IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Freeze-Frame with StaticNeRF: Uncertainty-Guided NeRF Map Reconstruction in Dynamic Scenes

Juhui Lee, Geonmo Yang, Seungjun Ma, and Younggun Cho

***Abstract*—Recent advances in neural representations have shown great promise for enabling high-fidelity dense mapping in robotics. Given the inherently dynamic nature of real-world environments, many studies have attempted to learn static scene representations from dynamic observations. However, existing methods often fail to remove subtly moving objects and struggle to accurately recover occluded static backgrounds, which leads to critical limitations in practice. Furthermore, when static neural maps are used for localization, dynamic content in query images must be handled effectively. To overcome these challenges, we propose a static neural mapping framework that is robust to diverse dynamic environments and capable of processing dynamic content during localization. We evaluated our approach through extensive experiments on both public and in-house datasets. Our method improves both dynamic object removal and localization robustness under dynamic conditions, and constitutes a significant step toward resilient robot navigation in real-world environments.**

***Index Terms*—Neural Radiance Fields, Mapping, Localization**

I. Introduction

NeRF [1] has emerged as a core technology in robotics, particularly for dense map representation. Unlike traditional mapping methods [2–4], NeRF [1] enables photorealistic view synthesis, making it attractive for applications demanding accurate and realistic scene understanding. In contrast to explicit maps, NeRF [1] uses an Multi Layer Perceptron (MLP) to model continuous radiance fields without storing discrete 3D points. This results in a more memory-efficient representation, especially beneficial for robots with limited onboard resources.

The inherently dynamic nature of real-world environments introduces two major challenges for neural mapping and localization: **(i) map corruption**, where dynamic objects are mistakenly fused into the map during reconstruction, and **(ii) query–map inconsistency**, where dynamic content appears in the query image but is absent from the pre-constructed map. Both issues degrade localization reliability by introducing mismatches between the query and the map.

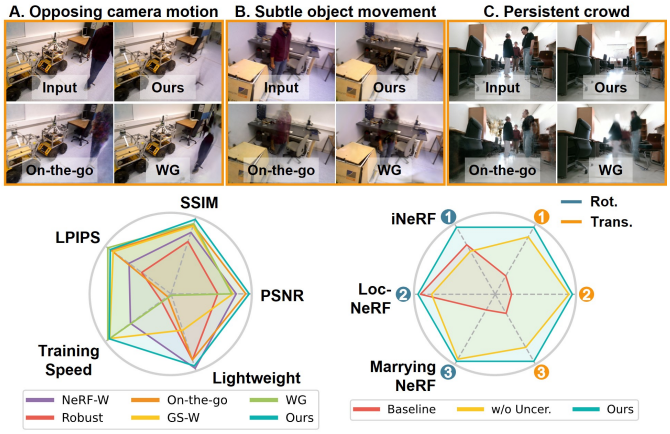

Fig. 1. **StaticNeRF.** The top row illustrates challenging real-world scenarios, while the bottom presents radar charts comparing rendering quality and model efficiency (left), and localization accuracy (right) across our method, a baseline, and an ablation without the uncertainty module. Our method effectively removes dynamic objects while achieving high quality, memory efficiency, and fast training, outperforming prior approaches [5, 7–9, 12]. By addressing key limitations of existing localization techniques [15–17], StaticNeRF improves localization accuracy in both rotation (blue) and translation (orange).

To mitigate map corruption, several works [5–14] have focused on recovering static backgrounds from dynamic scenes, with an emphasis on novel view synthesis rather than robust mapping for real-world robotics. These approaches are effective when static surfaces are frequently visible, but often fail when such regions are persistently occluded by dynamic objects (as discussed in Sec. III (Problem Formulation)). Given the complexity and diversity of robot trajectories in the real-world, a more robust and generalizable mapping strategy is still needed.

We present **StaticNeRF**, a static neural mapping pipeline robust to dynamic environments. StaticNeRF augments the radiance field with a lightweight Convolutional Neural Networks (CNN)-based uncertainty module, which (i) supplements static background regions that NeRF [1] alone cannot reconstruct, and (ii) identifies dynamic pixels in each query frame. In addition, a three-phase curriculum learning schedule stabilizes optimization and preserves the expressiveness of the transient field. Our main contributions are as follows (see Fig. 1):

- **Robust Static Mapping in Dynamic Scenes,** we present a NeRF [1]-based framework achieving reliable static scene reconstruction under challenging dynamic environments, where only limited valid supervisory signals are available.
- **Comprehensive Evaluation with Foundational Localization Methods,** we validate the effectiveness of our approach on both public and in-house datasets, and further

Received 29 June 2025; revised 17 September 2025; accepted 11 October 2025. This article was recommended for publication by Editor Abhinav Valada upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant (RS-2022-II220448), the National Research Foundation of Korea (NRF) grants (RS-2025-02217000 and RS-2025-24803365) funded by the Korea government (MSIT), and the Smart Manufacturing Innovation R&D Program (RS-2024-00448642) funded by the Ministry of SMEs and Startups. *(Corresponding author: Younggun Cho.)*

Juhui Lee, Geonmo Yang, and Younggun Cho are with the Department of Electrical and Computer Engineering, Inha University, Incheon, South Korea (e-mail: dlwngml6635@gmail.com, ygm7422@inha.edu, yg.cho@inha.ac.kr).

Seungjun Ma is with Hyundai Motor Company, Uiwang, South Korea (e-mail: sj.ma@hyundai.com).

Our supplementary materials and code are available at https://sparolab.github.io/research/staticnerf/.

Digital Object Identifier (DOI)

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

assess the reconstructed maps by incorporating them into several representative localization frameworks.

- **Efficiency and Reproducibility,** our approach is designed to be memory-efficient and relatively fast to train. To facilitate verification and further research, we make our implementation publicly available.

