ViP-NeRF: Visibility Prior for Sparse Input Neural Radiance Fields

NAGABHUSHAN SOMRAJ, Indian Institute of Science, India
RAJIV SOUNDARARAJAN, Indian Institute of Science, India

Neural radiance fields (NeRF) have achieved impressive performances in view synthesis by encoding neural representations of a scene. However, NeRFs require hundreds of images per scene to synthesize photo-realistic novel views. Training them on sparse input views leads to overfitting and incorrect scene depth estimation resulting in artifacts in the rendered novel views. Sparse input NeRFs were recently regularized by providing dense depth estimated from pre-trained networks as supervision, to achieve improved performance over sparse depth constraints. However, we find that such depth priors may be inaccurate due to generalization issues. Instead, we hypothesize that the visibility of pixels in different input views can be more reliably estimated to provide dense supervision. In this regard, we compute a visibility prior through the use of plane sweep volumes, which does not require any pre-training. By regularizing the NeRF training with the visibility prior, we successfully train the NeRF with few input views. We reformulate the NeRF to also directly output the visibility of a 3D point from a given viewpoint to reduce the training time with the visibility constraint. On multiple datasets, our model outperforms the competing sparse input NeRF models including those that use learned priors. The source code for our model can be found on our project page: https://nagabhushansn95.github.io/publications/2023/ViP-NeRF.html.

CCS Concepts: • **Computing methodologies** → **Rendering**; **Volumetric models**; *Computer vision*; *Computational photography*; 3D imaging; Reconstruction.

Additional Key Words and Phrases: neural rendering, novel view synthesis, sparse input NeRF, visibility prior, plane sweep volumes

ACM Reference Format:
Nagabhushan Somraj and Rajiv Soundararajan. 2023. ViP-NeRF: Visibility Prior for Sparse Input Neural Radiance Fields. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings (SIGGRAPH '23 Conference Proceedings), August 6–10, 2023, Los Angeles, CA, USA.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3588432.3591539

1 INTRODUCTION

The goal of novel view synthesis is to synthesize a scene from novel viewpoints given RGB images of a few other viewpoints and their relative camera poses. By representing the scene implicitly using multi-layer perceptrons (MLP) and employing volume rendering, neural radiance fields (NeRF) [Barron et al. 2021, 2022; Liu et al. 2022; Mildenhall et al. 2020] have achieved impressive view synthesis performance. Such superior performance is usually achieved when a large number of views is input to train the NeRF. However, in multiple applications such as virtual or augmented reality, telepresence, robotics, and autonomous driving, very few input images may be available for training [Niemeyer et al. 2022]. In such settings, external sensors or a pre-calibrated fixed camera array may be employed to obtain accurate camera poses. Thus, there is a need to train NeRFs with few input views referred to as the sparse input NeRF problem.

The key challenge with sparse input images is that the volume rendering equations in NeRF are under-constrained, leading to solutions that overfit the input views. This results in uncertain and inaccurate depth in the learned representation. Synthesized novel views in such cases contain extreme distortions such as blur, ghosting, and floater artifacts [Niemeyer et al. 2022; Roessle et al. 2022]. Recent works have proposed different approaches to constrain the training of NeRF to output visually pleasing novel views. While a few recent works [Yang et al. 2022; Zhang et al. 2021; Zhou and Tulsiani 2022] focus on training NeRF models on a specific category of objects such as chairs or airplanes, we focus on training category agnostic sparse input NeRF models [Niemeyer et al. 2022]. Such prior work can be broadly classified into conditional NeRF models and other regularization approaches.

The conditional NeRF models employ a latent representation of the scene obtained by pre-training on a large dataset of different scenes [Chen et al. 2021; Hamdi et al. 2022; Johari et al. 2022; Wang et al. 2021; Yu et al. 2021] to condition the NeRF. The latent prior helps overcome the limitation on the number of views by enabling the NeRF model to effectively understand the scene. Such an approach is popular even when only a single image of the scene is available as input to the NeRF [Cai et al. 2022; Lin et al. 2023; Xu et al. 2022]. Different from the above, MetaNeRF [Tancik et al. 2021] learns the latent information as initial weights of the NeRF MLPs by employing meta-learning. However, the pre-trained latent prior could suffer from poor generalization on a given target scene [Niemeyer et al. 2022]. Thus, we believe that there is a need to study the sparse-input NeRF without conditioning the NeRF on latent representations.

On the other hand, regularization based approaches constrain the NeRF training with novel loss functions to yield better solutions. DS-NeRF [Deng et al. 2022] uses sparse depth provided by a structure from motion (SfM) model as additional supervision for the NeRF. To provide richer dense supervision, DDP-NeRF [Roessle et al. 2022] completes the sparse depth map using a pre-trained convolutional neural network (CNN). However, the requirement of pre-training on a large dataset of scenes is cumbersome and the dense depth prior may suffer from generalization errors. RegNeRF [Niemeyer et al. 2022] and InfoNeRF [Kim et al. 2022] impose constraints to promote depth smoothness and reduce depth uncertainty respectively. However, in our experiments, we observe that these methods are still inferior to DS-NeRF on popular datasets. This motivates the exploration of other reliable features for dense supervision to constrain the NeRF in addition to sparse depth supervision.

In our work, we explore the use of regularization in terms of visibility of any pixel from a pair of viewpoints. Here visibility of a pixel refers to whether the corresponding object is seen in both the viewpoints. For example, foreground objects are typically visible in multiple views whereas the background objects may be partially

SIGGRAPH '23 Conference Proceedings, August 6–10, 2023, Los Angeles, CA, USA

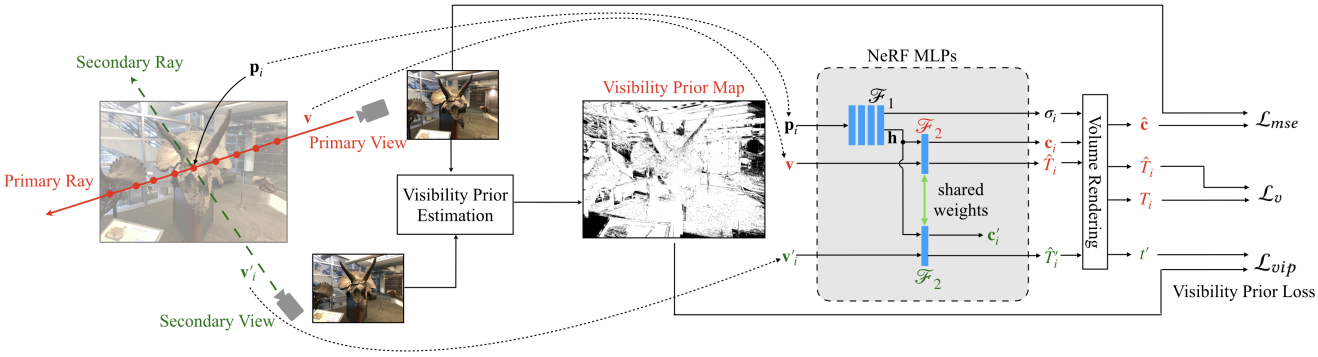


Fig. 1. Overview of ViP-NeRF architecture. Given the images from primary and secondary views, we estimate a visibility prior map in the primary view and use it to supervise the visibility of pixels as predicted by the NeRF. Specifically, we cast a ray through a randomly selected pixel in the primary view and sample 3D points along the ray. For every point $\mathbf{p}_i$, we use the NeRF MLPs to obtain its visibility in primary and secondary views, along with volume density σ_i and color $\mathbf{c}_i$. Volume rendering outputs visibility t' of the chosen pixel in the secondary view which is supervised by the visibility prior. $\mathcal{L}_v$ constrains the visibilities $\hat{T}_i$ output by network and T_i computed using volume rendering to be consistent with each other.

occluded. The visibility of a pixel in different views relies more on the relative depth of the scene objects than the absolute depth. We hypothesize that, given sparse input views, it may be easier to estimate the relative depth and visibility instead of the absolute depth. Thus, the key idea of our work is to regularize the NeRF with a dense visibility prior estimated using the given sparse input views. This allows the NeRF to learn better scene representation. We refer to our Visibility Prior regularized NeRF model as ViP-NeRF.

To obtain the visibility prior, we employ the plane sweep volumes (PSV) [Collins 1996] that have successfully been used in depth estimation [Gallup et al. 2007; Ha et al. 2016; Im et al. 2019; Yang and Pollefeys 2003] and view synthesis models [Zhou et al. 2018]. We create the PSV by warping one of the images to the view of the other at different depths (or planes) and compare them to obtain error maps. We determine a binary visibility map for each pixel based on the corresponding errors in the PSV. We regularize the NeRF training by using such a map as supervision for every pair of input views. We use the visibility prior in conjunction with the depth prior from DS-NeRF [Deng et al. 2022], where the former provides a dense prior on relative depth while the latter provides a sparse prior on absolute depth. Note that the estimation of our visibility prior does not require any pre-training on a large dataset.

Regularizing the NeRF with a dense visibility prior is computationally intensive and can lead to impractical training times. We reformulate the NeRF to directly and additionally output visibility to impose the regularization in a computationally efficient manner. We conduct experiments on two popular datasets to demonstrate the efficacy of the visibility prior for sparse input NeRF.

The main contributions of our work are as follows.

- We introduce visibility regularization to train the NeRF with sparse input views and refer to our model as ViP-NeRF.
- We estimate the dense visibility prior reliably using plane sweep volumes.
- We reformulate the NeRF MLP to output visibility thereby significantly reducing the training time.
- We achieve the state-of-the-art performance of sparse input NeRFs on multiple datasets.

2 RELATED WORK

Novel View Synthesis: Novel view synthesis methods typically use one or more input views to synthesize the scene from novel viewpoints. Recent pieces of work focus on obtaining volumetric 3D representations of the scene that can be computed once to render any viewpoint later. Zhou et al. [2018] propose multi-plane image (MPI) representations for view synthesis. Srinivasan et al. [2019] further extend this by infilling the occluded regions in the MPIs. Wiles et al. [2020] study an extreme case with a single input image and generate novel views by employing a monocular depth estimation network for scene reprojection. In contrast to the above explicit representations, neural radiance fields [Mildenhall et al. 2020] use an implicit representation through coordinate-based neural networks. Although NeRFs achieve excellent performance, they require dense input views for training. In this work, we focus on solving this problem, i.e. to train a NeRF given very few input views.

Sparse Input NeRF: Several recent works have studied sparse input NeRF by regularizing the NeRF with various priors. One of the early works, DietNeRF [Jain et al. 2021], hallucinates novel viewpoints during training and constrains the NeRF to generate novel views similar to the input images in the CLIP [Radford et al. 2021] representation space. DS-NeRF improves the performance by using fine-grained supervision at the pixel level using a sparse depth estimated by an SfM model [Deng et al. 2022]. DDP-NeRF further completes the sparse depth using a pre-trained network to obtain dense depth along with uncertainty estimates. Uncertainty modeling allows DDP-NeRF to relax the depth supervision at locations where the dense depth estimation is not confident. However, the completed depth may contain errors that may adversely affect the performance. DiffusioNeRF [Wynn and Turmukhambetov 2023] instead employs a pre-trained denoising diffusion model to regularize the distribution of RGB-D patches in novel viewpoints. In contrast, our work uses

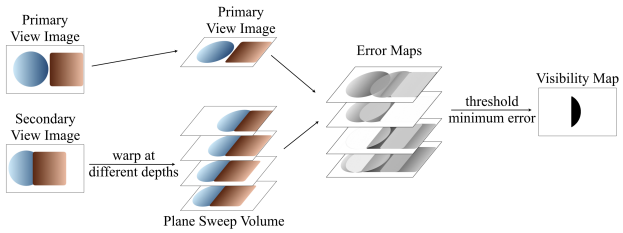


Fig. 2. A toy example to illustrate the computation of visibility prior. The scene contains a blue sphere and a brown box and the relative pose between the views is a translation in **x** direction. The secondary view image is warped to the primary view at different depth planes to create a PSV and compared with the primary view image to obtain error maps. We observe that the brown square and the blue circle are matched better in the second and third planes respectively leading to lower error (denoted as white) in the respective error maps. The minimum error across all the planes is thresholded to obtain the visibility prior map corresponding to the primary view image. The right portion of the sphere which is occluded in the secondary view image is denoted in black in the visibility map.

a more reliable visibility prior which can be estimated without the use of sophisticated CNNs and does not require pre-training on a large dataset of scenes.

