Instant Continual Learning of Neural Radiance Fields

Ryan Po Zhengyang Dong Alexander W. Bergman Gordon Wetzstein
Stanford University
{rlpo, awb, leozdong, gordonwz}@stanford.edu

Abstract

Neural radiance fields (NeRFs) have emerged as an effective method for novel-view synthesis and 3D scene reconstruction. However, conventional training methods require access to all training views during scene optimization. This assumption may be prohibitive in continual learning scenarios, where new data is acquired in a sequential manner and a continuous update of the NeRF is desired, as in automotive or remote sensing applications. When naively trained in such a continual setting, traditional scene representation frameworks suffer from catastrophic forgetting, where previously learned knowledge is corrupted after training on new data. Prior works in alleviating forgetting with NeRFs suffer from low reconstruction quality and high latency, making them impractical for real-world application. We propose a continual learning framework for training NeRFs that leverages replay-based methods combined with a hybrid explicit–implicit scene representation. Our method outperforms previous methods in reconstruction quality when trained in a continual setting, while having the additional benefit of being an order of magnitude faster.

1. Introduction

High-quality reconstruction and image-based rendering of 3D scenes is a long-standing research problem spanning the fields of computer vision [23, 36], computer graphics [7, 18], and robotics [3, 15, 41]. Recently, the introduction of Neural Radiance Fields (NeRFs) [39] has led to substantial improvements in this area through the use of differentiable 3D scene representations supervised with posed 2D images. However, NeRFs require access to all available views of the 3D scene during training, a condition that is prohibitive for automotive and remote sensing applications, among others, where data is sequentially acquired and an updated 3D scene representation should be immediately available. In such conditions, the scene representation must be trained in a continual setting, where the model is given access to a limited number of views at each stage of training, while still tasked with reconstructing the entire scene.

Figure 1. **Continual learning of NeRFs.** Conventionally, NeRFs are trained with access to all training views. However, for continual learning scenarios we must train on batches of input views without access to previously seen data (top). When trained in these settings, conventional methods suffer from catastrophic forgetting, leading to poor reconstructions (center). In contrast, our method reconstructs the entire scene with high quality (bottom).

When trained in a continual setting, NeRFs suffer from catastrophic forgetting [17], where previously learned knowledge is forgotten when trained on new incoming data. Recent work [53, 13] has shown promise in tackling catastrophic forgetting through replay-based methods. Such approaches aim to alleviate forgetting by storing information from previous tasks either explicitly or in a compressed representation, then revisiting this information during training

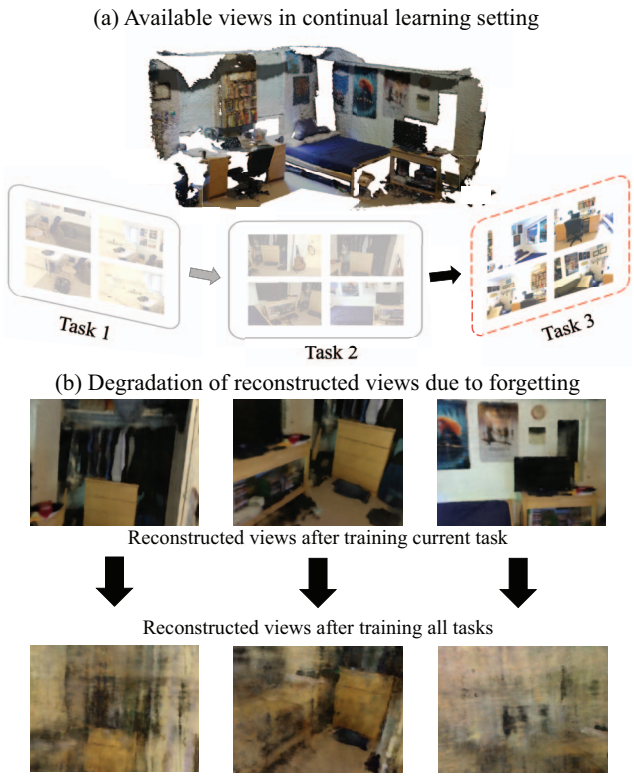

Figure 2. **Problem overview.** (a) Continual learning setting for training NeRFs. Instead of training the scene representation over all input views at once, the model is given 2D views of the scene in sequential batches. During a particular stage in training, the model is only given access to the most recently captured views. (b) Training NeRF is the continual setting leads to catastrophic forgetting. Previously learned 3D scene content is corrupted after training on newly captured views.

of subsequent tasks. Existing methods [68, 48] have seen success through the application of replay-based techniques in conjunction with NeRF for addressing the task of simultaneous mapping and localization (SLAM) [3], however such methods either suffer from memory scalability or latency issues.