II. Related Works

A. Novel View Synthesis in Dynamic Scenes

Neural rendering methods have focused on reconstructing static scenes in dynamic environments. Several NeRF [1]-based approaches decomposed scenes into static and dynamic components, with the dynamic content modeled via uncertainty [5] or time-conditioned features [6]. Further improvements in uncertainty estimation were achieved by incorporating DINOv2 [18] features, as demonstrated in [8], while adaptive loss reweighting was explored via Iteratively Reweighted Least Squares (IRLS) [7]. More recently, GS [19]-based methods have also tackled this challenge by using CNN-based architectures to learn visibility masks, often leveraging pretrained networks [9–11]. To further enhance separation, DINOv2 [18] features were adopted [12, 20], with training stability improved through an adaptive densification strategy [20]. Residual-based masking has also been augmented with auxiliary cues, either from SAM [21] features [22] or from text-to-image diffusion features [13]. Alternatively, other works have taken a representation-level direction without pretrained networks, focusing instead on opacity modulation [14] or explicit static–dynamic decomposition [23, 24]. While many prior efforts have focused on modeling dynamic content, often through the use of pretrained representations, they overlook a more fundamental question: *"Why is it challenging to fully remove dynamic objects, and which stages of the learning pipeline are responsible for this failure?"*

B. Visual Localization in NeRF

NeRF [1] has also been extended to visual localization. Gradient-based approaches [15, 25] iteratively refined the camera pose via gradient descent to minimize the photometric error. Absolute Pose Regression (APR)-based methods [26, 27] directly regressed the absolute camera pose from a query image. Monte Carlo Localization (MCL)-based approaches [17, 28] employed a particle filter framework, where odometry priors guided the proposal distribution and the observation function was defined by photometric error. Meanwhile, feature-based methods [16, 29] estimated camera pose by matching features between a neural map and the query image, followed by pose recovery using the Perspective-n-Point (PnP) algorithm [30]. More recently, a high-frequency visual odometry framework was proposed for joint mapping [31]. However, most of these methods assume static environments and exhibit significant performance degradation in the presence of dynamic objects.

III. Problem Formulation

Preliminaries. Before addressing the problem, we briefly review NeRF-W [5], which models a scene using two components: a static field s and a transient field t. We define a camera ray as

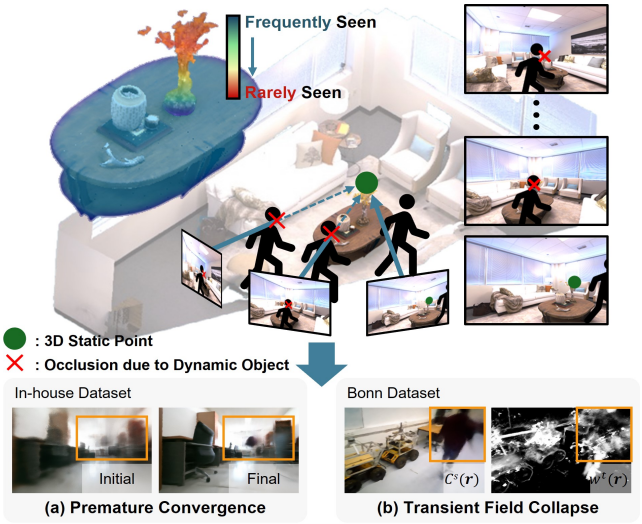

Fig. 2. **Problem Formulation.** (Top: illustration of typical failure cases observed in prior methods. Bottom: NeRF-W [5] results highlighting two training issues responsible for ineffective dynamic object removal in such scenarios.) In scenarios where static regions are rarely observed due to persistent occlusion by dynamic objects, existing methods often fail to remove dynamic content. This failure primarily stems from two learning issues: (a) **Premature Convergence**, where transient components are incorrectly and rapidly absorbed into the static field during early training, and (b) **Transient Field Collapse**, where the model gradually loses the capacity to represent dynamic regions.

$\mathbf{r}(z) = \mathbf{o} + z\mathbf{d}$, where z denotes the depth along the ray, $\mathbf{o}$ is the camera center, and $\mathbf{d}$ is a unit-norm viewing direction. At each of the N sampled points along the ray, each field predicts a color c_i^p and a density σ_i^p for $p \in \{s, t\}$. Additionally, the transient field estimates an uncertainty β_i. For a given ray $\mathbf{r}$, the rendering outputs of each field are computed as:

$$C^p(\mathbf{r}) = \sum_{i=1}^{N} w_i^p c_i^p, \quad \beta(\mathbf{r}) = \sum_{i=1}^{N} w_i^t \beta_i, \quad p \in \{s, t\}, \tag{1}$$

$$\text{where} \quad w_i^p = T_i\, \alpha_i^p, \quad T_i = \exp\left(- \sum_{j=1}^{i-1} (\sigma_j^s + \sigma_j^t)\delta_j \right). \tag{2}$$

Here, w_i^p denotes the sample weight at the i-th sample, $\alpha_i^p = 1 - \exp(-\sigma_i^p \delta_i)$ denotes the opacity, and $\delta_i = z_{i+1} - z_i$ represents the distance between two consecutive samples. Accordingly, the final output $\hat{C}(\mathbf{r})$ is obtained by summing the contributions from both the static and transient fields:

$$\hat{C}(\mathbf{r}) = C^s(\mathbf{r}) + C^t(\mathbf{r}). \tag{3}$$

NeRF-W [5] models aleatoric uncertainty based on the geometric consistency of observations across views:

$$\mathcal{L}_{\text{NeRF-W}}(\mathbf{r}) = \underbrace{\overbrace{\frac{\left\| C(\mathbf{r}) - \hat{C}(\mathbf{r}) \right\|_2^2}{2\beta(\mathbf{r})^2}}^{\text{Reconstruction Loss}} + \log \beta(\mathbf{r})}_{\text{Aleatoric Uncertainty Loss}} + \underbrace{\frac{\lambda_u}{N} \sum_{i=1}^{N} \sigma_i^t}_{\text{Regularization Loss}}, \tag{4}$$

where the last term applies regularization, scaled by a coefficient $\lambda_u = 0.01$. Static regions exhibit consistent visibility across multiple views, yielding low reconstruction errors. In contrast, dynamic objects cause inconsistent observations and