Instead of depth estimates as priors, RegNeRF [Niemeyer et al. 2022] regularizes the NeRF using depth smoothness constraints on the rendered patches in the hallucinated viewpoints. Different from depth regularization models, InfoNeRF [Kim et al. 2022] tries to circumvent overfitting by encouraging concentration of volume density along a ray. In addition, it also minimizes the variation of volume density distributions along rays of two nearby viewpoints. Although these constraints are meaningful, our visibility prior imposes constraints across multiple views and can exploit the structure of the problem more effectively.

Single Image NeRF: Recently, there is increased interest in training NeRFs with a single input image [Lin et al. 2023; Xu et al. 2022]. A common thread in single image NeRF models is to use an encoder to obtain a latent representation of the input image. A NeRF based decoder conditioned on the representation, outputs volume density and color at given 3D points. For example, pix2NeRF [Cai et al. 2022] combines π-GAN [Chan et al. 2021] with NeRF to render photo-realistic images of objects or human faces. Gao et al. [2020] focus on human faces alone and use a more structured approach by exploiting facial geometry. MINE [Li et al. 2021] combines NeRF with MPI by replacing the MLP based implicit representation with an MPI based explicit representation in the decoder. Lin et al. [2023] obtain a richer latent representation by fusing global and local features obtained using a vision transformer and CNN respectively. Different from the above models, Wimbauer et al. [2023] use the MLP decoder to predict volume density alone and obtain the color by directly sampling from the given images. However, a common drawback of these models is the need for pre-training. Thus the performance may be inferior when testing on a generic scene.

3 NERF PRELIMINARIES

We first provide a brief introduction to NeRF and define the notations for subsequent use. A neural radiance field is an implicit representation of a scene using two multi-layer perceptrons (MLP). Given a set of images of a scene with corresponding camera poses, a pixel $\mathbf{q}$ is selected at random, and a ray $\mathbf{r}$ is passed from the camera center $\mathbf{o}$ through $\mathbf{q}$. Let $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N$ be N randomly sampled 3D points along $\mathbf{r}$. If $\mathbf{d}$ is the direction vector of $\mathbf{r}$ and z_i is the depth of a 3D point $\mathbf{p}_i$, $i \in \{1, 2, \ldots, N\}$, then $\mathbf{p}_i = \mathbf{o} + z_i\mathbf{d}$. An MLP $\mathcal{F}_1$ is trained to predict the volume density σ_i at $\mathbf{p}_i$ as

$$\sigma_i, \mathbf{h}_i = \mathcal{F}_1(\mathbf{p}_i), \tag{1}$$

where $\mathbf{h}_i$ is a latent representation. A second MLP $\mathcal{F}_2$ then predicts the color using $\mathbf{h}_i$ and the viewing direction $\mathbf{v} = \mathbf{d}/\|\mathbf{d}\|$ as

$$\mathbf{c}_i = \mathcal{F}_2(\mathbf{h}_i, \mathbf{v}). \tag{2}$$

Let the distance between two consecutive samples $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ be $\delta_i = z_{i+1} - z_i$. The visibility or transmittance of $\mathbf{p}_i$ is then given by

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \delta_j \sigma_j\right). \tag{3}$$

The weight or contribution of $\mathbf{p}_i$ in rendering the color $\hat{\mathbf{c}}$ of pixel $\mathbf{q}$ is computed as

$$w_i = T_i\,(1 - \exp(-\delta_i \sigma_i)) \tag{4}$$

to obtain

$$\hat{\mathbf{c}} = \sum_{i=1}^{N} w_i \mathbf{c}_i. \tag{5}$$

The MLPs are trained using mean squared error loss with the true color $\mathbf{c}$ of $\mathbf{q}$ as

$$\mathcal{L}_{mse} = \|\mathbf{c} - \hat{\mathbf{c}}\|^2. \tag{6}$$

4 METHOD

We illustrate the outline of our model in Fig. 1. The core idea of our work is that when only a few multiview images are available for NeRF training, the visibility of a pixel in different views can be more reliably densely estimated as compared to its absolute depth. In this regard, we introduce visibility regularization to train the NeRF with sparse input views in Sec. 4.1. To impose the visibility regularization, we obtain a binary visibility prior map for every pair of input training images, which we explain in Sec. 4.2. Finally, to reduce the training time, we design a method to efficiently predict the visibility of a given pixel in different views in Sec. 4.3. Sec. 4.4 summarizes the various loss functions used in training our model.

4.1 Visibility Regularization

Recall from Sec. 3 that NeRF trains MLPs by picking a random pixel $\mathbf{q}$ and predicting the color of $\mathbf{q}$ using the MLPs and volume rendering. Without loss of generality, we refer to the view corresponding to the ray $\mathbf{r}$ passing through $\mathbf{q}$ as the primary view and choose any other view as a secondary view. NeRF then samples N candidate 3D points, $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N$, along $\mathbf{r}$. Let T_i' be the visibility of $\mathbf{p}_i$ from the secondary view, computed similar to Eq. (3). We define the visibility of pixel $\mathbf{q}$ in the secondary view, $t'(\mathbf{q})$, as the weighted visibilities

Table 1. Quantitative results on RealEstate-10K dataset.

Model	learned prior	2 views			3 views			4 views		
		LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑
InfoNeRF		0.6796	0.4653	12.30	0.6979	0.4024	11.15	0.6745	0.4298	11.52
DietNeRF	✓	0.5730	0.6131	15.90	0.5365	0.6190	16.60	0.5337	0.6282	16.89
RegNeRF		0.5307	0.5709	16.14	0.4675	0.6096	17.38	0.4831	0.6068	17.46
DS-NeRF		0.4273	0.7223	21.40	0.3930	0.7554	23.73	0.3961	0.7575	24.24
DDP-NeRF	✓	0.2527	0.7890	21.44	0.2240	0.8223	23.10	0.2190	0.8270	24.17
ViP-NeRF		**0.1704**	**0.8087**	**24.48**	**0.1441**	**0.8505**	**27.21**	**0.1386**	**0.8588**	**28.13**

Table 2. Quantitative results on NeRF-LLFF dataset.

Model	learned prior	2 views			3 views			4 views		
		LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑
InfoNeRF		0.7561	0.2095	9.23	0.7679	0.1859	8.52	0.7701	0.2188	9.25
DietNeRF	✓	0.7265	0.3209	11.89	0.7254	0.3297	11.77	0.7396	0.3404	11.84
RegNeRF		0.4402	0.4872	16.90	0.3800	0.5600	18.62	**0.3446**	0.6056	19.83
DS-NeRF		0.4548	0.5068	17.06	0.4077	0.5686	**19.02**	0.3825	0.6016	**20.11**
DDP-NeRF	✓	0.4223	**0.5377**	**17.21**	0.4178	0.5610	17.90	0.3821	0.5999	19.19
ViP-NeRF		**0.4017**	0.5222	16.76	**0.3750**	**0.5837**	18.92	0.3593	**0.6085**	19.57

of all the candidate 3D points $\mathbf{p}_i$ analogous to Eq. (5) as

$$t'(\mathbf{q}) = \sum_{i=1}^{N} w_i T'_i \quad \in [0,1], \tag{7}$$

where w_i are obtained through Eq. (4). We omit the dependence of w_i and T'_i on $\mathbf{q}$ in the above equation for ease of reading. We obtain a prior $\tau'(\mathbf{q}) \in \{0,1\}$ on the visibility $t'(\mathbf{q})$ as described in Sec. 4.2. We constrain the visibility $t'(\mathbf{q})$ to match the prior $\tau'(\mathbf{q})$. However, we find that the prior may be unreliable at pixels where $\tau' = 0$, as we describe in Sec. 4.2. Hence, we do not impose any visibility loss on such pixels and formulate our visibility prior loss as

$$\mathcal{L}_{vip}(\mathbf{q}) = \max(\tau'(\mathbf{q}) - t'(\mathbf{q}), 0). \tag{8}$$

Note that our loss function constrains the NeRF across pairs of views, unlike previous works which regularize [Niemeyer et al. 2022; Roessle et al. 2022] in a given view alone. We believe that this leads to a better regularization for synthesizing novel views.

4.2 Visibility Prior

Given primary and secondary views, our goal is to estimate whether every pixel in the primary view is also visible in the secondary view through a binary visibility prior $\tau'(\mathbf{q})$. We employ plane sweep volumes to compute the visibility prior. We illustrate the computation of the visibility prior with a toy example in Fig. 2. Here, we warp the image in the secondary view to the primary view using the camera parameters at different depths varying between the near depth $z_{\min}$ and far depth $z_{\max}$. We sample D depths uniformly in inverse depth similar to StereoMag [Zhou et al. 2018]. The set of warped images is referred to as plane sweep volume (PSV) [Huang et al. 2018].

Let $I^{(1)}$ be the image in the primary view and $I_k^{(2)}$ be the set of D warped images, where $k \in \{0, 1, \ldots, D-1\}$ denotes the plane index. We then compute the error map E_k of the warped secondary image with the primary image at each plane k of the PSV as

$$E_k = \|I^{(1)} - I_k^{(2)}\|_1, \tag{9}$$

where the norm is computed across the color channels. We determine the visibility prior τ' for pixel $\mathbf{q}$ by thresholding the minimum error across all the planes as

$$\begin{aligned} e(\mathbf{q}) &= \min_k E_k(\mathbf{q}), \\ \tau'(\mathbf{q}) &= \mathbb{1}_{\{\exp(-e(\mathbf{q})/\gamma) > 0.5\}}, \end{aligned} \tag{10}$$

where γ is a hyper-parameter.

Intuitively, for a given pixel $\mathbf{q}$, a lower error in any of the planes indicates the presence of a matching pixel in the secondary view, i.e. $\mathbf{q}$ is visible in the secondary view. Note that this holds true when the intensity of pixels does not change significantly across views, which is typical for most of the objects in real-world scenes [Li et al. 2021]. Consequently, the absence of a matching point across all the planes may indicate that $\mathbf{q}$ is not visible in the secondary view or $\mathbf{q}$ belongs to a highly specular object whose color varies significantly across different viewpoints. Thus, our prior is used to regularize the NeRF only in the first case above i.e. the pixels for which we find a match. Following the above procedure, we obtain the visibility prior for every pair of images obtained from the training set, by treating either image in the pair as the primary or the secondary view.

4.3 Efficient Prediction of Visibility

Recall that imposing $\mathcal{L}_{vip}$ in Eq. (8) requires computing visibility T'_i in the secondary view for every $\mathbf{p}_i$. A naive approach to compute

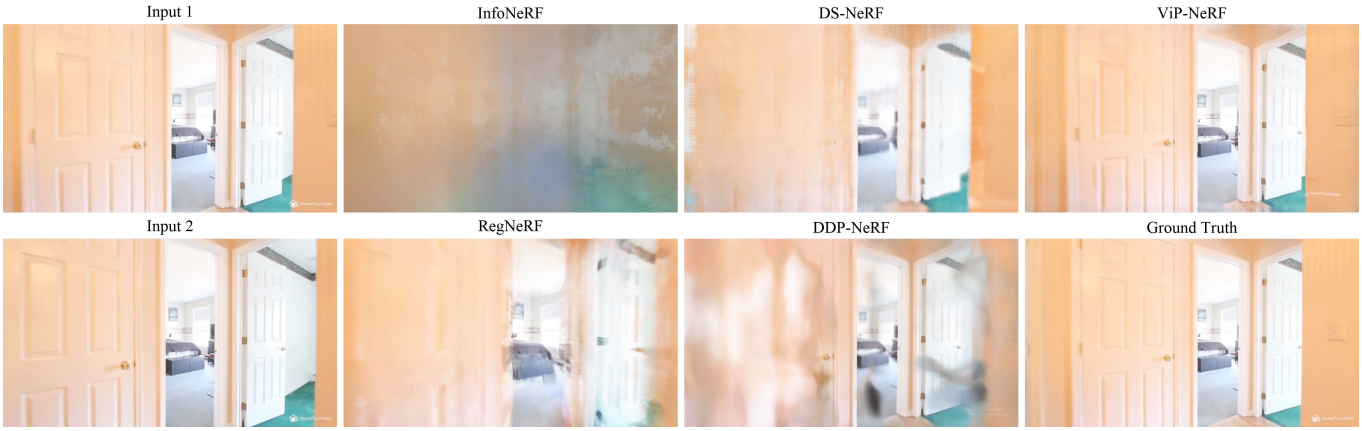


Fig. 3. Qualitative examples on RealEstate-10K dataset with two input views. We observe that the predictions of ViP-NeRF are close to the ground truth, while those of other models suffer from various distortions. In particular, DDP-NeRF blurs regions of the frame near the left door and contains black floater artifacts.

