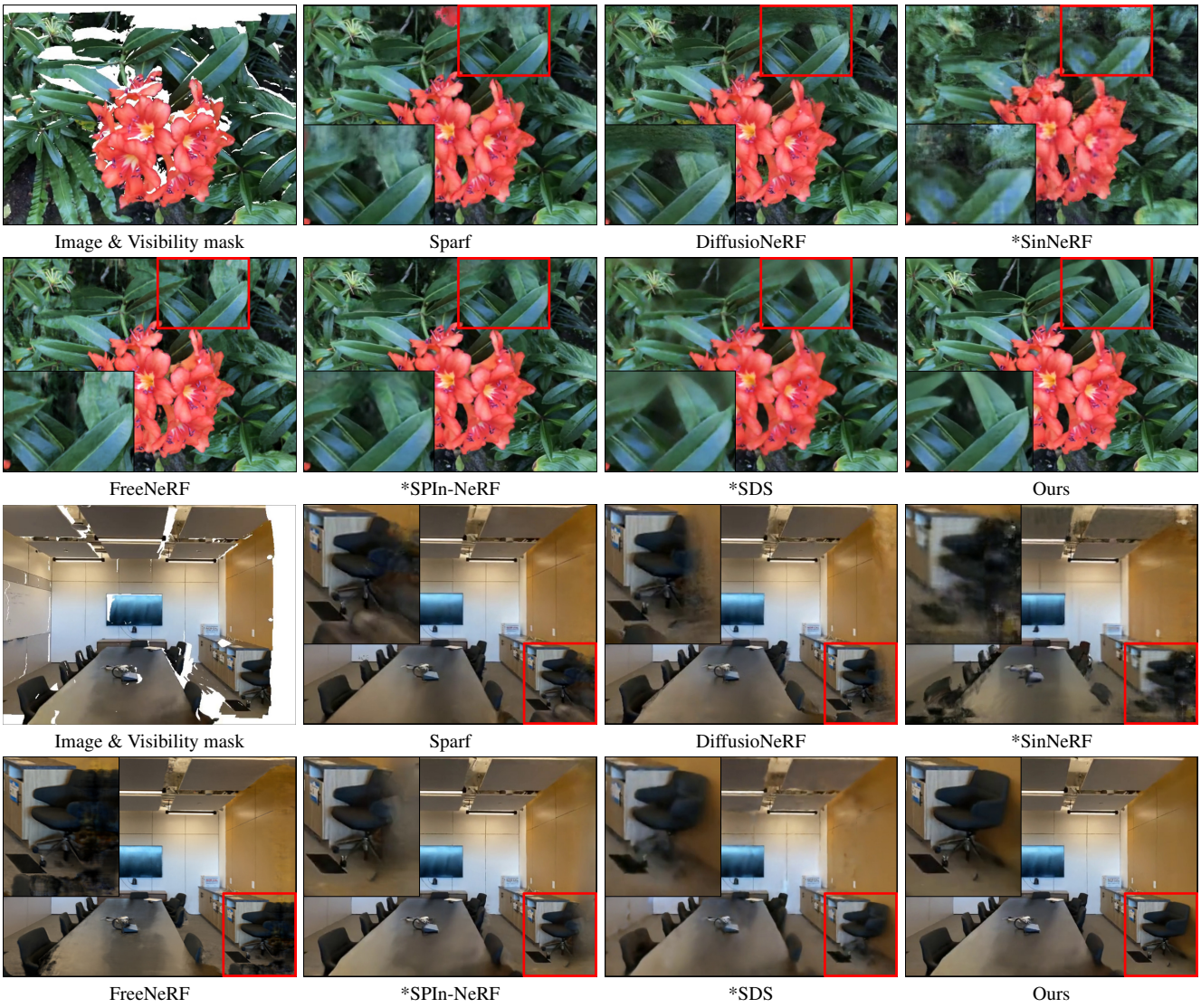

Figure 6. **Qualitative results of view extrapolation on LLFF dataset.** We present the image masked by the visibility mask on the left and the extrapolated results on the right. In comparison to the baselines, our results are significantly sharper and align coherently with the existing scene content. We are able to accurately reconstruct the structure of the leaves (top) and the partially observed chair (bottom), in contrast to the baselines, which yield blurry outcomes and struggle to differentiate between the foreground and background.

pervision from inpainted images, inadvertently introduces pattern artifacts. Additionally, while SDS loss can produce reasonable content, it often lacks the complexity of detail.

Compared to these methods, our technique excels in creating believable content that is both stylistically consistent and detailed.

Comparison of few-shot view synthesis: In Table 2, we demonstrate that our method outperforms other baselines in the few-shot view synthesis protocol with only 3 training views. This indicates that our approach can significantly reduce the number of required training views.

Ablations: In Figure 7, we illustrate the impact of removing components from our pipeline on the task of view extrapolation. The pretrained inpainting model struggles to fill masked regions with content that maintains consistent appearance and structure, leading to results that exhibit blurriness and color drift in NeRF, as depicted in Figure 7. By fine-tuning the inpainting model with the specific scene's captured photos, the diffusion model learns the scene's unique distribution, enabling it to more accurately generate content with consistent structure and appearance. Moreover, our enhancement model is capable of adding even greater detail than the fine-tuned inpainting model.

