NeRF *On-the-go*: Exploiting Uncertainty for Distractor-free NeRFs in the Wild

Weining Ren[1*] Zihan Zhu[1*] Boyang Sun[1] Jiaqi Chen[1] Marc Pollefeys[1,2] Songyou Peng[1,3]
[1]ETH Zürich [2]Microsoft [3]MPI for Intelligent Systems, Tübingen
nerf-on-the-go.github.io

Figure 1. **NeRF *On-the-go*.** Given casually captured image sequences or videos in the wild as inputs, the goal of this paper is to train a NeRF for static scenes and effectively remove all dynamic elements in the scenes (cars, trams, pedestrians, etc), i.e. distractors. Unlike existing methods such as NeRF-W [23] and RobustNeRF [35], which produce imperfect results, our method leverages the predicted uncertainty maps to effectively remove those distractors. This results in high-fidelity novel view synthesis on challenging dynamic scenes.

Abstract

Neural Radiance Fields (NeRFs) have shown remarkable success in synthesizing photorealistic views from multi-view images of static scenes, but face challenges in dynamic, real-world environments with distractors like moving objects, shadows, and lighting changes. Existing methods manage controlled environments and low occlusion ratios but fall short in render quality, especially under high occlusion scenarios. In this paper, we introduce NeRF On-the-go, a simple yet effective approach that enables the robust synthesis of novel views in complex, in-the-wild scenes from only casually captured image sequences. Delving into uncertainty, our method not only efficiently eliminates distractors, even when they are predominant in captures, but also achieves a notably faster convergence speed. Through comprehensive experiments on various scenes, our method demonstrates a significant improvement over state-of-the-art techniques. This advancement opens new avenues for NeRF in diverse and dynamic real-world applications.

1. Introduction

Novel View Synthesis (NVS) tackles the challenge of rendering a scene from previously unobserved viewpoints. Neural radiance fields (NeRFs) [26] have emerged as a groundbreaking paradigm for this task. This is because a NeRF can produce geometrically consistent and photorealistic renderings, even for complex scenarios with thin structures and semi-transparent objects.

Training a NeRF model requires a set of RGB images with given camera poses, and demands manual adjustments of camera settings, such as focal length, exposure, and white balance. More crucially, vanilla NeRFs operate under the assumption that the scene should remain completely static during the capture process, without any *distractors* such as moving objects, shadows, or other dynamic elements [35]. Nevertheless, the real world is inherently dynamic, making this distractor-free requirement often unrealistic to meet. Additionally, removing distractors from the captured data is non-trivial. The process involves per-pixel annotation for each image, a procedure that is very labor-intensive, especially for lengthy captures of large scenes. This underscores a key limitation in the practical application of NeRFs in dynamic, real-world environments.

Recently, several works [22, 34, 42, 47] have attempted to address the challenges. [34] and [42] use pre-trained semantic segmentation models for specific moving objects, but the model fails to segment undefined object classes.

* Equal contribution.

NeRF-W [22] optimizes pixel-wise uncertainty from randomly initialized embedding by volume rendering. Such a design is suboptimal since it neglects the prior information of the image and entangles the uncertainty with radiance field reconstruction. As a result, they need to introduce transient embeddings to account for distractors. The addition of a new degree of freedom complicates system tuning, leading to a Pareto-optimal scenario as discussed in [35]. Dynamic NeRF methods like D^2NeRF [47] can decompose static and dynamic scenes for video input, but underperform with sparse image inputs. More recently, RobustNeRF [35] models distractors as outliers and demonstrates impressive results in controlled and simple scenarios. Nevertheless, its performance significantly drops in complex, in-the-wild scenes. Interestingly, RobustNeRF also underperforms in scenarios without any distractors. This leads to a compelling research question:

Can we build a NeRF for in-the-wild scenes from casually captured images, regardless of the ratio of distractors?

Toward this goal, we introduce NeRF *On-the-go*, a versatile plug-and-play module designed for effective distractor removal, allowing rapid NeRF training from any casually captured images. Our method is grounded in three key aspects. First, we utilize DINOv2 features [29] for their robustness and spatial-temporal consistency in feature extraction, from which a small multi-layer perception (MLP) predicts per-sample pixel uncertainty. Second, our method leverages a structural similarity loss to improve uncertainty optimization, enhancing the distinction between foreground distractors and the static background. Third, we incorporate estimated uncertainty into NeRF's image reconstruction objective using a decoupled training strategy, which significantly enhances distractor elimination, particularly in high occlusion scenes. Our method demonstrates robustness across a wide range of scenarios, from confined indoor scenes with small objects to complex, large-scale street view scenes, and can effectively handle varying levels of distractors. Notably, we find that our *On-the-go* module can also significantly accelerate NeRF training up to one order of magnitude, compared with RobustNeRF. This efficiency, combined with its straightforward integration with modern NeRF frameworks, makes NeRF *On-the-go* an accessible and powerful tool for enhancing NeRF training in dynamic real-world settings.

2. Related Work

Uncertainty in Scene Reconstruction. Uncertainty has proven to enhance the robustness and reliability of a wide range of tasks such as monocular depth prediction [11, 32], semantic segmentation [13, 27], and simultaneous localization and mapping (SLAM) [5, 24, 36, 54]. In general, uncertainty can be divided into two categories: epistemic and aleatoric [16]. In the specific context of scene reconstruction, epistemic uncertainty generally arises from data limitations, such as restricted viewpoints. For instance, [40] utilizes ensemble learning to quantify epistemic uncertainty for exploring unobserved regions in next-best-view (NBV) planning for NeRF. Goli et al. [10] establishes a volumetric uncertainty field to remove the floaters from NeRF. On the other hand, aleatoric uncertainty comes from the inherent randomness of the data, such as the noise of measurement, appearance changes, and distractors in the scene. There are works [15, 30, 33] that utilize aleatoric uncertainty as a guiding principle for active learning and NBV planning for better NeRF training. Similarly, DebSDF [49] improves indoor scene reconstruction through an uncertainty map to mitigate the noise from monocular prior.