T_i' involves sampling up to N points along a secondary ray from the secondary view camera origin to $\mathbf{p}_i$ and querying the NeRF MLP $\mathcal{F}_1$ for each of these points. Thus, obtaining $t'(\mathbf{q})$ in Eq. (7) requires upto N^2 MLP queries, which increases the training time making it computationally prohibitive. We overcome this limitation by reformulating the NeRF MLP $\mathcal{F}_2$ to also output a view-dependent visibility of a given 3D point as,

$$\mathbf{c}_i, \hat{T}_i = \mathcal{F}_2(\mathbf{h}_i, \mathbf{v}); \quad \mathbf{c}_i', \hat{T}_i' = \mathcal{F}_2(\mathbf{h}_i, \mathbf{v}_i'), \tag{11}$$

where $\mathbf{v}_i'$ is the viewing direction of the secondary ray. We use the MLP output $\hat{T}_i'$ instead of T_i' in Eq. (7).

Note that to output $\hat{T}_i'$, we need not query $\mathcal{F}_1$ again and can reuse $\mathbf{h}_i$ obtained from Eq. (1). We only need to query $\mathcal{F}_2$ additionally and since $\mathcal{F}_2$ is a single layer MLP and significantly smaller than $\mathcal{F}_1$, the additional computational burden is negligible. Thus, directly obtaining the secondary visibility $\hat{T}_i'$ of $\mathbf{p}_i$ through Eq. (11) allows us to compute $t'(\mathbf{q})$ in Eq. (7) using only N queries of the MLP $\mathcal{F}_1$, as opposed to N^2 queries in the naive approach.

However, the use of $\hat{T}_i'$ in place of T_i' regularizes the NeRF training only if the two quantities are close to each other. Thus, we introduce an additional loss to constrain the visibility $\hat{T}_i$ output by $\mathcal{F}_2$ to be consistent with the visibility T_i computed using Eq. (3) as

$$\mathcal{L}_v = \sum_{i=1}^{N} \left(\left(\mathrm{SG}(T_i) - \hat{T}_i \right)^2 + \left(T_i - \mathrm{SG}(\hat{T}_i) \right)^2 \right), \tag{12}$$

where $\mathrm{SG}(\cdot)$ denotes the stop-gradient operation. The first term in the above loss function uses T_i as a target and brings $\hat{T}_i$ closer to it. On the other hand, since $\hat{T}_i$ gets additionally updated directly based on the visibility prior, the second term helps transfer such updates to $\mathcal{F}_1$ more efficiently than backpropagation through $\mathcal{F}_2$.

4.4 Overall Loss

Similar to DS-NeRF [Deng et al. 2022], we also use the sparse depth given by an SfM model to supervise the NeRF as

$$\mathcal{L}_{sd} = \|z - \hat{z}\|^2, \tag{13}$$

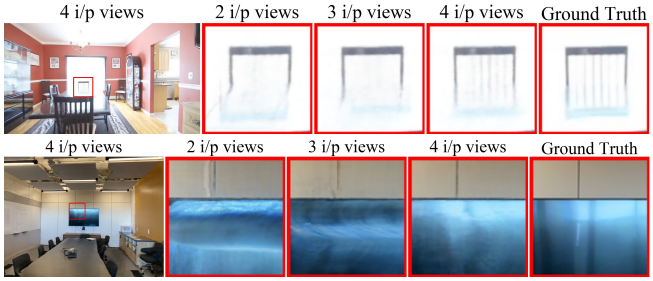


Fig. 4. Qualitative examples on RealEstate-10K and NeRF-LLFF dataset with two, three, and four input views. We observe that ViP-NeRF models specular regions better as the number of input views increases. For example, in the first row, the reflection of the chair is better reconstructed as the number of views increases.

where z is the depth provided by the SfM model, $\hat{z} = \sum_i w_i z_i$ is the depth estimated by NeRF and w_i are obtained in Eq. (4). Our overall loss for ViP-NeRF is a linear combination of the losses obtained in Eq. (6), Eq. (8), Eq. (12) and Eq. (13) as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{mse} + \lambda_2 \mathcal{L}_{sd} + \lambda_3 \mathcal{L}_{vip} + \lambda_4 \mathcal{L}_v, \tag{14}$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are hyper-parameters. We note that $\mathcal{L}_{vip}$ is always employed in conjunction with $\mathcal{L}_v$ to make the learning computationally tractable.

5 EXPERIMENTS

5.1 Evaluation Setup

We conduct experiments on two different datasets, namely RealEstate-10K and NeRF-LLFF. We evaluate all the models in the more challenging setup of 2, 3, or 4 input views, unlike prior work which use 9–18 input views [Jain et al. 2021; Roessle et al. 2022]. The test set is retained to be the same across all different settings for both datasets.

RealEstate-10K [Zhou et al. 2018] dataset is commonly used to evaluate view synthesis models [Han et al. 2022; Tucker and

Table 3. Comparison of reliability of priors used in different models. The reference visibility is obtained using NeRF trained with dense input views.

	RealEstate-10K			NeRF-LLFF		
model	**Prec.** ↑	**Rec.** ↑	**F1** ↑	**Prec.** ↑	**Rec.** ↑	**F1** ↑
ViP-NeRF	0.97	**0.83**	**0.89**	0.82	**0.85**	**0.83**
DDP-NeRF	**0.98**	0.53	0.66	**0.86**	0.33	0.47

Table 4. Evaluation of depth estimated by different models with two input views. The reference depth is obtained using NeRF trained with dense input views. The depth RMSE on the two datasets are of different orders on account of different depth ranges.

	RealEstate-10K		NeRF-LLFF	
model	**RMSE** ↓	**SROCC** ↑	**RMSE** ↓	**SROCC** ↑
ViP-NeRF	**1.6411**	**0.7702**	**45.6314**	**0.6184**
DDP-NeRF	1.7211	0.7544	46.6268	0.6136

Snavely 2020] and contains videos of camera motion, both indoor and outdoor. The dataset also provides the camera intrinsics and extrinsics for all the frames. For our experiments, we choose 5 scenes from the test set, each containing 50 frames with a spatial resolution of 1024×576. In each scene, we reserve every 10^{th} frame for training and use the remaining 45 frames for testing. Please refer to the supplementary for more details on the choice of scenes.

NeRF-LLFF [Mildenhall et al. 2019] dataset is used to evaluate the performance of various NeRF Models including sparse input NeRF models. It consists of 8 forward-facing scenes with a variable number of frames per scene at a spatial resolution of 1008×756. Following RegNeRF [Niemeyer et al. 2022], we use every 8^{th} frame for testing. For training, we pick 2, 3 or 4 frames uniformly among the remaining frames following RegNeRF [Niemeyer et al. 2022].

Evaluation measures: We quantitatively evaluate the methods using LPIPS [Zhang et al. 2018], structural similarity (SSIM) [Wang et al. 2004], and peak signal to noise ratio (PSNR) measures. For LPIPS, we use the v0.1 release with the AlexNet [Krizhevsky et al. 2012] backbone as suggested by the authors.

5.2 Comparisons and Implementation Details

We compare the performance of our model with other sparse input NeRF models such as DDP-NeRF [Roessle et al. 2022] and DietNeRF [Jain et al. 2021] which use learned priors to constrain the NeRF training. We also compare with DS-NeRF [Deng et al. 2022], InfoNeRF [Kim et al. 2022], and RegNeRF [Niemeyer et al. 2022] that do not use learned priors. We train the models for 50k iterations on both datasets using the code provided by the respective authors.

For ViP-NeRF, we use Adam optimizer with a learning rate of 5e-4 that exponentially decays to 5e-6 following NeRF [Mildenhall et al. 2020]. We set the loss weights such that the magnitudes of all the losses are of similar order after scaling. Specifically, we set $\lambda_1 = 1, \lambda_2 = 0.1, \lambda_3 = 0.001$ and $\lambda_4 = 0.1$. For visibility prior estimation, we set $D = 64$ and $\gamma = 10$. Since we require $\hat{T}_i'$ to be close to T_i' while using $\hat{T}_i'$ to compute $\mathcal{L}_{vip}$, we impose $\mathcal{L}_{vip}$ after 20k iterations. We train our models on a single NVIDIA RTX A4000 16GB GPU.

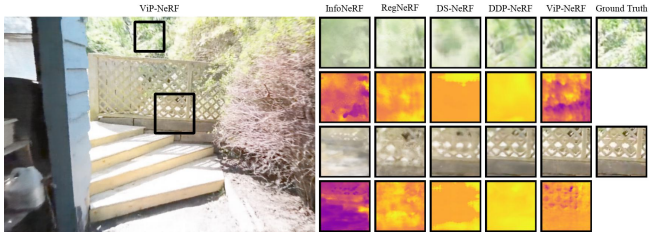


Fig. 5. Estimated depth map on RealEstate-10K dataset with two input views. We find that ViP-NeRF is better in both frame synthesis and depth estimation compared to the competing models. For example, in the first row, the depth estimated by DDP-NeRF is smooth which may be leading to a loss of sharpness in synthesizing the shrubs. In contrast, ViP-NeRF predictions are sharper. For better visualization, we show inverse depth and normalize it to set the maximum value to unity.

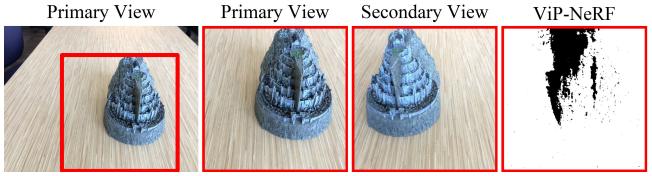


Fig. 6. Visualization of the visibility map predicted by ViP-NeRF. White indicates the regions of the 'Primary View' which are visible in the 'Secondary View' and black indicates the occluded regions. From the primary and secondary views, we observe that the left part of the fortress and the neighboring portion of the wood are hidden in the secondary view. ViP-NeRF is able to reasonably determine the visible and occluded regions.

Table 5. Ablation experiments on both the datasets with two input views.

	RealEstate-10K		NeRF-LLFF	
model	**LPIPS** ↓	**SSIM** ↑	**LPIPS** ↓	**SSIM** ↑
ViP-NeRF	**0.1704**	**0.8087**	**0.4017**	**0.5222**
w/o sparse depth	0.2754	0.7588	0.5056	0.4631
w/o dense visibility	0.4273	0.7223	0.4548	0.5068

5.3 Results

We show the quantitative performance of ViP-NeRF and other competing models on RealEstate-10K and NeRF-LLFF datasets in Tabs. 1 and 2. Our model outperforms all the competing models, particularly in terms of the perceptual metric, LPIPS. ViP-NeRF even outperforms models such as DDP-NeRF and DietNeRF which involve pre-training on a large dataset. Fig. 3 shows qualitative comparisons on a scene from the RealEstate-10K dataset, where we observe significantly better synthesis by our model as compared to the competing models. We show more qualitative comparisons in Figs. 7 to 10 in the figure only pages at the end of this manuscript. In these samples, we find that ViP-NeRF removes most of the floater artifacts and successfully retains the shapes of objects.

In Fig. 4, we qualitatively compare the predictions of our model with different numbers of input views. We observe that ViP-NeRF estimates the geometry reasonably well with even two input views. However, with more input views, the performance of ViP-NeRF

improves in reflective or specular regions. Fig. 6 visualizes the visibility map predicted by ViP-NeRF, where we observe that it is able to accurately predict the regions in the primary image which are visible and occluded in the secondary image.