In this work, we tackle the task of continually learning NeRFs by leveraging the benefits of replay-based techniques. Specifically, we acknowledge that a trained NeRF itself is a compressed representation of all previously observed 2D views. By freezing a copy of the scene representation after the training of each task, we essentially have access to pseudo ground truth RGB values for all previously seen data by querying this oracle. We also modify the underlying neural scene representation architecture motivated by one key insight: catastrophic forgetting is a fundamental problem faced by neural networks. Therefore, the fully implicit (MLP) representation used by NeRF is fundamentally ill-suited for the task of continual learning. We minimize the reliance of our underlying scene model on the decoder neural network by using a hybrid implicit–explicit representation. By replacing the frequency encoding in NeRF with a multi-resolution hash encoding [40], we greatly reduce the size of the decoder multilayer perceptron (MLP), minimizing the effects of catastrophic forgetting.

As an additional benefit, our method is also an order of magnitude faster than previous replay-based methods [13]. This enables fast continual scene fitting, as our method can learn additional 3D scene content from new input views in as little as 5 seconds (see Section 5.4 for details).

2. Related Work

Neural radiance fields. Scene representation networks [51] and neural rendering [57, 58] have emerged as a family of techniques enabling effective 3D scene reconstruction. Given a set of images and corresponding ground truth camera poses, neural radiance fields (NeRFs) [39], for example, optimizes a underlying scene representation by casting rays, sampling the scene volume and aggregating sampled color and density values to synthesize an image. The success of NeRFs has spawned a line of works on improving the quality and efficiency of the method [5, 4, 11, 20, 24, 6, 32, 33, 38, 40, 45, 55, 56, 60, 61, 64, 65, 67], while extending the method to a range of applications [62, 12, 34, 43, 19, 22, 42, 53, 68]. NeRFs leverage a neural implicit representation (NIR) [50] in the form of a simple, yet effective multi-layer pereceptron (MLP) to represent the 3D scene. Many follow-up works improve on the underlying NIR, enabling features such as real-time rendering [45, 66, 8] and faster training [40, 33, 65, 10]. A key limitation for the training of NeRFs is the assumption that all input images of the target scene are available during training. In scenarios such as autonomous vehicle or drone footage captures, this assumption no longer holds as data is sequentially acquired and an updated 3D representation should be immediately available. NeRFs trained on sequential data suffer from catastrophic forgetting [46]. Our method overcomes this limitation, providing a high quality reconstruction of the entire scene, while imparting minimal computational and memory overhead.

Continual learning. Continual learning is a long-standing problem in the field of machine learning, where partial training data is available at each stage of training. As mentioned above, NeRFs trained in a continual learning setting suffers from catastrophic forgetting [46]. Existing work in this field fall into three main categories [29]: parameter regularization [31, 59, 1, 25], parameter isolation [2, 63, 37, 16] and data replay [26, 44, 47, 49, 35, 9]. Parameter isolation methods aim at combating catastrophic forgetting by attempting to learn a sub-network for each task, while parameter regularization methods identify parameters important for preserving old knowledge and penalizing changes to them. Finally, data replay methods preserve previous knowledge by storing

