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) [6] 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 [6]) to capture scene-specific, fine-grained details and utilizes 2D diffusion models [9] 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. Our qualitative and quantitative evaluation show significant gains over previous work.

2. Preliminaries

Neural radiance fields: A neural radiance field (NeRF [6]) is an implicit scene representation that can be rendered into a 2-d image and depth map using $\mathbf{C}(\mathbf{r}) = \sum_{i=1}^{N_\mathbf{r}} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i$. and $\mathbf{D}(\mathbf{r}) = \sum_{i=1}^{N_\mathbf{r}} T_i(1 - \exp(-\sigma_i\delta_i))s_i$.. It can be supervised through target pixel colors via $L^{\text{rgb}} = \sum_\mathbf{r} \|\mathbf{C}^{\text{target}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2$. Besides, it can be supervised by $L^{\text{depth}} = \sum_\mathbf{r} \|\mathbf{D}^{\text{target}}(\mathbf{r}) - \mathbf{D}(\mathbf{r})\|_2^2$ if traget depth is available.

Diffusion models: Diffusion models [5, 9–11] 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. It is trained by $L^{diffusion} = \mathbb{E}_{x,t,\epsilon}\|\epsilon_\Psi(x_t, t, l) - \epsilon\|_2^2$.

3. 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.

3.1. 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. 2). Due to the lack of dense multi-view images for effective regularization of the underlying 3D space, we utilize the method proposed in [12] 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 [7] 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 [9], which we further fine-tune on a per-scene basis. 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. 3.2).

For each virtual view, we also inpaint the depth conditioned on the inpainted color image using a depth completion network (see Sec. 3.2).

Once the image and depth for the virtual view are inpainted, they are used to further supervise the NeRF through $L^{\text{rgb}}_{\text{inpaint}}$ and L^{depth} respectively (see Fig. 2). $L^{\text{rgb}}_{\text{inpaint}}$ is computed as follows:

$$L^{\text{rgb}}_{\text{inapint}} = \sum_\mathbf{r} w(t)|\mathbf{C}^{\text{inpaint}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})|, \tag{1}$$

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

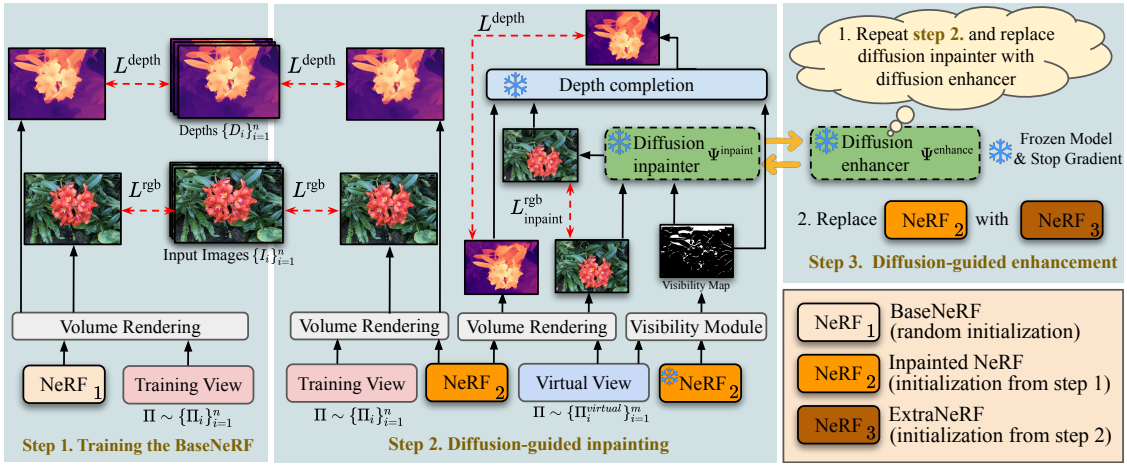

Figure 2. **Overview of our method:** We start from n input images, their camera poses, and depth maps (predicted as described in Sec. 3). In Step 1, we train a BaseNeRF by supervising with this input data. In Step 2, we add supervision from virtual views. We repeatedly inpaint the areas that are unsupervised 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.