Dense depth vs dense visibility: The key idea of our paper is that it may be possible to reliably estimate dense visibility than dense depth. From Tab. 1, we find that ViP-NeRF outperforms DDP-NeRF consistently, which indicates that the dense visibility prior we compute without any pre-training is superior to the learned dense depth prior used by DDP-NeRF. Further from Tab. 2, we observe that ViP-NeRF consistently improves over DS-NeRF in terms of LPIPS and SSIM, whereas DDP-NeRF does not. This may be due to the domain shift between the training dataset of DDP-NeRF and the LLFF dataset, resulting in no performance improvement over DS-NeRF. Thus, we conclude that augmenting sparse depth with dense visibility leads to better view synthesis performance than dense completion of the sparse depth. We further validate this conclusion by comparing the two priors in the following.

Validating priors: We compare the reliability of the dense visibility prior used in our model against the dense depth prior from DDP-NeRF. For this comparison, we convert the dense depth to visibility and compare it with the visibility prior of our approach. Specifically, we warp the image in the secondary view to the primary view using the dense depth prior and compute the visibility map similar to Eq. (10). We compare the visibility maps obtained using dense depth and our approach with the visibility map predicted by a NeRF model trained with dense input views. We evaluate the visibility maps in terms of precision, recall, and F1 score.

From Tab. 3, we observe that our approach significantly outperforms DDP-NeRF prior in terms of the recall and F1 score, while performing similarly in terms of precision. A high precision of our prior indicates that it makes very few mistakes when imposing $\mathcal{L}_{vip}$. On the other hand, a high recall shows that our prior is able to capture most of the visible regions where $\mathcal{L}_{vip}$ needs to be imposed. On the contrary, a low recall for the DDP-NeRF prior indicates that large regions that are actually visible in the secondary view are marked as occluded by the dense depth prior. Consequently, this indicates the presence of a large number of pixels with inaccurate depth in the prior of DDP-NeRF. Thus, we conclude that our visibility prior is more reliable than the dense depth prior from DDP-NeRF for training the NeRF.

As discussed in Sec. 1, visibility is related to relative depth, and thus a prior on visibility only constrains the relative depth ordering of the objects. On the other hand, the dense depth prior constrains the absolute depth, perhaps incorrectly. Thus the visibility prior provides more freedom to the NeRF in reconstructing the 3D geometry and is also more reliable compared to the depth prior. This may explain the superior performance of visibility regularization over dense depth regularization.

Evaluation of estimated depth: It is believed that better performance in synthesizing novel views is directly correlated with the accuracy of depth estimation [Deng et al. 2022]. Thus, we compare our model with DDP-NeRF on their ability to estimate absolute depth correctly using root mean squared error (RMSE). We also evaluate the models on their ability to estimate the relative depth of the scene correctly using spearman rank-order correlation coefficient (SROCC) [Corder and Foreman 2014], which computes the linear correlation between ranks of the estimated pixel depths with that of the ground truth depth. Due to the unavailability of ground truth depth on both the datasets, we train a NeRF model with dense input views and use its predicted depth as a pseudo ground truth. From Tab. 4, we observe that our model consistently outperforms DDP-NeRF both in terms of absolute and relative depth. Fig. 5 shows that the depth estimated by DDP-NeRF is smooth in textured regions, which may be leading to blur in the synthesized frame. In contrast, the dense visibility prior used in our model allows NeRF to predict sharp depth in such regions leading to sharper frame predictions.

Ablations: We analyze the contributions of dense visibility and sparse depth priors in ViP-NeRF, by disabling them one at a time. From Tab. 5 and Fig. 11a, we find that removing either priors leads to a drop in performance on both the datasets. This suggests that the dense visibility prior may be providing information that is complementary to the sparse depth prior. For a more fine-grained analysis, we compare the LPIPS scores on individual scenes in Fig. 11b. We observe that the addition of dense visibility prior over sparse depth prior leads to an improvement in the performance on all the scenes. Further, we find that our model with dense visibility prior alone is able to achieve impressive performance, especially on the RealEstate-10K dataset.

5.4 Limitations and Future Work

Our visibility prior constrains only the regions visible in at least two of the input views. As a result, we observe inaccurate depth estimation in the regions that are visible in only one of the input images. However, such regions account for a very small portion of the scene and reduce further with three or four input views. Further, ViP-NeRF may fail to synthesize disoccluded regions that can occur in sparse-input view synthesis, similar to RegNeRF. It would be interesting to explore the use of generative NeRF models such as pix2NeRF [Cai et al. 2022] to synthesize such disocclusions.

Our approach to estimating the visibility prior may not account for significant color changes that can occur when the scene contains highly specular surfaces. We do not impose any loss on such pixels. It would be interesting to analyze if pre-training a network on a large dataset to estimate visibility can provide more supervision in specular regions. Moreover, it would be interesting to see if pre-training a network to predict dense visibility generalizes better when compared to depth completion. Also, we observe in Tab. 2 that adding a new view leads to a significant improvement in performance as compared to adding new regularizations. Thus, one could explore hallucinating new views using generative models and use the hallucinated views for additional supervision.

6 CONCLUSION

We study the problem of training NeRFs in sparse input scenarios, where the NeRF tends to overfit the input views and learn incorrect geometry. We propose a prior on the visibility of pixels in other viewpoints to regularize the training and mitigate such errors. The visibility prior obtained using a plane sweep volume is more reliable as compared to the depth prior estimated using pre-trained networks. We reformulate the NeRF MLPs to additionally output visibility to

compute the visibility prior loss in a time-efficient manner. ViP-NeRF achieves state-of-the-art performance on two commonly used datasets for novel view synthesis.

ACKNOWLEDGMENTS

This work was supported in part by a grant from Qualcomm. The first author was supported by the Prime Minister's Research Fellowship awarded by the Ministry of Education, Government of India. The authors would also like to thank Suhas Srinath and Nithin Babu for the valuable discussions.

REFERENCES

Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. 2016. Large-Scale Data for Multiple-View Stereopsis. *International Journal of Computer Vision (IJCV)* (2016), 1–16.

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. 2022. Pix2NeRF: Unsupervised Conditional p-GAN for Single Image to Neural Radiance Fields Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. 2021. Pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. 2021. MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo. *arXiv e-prints*, Article arXiv:2103.15595 (March 2021), arXiv:2103.15595 pages. arXiv:2103.15595

R.T. Collins. 1996. A Space-Sweep Approach to True Multi-Image Matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.1996.517097

Gregory W Corder and Dale I Foreman. 2014. *Nonparametric statistics: A step-by-step approach.*

Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. 2022. Depth-Supervised NeRF: Fewer Views and Faster Training for Free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. 2007. Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2007.383245

Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. 2020. Portrait Neural Radiance Fields from a Single Image. *arXiv e-prints*, Article arXiv:2012.05903 (2020), arXiv:2012.05903 pages. arXiv:2012.05903

Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. 2016. High-Quality Depth From Uncalibrated Small Motion Clip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Abdullah Hamdi, Bernard Ghanem, and Matthias Nießner. 2022. SPARF: Large-Scale Learning of 3D Sparse Radiance Fields from Few Input Images. *arXiv e-prints*, Article arXiv:2212.09100 (2022), arXiv:2212.09100 pages. arXiv:2212.09100

Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. 2022. Single-View View Synthesis in the Wild with Learned Adaptive Multiplane Images. In *Proceedings of the ACM SIGGRAPH*.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. 2018. DeepMVS: Learning Multi-View Stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. 2019. DPSNet: End-to-end Deep Plane Sweep Stereo. *arXiv e-prints*, Article arXiv:1905.00538 (2019), arXiv:1905.00538 pages. arXiv:1905.00538

Ajay Jain, Matthew Tancik, and Pieter Abbeel. 2021. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. 2022. GeoNeRF: Generalizing NeRF With Geometry Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mijeong Kim, Seonguk Seo, and Bohyung Han. 2022. InfoNeRF: Ray Entropy Minimization for Few-Shot Neural Volume Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. 2021. MINE: Towards Continuous Depth MPI With NeRF for Novel View Synthesis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Kai-En Lin, Yen-Chen Lin, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. 2023. Vision Transformer for NeRF-Based View Synthesis From a Single Input Image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. 2022. Neural Rays for Occlusion-Aware Image-Based Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)* 38, 4 (July 2019), 1–14. https://doi.org/10.1145/3306346.3322980

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. 2022. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis From Sparse Inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. 2022. Dense Depth Priors for Neural Radiance Fields From Sparse Input Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the Boundaries of View Extrapolation With Multiplane Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. 2021. Learned Initializations for Optimizing Coordinate-Based Neural Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Richard Tucker and Noah Snavely. 2020. Single-View View Synthesis With Multiplane Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. 2021. IBRNet: Learning Multi-View Image-Based Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)* 13, 4 (2004), 600–612.

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. SynSin: End-to-End View Synthesis From a Single Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. 2023. Behind the Scenes: Density Fields for Single View Reconstruction. *arXiv e-prints*, Article arXiv:2301.07668 (2023), arXiv:2301.07668 pages. arXiv:2301.07668

Jamie Wynn and Daniyar Turmukhambetov. 2023. DiffusioNeRF: Regularizing Neural Radiance Fields with Denoising Diffusion Models. *arXiv e-prints*, Article arXiv:2302.12231 (2023), arXiv:2302.12231 pages. arXiv:2302.12231

Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. 2022. SinNeRF: Training Neural Radiance Fields on Complex Scenes from a Single Image. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ruigang Yang and M. Pollefeys. 2003. Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2003.1211356

Zhenpei Yang, Zhile Ren, Miguel Angel Bautista, Zaiwei Zhang, Qi Shan, and Qixing Huang. 2022. FvOR: Robust Joint Shape and Pose Optimization for Few-View Object Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelNeRF: Neural Radiance Fields From One or Few Images. In *Proceedings of the IEEE Conference on*

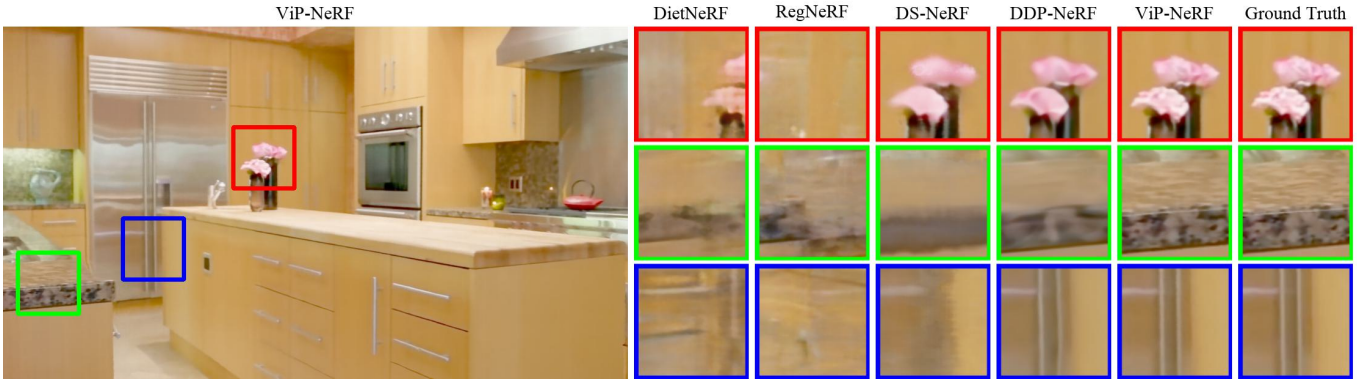


Fig. 7. Qualitative examples on RealEstate-10K dataset with two input views. We observe sharp predictions by ViP-NeRF while predictions by other models suffer from blur and other artifacts. In particular, DDP-NeRF predictions contain blurred flowers (first row) and blurred tiles (second row).