Closely related to us, NeRF-W [23] was pioneering to eliminate transient objects and address variable illumination in unstructured internet photo collections, achieved by introducing transient and appearance embeddings. Follow-up works like Ha-NeRF [4] hallucinates NeRFs from unconstrained tourism images, while Neural Scene Chronology [21] reconstructs temporal-varying chronology from time-stamped Internet photos. Building upon previous formulation for aleatoric uncertainty, we innovate by integrating DINOv2 features into uncertainty prediction, which enhances the quality of predicted uncertainty. In a recent work, Kim et al. [17] also presents a similar DINO-based uncertainty prediction approach, but directly adapts for NeRF-W [23] to a pose-free condition. In contrast, we focus on refining NeRF training to effectively handle distractors from casually-captured image sequences.

SLAM and SfM in Dynamic Scenes. Handling dynamic scenes has been studied for years in the literature of SLAM and SfM. Classical methods exclude pixels associated with dynamic objects with robust kernel function [7, 28] or RANSAC [37, 38]. However, such hand-craft features are effective in scenarios with a low occlusion ratio but struggle at in-the-wild scenes. To address this, recent advances have integrated additional information. This includes external segmentation or detection modules for predefined classes [55, 57–59], utilization of optical or scene flow [2, 8, 41, 43, 56, 61], and geometry-based approaches using clustering and epipolar line distance [3, 12, 58].

NeRF in Dynamic Scenes. Recent NeRF methods focus on reconstructing both static and dynamic components from a video sequence [6, 9, 19, 20, 31, 44, 47, 48] enabling novel view synthesis at arbitrary timestamps. Although primarily designed for video inputs, these methods often underperform with photo collection sequences [35]. Additionally, separating static and dynamic components can be time-consuming and requires extensive hyperparameter tuning. A notable example in this realm is EmerNeRF [53], which also employs the DINOv2 [29] features. However, they use

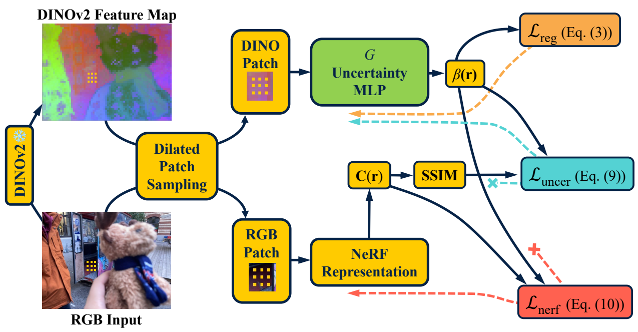


Figure 2. **Pipeline.** A pre-trained DINOv2 network extracts feature maps from posed images, followed by a dilated patch sampler that selects rays. The uncertainty MLP G then takes the DINOv2 features of these rays as inputs to generate the uncertainties $\beta(\mathbf{r})$. Three losses (on the right) are used to optimize G and the NeRF model. Note that the training process is facilitated by detaching the gradient flows as indicated by the colored dashed lines.

them for enhanced scene decomposition, while we use them as a strong prior knowledge for distractor removal.

RobustNeRF, to our knowledge the only method that also targets static scene reconstruction from dynamic scenes, uses Iteratively Reweighted Least Squares for outlier verification. Compared with it, our method can deal with more complex scenes with various levels of occlusions.

3. Method

We start by showing how to utilize per-pixel DINO features for uncertainty prediction (Sec. 3.1). Subsequently, we show a novel approach for learning uncertainty to remove distractors in NeRF (Sec. 3.2). We further introduce our decoupled optimization scheme for uncertainty prediction and NeRF (Sec. 3.3). Finally, we illustrate why sampling method is important in distractor-free NeRF training (Sec. 3.4). An overview of our pipeline is depicted in Fig. 2.

3.1. Uncertainty Prediction with DINOv2 Features

Our primary objective is to effectively identify and eliminate recurring distractors–those that appear across multiple images. To achieve this, we take advantage of DINOv2 [29] features, which have shown to be able to maintain spatial-temporal consistency across views.

We begin with extracting DINOv2 features for each input RGB image. Next, these features serve as inputs to a small MLP to predict the uncertainty value for each pixel. To further enforce the consistency of our uncertainty prediction, we incorporate a regularization term.

Image Feature Extraction. For RGB images with a resolution of $H \times W$, we derive per-pixel features through a pre-trained DINOv2 feature extractor $\mathcal{E}$:

$$\mathcal{F}_i = \mathcal{E}(\mathcal{I}_i),\ \mathcal{E} \in \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times C} \tag{1}$$

where i spans all training images, and C denotes feature dimension. This module also upsamples the feature maps to the original resolution by nearest-neighbor sampling.

Uncertainty Prediction. Once we obtain the 2D DINOv2 feature maps, we proceed to determine the uncertainty of each sampled ray $\mathbf{r}$. We first query its corresponding feature $\mathbf{f} = \mathcal{F}_i(\mathbf{r})$, and then input it to a shallow MLP to estimate the uncertainty for this ray $\beta(\mathbf{r}) = G(\mathbf{f})$, where G is the uncertainty MLP. In the subsequent sections, we will demonstrate how this predicted uncertainty $\beta(\mathbf{r})$ is integrated into the optimization process as a weighting function, which plays a crucial role in refining the NeRF model, particularly in handling and mitigating the impact of distractors in the scene.