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

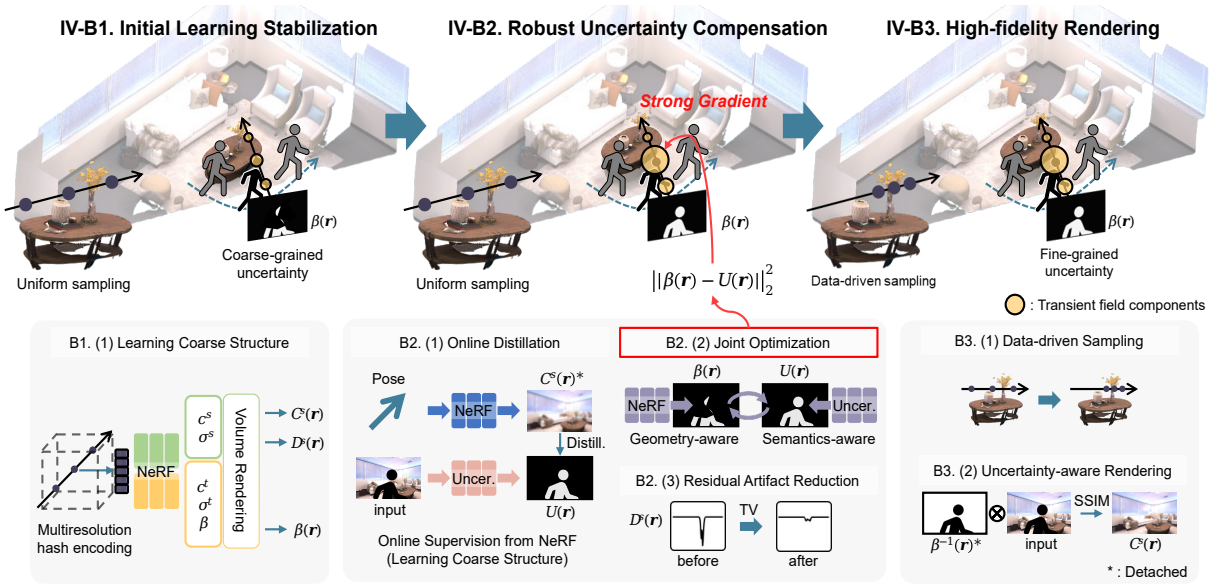

Fig. 3. **Overview of our proposed static neural rendering pipeline.** The top row illustrates stage-wise scene evolution during training, and the bottom row presents the corresponding method components aligned with each stage. Our framework follows a curriculum learning strategy consisting of three functional stages: (1) **Initial Training** involves learning coarse geometric and photometric representations through uniform sampling. (2) **Uncertainty Compensation** incorporates a CNN-based uncertainty network to address ambiguities that NeRF alone cannot resolve, enabling better disentanglement of static and transient fields through joint optimization. (3) **High-fidelity Rendering** adopts a data-driven sampling strategy and focuses on high-quality rendering.

high reconstruction errors. These high-error regions are thus modeled with greater uncertainty, which in turn down-weights their contribution to the reconstruction loss during optimization.

NeRF-W [5] relies on the visibility of static regions across most views, and fails when this assumption is violated by dominant or persistent dynamic content (see Fig. 2). Static points show consistent appearances across views, with minor variations explained by view-dependent effects, whereas dynamic objects exhibit irregular patterns. This distinction should allow the model to separate static and dynamic content even with limited supervision, and failure to do so suggests that the issue lies in the learning process rather than in the data. To achieve high-fidelity rendering while disentangling static and transient components, the optimization process inherently encourages the transient field to diminish progressively throughout training. Therefore, under this optimization trajectory with sparse supervision, we identify two inherent problems, formulated as follows:

*Problem 1 (**Premature convergence**):* In the early stage of training, transient objects are incorrectly absorbed into the static field. A key cause of this issue is the early use of data-driven sampling, which allocates more samples to high-density regions. Before the static and transient fields are properly separated, this strategy concentrates samples on dynamic content. As a result, overly fast optimization leads to underestimated reconstruction loss in dynamic regions, producing overconfident uncertainty estimates and reinforcing premature convergence.

*Problem 2 (**Transient field collapse**):* Another issue arises even with proper initialization: the transient field gradually diminishes throughout training, a trend that is further exacerbated by the regularization term in Eq. (4). Since the expressiveness of the transient field is governed by the transient weights w_i^t, this attenuation highlights the need for a compensatory mechanism to preserve meaningful transient representations.

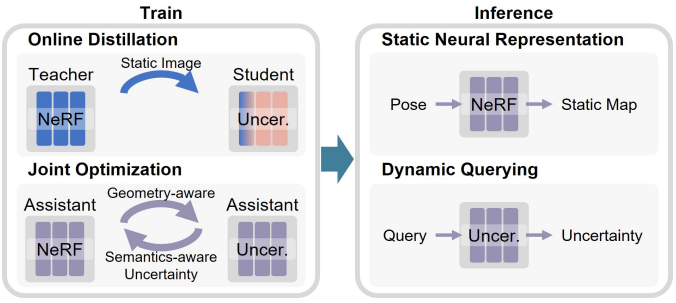

Fig. 4. **Overview of our integrated system designed for robustness in dynamic environments.** The uncertainty network is initially trained via online distillation from a NeRF teacher and jointly optimized with NeRF to enhance transient suppression. At inference time, the uncertainty network identifies dynamic objects in the query image as high-uncertainty regions, contributing to improved performance when integrated with localization.

IV. StaticNeRF: Robust, Fast, and Efficient Neural Rendering in Real-World Dynamic Scenes

We propose **StaticNeRF**, as illustrated in Fig. 3, an approach that effectively removes transient content in diverse real-world dynamic environments. To support real-time robotics applications, our model adopts multiresolution hash encoding [32], enabling a compact architecture that significantly accelerates training. Inspired by NeRF-W [5], we decompose the scene into static and transient fields and optimize the model using an aleatoric uncertainty loss in Eq. (4). Here, we learn image-specific appearance embeddings to account for illumination variations. Each embedding vector serves as an additional input to the network that extracts the static color c_i^s.

A. Integrated System for Dynamic Environments

As shown in Fig. 4, we integrate a CNN-based uncertainty network into our framework to enhance robustness, serving two

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

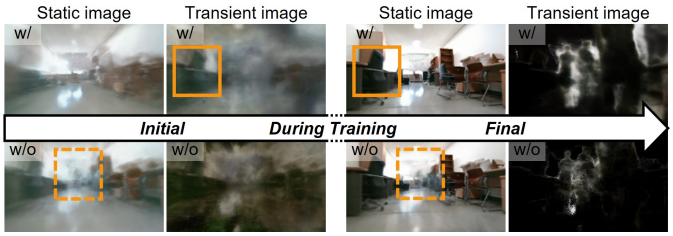

Fig. 5. **Effect of Gaussian noise injection on premature convergence.** Solid boxes represent the static object that migrated from the transient to the static field, whereas dashed boxes indicate dynamic objects that failed to separate from the static field. Without noise injection, prematurely converged transient content remains throughout training, hindering proper separation.