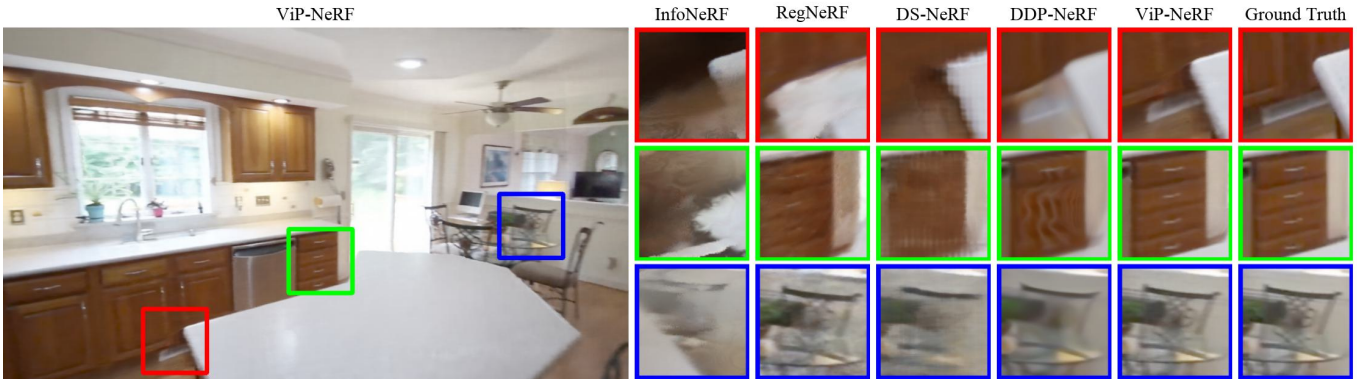


Fig. 8. Qualitative examples on RealEstate-10K dataset with three input views. We find that ViP-NeRF is able to reconstruct novel views significantly better than the competing models. DDP-NeRF extends parts of the white table and fails to reconstruct the drawer handles accurately in the first and second examples. In the third example, DDP-NeRF fails to reconstruct thin objects in the chair.

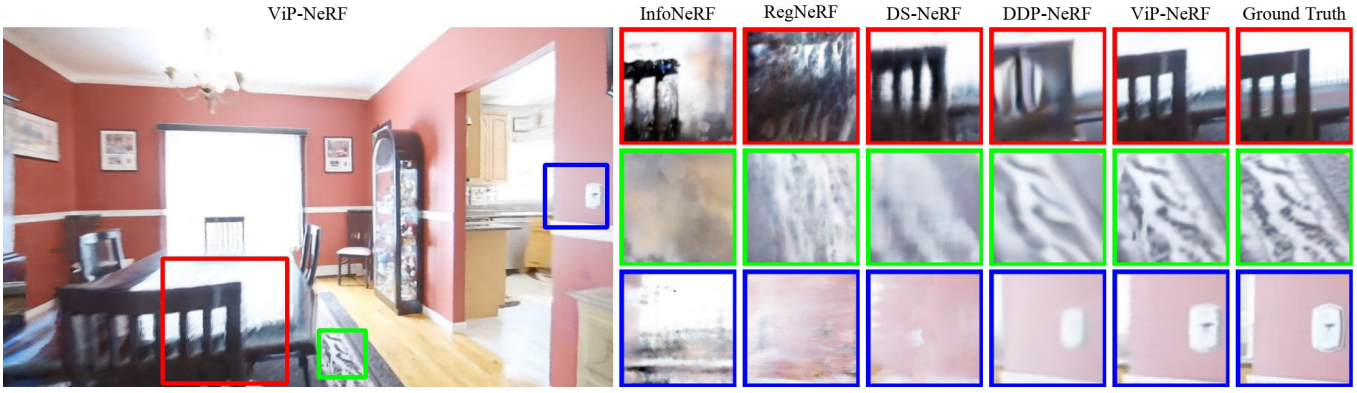


Fig. 9. Qualitative examples on RealEstate-10K dataset with four input views. In the first example, DDP-NeRF fails to retain the structure of the chair while it blurs the texture of the carpet in the second example. We observe even more severe distortions among the predictions of other models.

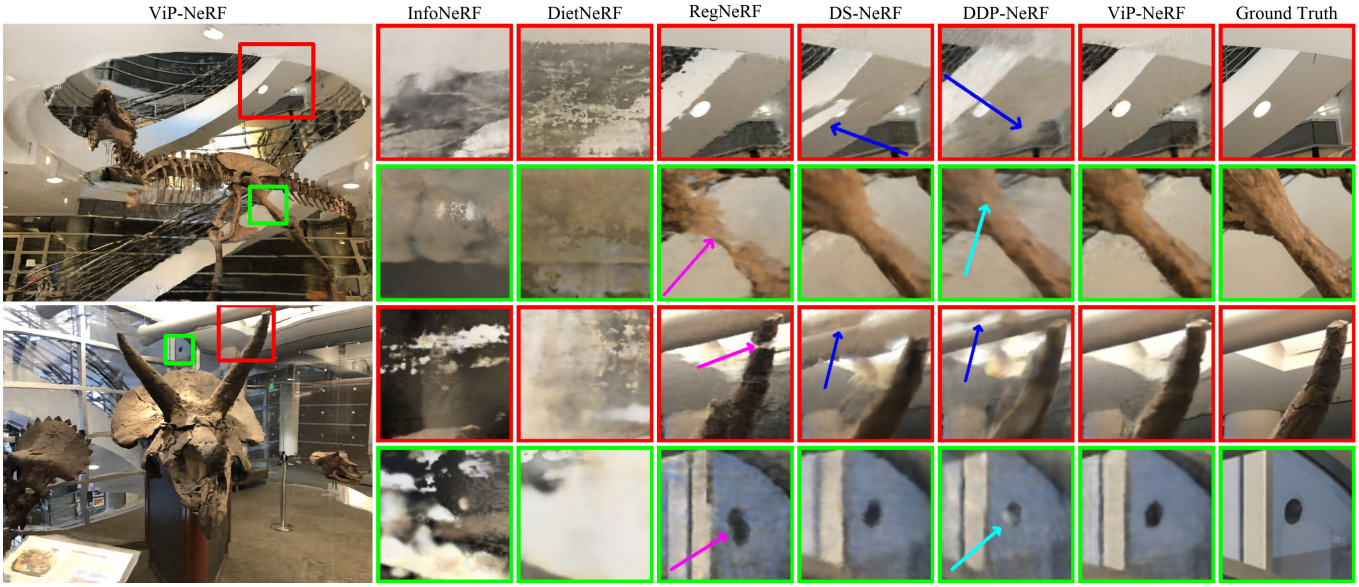
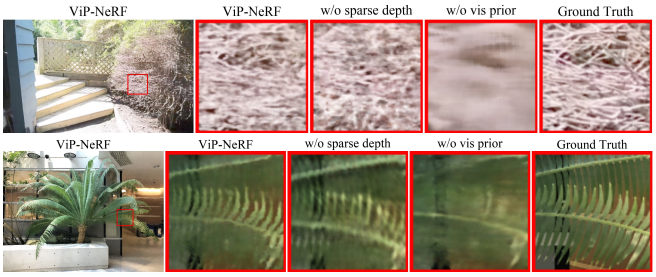
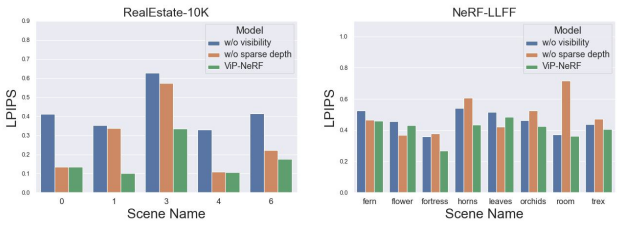


Fig. 10. Qualitative examples on NeRF-LLFF dataset with two input views. In the first and third examples, we observe floater artifacts (blue arrows) in the predictions of DS-NeRF and DDP-NeRF, which are mitigated in the predictions of ViP-NeRF. We find that RegNeRF fails to capture thin t-rex bone in the second example and breaks the horn into two pieces in the third example (magenta arrows). Cyan arrows indicate color changes in the predictions of DDP-NeRF in the second and fourth examples. We note that predictions by our model do not suffer from the above described artifacts.



(a) Qualitative examples for ablations on RealEstate-10K and NeRF-LLFF dataset. We observe that the absence of dense visibility prior leads to significant blur in the predicted frames. While the reconstruction is reasonable without the sparse depth prior, we obtain the best reconstructions when using both the priors.



(b) Scene-wise LPIPS scores of ViP-NeRF and the ablated models. Note that lower LPIPS scores are better. ViP-NeRF performs better than both the ablated models in most cases leading to overall better performance.

Fig. 11. Qualitative and quantitative comparisons of ablated models on both RealEstate-10K and NeRF-LLFF datasets.

Computer Vision and Pattern Recognition (CVPR).

Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. 2021. NeRS: Neural Reflectance Surfaces for Sparse-view 3D Reconstruction in the Wild. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS).*

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo Magnification: Learning View Synthesis Using Multiplane Images. *ACM Transactions on Graphics (TOG)* 37, 4 (July 2018).

Zhizhuo Zhou and Shubham Tulsiani. 2022. SparseFusion: Distilling View-conditioned Diffusion for 3D Reconstruction. *arXiv e-prints*, Article arXiv:2212.00792 (2022), arXiv:2212.00792 pages. arXiv:2212.00792

Supplement

Table 6. Original name of RealEstate-10K videos we selected for experiments and their updated names.

Original Name	Updated Name	Timestamp
000c3ab189999a83	0	53453400
000db54a47bd43fe	1	227894333
0017ce4c6a39d122	3	46012000
002ae53df0e0afe2	4	61144000
0043978734eec081	6	54387667

The contents of this supplement include

A. Details of RealEstate-10K dataset.
B. Implementation details of the competing sparse input NeRF models.
C. Comparisons on DTU dataset.
D. Video examples on RealEstate-10K and NeRF-LLFF datasets.
E. Additional comparisons between ViP-NeRF and DDP-NeRF.
F. Additional analysis.

A DATASET DETAILS

RealEstate-10K [Zhou et al. 2018] is a large database consisting of about 80,000 video segments, each containing more than 30 frames. This dataset was proposed to train traditional deep learning models which require training on a large number of videos and hence the dataset is further divided into train and test splits. Since NeRF based models optimize the networks on individual scenes, we select five videos from the test set to evaluate the NeRF models. For easy reference, we rename the videos as scene numbers starting from zero. The dataset provides only links to the videos on YouTube and hence we discard the videos which are no longer available. Further, we also discard the videos which are less than 50 frames in length. We then select the first five videos and choose a random segment of 50 frames within the videos. In Tab. 6, we provide the mapping between the original name of the videos we selected and the updated names, along with the timestamp of the first frame in the video segment. Please refer to Zhou et al. [2018] for more details on obtaining the data.

In each scene, we reserve 5 frames for training and use the remaining 45 frames for testing. Specifically, frames 10, 20, 30, 0 and 40 are reserved for training. For training with n views, we choose the first n frames from the above list.

B IMPLEMENTATION DETAILS

We use the official code releases by the respective authors and train the models on both RealEstate-10K and NeRF-LLFF datasets [Mildenhall et al. 2019]. In the following, we provide details of any changes we make on top of the respective code releases.

InfoNeRF [Kim et al. 2022]: The code release for InfoNeRF uses test viewpoints during training. For a fair comparison, we replace the test poses with poses interpolated from train poses.

RegNeRF [Niemeyer et al. 2022]: We found an inconsistency between the description in the paper and the implementation (possibly a bug) in RegNeRF code, where they use only a single hallucinated view instead of multiple. We fix this and train the RegNeRF model.

Table 7. Quantitative results on DTU dataset. RegNeRF+ uses test camera poses during training.

Model	**learnt prior**	**LPIPS ↓**	**SSIM ↑**	**PSNR ↑**
InfoNeRF		0.6649	0.2659	8.67
DietNeRF	✓	0.7686	0.2790	7.36
RegNeRF		0.7808	0.2327	7.25
RegNeRF+		0.4378	0.5310	12.73
DS-NeRF		0.5136	0.4841	11.99
DDP-NeRF	✓	0.5542	0.4544	11.40
ViP-NeRF		**0.4876**	**0.5057**	**12.04**

C COMPARISONS ON DTU DATASET

DTU [Aanæs et al. 2016] is a commonly used benchmark dataset for conditional NeRF models [Chen et al. 2021; Yu et al. 2021]. Nonetheless, a few sparse input NeRF models benchmark on the DTU dataset as well. We use the train and test sets defined by pixelNeRF [Yu et al. 2021]. Specifically, there are 15 scenes and around 49 frames per scene of which 40 frames are used for testing, and 9 frames are reserved for training.

We found that RegNeRF uses the test viewpoints as the hallucinated viewpoints for training on DTU scenes. However, this would be an unfair comparison since other models including ours do not use test camera poses during training. Hence we remove such viewpoints and train the RegNeRF model. Nonetheless, we report the performance of RegNeRF that uses test views during training as 'RegNeRF+' in Tab. 7.