Uncertainty Regularization. To enforce spatial-temporal consistency in uncertainty predictions, we introduce a regularization term based on the cosine similarity of feature vectors within a minibatch. Specifically, for each sampled ray $\mathbf{r}$, we define a neighbor set $\mathcal{N}(\mathbf{r})$ consisting of rays in the same batch whose associated feature vectors exhibit high similarity to the feature $\mathbf{f}$ of $\mathbf{r}$. This neighbor set is formed by selecting rays that meet a specified cosine similarity threshold η:

$$\mathcal{N}(\mathbf{r}) = \{\mathbf{r}' |\cos(\mathbf{f}, \mathbf{f}') > \eta\}$$

where $\mathbf{f}'$ is the associated feature of $\mathbf{r}'$. The refined uncertainty for a ray $\mathbf{r}$ is computed as the average of $\mathcal{N}(\mathbf{r})$:

$$\bar{\beta}(\mathbf{r}) = \frac{1}{|\mathcal{N}(\mathbf{r})|} \sum_{r' \in \mathcal{N}(\mathbf{r})} \beta(\mathbf{r}') \tag{2}$$

To reinforce consistency, we introduce a regularization term that penalizes the variance of uncertainty within $\mathcal{N}(\mathbf{r})$:

$$\mathcal{L}_{\text{reg}}(\mathbf{r}) = \frac{1}{|\mathcal{N}(\mathbf{r})|} \sum_{r' \in \mathcal{N}(\mathbf{r})} (\bar{\beta}(\mathbf{r}) - \beta(\mathbf{r}'))^2. \tag{3}$$

This regularization aims to smooth out abrupt changes in uncertainty predictions across similar features from rays across images, thereby enhancing the overall robustness and consistency of the uncertainty estimation process.

3.2. Uncertainty for Distractor Removal in NeRF

We hypothesize that pixels correlating with dynamic elements (distractors) should have high uncertainty, whereas static regions should have low uncertainty. This premise allows us to effectively integrate predicted uncertainty into NeRF training objectives, aiming to progressively filter out distractors for enhanced novel view synthesis.

We will analyze the potential issue of the classical way of incorporating uncertainty into the loss function for NeRF. Finally, we introduce a simple yet effective modification, to incorporate uncertainty, for robust distractor removal.

Uncertainty Convergence Analysis. Uncertainty prediction has been widely used in different fields, including NeRF-based novel view synthesis. For example, in the seminal work NeRF in the Wild [23], their loss is written as [*]:

$$\mathcal{L}(\mathbf{r}) = \frac{\|\mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r})\|^2}{2\beta^2(\mathbf{r})} + \lambda_1 \log \beta(\mathbf{r}) \tag{4}$$

Here, $\mathbf{C}(\mathbf{r})$ and $\hat{\mathbf{C}}(\mathbf{r})$ represent the input and rendered RGB values. The uncertainty $\beta(\mathbf{r})$ is treated as a weight function. The regularization term is crucial for balancing the first term and preventing the trivial solution where $\beta(\mathbf{r}) = \infty$.

Here we present a simple analysis to understand how the uncertainty changes wrt. the loss function, we first take the partial derivative wrt. $\beta(\mathbf{r})$:

$$\frac{d\mathcal{L}(\mathbf{r})}{d\beta(\mathbf{r})} = -\frac{\|\mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r})\|^2}{\beta(\mathbf{r})^3} + \lambda_1 \frac{1}{\beta(\mathbf{r})} \tag{5}$$

Setting this derivative to 0, we derive the closed-form solution for the optimal uncertainty:

$$\frac{d\mathcal{L}(\mathbf{r})}{d\beta(\mathbf{r})} = 0 \Rightarrow \beta(\mathbf{r}) = \sqrt{\frac{1}{\lambda_1}} \|\mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r})\| \tag{6}$$

This reveals an important relationship between uncertainty prediction and the error between the rendered and input colors. Specifically, the optimal uncertainty is directly proportional to this error term.

However, a challenge arises when employing the ℓ_2 loss as shown in Eq. (4), particularly when the color of distractors and background is close (as illustrated in Fig. 3 (d)). In such cases, the predicted uncertainty in those regions will also be low according to Eq. (6). This impedes the effectiveness of uncertainty-based distractor removal, and leads to cloud artifacts in the rendered images.

Recognizing the limitation inherent in the ℓ_2 RGB loss, we propose a new loss for better uncertainty learning, so that the predicted uncertainty can discriminate between distractors and static background more effectively.

SSIM-Based Loss for Enhanced Uncertainty Learning. The structural similarity index (SSIM) is comprised of three measurements: luminance, contrast, and structure similarities. These components capture local structural and contractual differences, which is crucial for distinguishing between scene elements. This is verified in Fig. 3, where SSIM is effective in detecting distractors by incorporating these three components together. An SSIM loss can be formulated as:

$$\begin{aligned} \mathcal{L}_{\text{SSIM}} &= 1 - \text{SSIM}(P, \hat{P}) \\ &= 1 - L(P, \hat{P}) \cdot C(P, \hat{P}) \cdot S(P, \hat{P}) \end{aligned} \tag{7}$$

where P and $\hat{P}$ are patches sampled from the input and rendered images $\mathbf{C}(\mathbf{r})$ and $\hat{\mathbf{C}}(\mathbf{r})$, respectively. L, C, S refer to the luminance, contrast, and structure similarities between P and $\hat{P}$. We further modify Eq. (7) as:

$$\mathcal{L}_{\text{SSIM}} = (1 - L(P, \hat{P})) \cdot (1 - C(P, \hat{P})) \cdot (1 - S(P, \hat{P})) \tag{8}$$

Compared to Eq. (7), our reformulation in Eq. (8) places greater emphasis on the differences between dynamic and static elements. Consequently, this enhances the disparity in uncertainty, facilitating more effective optimization of uncertainty. The mathematical proof and comparisons between Eq. (7) and Eq. (8) are included in the supplements.