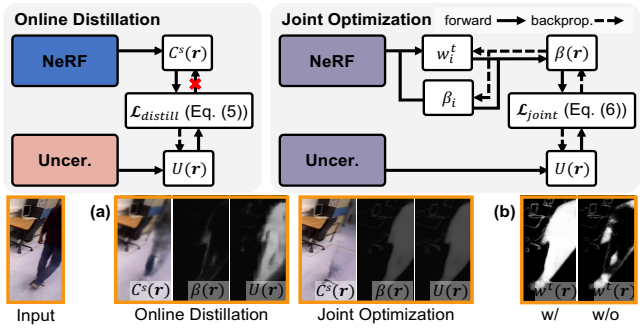

Fig. 6. **Illustration of gradient flow and the effect of joint optimization.** Online distillation supervises the uncertainty network solely based on NeRF predictions, whereas joint optimization introduces bi-directional feedback, enabling semantics-aware uncertainty to refine the NeRF representation. (a) Visual comparison of network outputs after online distillation and joint optimization. (b) Comparison of $w^t(\mathbf{r})$ with and without joint optimization, demonstrating its effectiveness in preserving transient signals.

primary purposes. Note that this is not a pretrained model but a lightweight CNN (approx. 8MB) with 10 convolutional layers, designed specifically for our task. First, it detects dynamic objects in the query image during localization by modeling aleatoric uncertainty. In the absence of explicit Ground Truth (GT) annotations indicating dynamic objects, we adopt a distillation strategy in which the NeRF, leveraging its strong geometric and photometric modeling capacity, serves as a teacher. The CNN is trained to capture discrepancies between the input and the NeRF-predicted static image $C^s(\mathbf{r})$. Through this training, the network learns to output high uncertainty in regions that deviate from the static map at inference time. Second, the uncertainty network compensates for dynamic regions that NeRF alone fails to model. As highlighted in [33], CNN-based architectures exhibit strong capabilities in local feature modeling and generalization. Building upon these strengths, our uncertainty network is trained with two consistency objectives: (i) intra-image consistency, by assigning coherent uncertainty values to pixels that belong to the same object within a single image, and (ii) cross-view consistency, by generating similar uncertainty responses for the same object observed from different viewpoints across the dataset. These dual consistencies stabilize uncertainty estimation across space and view, enabling the CNN to capture semantics-aware cues. This semantic reasoning complements NeRF's geometry-aware, point-wise uncertainty and effectively mitigates its limitations under challenging dynamic conditions.

B. Curriculum Learning

As discussed in Sec. III (Problem Formulation), establishing stable learning in the early stages of training is crucial. Equally important is preserving the representational capacity of the transient field, allowing it to continuously capture transient objects throughout training. To this end, we adopt curriculum learning structured into three sequential phases. In the initial phase (Sec. IV-B1), the model focuses on learning the coarse structure of the scene while preventing the premature convergence of transient objects into the static field. As training progresses into the mid phase (Sec. IV-B2), we incorporate a CNN-based uncertainty network as a compensation mechanism to stabilize uncertainty estimation. Finally, in the last phase (Sec. IV-B3), the model aims to enhance rendering quality based on disentangled fields and stabilized uncertainty estimates.

1) Initial Learning Stabilization: To mitigate the *Problem 1*, we initially adopt a uniform sampling strategy. However, this alone is insufficient to fully resolve the issue. While prior work has focused on the regularization effects of noise in hidden activations [34], we observed that injecting Gaussian noise directly into the density fields also helps stabilize training and mitigate premature convergence. As shown in Fig. 5, adding Gaussian noise may initially cause some static objects to be partially assigned to the transient field. However, they gradually return to the static field as training progresses, due to the inherent optimization behavior of NeRF and the regularization term in the transient density, as defined in Eq. (4). In contrast, without Gaussian noise, transient objects mistakenly absorbed into the static field during early training tend to remain entangled in the static representation even in the later training stages.

2) Robust Uncertainty Compensation: In the previous step (Sec. IV-B1), NeRF captures the coarse structure of the scene, thereby acquiring sufficient representational capacity to serve as a teacher for supervising the subsequent module.

Online Distillation. As illustrated in Fig. 4, the CNN-based network is trained as a student model to predict the aleatoric uncertainty, according to the following equation:

$$\mathcal{L}_{\text{distill}}(\mathbf{r}) = \frac{\|C(\mathbf{r}) * B_{11\times 11} - C^s(\mathbf{r})\|_2^2}{2U(\mathbf{r})^2} + \log U(\mathbf{r}), \quad (5)$$

where $U(\mathbf{r})$ denotes the uncertainty predicted by the CNN-based network at the pixel corresponding to ray $\mathbf{r}$. It learns to model the discrepancy between the input image $C(\mathbf{r})$ and the static image $C^s(\mathbf{r})$ predicted by the NeRF teacher network. However, since the teacher network is trained to capture coarse geometry, it lacks the ability to represent high-frequency details. To address this, a box filter $B_{11\times 11}$ is applied to the input image, encouraging the network to associate uncertainty primarily with structural discrepancies and preventing fine-grained textures from being misinterpreted as uncertain regions.

Joint Optimization. As discussed in the *Problem 2*, accurately predicting transient weights w_i^t is crucial for reconstructing a clean static image. Therefore, we propose joint optimization of a CNN-based uncertainty network and NeRF as follows:

$$\mathcal{L}_{\text{joint}}(\mathbf{r}) = \|\beta(\mathbf{r}) - U(\mathbf{r})\|_2^2. \quad (6)$$

This strategy is effective from two perspectives. First, as discussed in Section IV-A, the CNN-based semantics-aware uncertainty complements the NeRF-based geometry-aware uncer-

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1 Curriculum Learning

Ensure: Optimized NeRF θ^*_{NeRF}, Uncertainty Network $\theta^*_{Uncer.}$.
Initialize $\theta_{NeRF}, \theta_{Uncer.}$.
for training progress $\tau = 0\%$ to 100% **do**
  $\lambda_1 \leftarrow \lambda_{\text{NeRF-W}}$ ▷ Step 1: Initial Training
  $\mathcal{L}_1 \leftarrow \lambda_{\text{NeRF-W}} \cdot \mathcal{L}_{\text{NeRF-W}}$, $\mathcal{L} \leftarrow \mathcal{L}_1$
  if $\tau \geq 25\%$ **then** ▷ Step 2.1: Online Distillation
    $\lambda_2 \leftarrow \lambda_1 + \lambda_{\text{distill}}$
    $\lambda_{\text{distill}} \leftarrow \frac{\lambda_1}{\lambda_2} \cdot \lambda_{\text{distill}}$
    $\mathcal{L}_2 \leftarrow \mathcal{L}_1 + \lambda_{\text{distill}} \cdot \mathcal{L}_{\text{distill}}$, $\mathcal{L} \leftarrow \mathcal{L}_2$
  else if $\tau \geq 30\%$ **then** ▷ Step 2.2: Joint Optimization
    $\lambda_3 \leftarrow \lambda_2 + \lambda_{\text{joint}}$
    $\lambda_{\text{distill, joint}} \leftarrow \frac{\lambda_2}{\lambda_3} \cdot \lambda_{\text{distill, joint}}$
    $\mathcal{L}_3 \leftarrow \mathcal{L}_2 + \lambda_{\text{joint}} \cdot \mathcal{L}_{\text{joint}}$, $\mathcal{L} \leftarrow \mathcal{L}_3$
  else if $\tau \geq 40\%$ **then** ▷ Step 2.3: Residual Artifact Reduction
    $\lambda_4 \leftarrow \lambda_3 + \lambda_{\text{TV}}$
    $\lambda_{\text{distill, joint, TV}} \leftarrow \frac{\lambda_3}{\lambda_4} \cdot \lambda_{\text{distill, joint, TV}}$
    $\mathcal{L}_4 \leftarrow \mathcal{L}_3 + \lambda_{\text{TV}} \cdot \mathcal{L}_{\text{TV}}$, $\mathcal{L} \leftarrow \mathcal{L}_4$
  else if $\tau \geq 60\%$ **then** ▷ Step 3: High-fidelity Rendering
    $\lambda_5 \leftarrow \lambda_4 + \lambda_{\text{SSIM}} + \lambda_{\text{prop}}$
    $\lambda_{\text{distill, joint, TV, SSIM, prop}} \leftarrow \frac{\lambda_4}{\lambda_5} \cdot \lambda_{\text{distill, joint, TV, SSIM, prop}}$
    $\mathcal{L}_5 \leftarrow \mathcal{L}_4 + \lambda_{\text{SSIM}} \cdot \mathcal{L}_{\text{SSIM}} + \lambda_{\text{prop}} \cdot \mathcal{L}_{\text{prop}}$, $\mathcal{L} \leftarrow \mathcal{L}_5$
  end if
  Update $\theta_{NeRF}, \theta_{Uncer.}$ using $\mathcal{L}$
end for
return $\theta^*_{NeRF}, \theta^*_{Uncer.}$