We show quantitative comparisons between the competing models in Tab. 7 and qualitative comparisons in Fig. 12. ViP-NeRF outperforms all the competing models that do not use test camera poses during training, including DDP-NeRF and DietNeRF which employ pre-training. Even qualitatively, we find that the predictions of ViP-NeRF are closer to the ground truth without many artifacts seen in the predictions of the other models.

D VIDEO COMPARISONS

Along with this supplementary, we attach a few videos to compare ViP-NeRF with the competing models such as DS-NeRF [Deng et al. 2022], DDP-NeRF [Roessle et al. 2022] and RegNeRF [Niemeyer et al. 2022]. Kindly use the attached 'VideoSamples.html' file to view all the videos in a single browser window.

E ADDITIONAL COMPARISONS

Here, we show more comparisons with the second-best-performing model, DDP-NeRF, on both datasets. Specifically, out of 5 scenes from RealEstate-10K, Figs. 3 and 7 in the main paper show the comparisons on two of the scenes. Fig. 13 shows qualitative comparisons

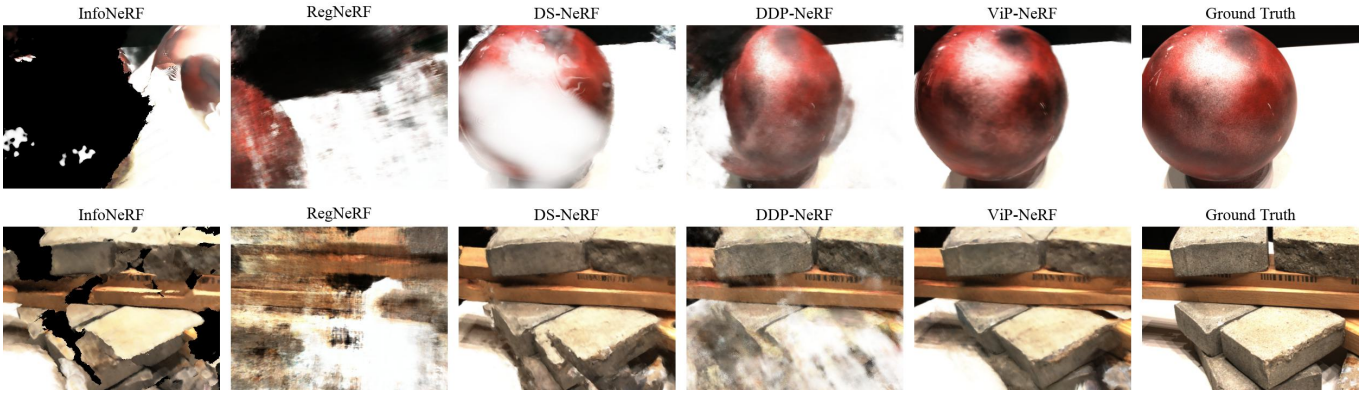


Fig. 12. Qualitative examples on DTU dataset. We observe various distortions among the predictions by competing models. For example, in the second row, the DS-NeRF prediction has a ghosting artifact in one of the bricks. However, predictions by our model are significantly closer to the ground truth.

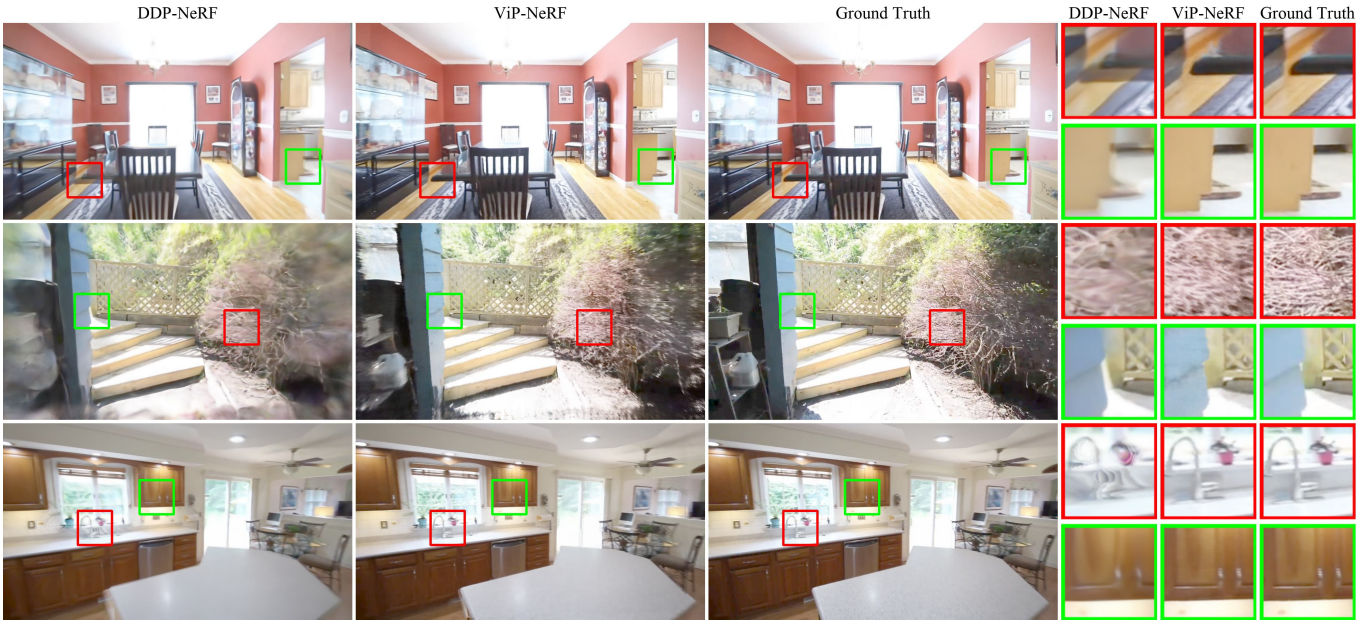


Fig. 13. Qualitative examples on RealEstate-10K dataset with two input views. In the first example, we see artifacts such as ghosting and blur in the frame predicted by DDP-NeRF. In the second example, we observe that DDP-NeRF infers incorrect geometry due to which the objects are placed incorrectly in the synthesized view. However, the predictions by ViP-NeRF do not suffer from such artifacts and are significantly sharper.

on the remaining three scenes. We observe that ViP-NeRF synthesizes superior quality frames compared to DDP-NeRF in all the scenes. On the NeRF-LLFF dataset, the figures in the main paper show ViP-NeRF predictions on 4 out of 8 scenes in the dataset. Figs. 14 and 15 shows ViP-NeRF predictions on the remaining four scenes in the NeRF-LLFF dataset.

Fig. 10 in the main paper shows qualitative comparisons with two input views on the NeRF-LLFF dataset. Figs. 14 and 15 show qualitative comparisons with three and four input views, respectively. From the figures, we observe that ViP-NeRF synthesizes sharper frames as compared to DDP-NeRF while maintaining better structures of the objects. Finally, Fig. 16 shows more comparisons with DDP-NeRF on the DTU dataset. We find that while ViP-NeRF synthesizes plausible frames, DDP-NeRF predictions suffer from floating white/black clouds that occlude the objects of interest.

In Tabs. 9 to 14, we provide scene-wise quantitative performance comparisons.

F ADDITIONAL ANALYSIS

F.1 PSV based Dense Depth Prior

Here, we ask whether the improvement in view synthesis performance is because the visibility prior can be reliably estimated or due to

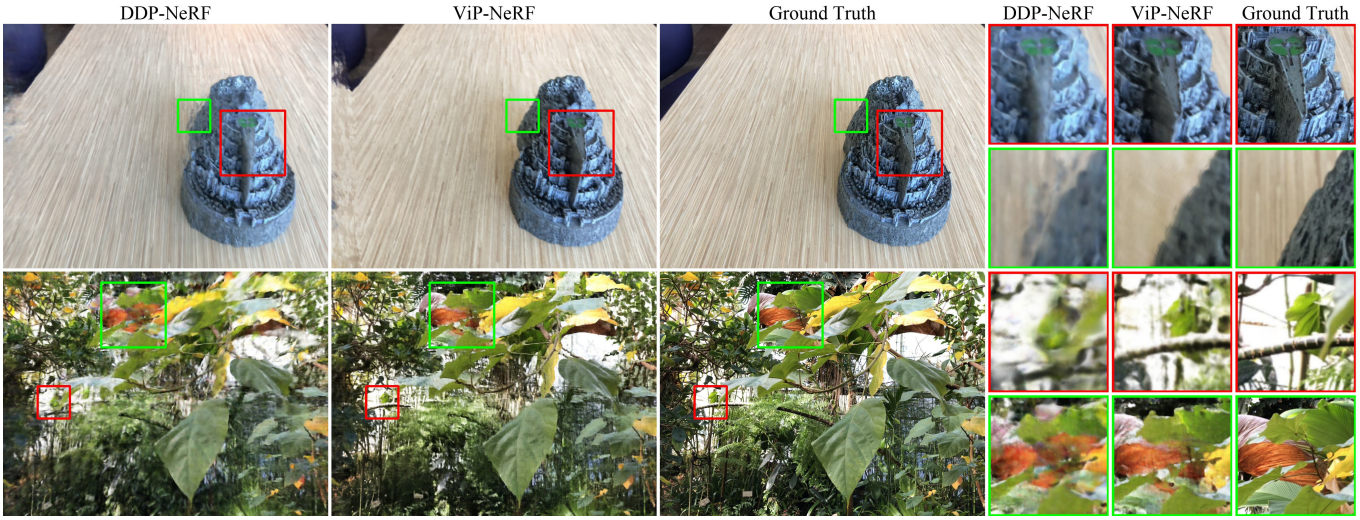


Fig. 14. Qualitative examples on NeRF-LLFF dataset with three input views. In the first example, we find that DDP-NeRF prediction has a different global color than the ground truth. In addition, the angle of the sharp triangular object on the fortress is changed. We also observe floating blue clouds outside the fortress and blur in other regions. In the second example, we notice that DDP-NeRF is unable to infer the positions of the objects (horizontal stem and the orange leaf) correctly and instead places them at incorrect positions or breaks them into multiple parts. On the other hand, ViP-NeRF is able to synthesize the novel views reasonably well.

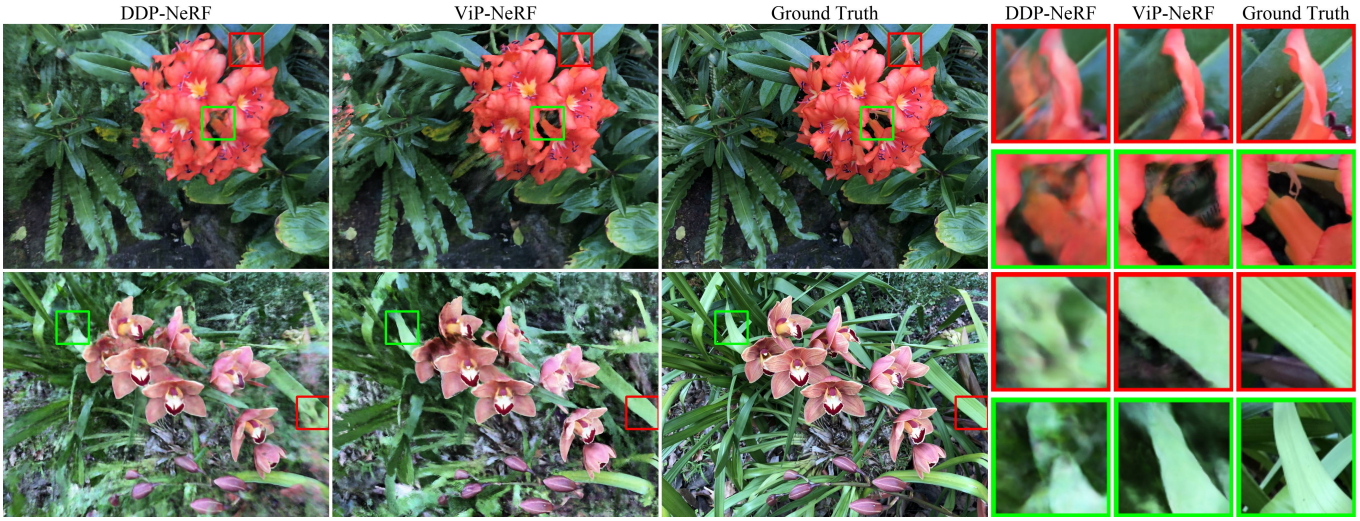


Fig. 15. Qualitative examples on NeRF-LLFF dataset with four input views. We find that DDP-NeRF predictions contain floaters (orange in the first example and black in the second), whereas ViP-NeRF predictions are devoid of such artifacts. We also observe that ViP-NeRF finds difficulty in synthesizing disoccluded regions such as the flower stem in the first example. Nonetheless, ViP-NeRF predicts plausible solutions in such cases.

the use of plane sweep volumes. To answer this, we experiment with obtaining a dense depth prior using the plane sweep volume similar to our visibility map computation. Specifically, we estimate dense depth as $z_p(\mathbf{q}) = \arg\min_k E_k(\mathbf{q})$ similar to Eq. 10 and use it to supervise NeRF similar to Eq. 13. We show the quantitative and qualitative performance of this model in Tab. 8 and Fig. 17, where we find that its performance significantly deteriorates. This supports our hypothesis that estimating depth accurately is a very hard problem that may require pre-training on a large dataset, but reliably estimating visibility appears to be relatively easier.