Building on this updated SSIM formulation, we introduce a new loss tailored for uncertainty learning:

$$\mathcal{L}_{\text{uncer}}(\mathbf{r}) = \frac{\mathcal{L}_{\text{SSIM}}}{2\beta(\mathbf{r})^2} + \lambda_1 \log \beta(\mathbf{r}) \tag{9}$$

This loss is a simple modification of Eq. (4), adapted for better uncertainty learning. $\mathcal{L}_{\text{uncer}}$ is specifically applied to train the uncertainty estimation MLP G. This is crucial as it allows us to decouple the training of the NeRF model from uncertainty prediction. Such decoupling ensures that the learned uncertainty is robust to various types of distractors. Please refer to Table 4 for an ablation for $\mathcal{L}_{\text{uncer}}$.

Note that a recent work S3IM [50] also uses SSIM for NeRF training, but their loss is tailored for static scenes, whereas ours is designed for better uncertainty learning. Also, S3IM employs stochastic sampling to identify non-local structural similarities, while we use dilated sampling to focus on local structures for distractor removal.

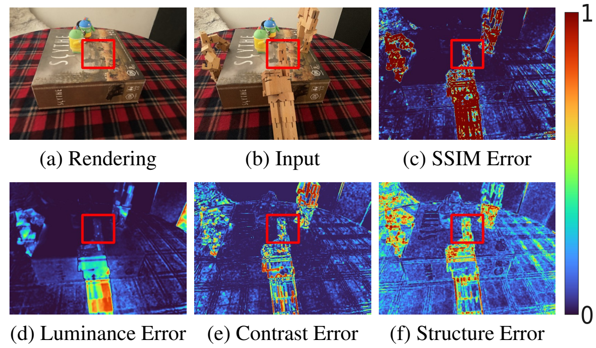
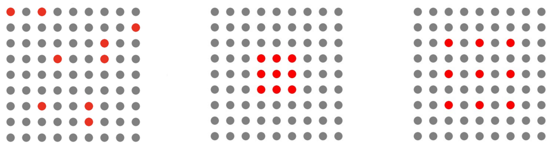


Figure 3. **SSIM Can Effectively Distinguish Distractors.** In this scene from [35], the 3 wooden robots are the dynamic elements. SSIM pinpoints distractors by leveraging discrepancies in three measurements including luminance, contrast, and structure. Conversely, relying solely on the ℓ_2 error between RGB values (luminance error) proves challenging, especially when the distractors and background have similar colors. The color bar on the right side indicates the correspondence for error interpretation.

3.3. Optimization

As mentioned above, it is crucial to separately optimize the uncertainty prediction module and NeRF model. For optmization of the uncertainty prediction MLP, we employ

[*]We omit their regularization term for transient density.



(a) Random [23] (b) Patch [35] (c) **Dilated Patch**

Figure 4. **Comparison of Different Ray Sampling Strategies.** In contrast to random sampling and patch sampling, dilated patch sampling can improve training efficiency and uncertainty learning.

$\mathcal{L}_{\text{uncer}}$ in Eq. (9) and $\mathcal{L}_{\text{reg}}$ in Eq. (3). In parallel, we train the NeRF model with the following:

$$\mathcal{L}_{\text{nerf}}(\mathbf{r}) = \frac{\|\mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r})\|^2}{2\beta^2(\mathbf{r})} \tag{10}$$

This loss, essentially Eq. (4) without the regularization term, is used because $\mathcal{L}_{\text{uncer}}$ already prevents trivial solutions for uncertainty ($\beta(\mathbf{r}) = \infty$). The parallel training process is facilitated by detaching the gradient flow from $\mathcal{L}_{\text{uncer}}$ to NeRF representation, and $\mathcal{L}_{\text{nerf}}$ to the uncertainty MLP G as illustrated in Fig. 2. Note that we also follow RobustNeRF [35] and include the interval loss and distortion loss from Mip-NeRF 360 [1] for NeRF training, which we omit here for simplicity. Our overall objectives integrate all losses together, denoted as:

$$\lambda_2 \mathcal{L}_{\text{nerf}}(\mathbf{r}) + \lambda_3 \mathcal{L}_{\text{uncer}}(\mathbf{r}) + \lambda_4 \mathcal{L}_{\text{reg}}(\mathbf{r}) \tag{11}$$

where each term is weighted by a corresponding λ.

3.4. Dilated Patch Sampling

In this section, we delve into the ray sampling strategy, a key factor in the efficacy of NeRF training, particularly in achieving distractor-free results.

RobustNeRF has demonstrated the efficacy of patch-based ray sampling (Fig. 4 (b)) over random sampling (Fig. 4 (a)). However, this approach has its limitations, primarily due to the small size of the sampled patches (e.g. 16×16). Especially when the batch size is small due to the constraint of GPU memory, this small context can restrict the network's learning capacity to remove distractors, impacting optimization stability and convergence speed.

To tackle the issue, we utilize dilated patch sampling [14, 25, 39, 45, 51, 52], depicted in Fig. 4 (c). This strategy involves sampling rays from a dilated patch. By enlarging the patch size, we can significantly increase the amount of contextual information available in each training iteration.

Our empirical findings in Table 3 show that dilated patch sampling not only accelerates the training process, but also yields superior performance in distractor removal.

4. Experiments

RobustNeRF Dataset. There are four sequences with toys-on-the-table settings. However, note that we are unable to include the *Crab* scene since it is not released. Meanwhile, we put comparisons on *Baby Yoda* scene in supplements, since each image in this sequence contains a distinct set of distractors, which is different from our setting.

On-the-go **Dataset.** To rigorously evaluate our approach in real-world indoor and outdoor settings, we captured a dataset that contains 12 casually captured sequences, including 10 outdoor and 2 indoor scenes, with varying ratios of distractors (from 5% to over 30 %). For quantitative evaluation, we select 6 sequences representing different occlusion rates, as shown in Fig. 5. More details and results for this dataset are available in supplements.