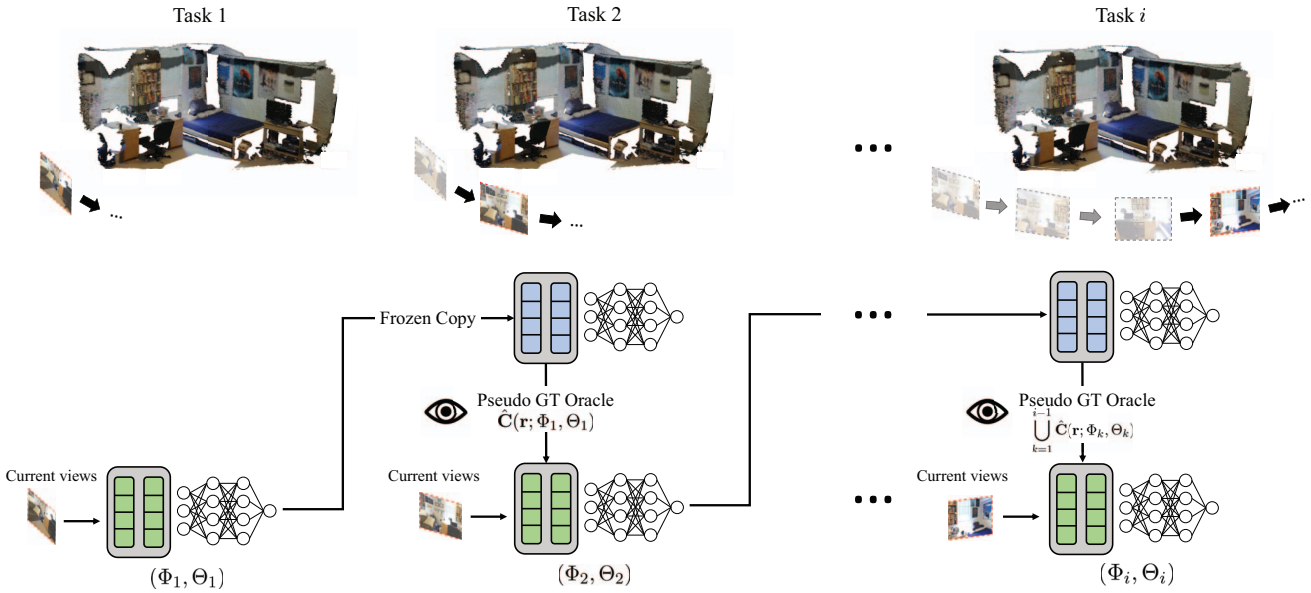

Figure 3. **Memory replay through NeRF distillation.** Scene representation is sequentially trained on sequentially acquired views. After each stage of training, a frozen copy of the scene parameters is stored. While optimizing for the next set of incoming images, the frozen network is queried to obtain pseudo ground truth values. The current network (Φ_i, Θ_i) is trained on a mixed objective that minimizes photometric loss with respect to ground truth images from the current task, and pseudo ground truth values for previous tasks (Equation 4).

a subset of previous training data. Subsequent tasks are then optimized over old and new incoming data. Our proposed method leverages a self-distillation method similar to previous data replay approaches, storing pseudo ground truth values for all previous training with minimal memory usage.

SLAM & continual learning of NeRFs. Works in the field of simultaneous mapping and localization (SLAM) [3] aim at reconstructing a 3D scene from a continuous stream of images, similar to the continual learning setting. Recent works [53, 68, 48] combine NIRs and traditional SLAM-based methods with promising results. These methods fall under the data replay category, as they approach the task of continual learning by explicitly storing key-frames from previous image streams. Storing data explicitly can be expensive, and designing an appropriate importance heuristic for selecting key-frames is non-trivial. In contrast, our approach stores previous data as an implicitly defined generator, greatly reducing memory overhead.

3. Continual Learning of NeRFs

Before we explain the details of our proposed method, it is important to first formally establish the task of continual learning of NeRFs. We consider the scenario where t sets of image data come in sequentially, represented by $\{\mathcal{I}_1, \ldots, \mathcal{I}_t\}$. Each image data set is represented by $\mathcal{I}_i = (\mathbf{I}_i, \mathbf{R}_i)$, where $\mathbf{I}_i$ represents the per-pixel RGB values of the image data and $\mathbf{R}_i$ represent the camera rays corresponding to each image pixel. Note that $\mathbf{R}_i$ can either be explicitly stored as values in $\mathbb{R}^6$ (ray origin and direction) or implicitly through camera extrinsic and intrinsic matrices.

The objective of our optimization remains the same, we wish to minimize reconstruction loss across all provided ground truth views in $\{\mathcal{I}_1, \ldots, \mathcal{I}_t\}$. However, the training procedure differs from conventional NeRF training. Training is performed sequentially as illustrated in Figure 2a. At a given stage of training, our model is only given access to a subset of all of the RGB images (visualized in Figure 2b), but access to ray information from all previous tasks. Formally, at time step i, the model is able to access $\mathbf{I}_i$ and $\{\mathbf{R}_1, \ldots, \mathbf{R}_i\}$. Note that this formulation is slightly different from prior works such as MEIL-NeRF [13] where access to ray information is also constrained. However, we believe this constraint is unwarranted since ray information can be stored implicitly for every input view with only 6 scalar values[1], assuming all input views share the same camera intrinsics. Similar to prior work [13], our method is based on self distillation [21], therefore we also assume that we have access to a frozen copy of our trained representation from the previous task.