F.2 Distant Views

To understand how the performance of ViP-NeRF varies when the training images are farther apart with more occlusions, we train ViP-NeRF with 2 input views that are 10, 20, 30, and 40 frames apart on the RealEstate-10K dataset. We show the performance of ViP-NeRF

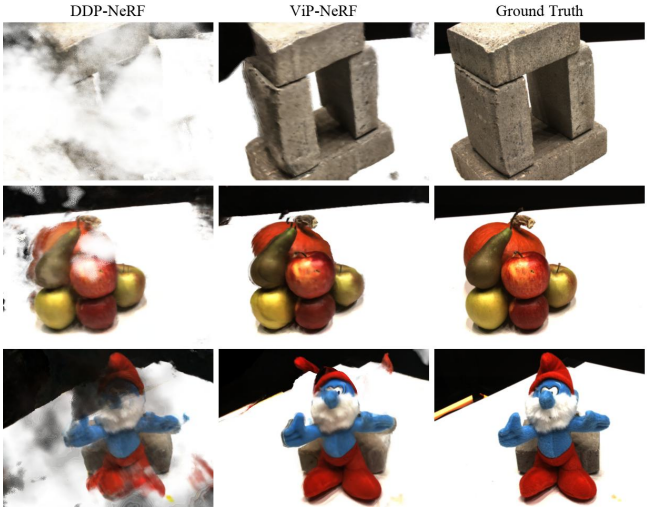


Fig. 16. Qualitative examples on DTU dataset with two input views. DDP-NeRF predictions contain significant floating clouds in all three examples, whereas ViP-NeRF produces more realistic novel views.

Table 8. Comparison between NeRF models trained with visibility and depth priors obtained using plane sweep volume.

model	RealEstate-10K		NeRF-LLFF	
	LPIPS ↓	**SSIM ↑**	**LPIPS ↓**	**SSIM ↑**
ViP-NeRF	**0.1704**	**0.8087**	**0.4017**	**0.5222**
DS-NeRF + PSV dense depth	0.7453	0.4247	0.6238	0.3878

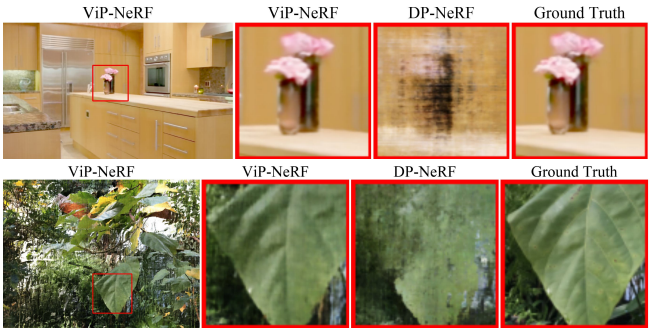


Fig. 17. Qualitative examples on RealEstate-10K and NeRF-LLFF datasets with visibility and depth prior (DP-NeRF) obtained using our method. We observe significant distortions in the predictions of DP-NeRF due to unreliable depth prior.

in terms of LPIPS and SSIM in Fig. 18. For reference, we also conduct a similar experiment with DDP-NeRF and report its performance. We note that the performance of ViP-NeRF stays relatively stable across different settings. We observe a small improvement in performance initially, which may be due to the availability of new regions during training when the train views are spread more apart. However, the performance tends to drop in small amounts with a further increase in distance between the training views. Further, ViP-NeRF outperforms DDP-NeRF in all cases.

F.3 Accuracy of Predicted Visibility

Recall that we use Eq. 12 during training to enforce the visibility $\hat{T}_i$ output by $\mathcal{F}_2$ to be consistent with the visibility T_i computed by volume rendering. In Fig. 19, we plot $\mathcal{L}_v$ against the iteration number as the training progresses. We observe that the loss uniformly decreases to a very small value close to zero indicating that $\hat{T}_i$ and T_i are indeed close to each other. This helps effective imposition of our visibility prior.

G PERFORMANCE ON INDIVIDUAL SCENES

In Tabs. 9 to 14, we report the performance of various models on individual scenes.

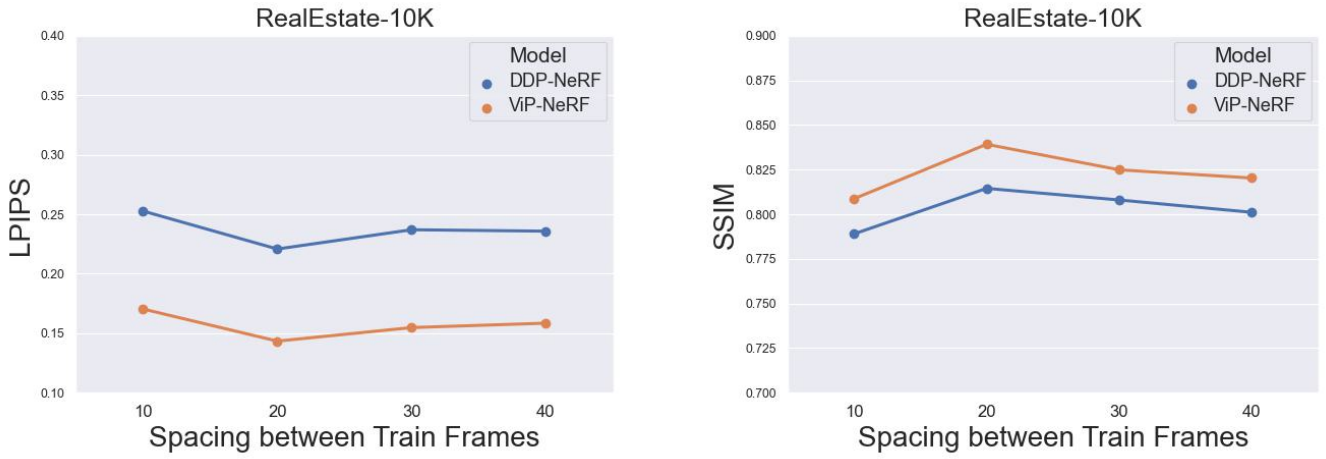


Fig. 18. Quantitative comparison of the performance of DDP-NeRF and ViP-NeRF models with increasing distance between the training views. The x-axis denotes the frames skipped between the two training frames.

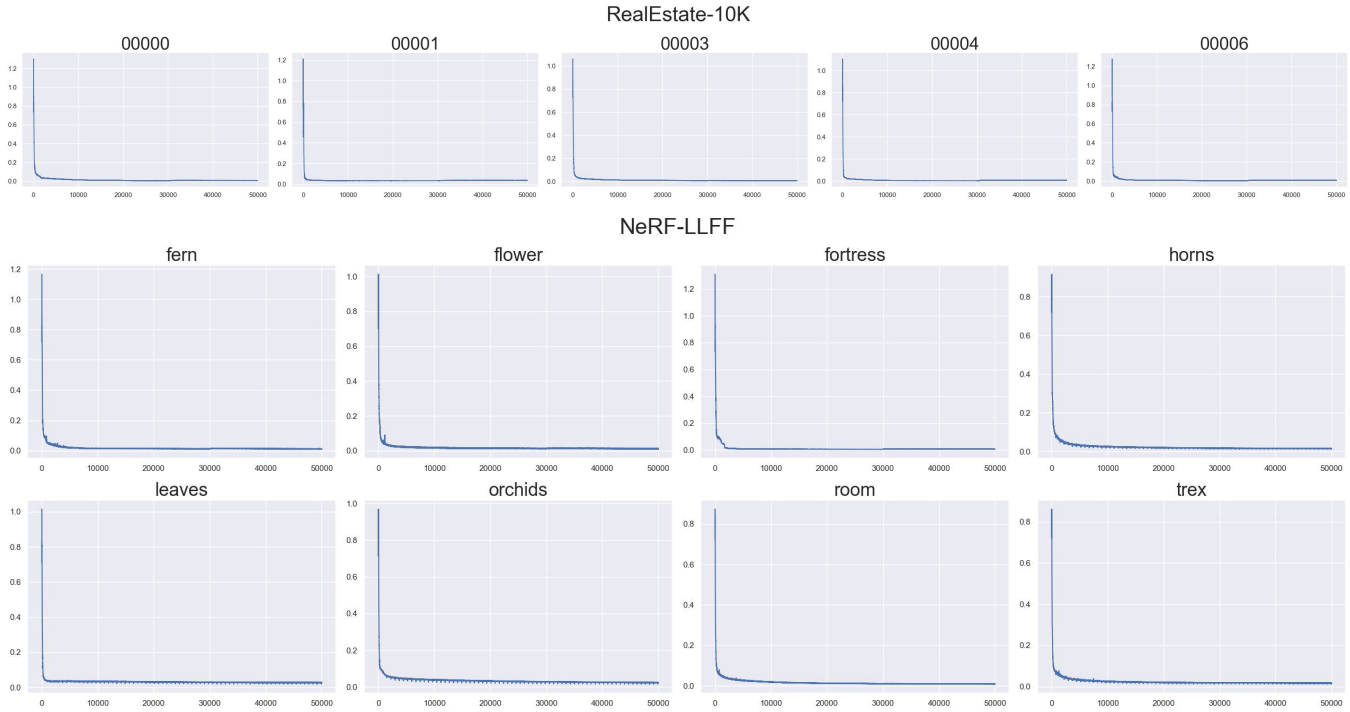


Fig. 19. Loss curves of $\mathcal{L}_v$ during training on individual scenes of both RealEstate-10K and NeRF-LLFF datasets.