Baselines. We compare our method with Mip-NeRF 360 [1], D^{2}NeRF [47], NeRF-W [23]†, Ha-NeRF [4]‡, RobustNeRF [35]§, and Mip-NeRF 360 + SAM, a baseline that we design to exclude dynamic objects in images with SAM [18], and train a NeRF on static parts. Refer to supplements for more details.

Metrics. We adopt the widely used PSNR, SSIM [46] and LPIPS [60] for the evaluation of novel view synthesis.

Figure 5. ***On-the-go*** **Dataset.** Sample training images showing the distractors in several scenes of our self-captured dataset.

4.1. Evaluation

On-the-go **Dataset.** We extend our evaluation on our *On-the-go* dataset, as depicted in Fig. 5 and Table 1. Compared to our method, RobustNeRF often fails to retain fine details in low to medium-occlusion scenarios, and struggles to eliminate distractors in high-occlusion settings. Besides, we notice that even after tuning the hyperparameter of outlier ratios for highly-occluded scenes, RobustNeRF still shows inferior performance. Please refer to the supplements.

Unlike RobustNeRF, NeRF-W and Ha-NeRF show proficiency in removing distractors at low and medium occlu-

† https://github.com/kwea123/nerf_pl/tree/nerfw
‡ https://github.com/rover-xingyu/Ha-NeRF
§ https://github.com/google-research/multinerf

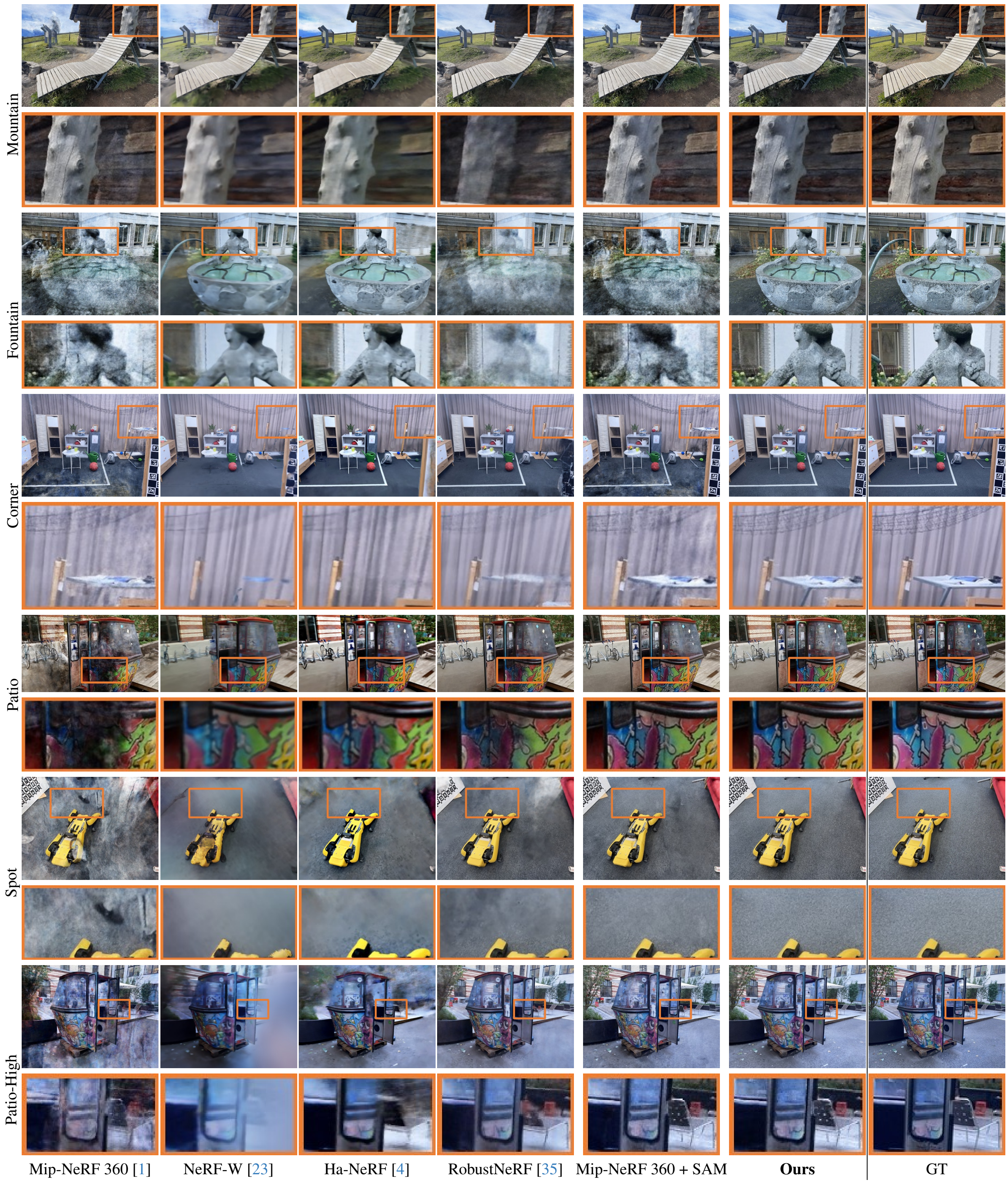


Figure 6. **Novel View Synthesis Results on Our *On-the-go* Dataset.** Our method constantly outperforms baseline methods on scenes with various ratios of distractors, from confined indoor scenes with objects to large outdoor scenes.