tainty. This synergy enables the estimation of regions that are otherwise difficult to capture with NeRF alone. Second, it helps maintain sufficient representational capacity in the transient field. From Eq. (1), the uncertainty term $\beta(\mathbf{r})$ is defined as a function of the transient weights w_t^i. Therefore, by the chain rule, improving uncertainty estimation indirectly contributes to better transient weights optimization (see Fig. 6) as follows:

$$w^t(\mathbf{r}) = \sum_{i=1}^{N} w_i^t, \quad \frac{\partial \mathcal{L}_{\text{joint}}(\mathbf{r})}{\partial w^t(\mathbf{r})} = \frac{\partial \mathcal{L}_{\text{joint}}(\mathbf{r})}{\partial \beta(\mathbf{r})} \cdot \frac{\partial \beta(\mathbf{r})}{\partial w^t(\mathbf{r})}. \tag{7}$$

Residual Artifact Reduction. Although joint optimization mitigates most dynamic artifacts, some residual signals may remain in the static field. To further suppress these artifacts, we apply a Total Variation (TV) loss to the predicted static depth $D^s(\mathbf{r})$. Specifically, the loss is computed over 11×11 patches, using a dilation rate of 4 pixels.

3) High-Fidelity Rendering: In the final stage, we focus on enhancing rendering quality by leveraging disentangled neural fields and the accurately estimated uncertainty from previous steps. To capture finer details, we adopt a data-driven sampling strategy [35], which replaces the earlier uniform sampling with a more targeted distribution. Additionally, to compensate for the smoothing effect introduced by the TV loss and to further improve geometric accuracy, we introduce an uncertainty-aware Structural Similarity Index (SSIM) loss as follows:

$$\mathcal{L}_{\text{SSIM}}(\mathbf{P}) = \frac{1}{\beta(\mathbf{P})} \left(1 - \text{SSIM}(C(\mathbf{P}), C^s(\mathbf{P}))\right), \tag{8}$$

where $\mathbf{P}$ denotes a local image patch (a set of rays $\mathbf{r}$), defined consistently with the TV loss to maintain computational efficiency. SSIM is computed before aggregation, where the inverse uncertainty term $\beta(\mathbf{P})^{-1}$ serves as a weighting factor to attenuate the impact of dynamic regions. Gradient flow through $\beta(\mathbf{P})$ is detached to prevent the loss from influencing the uncertainty estimation.

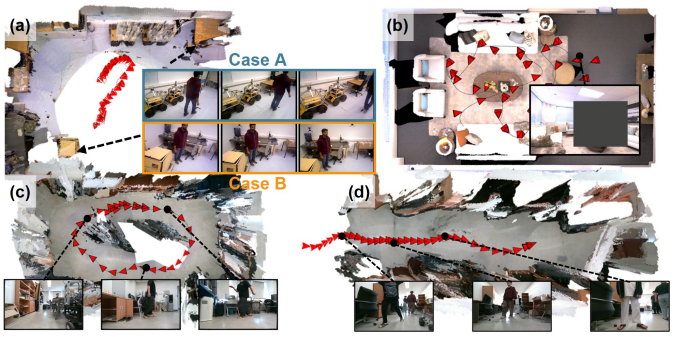

Fig. 7. **Visualization of data sequences used for evaluation.** (a) Bonn [36] dataset, where dynamic objects exhibit limited motion compared to the camera, resulting in sparse observations of the static background (covering Case A and B as shown in Fig. 1). (b) Replica [37] dataset, augmented with large, randomly placed masking boxes to simulate occlusions. (c, d) In-house indoor datasets collected in *lab* and *storeroom* environments.

C. Training Objective

We adopt a unified curriculum learning strategy (outlined in Algorithm 1), enabling consistent application across datasets without customization. As training advances, additional loss terms are incrementally introduced to guide the model toward increasingly complex objectives. In particular, once the training progress τ exceeds 60%, we incorporate the proposal loss $\mathcal{L}_{\text{prop}}$ to initiate training of the proposal network [35], thereby transitioning from uniform to data-driven sampling. To ensure that earlier objectives remain influential, we assign lower weights to newly introduced loss terms and set the weights as follows: $\lambda_{\text{distill}} = 1$, $\lambda_{\text{joint}} = 1$, $\lambda_{\text{TV}} = 0.01$, $\lambda_{\text{SSIM}} = 10$, and $\lambda_{\text{prop}} = 1$.

V. Experimental Results

This section begins with the experimental setup (Sec. V-A) and then evaluates the proposed static neural representation (Sec. V-B). To further evaluate the quality of our map, we employ multiple localization methods (Sec. V-C). Finally, we present an ablation study that analyzes the contribution of each key component in our neural rendering pipeline (Sec. V-D).

A. Experimental Setup

1) Datasets: We evaluated our method on both public and in-house datasets, as illustrated in Fig. 7. For public benchmarks, we used the Replica dataset [37] (*room0*, *office0*), the On-the-go dataset [8] (*corner*, *patio_high*), the Wild-SLAM dataset [10] (*table_tracking1*, *crowd*), and the Bonn dataset [36] (*person_tracking*, *person_tracking2*, *crowd2*, *crowd3*). In addition, we collected in-house datasets using a real-world mobile robot, comprising eight sequences: four in a corridor-like *storeroom* and four in a *laboratory*. In the Replica dataset, transient occluders were simulated by overlaying randomly positioned and colored square patches on the training images.