Figure 7. **Ablation study.** Inpainting disoccluded regions with a pretrained diffusion model results in blurry and color-drifted outcomes. However, fine-tuning the model on specific scenes reduces these issues, as the fine-tuned model captures scene-specific statistics more accurately. Additionally, our enhancement model further enables fine-grained details in NeRF-rendered images, producing sharper results.

Figure 8. **Qualitative results on Tanks&Temples.** While only a very small portion of the deck and the tires of the truck is visible from input viewpoints (left), our model is still able to synthesis the missing content (right).

Table 4. Quantitative comparison on Tanks & Temples Dataset.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓	KID ↓	MUSIQ ↑
Sparf	**19.21**	0.576	0.518	0.149	36.07
*SPIn-NeRF	18.28	0.683	0.348	**0.042**	37.63
Ours	19.20	**0.718**	**0.312**	0.065	**40.85**

Additionally, the results in Table 3 further demonstrate the effectiveness of each component within our pipeline.

5.3. Tanks & Temples

Given that this dataset features a significant proportion of pixels with extremely large depth values, we limit our comparison to methods equipped to handle unbounded scenes. In Table 4, our method surpasses others in LPIPS and MUSIQ scores, signifying superior visual quality of our results. However, our KID score falls short of SPIn-NeRF's. We hypothesize that this is likely due to the dataset's small size that may be insufficient to accurately estimate the test set's distribution. Furthermore, we showcase an example highlighting our model's ability to synthesize extensive areas of content that are unobservable from the input viewpoints (see Figure 8).

6. Conclusion

In this paper, we present an approach to broadening viewing range for a NeRF captured from a small, narrowly grouped set of input images. Our method, dubbed as ExtraNeRF, uses a pretrained diffusion model to produce new detail in two ways: first to inpaint given a visibility mask computed from the NeRF itself, then to enhance detail. We find that per-scene fine-tuning, design of enhancement model, and our data collections are critical for achieving good results. We set a new SOTA for view extrapolation on the LLFF dataset and Tanks & Temples dataset.

Acknowledgment: This work was supported by the UW Reality Lab, Meta, Google, OPPO, and Amazon.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 3, 5

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023. 3

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 6

[4] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3. 1–a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023. 5

[5] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *ICCV*, 2021. 2

[6] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In *arXiv*, 2023. 6

[7] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. 2013. 2

[8] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *SIGGRAPH*, 1993. 2

[9] Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20637–20647, 2023. 6

[10] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 3

[11] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. 2

[12] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *SIGGRAPH*, 1996. 2

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2, 3

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3

[15] Ronghang Hu, Nikhila Ravi, Alexander C Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *ICCV*, 2021. 2

[16] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021. 2

[17] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5148–5157, 2021. 6

[18] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 6

[19] Johannes Kopf, Michael F Cohen, and Richard Szeliski. First-person hyper-lapse videos. *TOG*, 2014. 2

[20] Sangyun Lee, Hyungjin Chung, Jaehyeon Kim, and Jong Chul Ye. Progressive deblurring of diffusion models for coarse-to-fine image synthesis. *arXiv*, 2022. 2

[21] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996. 2

[22] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2

[23] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *CVPR*, 2023. 2

[24] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2

[25] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv*, 2023. 2

[26] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv*, 2019. 2

[27] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv*, 2023. 2

[28] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 2

[29] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 6

[30] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G Derpanis, and Igor Gilitschenski. Reference-guided controllable inpainting of neural radiance fields. 6

[31] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor

Gilitschenski, and Alex Levinshtein. SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2023. 6

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

[33] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. 2, 6

[34] Julien Philip and Valentin Deschaintre. Floaters no more: Radiance field gradient scaling for improved near-camera training. 2023. 3

[35] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 4, 6

[36] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv*, 2023. 2

[37] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2

[38] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[39] Chris Rockwell, David F Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *ICCV*, 2021. 2

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 3, 5

[41] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 3, 5

[42] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, 2022. 2

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[44] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 2, 6

[45] Yuan Shen, Wei-Chiu Ma, and Shenlong Wang. Sgam: Building a virtual 3d world through simultaneous generation and mapping. *NeurIPS*, 2022. 2

[46] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv*, 2023. 2

[47] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv*, 2023. 2

[48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2, 3

[49] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv*, 2020. 2, 3

[50] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021. 2

[51] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. *arXiv preprint arXiv:2310.16818*, 2023. 2

[52] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *ICCV*, 1998. 2

[53] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *arXiv*, 2023. 2

[54] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023. 2, 3, 6

[55] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *CVPR*, 2023. 2

[56] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[57] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *ECCV*, 2018. 2

[58] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[59] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 2

[60] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv*, 2022. 2

[61] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *CVPR*, 2022. 2

[62] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 2

[63] Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4180–4189, 2023. 2, 6

[64] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*, 2021. 2

[65] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 6

[66] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

[67] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 6

[68] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[69] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ToG*, 2021. 2

[70] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022. 2

[71] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Antonio Torralba, and Aude Oliva. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*, 2016. 5

[72] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv*, 2018. 2

[73] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *TOG*, 2004. 2