	Low Occlusion						Medium Occlusion						High Occlusion					
	Mountain			Fountain			Corner			Patio			Spot			Patio-High		
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
Mip-NeRF 360 [1]	0.295	0.601	19.64	0.556	0.290	13.91	0.345	0.660	20.41	0.421	0.503	15.48	0.469	0.306	17.82	0.486	0.432	15.73
NeRF-W [23]	0.491	0.492	18.07	0.546	0.410	17.20	0.349	0.708	20.21	0.445	0.532	17,55	0.690	0.384	16.40	0.606	0.349	12.99
Ha-NeRF [4]	0.499	0.485	18.64	0.569	0.393	16.71	0.367	0.684	19.23	0.393	0.543	16.82	0.599	0.460	17.85	0.505	0.463	16.67
RobustNeRF [35]	0.383	0.496	17.54	0.576	0.318	15.65	0.244	0.764	23.04	0.251	0.718	20.39	0.391	0.625	20.65	0.366	0.578	20.54
Mip-NeRF 360 + SAM	**0.258**	0.642	**20.20**	0.556	0.287	13.65	0.332	0.670	20.65	0.227	0.738	20.83	0.323	0.542	21.08	0.326	0.576	20.13
Ours	0.259	**0.644**	20.15	**0.314**	**0.609**	**20.11**	**0.190**	**0.806**	**24.22**	**0.219**	**0.754**	**20.78**	**0.189**	**0.787**	**23.33**	**0.235**	**0.718**	**21.41**

Table 1. **Novel View Synthesis Results on Our *On-the-go* Dataset.** We show quantitative comparison between our methods and baselines.

sion levels, but this effectiveness comes at the cost of reduced image quality. This trade-off is a consequence of its transient embedding approach, as discussed in [30, 35]. Furthermore, NeRF-W and Ha-NeRF struggle notably at higher occlusion ratios. In such cases, their per-image transient embeddings are unable to adequately model distractors, leading to a noticeable performance drop. The Mip-NeRF 360 combined with SAM method works well in simple scenes like *Mountain*, where distractors are easy to segment. However, its effectiveness diminishes in more complex scenes. In contrast, we exhibit versatility across scenes with various occlusion ratios, and can consistently produce high-quality renderings.

Comparison on RobustNeRF Dataset [35]. As shown in Table 2, our method exhibits superior performance quantitatively and qualitatively over all baselines. RobustNeRF's hard-thresholding approach tends to overlook complex structures with limited observations, such as the shoes and carpet in the *Android* scene. Moreover, we observed that they underperform in scenarios involving plane surfaces with view-dependent effects, e.g. the wooden texture on the table with view-dependent highlight in *Statue* scene. Note that Mip-NeRF 360 + SAM requires a tedious process of manually selecting every distractor in each image using SAM [18], but it still struggles with capturing thin structures, shadows, and reflections.

4.2. Ablation Study

All ablations are conducted on the challenging highly-occluded “Patio-High” scene in our *On-the-go* dataset.

Patch Dilation. Here we test different dilation rates for our dilation patch sampling, as shown in Table 3. Within a range from 1 to 4, a higher dilation rate results in much faster convergence and better rendering quality. This verifies our hypothesis in Sec. 3.4 that increasing the contextual information within patches can effectively boost performance. However, when the dilation rate is above 4, uncertainty optimization begins to collapse. It is likely because higher dilation rates cause patches to lose semantic information. This occurs as the sampling now becomes more akin to random sampling, negatively impacting the learning of uncertainty. Further details and analysis on patch size and dilation rate across different sequences are available in the supplements.

	Statue			Android		
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
Mip-NeRF 360 [1]	0.36	0.66	19.09	0.40	0.65	19.35
D^{2}NeRF [47]	0.48	0.49	19.09	0.43	0.57	20.61
RobustNeRF [35]	0.28	0.75	20.89	0.31	0.65	21.72
RobustNeRF* [35]	0.27	0.73	21.13	0.22	0.73	22.83
Mip-NeRF 360 + SAM	**0.23**	0.74	21.30	0.23	0.71	22.62
Ours	0.24	**0.77**	**21.58**	**0.21**	**0.75**	**23.50**

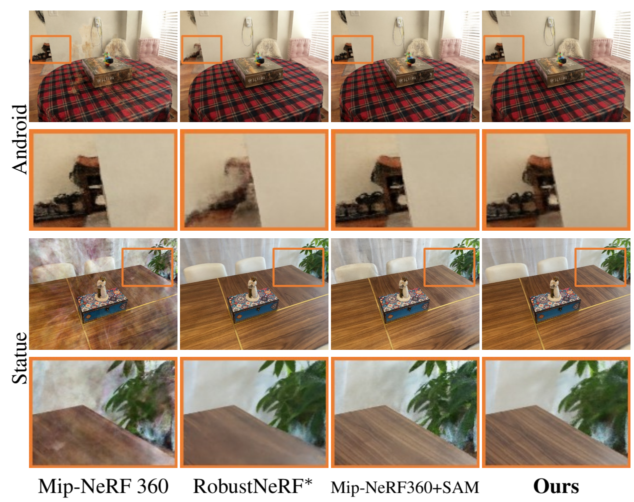


Table 2. **Novel View Synthesis Results on the RobustNeRF Dataset.** The numbers for Mip-NeRF 360 [1], D^{2}NeRF [47] and RobustNeRF [35] are taken from [35]. RobustNeRF* [35] denotes our own run using the official code release.

Loss Functions. In Table 4, we ablate on different training losses. In (b), SSIM proves more adept at differentiating distractors with static elements compared to ℓ_2 loss. In (c), we train the uncertainty MLP and NeRF together. This results in a significant performance drop, indicating the effectiveness of our decoupled training approach. Moreover, we find from (a) that omitting $\mathcal{L}_{reg}$ will negatively impact the rendering quality of certain views. Additional studies on various sequences are available in the supplements.

4.3. Analysis

Fast Convergence. Fig. 7 presents a comparison between RobustNeRF and ours during training processes. Thanks to our uncertainty prediction pipeline and dilated patch sam-

	LPIPS↓	SSIM↑	PSNR↑
1	0.451	0.515	17.82
2	0.262	0.692	20.70
4	**0.235**	**0.718**	**21.41**
8	0.392	0.529	18.22
16	0.477	0.439	16.08

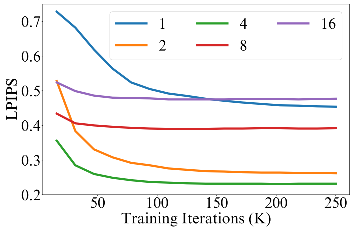


Table 3. **Ablations on Patch Dilation Rates**. Comparisons of various dilation rates for the dilated patch sampling, with a patch size of 32×32.