2) Implementation Details: All models were trained on an NVIDIA RTX A6000 GPU, and localization experiments were conducted on an NVIDIA RTX 3090 GPU. The proposed method was trained with a batch size of 1024.

B. Static Neural Representations

1) Setup: The proposed method aims to remove moving objects while modeling per-image illumination. To evaluate

Fig. 8. **Qualitative results corresponding to Table I-(i) are shown using scenes from On-the-go [8] (O), Wild-SLAM [38] (W), Bonn [36] (B), and our in-house datasets (I).** Enlarged regions, marked by orange boxes, are provided for visual clarity. In comparison, our method consistently removes dynamic objects and reconstructs clean static backgrounds, whereas baseline methods [5, 7–9, 12, 38] often retain residual artifacts.

TABLE I
QUANTITATIVE EVALUATION OF NOVEL VIEW SYNTHESIS QUALITY USING LPIPS(↓), SSIM(↑), AND PSNR(↑).

Method	Replica [37] + Occluders			On-the-go [8]			Wild-SLAM [38]			Bonn [36]			In-house: Storeroom			In-house: Lab		
	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR
(i) Evaluation of dynamic object removal and rendering quality under consistent illumination conditions.																		
NeRF-W [5]	0.357	0.754	**24.11**	0.449	0.527	21.18	0.311	0.733	22.47	0.322	0.800	21.00	0.343	**0.815**	**22.99**	0.399	0.790	**21.44**
Robust [7]	0.446	0.642	17.29	0.547	0.455	18.60	0.310	0.736	22.68	0.383	0.781	19.44	0.502	0.717	16.03	0.493	0.709	15.27
On-the-go [8]	**0.231**	**0.847**	**27.63**	0.331	0.599	**22.77**	0.236	0.757	23.94	0.273	0.821	**21.33**	0.331	0.812	**22.52**	0.370	0.788	**21.43**
GS-W [9]	0.244	0.832	22.85	**0.251**	**0.672**	**25.24**	**0.201**	**0.799**	**24.06**	**0.249**	**0.854**	**23.30**	**0.262**	**0.854**	20.34	**0.291**	**0.808**	20.98
WG [12]	**0.207**	**0.861**	22.43	**0.305**	**0.621**	22.41	**0.212**	**0.772**	**24.22**	**0.268**	**0.826**	19.79	0.344	0.806	17.43	**0.293**	**0.810**	18.61
WG-SLAM [38]	–	–	–	–	–	–	0.236	0.719	20.08	0.270	0.751	20.41	**0.269**	0.652	17.44	0.323	0.787	17.77
Ours	**0.224**	**0.914**	**28.90**	**0.272**	**0.646**	**23.58**	**0.185**	**0.788**	**25.36**	**0.219**	**0.840**	**23.64**	**0.263**	**0.833**	**23.60**	**0.298**	**0.814**	**21.84**
(ii) Evaluation of dynamic object removal and rendering quality under varying illumination, enabled by learned appearance embeddings.																		
NeRF-W [5]	0.471	0.663	**17.65**	0.486	0.502	21.06	0.310	**0.737**	**23.60**	0.342	0.795	19.81	**0.366**	**0.799**	**21.06**	0.421	**0.778**	**18.50**
GS-W [9]	**0.381**	**0.705**	15.36	**0.271**	**0.661**	**24.76**	**0.228**	**0.784**	**23.75**	**0.242**	**0.864**	**23.54**	**0.293**	**0.842**	**20.00**	**0.309**	**0.806**	**20.68**
WG [12]	**0.408**	**0.673**	**16.38**	**0.383**	**0.570**	**21.08**	**0.295**	0.723	19.64	**0.257**	**0.836**	**19.87**	0.384	0.776	15.16	**0.366**	0.756	15.05
Ours	**0.219**	**0.915**	**28.90**	**0.271**	**0.650**	**23.51**	**0.186**	**0.786**	**25.17**	**0.198**	**0.840**	**25.62**	**0.261**	**0.835**	**23.69**	**0.313**	**0.808**	**21.74**

The **best**, **second-best**, and **third-best** performing methods are highlighted. Our method significantly outperforms all baselines.

TABLE II
MEMORY AND TRAINING TIME COMPARISON OF DIFFERENT METHODS.

Method	**Ours**	NeRF-W [5]	Robust [7]	On-the-go [8]	GS-W [9]	WG [12]
Memory [MB]	13.4	4.94	34.3	34.4	138	259
Training Time	38m	3h 15m	7h 3m	7h 48m	41m	27m

these capabilities, we performed two experiments, as shown in Table I: dynamic object removal (i) under consistent illumination conditions, and (ii) under varying illumination conditions using learned appearance embeddings. The second condition introduces a more challenging task, where varying illumination complicates the removal of dynamic content due to its entanglement with appearance features. To simulate varying illumination, three brightness levels (1.00, 0.75, 0.25) were applied cyclically across frames, except for the Bonn datasets.

2) Baseline Approaches: We compared our method against several baselines, including NeRF-W [5], RobustNeRF [7], NeRF On-the-go [8], GS-W [9], and WildGaussians [12]. Since our evaluation spans diverse sequential data, we incorporated WildGS-SLAM [38] and trained all methods on identical image sets to ensure a fair comparison. All baselines were included in the first experiment, whereas only appearance-embedding-based methods [5, 9, 12] were considered in the second experiment.

3) Evaluation Protocol: As shown in Table I, we report the average performance across sequences for each dataset using PSNR, SSIM [39], and LPIPS [40] as evaluation metrics. For the Replica dataset, all pixels with static GT were used, whereas for the other datasets, only static regions were considered with explicit dynamic masks [41].

4) Results: As shown in Fig. 8 and Table I, our method outperforms existing baselines by achieving robust dynamic object removal and high-fidelity rendering. As reported in Table I-(ii), especially for the Bonn dataset, learning appearance embeddings enables the model to capture illumination variations across frames, leading to improved performance under varying lighting conditions. Baselines generally perform moderately on the Replica, On-the-go, and Wild-SLAM datasets, where static backgrounds are clearly visible, but fail to generalize to other datasets. RobustNeRF [7] relies on photometric error history, making it prone to error accumulation from early misconvergence. Although pretrained features enhance uncertainty

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

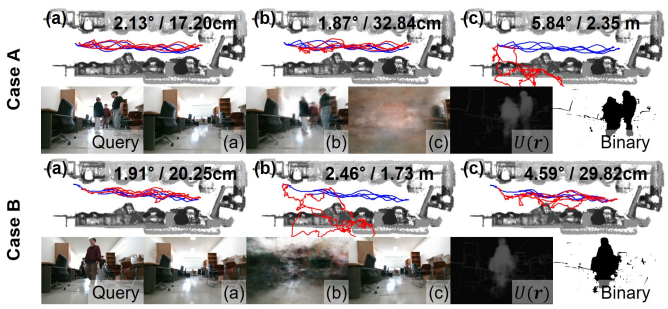

Fig. 9. **Online localization results using Loc-NeRF [17].** In Case A, localization was performed on the same sequence used for mapping, while Case B used a different sequence. We compared (a) our method, (b) the baseline, and (c) a variant without the uncertainty network. For each case, the top row visualizes the predicted (red) and reference trajectory (blue), while the bottom row shows the rendered image at the predicted pose, the estimated uncertainty $U(\mathbf{r})$, and its thresholded binary mask. Rotation and translation errors are reported in the top right corner of each result.