Table 9. Per-scene performance of various models with two input views on RealEstate-10K dataset. The three rows show LPIPS, SSIM, and PSNR scores, respectively.

model \ scene name	**0**	**1**	**3**	**4**	**6**	**average**
	0.6545	0.5037	0.7386	0.7982	0.7033	0.6796
InfoNeRF	0.4863	0.7458	0.3002	0.3990	0.3953	0.4653
	11.85	16.19	11.59	10.55	11.32	12.30
	0.6242	0.4765	0.7154	0.5885	0.4604	0.5730
DietNeRF	0.5743	0.8298	0.3065	0.6632	0.6919	0.6131
	13.16	18.13	12.11	16.86	19.22	15.90
	0.5212	0.4305	0.6050	0.5427	0.5542	0.5307
RegNeRF	0.5416	0.8163	0.2549	0.6797	0.5618	0.5709
	14.83	19.11	12.88	17.78	16.13	16.14
	0.4128	0.3535	0.6269	0.3298	0.4133	0.4273
DS-NeRF	0.7128	0.8715	0.4210	0.8354	0.7705	0.7223
	18.99	23.35	**17.08**	25.48	22.10	21.40
	0.2265	0.2081	0.4788	0.1350	0.2150	0.2527
DDP-NeRF	0.7985	0.9222	**0.4663**	0.8964	**0.8615**	0.7890
	20.79	22.16	16.89	24.43	22.91	21.44
	0.1358	0.3363	0.5723	0.1100	0.2225	0.2754
ViP-NeRF w/o sparse depth	**0.8807**	0.8456	0.3422	0.9070	0.8186	0.7588
	25.54	19.89	13.26	**27.98**	24.48	22.23
	0.1351	**0.1008**	**0.3348**	**0.1053**	**0.1762**	**0.1704**
ViP-NeRF	0.8802	**0.9425**	0.4567	**0.9085**	0.8558	**0.8087**
	25.03	**26.81**	17.05	27.91	**25.60**	**24.48**

Table 10. Per-scene performance of various models with three input views on RealEstate-10K dataset. The three rows show LPIPS, SSIM, and PSNR scores, respectively.

model \ scene name	**0**	**1**	**3**	**4**	**6**	**average**
	0.7226	0.6075	0.6846	0.7916	0.6831	0.6979
InfoNeRF	0.3806	0.5669	0.2155	0.3991	0.4496	0.4024
	10.07	13.06	10.09	10.82	11.70	11.15
	0.5565	0.5400	0.6475	0.5258	0.4126	0.5365
DietNeRF	0.5841	0.7861	0.3068	0.7022	0.7156	0.6190
	14.86	16.83	12.56	18.11	20.65	16.60
	0.4885	0.3832	0.5947	0.5123	0.3585	0.4675
RegNeRF	0.5625	0.8200	0.2852	0.6851	0.6951	0.6096
	15.16	20.11	13.41	18.57	19.67	17.38
	0.3302	0.3241	0.5925	0.3245	0.3939	0.3930
DS-NeRF	0.7772	0.8902	0.4540	0.8460	0.8096	0.7554
	22.09	26.61	18.30	26.94	24.70	23.73
	0.1868	0.1644	0.4665	0.1067	0.1955	0.2240
DDP-NeRF	0.8475	0.9406	0.5229	0.9159	0.8848	0.8223
	22.17	23.53	17.94	27.46	24.39	23.10
	0.1022	0.3194	0.3119	0.0720	0.1252	0.1861
ViP-NeRF w/o sparse depth	0.9046	0.8284	**0.5611**	0.9277	0.9157	0.8275
	27.64	21.04	**18.77**	31.61	29.39	25.69
	0.0977	**0.1302**	**0.3083**	**0.0682**	**0.1163**	**0.1441**
ViP-NeRF	**0.9094**	**0.9408**	0.5515	**0.9319**	**0.9190**	**0.8505**
	27.59	**28.31**	18.59	**32.15**	**29.40**	**27.21**

Table 11. Per-scene performance of various models with four input views on RealEstate-10K dataset. The three rows show LPIPS, SSIM, and PSNR scores, respectively.

model \ scene name	0	1	3	4	6	average
InfoNeRF	0.6386	0.6315	0.7153	0.7993	0.5879	0.6745
	0.5111	0.4900	0.2117	0.3767	0.5596	0.4298
	12.46	10.99	9.72	10.13	14.28	11.52
DietNeRF	0.5724	0.4908	0.6502	0.5386	0.4164	0.5337
	0.5974	0.8224	0.3031	0.6964	0.7219	0.6282
	14.45	18.75	12.89	17.85	20.50	16.89
RegNeRF	0.5054	0.3902	0.6256	0.5229	0.3711	0.4831
	0.5577	0.8200	0.2834	0.6775	0.6952	0.6068
	15.32	20.27	13.54	18.29	19.87	17.46
DS-NeRF	0.3657	0.3161	0.6193	0.3039	0.3755	0.3961
	0.7671	0.8971	0.4562	0.8517	0.8152	0.7575
	22.54	27.41	18.65	27.44	25.17	24.24
DDP-NeRF	0.1834	0.1307	0.4746	0.1049	0.2015	0.2190
	0.8479	**0.9490**	0.5394	0.9161	0.8828	0.8270
	22.26	26.08	18.65	28.81	25.04	24.17
ViP-NeRF w/o sparse depth	0.1022	0.3194	0.3119	0.0720	0.1252	0.1861
	0.9046	0.8284	**0.5611**	0.9277	0.9157	0.8275
	27.64	21.04	**18.77**	31.61	29.39	25.69
ViP-NeRF	**0.0977**	**0.1302**	**0.3083**	**0.0682**	**0.1163**	**0.1441**
	0.9094	0.9408	0.5515	**0.9319**	**0.9190**	**0.8505**
	27.59	**28.31**	18.59	**32.15**	**29.40**	**27.21**

Table 12. Per-scene performance of various models with two input views on NeRF-LLFF dataset. The three rows show LPIPS, SSIM, and PSNR scores, respectively.

model \ scene name	fern	flower	fortress	horns	leaves	orchids	room	trex	average
InfoNeRF	0.7805	0.6880	0.8156	0.7649	0.6422	0.6908	0.8044	0.7941	0.7561
	0.2635	0.1890	0.1808	0.2020	0.0987	0.1108	0.3757	0.2115	0.2095
	11.00	10.97	6.48	8.86	9.47	9.43	10.76	8.44	9.23
DietNeRF	0.7663	0.6935	0.6621	0.7574	0.6720	0.7256	0.7674	0.7496	0.7265
	0.2883	0.2590	0.4424	0.2819	0.1196	0.1470	0.5153	0.3674	0.3209
	12.30	12.18	14.22	10.71	10.58	10.57	13.09	11.30	11.89
RegNeRF	0.5067	0.4408	0.3838	0.5301	**0.3590**	0.4595	0.3955	0.4308	0.4402
	0.4681	**0.5067**	0.4621	0.4277	**0.3637**	0.3018	0.7306	**0.5388**	0.4872
	16.51	**16.92**	20.53	15.91	14.51	13.88	18.59	**16.69**	16.90
DS-NeRF	0.5249	0.4578	0.3591	0.5395	0.5174	0.4614	0.3727	0.4386	0.4548
	0.4681	0.4455	0.6316	0.4816	0.2429	0.3190	**0.7607**	0.5295	0.5068
	16.69	16.17	**23.10**	16.64	12.68	13.86	**18.94**	15.85	17.06
DDP-NeRF	0.4715	0.4803	**0.1888**	0.4973	0.5550	0.4438	**0.3290**	0.4660	0.4223
	0.4940	0.4535	**0.7600**	0.5167	0.2282	**0.3682**	0.7572	0.5358	**0.5377**
	17.35	16.18	22.77	17.10	12.65	**15.12**	18.68	15.76	**17.21**
ViP-NeRF w/o sparse depth	0.4665	**0.3673**	0.3779	0.6068	0.4210	0.5252	0.7174	0.4706	0.5056
	0.4719	0.5044	0.5286	0.4279	0.3575	0.2913	0.5191	0.5241	0.4631
	16.60	16.78	21.22	15.65	**14.76**	14.21	13.76	16.50	16.28
ViP-NeRF	**0.4605**	0.4297	0.2689	**0.4356**	0.4842	**0.4238**	0.3626	**0.4052**	**0.4017**
	0.4586	0.4355	0.6869	**0.5353**	0.2172	0.3592	0.7371	0.5383	0.5222
	16.45	14.65	22.36	**17.25**	11.90	14.21	18.11	16.13	16.76

Table 13. Per-scene performance of various models with three input views on NeRF-LLFF dataset. The three rows show LPIPS, SSIM, and PSNR scores, respectively.

model \ scene name	fern	flower	fortress	horns	leaves	orchids	room	trex	average
	0.8583	0.6949	0.8272	0.7555	0.6518	0.7121	0.8312	0.7885	0.7679
InfoNeRF	0.2144	0.2308	0.1680	0.1800	0.1083	0.0834	0.2911	0.1768	0.1859
	7.51	10.75	5.22	8.87	9.92	8.28	8.94	8.75	8.52
	0.7977	0.6705	0.6984	0.7458	0.6672	0.7478	0.7620	0.7223	0.7254
DietNeRF	0.3207	0.3052	0.4098	0.2946	0.1266	0.1493	0.5518	0.3514	0.3297
	12.01	13.08	12.37	11.10	10.57	10.13	12.62	11.88	11.77
	0.4874	**0.2855**	0.3340	0.4706	**0.4040**	0.4540	0.2782	0.3685	0.3800
RegNeRF	0.4834	0.5764	0.6222	0.5177	0.3645	0.3058	0.8074	**0.6214**	0.5600
	17.84	19.48	22.62	18.12	14.61	14.11	20.90	**18.42**	18.62
	0.5146	0.3064	0.3024	0.5185	0.5533	0.4811	**0.2528**	0.4057	0.4077
DS-NeRF	0.5191	**0.6383**	0.6972	0.5131	0.2484	0.3289	**0.8335**	0.5861	0.5686
	18.64	**21.35**	**24.63**	17.59	12.85	14.15	**22.92**	17.31	**19.02**
	0.5186	0.3177	**0.2216**	0.5187	0.5602	0.4860	0.3328	0.4512	0.4178
DDP-NeRF	**0.5322**	0.6216	**0.7459**	0.5252	0.2396	**0.3518**	0.7659	0.5397	0.5610
	18.61	20.34	22.50	17.43	12.84	**15.19**	18.65	16.26	17.90
	0.6117	0.4248	0.2978	0.6195	0.4625	0.5381	0.4759	0.4739	0.4855
ViP-NeRF w/o sparse depth	0.4362	0.4961	0.6455	0.4585	**0.3743**	0.2938	0.7304	0.5127	0.5110
	16.31	18.15	23.66	16.34	**15.12**	14.37	19.21	16.60	17.71
	0.5529	0.2888	0.2354	**0.4632**	0.4842	**0.4416**	0.2937	**0.3483**	**0.3750**
ViP-NeRF	0.4958	0.6344	0.7412	**0.5619**	0.2522	0.3441	0.8116	0.6061	**0.5837**
	17.49	20.82	24.12	**18.27**	12.61	14.24	21.97	18.16	18.92

Table 14. Per-scene performance of various models with four input views on NeRF-LLFF dataset. The three rows show LPIPS, SSIM, and PSNR scores, respectively.

model \ scene name	fern	flower	fortress	horns	leaves	orchids	room	trex	average
	0.7962	0.6818	0.8686	0.7722	0.6574	0.7578	0.7812	0.7973	0.7701
InfoNeRF	0.1901	0.2296	0.1608	0.1853	0.0904	0.0851	0.4544	0.2589	0.2188
	9.83	11.54	4.72	8.81	9.30	8.12	11.89	10.10	9.25
	0.8035	0.6901	0.7050	0.7758	0.6992	0.7700	0.7314	0.7486	0.7396
DietNeRF	0.3473	0.2921	0.4186	0.3036	0.1366	0.1565	0.5944	0.3507	0.3404
	12.87	12.57	12.62	10.81	10.81	10.15	13.91	11.15	11.84
	0.3825	0.2981	0.3904	**0.3772**	**0.3384**	0.4463	0.2105	**0.3454**	**0.3446**
RegNeRF	**0.6221**	**0.6378**	0.5383	**0.6230**	**0.4248**	0.3315	0.8657	**0.6503**	0.6056
	20.87	19.80	22.23	**20.10**	**15.93**	14.73	23.84	**18.75**	19.83
	0.3945	0.3165	0.3601	0.4569	0.4684	0.4521	**0.1948**	0.4307	0.3825
DS-NeRF	0.6172	0.6285	0.6431	0.5766	0.3721	0.3803	**0.8794**	0.5884	0.6016
	20.96	**20.69**	**24.05**	19.52	15.81	15.40	**25.35**	17.31	**20.11**
	0.4593	0.3334	**0.2080**	0.4718	0.4921	0.4584	0.2729	0.4177	0.3821
DDP-NeRF	0.5849	0.6118	**0.7136**	0.5695	0.3533	**0.3937**	0.8227	0.6031	0.5999
	19.75	19.83	22.99	19.00	15.02	**15.72**	21.82	17.57	19.19
	0.4951	0.3100	0.3733	0.5117	0.4615	0.5283	0.3130	0.4061	0.4197
ViP-NeRF w/o sparse depth	0.5754	0.6116	0.5998	0.5470	0.3778	0.3106	0.8194	0.6215	0.5763
	19.16	19.49	23.19	18.20	15.68	14.65	22.00	18.65	19.15
	0.4298	**0.2970**	0.2970	0.4324	0.4316	**0.4295**	0.2607	0.3462	0.3593
ViP-NeRF	0.5788	0.6248	0.6866	0.5801	0.3760	0.3869	0.8402	0.6363	**0.6085**
	19.35	19.82	23.81	19.00	14.96	15.13	23.19	18.62	19.58