	LPIPS↓	SSIM↑	PSNR↑
(a) w/o $\mathcal{L}_{\text{reg}}$	0.261	0.698	21.02
(b) ℓ_2 in $\mathcal{L}_{\text{uncer}}$	0.437	0.492	17.13
(c) $\mathcal{L}_{\text{uncer}}$ for NeRF	0.496	0.437	16.70
Ours	**0.235**	**0.718**	**21.41**

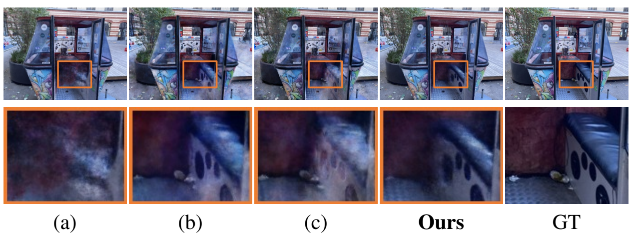


Table 4. **Ablations on Loss Functions.** We compare different loss choices for training our system.

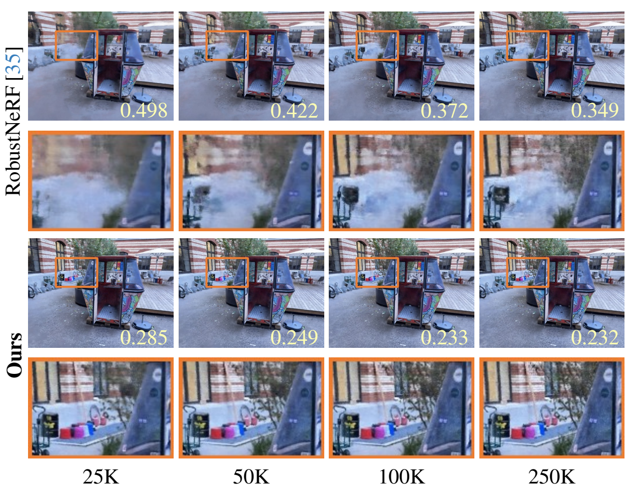


Figure 7. **Convergence Speed Comparison.** LPIPS metrics are included in images. Our method can already capture better details at 25K iterations than RobustNeRF at 250K iterations.

pling, we show notably faster convergence. It can be noticed that we can already capture fine details from the early stages of training, see ours at 25K and RobustNeRF at 250K.

Applicability to Static Scenes. After showcasing our efficacy in building a NeRF from dynamic scenes, we explore whether it is directly adaptable to static scenes. We evaluate

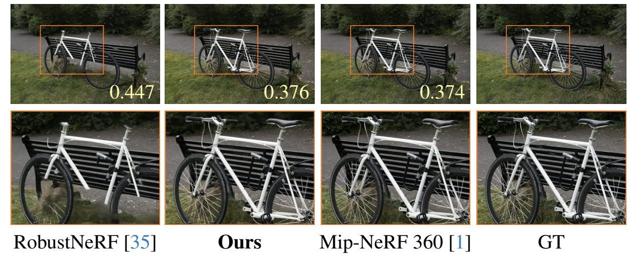


Figure 8. **Performance on Static Scenes.** LPIPS metrics are included in images. Our performance is much better than RobustNeRF and on par with the SOTA method [1].

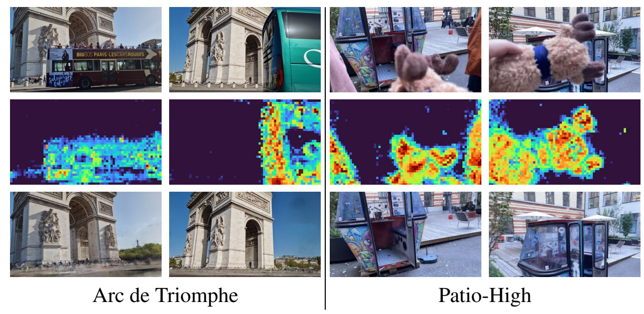


Figure 9. **Handling Large Obstructions.** From top to bottom: input frames, our uncertainty maps, our rendering results.

using a static scene from the Mip-NeRF 360 [1] dataset. As illustrated in Fig. 8, we indeed achieve great performance as Mip-NeRF 360 [1]. In contrast, RobustNeRF fails to capture certain parts of the bicycle, since one of their key designs involves omitting at least some portions of a scene.

Large Obstructions. In Fig. 9, we further show that our method can faithfully model the large obstructions with our predicted uncertainty, and effectively remove them.

5. Conclusions

We introduce NeRF *On-the-go*, a versatile method that enables effective and efficient distractor removal in dynamic real-world scenes containing various levels of distractors. Our method represents a step towards realizing the full potential of NeRF in practical, in-the-wild applications.

Limitation. While our method shows robustness on diverse real-world scenes, we suffer in predicting correct uncertainties for regions with strong view-dependent effects, such as highly reflective surfaces like windows and metals. Integrating additional prior knowledge into the optimization process could potentially be beneficial.