TABLE III
EVALUATION OF MAP QUALITY VIA LOCALIZATION PERFORMANCE (R: REPLICA, W: WILD-SLAM, B: BONN, I: IN-HOUSE).

iNeRF [15]				
Method	R-office0	W-table_tracking1	B-crowd2	B-crowd3
Ours	**1.40 (90) / 0.40 (70)**	**0.81 (75) / 0.92 (65)**	**1.40 (95) / 1.69 (35)**	**0.72 (80) / 1.67 (60)**
Baseline	1.97 (75) / 1.46 (50)	1.91 (65) / 1.11 (35)	4.02 (10) / -	2.90 (25) / -
w/o Uncer.	2.18 (75) / 0.61 (35)	1.00 (50) / 2.01 (45)	3.56 (5) / -	4.41 (10) / -
Marrying NeRF [16]				
Method	R-office0	W-table_tracking1	B-crowd2	B-crowd3
Ours	**0.29** (80) / **0.89** (70)	**0.94 (95)** / 2.92 **(60)**	**0.80** (20) / **1.61 (15)**	**0.76** (15) / **1.56** (10)
Baseline	1.24 (65) / 3.13 (10)	2.10 (70) / -	2.13 (25) / -	1.79 **(55)** / 4.65 (5)
w/o Uncer.	0.33 **(85)** / 1.54 **(80)**	1.11 (85) / **2.90** (35)	1.83 **(30)** / 2.21 (10)	0.77 (30) / 2.15 **(15)**
Loc-NeRF [17]				
Method	I-lab0	I-lab1	I-storeroom0	I-storeroom1
Ours	**2.24 / 11.0**	1.94 / **8.49**	**1.26 / 12.6**	**1.05 / 8.13**
Baseline	–	2.24 / 16.9	1.31 / 59.6	1.32 / 23.8
w/o Uncer.	–	**1.87** / 8.87	1.53 / 15.3	1.34 / 9.77

Values represent rotation and translation errors (degree / cm), with parentheses indicating success rate (%).

generalization, On-the-go [8] and GS-based methods [9, 12] remain tightly coupled with uncertainty estimation. WildGS-SLAM [38] inherits the limitation of WG [12] in handling dynamic content, and this drawback is further exacerbated by the limited diversity of data in each training iteration. In particular, GS-based methods suffer from severe premature convergence due to aggressive optimization strategies. Ultimately, these approaches fail to overcome the fundamental limitations, resulting in suboptimal performance across diverse environments.

Table II presents the memory consumption and training time. Explicit representation methods [9, 12] incur high memory usage, whereas our implicit approach achieves higher efficiency and reduced training time.

C. Evaluation via Localization Tasks

1) Setup: To evaluate our mapping pipeline, we conducted evaluations using localization tasks under three different settings: **(i) Ours**, using a static neural map with an uncertainty module; **(ii) Baseline**, the original localization method without handling dynamic content; **(iii) w/o Uncer.**, the same static neural map without the uncertainty module.

2) Baseline Approaches: We selected three representative localization baselines. iNeRF [15] performs gradient-based optimization using photometric error, which is well-suited to dense neural representations [1, 19]. Marrying NeRF [16] employs

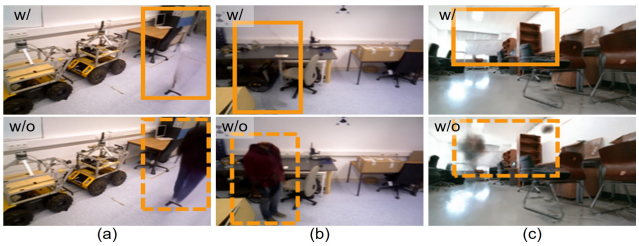

Fig. 10. **Qualitative ablation results on key components.** (Solid boxes: w/, Dashed boxes: w/o.) We present results on three representative sequences, each selected to clearly highlight the effect of (a) joint optimization, (b) sampling scheduling, and (c) residual artifact reduction.

TABLE IV
QUANTITATIVE RESULTS OF THE ABLATION STUDY.

	Ours	w/o Joint optim.	w/o Sampling sche.	w/o Noise reduc.
LPIPS ↓	0.250 / 0.153	0.262 / 0.155	0.400 / 0.143	0.247 / 0.154
SSIM ↑	0.893 / 0.911	0.886 / 0.910	0.807 / 0.913	0.894 / 0.911
PSNR ↑	25.95 / 29.28	24.51 / 29.13	17.27 / 29.48	25.53 / 29.23

Results are reported as Replica (*room_0*) / Bonn (*balloon*).

feature matching followed by a PnP step, enabling us to assess whether dynamic content affects geometry-based methods. Loc-NeRF [17] incorporates a particle filter for sequential pose estimation, making it suitable for evaluation in real-time scenarios. To ensure consistency in rendering quality and inference speed, we reimplemented all three baselines with multiresolution hash encoding. iNeRF and Marrying NeRF were evaluated on the Replica, Wild-SLAM, and Bonn datasets, which satisfy their assumptions on baseline distance and viewpoint diversity, whereas Loc-NeRF was tested on our in-house datasets that provide real-time wheel odometry. The reference trajectory for evaluation was obtained using FastLIO [42], and evaluation followed the original protocols of each method.

3) Evaluation Protocol: We evaluated pose accuracy using the average rotation error (degrees) and translation error (centimeters), as reported in Table III. Following the original evaluation protocol, iNeRF and Marrying NeRF were additionally assessed with success rates (shown in parentheses).

4) Results: Table III presents localization performance on both public and in-house datasets. iNeRF [15], which depends on pixel-wise photometric consistency, is highly susceptible to dynamic content. Marrying NeRF [16] also shows degraded performance due to unstable feature matching and reduced map quality in dynamic scenes. However, without the uncertainty module, dynamic objects in the query seldom match the static map, resulting in only limited degradation. Fig. 9 illustrates the qualitative evaluation of Loc-NeRF [17] across two scenarios: *Case A*, where the same sequence (*storeroom2*) is used for both mapping and localization, and *Case B*, where different sequences (*storeroom2* for mapping and *storeroom0* for localization) are used. In *Case A*, the baseline performs relatively well due to consistent object placement, but still fails without the uncertainty module. In *Case B*, mismatched sequences lead to significant failures in both the baseline and the variant without uncertainty. In contrast, our method remains robust across all three evaluations by filtering dynamic content during mapping and handling it in query images, resulting in consistent localization accuracy.