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 [8], 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.

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 $L^{\text{rgb}}_{\text{inpaint}}$ with $L^{\text{rgb}}_{\text{enhance}}$. $L^{\text{rgb}}_{\text{enhance}}$ is almost identical to $L^{\text{rgb}}_{\text{inpaint}}$ except that we replace $\mathbf{C}^{\text{inpaint}}$ with $\mathbf{C}^{\text{enhance}}$ (i.e. enhanced colors).

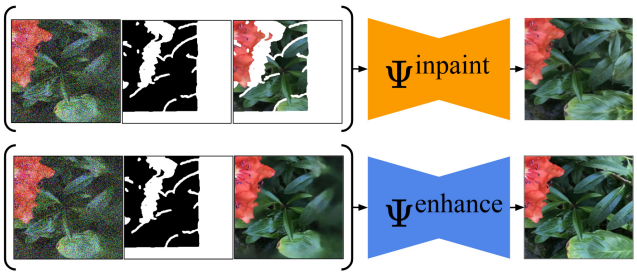

Figure 3. The input triplet of diffusion model consists of noisy-image, mask, and an guidance image. While masked pixels of guidance images of Ψ^{inpaint} are erased, they are preserved as the guidance for Ψ^{enhance}.

3.2. 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. 2, 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.

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

Masked Depth | Guidance Image | Completed Depth

Figure 4. 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.

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. 4). 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 [17] (see Suppl. for details).

4. Experiments

4.1. Experimental setup

LLFF Datasets: We utilize the LLFF dataset to demonstrate the effectiveness of our method. To assess performance in the task of view extrapolation, 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).

Metrics: For the FR metrics, we exclusively use LPIPS[16], KID [3], and also include PSNR and SSIM [13] for a comprehensive assessment.

Baselines: We compare our method with six related baselines for which code is available: (1) Sparf [12], one of the state-of-the-art (SOTA) methods for sparse view reconstruction. (2) FreeNeRF [15], another SOTA in sparse view reconstruction. (2) DiffusioNeRF [14], which employs a patch-wise diffusion model to provide RGB and depth supervision for a NeRF. (3) *SDS [8] 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
*SDS	20.56	0.654	0.338	0.0351	49.35
Ours	**20.76**	**0.688**	**0.269**	**0.0154**	**54.13**

4.2. LLFF

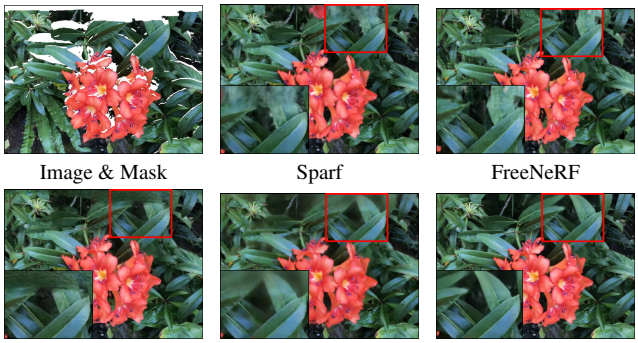

Image & Mask | Sparf | FreeNeRF

DiffusioNeRF | *SDS | Ours

Figure 5. **Qualitative results of view extrapolation.**

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 5 presents a qualitative comparison, highlighting the distinctions between our method and competing approaches.

While Sparf 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. Using *SDS loss only 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.

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. 2

[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. 2

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 4

[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. 4

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

[6] 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

[7] Julien Philip and Valentin Deschaintre. Floaters no more: Radiance field gradient scaling for improved near-camera training. 2023. 2

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

[9] 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

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

[11] 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

[12] 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, 4

[13] 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. 4

[14] 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. 4

[15] 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. 4

[16] 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. 4

[17] 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. 4