Acknowledgements. We thank the Max Planck ETH Center for Learning Systems (CLS) for supporting Songyou Peng. We also thank Yiming Zhao and Clément Jambon for helpful discussions.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 5, 6, 7, 8

[2] Ioan Andrei Bârsan, Peidong Liu, Marc Pollefeys, and Andreas Geiger. Robust dense mapping for large-scale dynamic environments. In *ICRA*, 2018. 2

[3] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *RA-L*, 2018. 2

[4] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *CVPR*, 2022. 2, 5, 6, 7

[5] Gabriele Costante and Michele Mancini. Uncertainty estimation for data-driven visual odometry. *IEEE Transactions on Robotics*, 2020. 2

[6] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *ICCV*, 2021. 2

[7] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *PAMI*, 2017. 2

[8] Daniela Esparza and Gerardo Flores. The stdyn-slam: a stereo vision and semantic segmentation approach for vslam in dynamic outdoor environments. *IEEE Access*, 2022. 2

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

[10] Lily Goli, Cody Reading, Silvia Selllán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' rays: Uncertainty quantification for neural radiance fields. In *CVPR*, 2024. 2

[11] Julia Hornauer and Vasileios Belagiannis. Gradient-based uncertainty for monocular depth estimation. In *ECCV*, 2022. 2

[12] Jiahui Huang, Sheng Yang, Zishuo Zhao, Yu-Kun Lai, and Shi-Min Hu. Clusterslam: A slam backend for simultaneous rigid body clustering and motion estimation. In *CVPR*, 2019. 2

[13] Po-Yu Huang, Wan-Ting Hsu, Chun-Yueh Chiu, Ting-Fan Wu, and Min Sun. Efficient uncertainty estimation for semantic segmentation in videos. In *ECCV*, 2018. 2

[14] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021. 5

[15] Liren Jin, Xieyuanli Chen, Julius Rückin, and Marija Popović. Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. In *IROS*, 2023. 2

[16] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017. 2

[17] Injae Kim, Minhyuk Choi, and Hyunwoo J Kim. Up-nerf: Unconstrained pose-prior-free neural radiance fields. In *NeurIPS*, 2023. 2

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 5, 7

[19] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *CVPR*, 2022. 2

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

[21] Haotong Lin, Qianqian Wang, Ruojin Cai, Sida Peng, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural scene chronology. In *CVPR*, 2023. 2

[22] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 1, 2

[23] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 1, 2, 4, 5, 6, 7

[24] Nathaniel Merrill, Yuliang Guo, Xingxing Zuo, Xinyu Huang, Stefan Leutenegger, Xi Peng, Liu Ren, and Guoquan Huang. Symmetry and uncertainty-aware object slam for 6dof object pose estimation. In *CVPR*, 2022. 2

[25] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In *ECCV*, 2022. 5

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

[27] Jishnu Mukhoti, Joost van Amersfoort, Philip HS Torr, and Yarin Gal. Deep deterministic uncertainty for semantic segmentation. In *CVPR*, 2023. 2

[28] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 2017. 2

[29] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3

[30] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *ECCV*, 2022. 2, 7

[31] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. *ACM TOG*, 2021. 2

[32] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*, 2020. 2

[33] Yunlong Ran, Jing Zeng, Shibo He, Jiming Chen, Lincheng Li, Yingfeng Chen, Gimhee Lee, and Qi Ye. Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *RA-L*, 2023. 2

[34] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022. 1

[35] Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *CVPR*, 2023. 1, 2, 4, 5, 6, 7, 8

[36] Erik Sandström, Kevin Ta, Luc Van Gool, and Martin R Oswald. Uncle-slam: Uncertainty learning for dense neural slam. In *ICCV Workshops*, 2023. 2

[37] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[38] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 2

[39] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 5

[40] Jianxiong Shen, Ruijie Ren, Adria Ruiz, and Francesc Moreno-Noguer. Estimating 3d uncertainty field: Quantifying uncertainty for neural radiance fields. In *ICRA*, 2024. 2

[41] Shihao Shen, Yilin Cai, Wenshan Wang, and Sebastian Scherer. Dytanvo: Joint refinement of visual odometry and motion segmentation in dynamic environments. In *ICRA*, 2023. 2

[42] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, 2022. 1

[43] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *NeurIPS*, 2021. 2

[44] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021. 2

[45] Cong Wang, Di Kang, Yan-Pei Cao, Linchao Bao, Ying Shan, and Song-Hai Zhang. Neural point-based volumetric avatar: Surface-guided neural points for efficient and photorealistic volumetric head avatar. In *SIGGRAPH Asia Conference Papers*, 2023. 5

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 5

[47] Tianhao Wu, Fangcheng Zhong, Forrester Cole, Andrea Tagliasacchi, and Cengiz Oztireli. D2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. In *NeurIPS*, 2022. 1, 2, 5, 7

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

[49] Yuting Xiao, Jingwei Xu, Zehao Yu, and Shenghua Gao. Debsdf: Delving into the details and bias of neural indoor scene reconstruction. *arXiv preprint arXiv:2308.15536*, 2023. 2

[50] Zeke Xie, Xindi Yang, Yujie Yang, Qi Sun, Yixiang Jiang, Haoran Wang, Yunfeng Cai, and Mingming Sun. S3im: Stochastic structural similarity and its unreasonable effectiveness for neural fields. In *ICCV*, 2023. 4

[51] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *ECCV*, 2022. 5

[52] Shiyao Xu, Lingzhi Li, Li Shen, and Zhouhui Lian. Desrf: Deformable stylized radiance field. In *CVPR*, 2023. 5

[53] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. In *ICLR*, 2024. 2

[54] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *CVPR*, 2020. 2

[55] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 2019. 2

[56] Weicai Ye, Xingyuan Yu, Xinyue Lan, Yuhang Ming, Jinyu Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Deflowslam: Self-supervised scene motion decomposition for dynamic dense slam. *arXiv preprint arXiv:2207.08794*, 2022. 2

[57] Weicai Ye, Xinyue Lan, Shuo Chen, Yuhang Ming, Xingyuan Yu, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Pvo: Panoptic visual odometry. In *CVPR*, 2023. 2

[58] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *IROS*, 2018. 2

[59] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. Vdo-slam: a visual dynamic object-aware slam system. *arXiv preprint arXiv:2005.11052*, 2020. 2

[60] 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. 5

[61] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *ECCV*, 2022. 2