D. Ablation Study

The effectiveness of the three key components in our neural rendering pipeline is demonstrated in Fig. 10 and Table IV. **Joint optimization** enhances the reconstruction of static fields that NeRF alone fails to recover, by incorporating semantics-aware uncertainty estimated from a CNN-based network. This additional guidance allows the system to better distinguish between static and dynamic content during optimization. **Sampling scheduling** is designed to mitigate premature convergence without sacrificing fidelity. Although applying data-driven sampling from the beginning may appear beneficial for improving visual quality, we observed that deferring it to the later stage yields comparable fidelity while avoiding instability in the early phase of training. **Residual artifact reduction** targets subtle noise that remains even after joint optimization. It effectively suppresses such residual artifacts without degrading geometric accuracy, resulting in cleaner and more consistent reconstructions.

VI. Conclusions

We introduce **StaticNeRF**, a unified framework designed for robust performance in dynamic environments, combining a novel curriculum learning strategy for static neural mapping with our proposed dynamic querying module. Specifically, we validated the effectiveness of this approach through real-world robotic navigation experiments.

Limitations. Our current uncertainty network for dynamic querying tends to make overly conservative predictions when encountering novel dynamic objects that resemble the static background. To overcome this, future work should aim to better capture moderately ambiguous regions.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 2002.

[3] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "Elasticfusion: Dense slam without a pose graph." in *Robotics: science and systems*, vol. 11, no. 3. Rome, 2015.

[4] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[5] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.

[6] T. Wu, F. Zhong, A. Tagliasacchi, F. Cole, and C. Oztireli, "D^ 2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video," *Advances in neural information processing systems*, vol. 35, pp. 32 653–32 666, 2022.

[7] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi, "Robustnerf: Ignoring distractors with robust losses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 626–20 636.

[8] W. Ren, Z. Zhu, B. Sun, J. Chen, M. Pollefeys, and S. Peng, "Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8931–8940.

[9] D. Zhang, C. Wang, W. Wang, P. Li, M. Qin, and H. Wang, "Gaussian in the wild: 3d gaussian splatting for unconstrained image collections," *arXiv preprint arXiv:2403.15704*, 2024.

[10] J. Xu, Y. Mei, and V. Patel, "Wild-gs: Real-time novel view synthesis from unconstrained photo collections," *Advances in Neural Information Processing Systems*, vol. 37, pp. 103 334–103 355, 2024.

[11] Y. Wang, J. Wang, and Y. Qi, "We-gs: An in-the-wild efficient 3d gaussian representation for unconstrained photo collections," *arXiv preprint arXiv:2406.02407*, 2024.

[12] J. Kulhanek, S. Peng, Z. Kukelova, M. Pollefeys, and T. Sattler, "Wildgaussians: 3d gaussian splatting in the wild," *arXiv preprint arXiv:2407.08447*, 2024.

[13] S. Sabour *et al.*, "Spotlesssplats: Ignoring distractors in 3d gaussian splatting," *arXiv preprint arXiv:2406.20055*, 2024.

[14] H. Dahmani, M. Bennehar, N. Piasco, L. Roldao, and D. Tsishkou, "Swag: Splatting in the wild images with appearance-conditioned gaussians," in *European Conference on Computer Vision*. Springer, 2024, pp. 325–340.

[15] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1323–1330.

[16] R. Chen, Y. Cong, and Y. Ren, "Marrying nerf with feature matching for one-step pose estimation," *arXiv preprint arXiv:2404.00891*, 2024.

[17] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-nerf: Monte carlo localization using neural radiance fields," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4018–4025.

[18] M. Oquab *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[19] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[20] C. Fu *et al.*, "Robustsplat: Decoupling densification and dynamics for transient-free 3dgs," *arXiv preprint arXiv:2506.02751*, 2025.

[21] A. Kirillov *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[22] P. Ungermann, A. Ettenhofer, M. Nießner, and B. Roessle, "Robust 3d gaussian splatting for novel view synthesis in presence of distractors," in *DAGM German Conference on Pattern Recognition*. Springer, 2024, pp. 153–167.

[23] Y. Wang, M. Klasson, M. Turkulainen, S. Wang, J. Kannala, and A. Solin, "Desplat: Decomposed gaussian splatting for distractor-free rendering," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 722–732.

[24] W. Park, M. Nam, S. Kim, S. Jo, and S. Lee, "Forestsplats: Deformable transient field for gaussian splatting in the wild," *arXiv preprint arXiv:2503.06179*, 2025.

[25] Z. Li, K. Fu, H. Wang, and M. Wang, "Pi-nerf: a partial-invertible neural radiance fields for pose estimation," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 7826–7836.

[26] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Lens: Localization enhanced by nerf synthesis," in *Conference on Robot Learning*. PMLR, 2022, pp. 1347–1356.

[27] S. Chen, Z. Wang, and V. Prisacariu, "Direct-posenet: Absolute pose regression with photometric consistency," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1175–1185.

[28] M. Kong, S. Lee, J. Lee, and E. Kim, "Fast global localization on neural radiance field," *arXiv preprint arXiv:2406.12202*, 2024.

[29] A. Moreau, N. Piasco, M. Bennehar, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Crossfire: Camera relocalization on self-supervised features from an implicit representation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 252–262.

[30] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Ep n p: An accurate o (n) solution to the p n p problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.

[31] J. Naumann, B. Xu, S. Leutenegger, and X. Zuo, "Nerf-vo: Real-time sparse visual odometry with neural radiance fields," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 7278–7285, 2024.

[32] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[33] Y. Zhao, G. Wang, C. Tang, C. Luo, W. Zeng, and Z.-J. Zha, "A battle of network structures: An empirical study of cnn, transformer, and mlp," *arXiv preprint arXiv:2108.13002*, 2021.

[34] A. Camuto, M. Willetts, U. Simsekli, S. J. Roberts, and C. C. Holmes, "Explicit regularisation in gaussian noise injections," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 603–16 614, 2020.

[35] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[36] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss, "Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7855–7862.

[37] J. Straub *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[38] J. Zheng, Z. Zhu, V. Bieri, M. Pollefeys, S. Peng, and I. Armeni, "Wildgs-slam: Monocular gaussian splatting slam in dynamic environments," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 11 461–11 471.

[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[40] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2017, pp. 2961–2969.

[42] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.