4. Method

In this section, we first provide a brief recap behind the formulation of NeRFs [39], then introduce our solution to catastrophic forgetting in the context of training NeRFs in a continual setting. There are two main contributors to our

[1]Camera extrinsic matrices can be implicitly stored in the form $(t_x, t_y, t_z, r_x, r_y, r_z)$, where t_x, t_y, t_z represents the position of the camera optical center and r_x, r_y, r_z the orientation of the camera.

solution: namely, the use of a hybrid feature representation (Section 4.2) and task specific network distillation (Section 4.3).

4.1. NeRF preliminaries

Neural radiance fields (NeRFs) [39] represent a 3D scene through an implicit function from a point in 3D space $\mathbf{x} = (x, y, z)$ along with a corresponding viewing direction $\mathbf{d} = (\theta, \phi)$ to a density value σ and RGB color $\mathbf{c} = (r, g, b)$. Conventionally, NeRFs are represented with an MLP characterized by its parameters Θ, giving the mapping

$$F_\Theta : (\mathbf{x}, \mathbf{d}) \mapsto (\sigma, \mathbf{c}). \tag{1}$$

Novel views of the 3D scene are generated through volume rendering [28] of the 5D radiance field. Given an image pixel with the corresponding ray $\mathbf{r} = (\mathbf{r}_o, \mathbf{r_d})$, by sampling points $\mathbf{x}_i$ along this ray and evaluating the radiance field values $(\sigma_i, \mathbf{c}_i)$ at these points, the color associated with this ray can be recovered. With $\mathbf{N}$ sampled points, the RGB color of a ray $\mathbf{r}$ is obtained by

$$\hat{\mathbf{C}}(\mathbf{r}; \Theta) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \tag{2}$$

where δ_i represents the distance between the i^{th} and $(i{+}1)^{th}$ sampled point and T_i represents the accumulated transmittance from r_o to the current sample point, given by $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$.

4.2. Multi-resolution hash encoding

Prior work [40] has found success in replacing the fully implicit F_Θ with a hybrid representation, leading to faster convergence rates along with better memory and computational efficiency. Hybrid representations map 3D coordinates to an explicitly defined feature space before passing these features into a significantly smaller implicit MLP decoder to obtain density and RGB values. We leverage these explicit feature mappings to alleviate the effects of catastrophic forgetting.

Multi-resolution feature grids. Following Instant-NGP [40], we map 3D coordinates to explicit features arranged into L levels, each level containing a maximum of T features, with each feature having a dimensionality of F. Each level stores features corresponding to vertices of a 3D grid with fixed resolution. Consider a single feature level l: the queried 3D coordinate $\mathbf{x}$ is first scaled to match the native resolution of l, and the neighboring 2^3 vertices from the fixed resolution 3D grid are identified. Each vertex of interest is mapped to an entry in the l^{th} level feature array and the final feature value corresponding to $\mathbf{x}$ is obtained through tri-linear interpolation. This feature value is then passed into an implicit function represented by an MLP, mapping from feature space to density and RGB values.

Forgetting in explicit features. Consider the case where T matches the total number of vertices at each grid resolution, such that a 1:1 mapping exists between grid vertices and feature embeddings. In the continual setting, features are only updated when the corresponding voxel is visible in the training views of the current task, whereas other features remain constant, unaffected by catastrophic forgetting. This is in stark contrast to the global updates observed in fully implicit representations such as in NeRF [39]. In NeRF, each network parameter influences radiance and density values at every point in 3D space, and training on new data points overwrites information learned in the entire scene, even for regions not visible in the current training views.

Hashed feature tables. In an effort to lower memory usage at higher grid resolutions, Instant-NGP proposes a hashed encoding scheme. At finer levels, a hash function $h : \mathcal{Z}^d \mapsto \mathcal{Z}_T$ is used to index into the feature array, effectively acting as a hash table. Following prior work [40], we use a spatial hash function of the form

$$h(\mathbf{x}) = \left(\bigoplus_{i=1}^{d} x_i \pi_i \right) \bmod T, \tag{3}$$

where $\bigoplus$ represents the bit-wise XOR operator and π_i are unique, large primes.

In contrast to dense feature grids, hashed feature tables suffer from catastrophic forgetting in the feature space due to hash collisions. Consider a single task $\mathcal{I}_i$. A vertex v_1 visible in $\mathcal{I}_i$ may share the same hash table entry as another vertex v_2 that is not visible in $\mathcal{I}_i$. During training, the training objective will only optimize the shared hash table entry for the current task $\mathcal{I}_i$, learning the correct feature value for v_1, while forgetting any information learnt for v_2. The effects of forgetting are dependent on the frequency of hash collisions between grid vertices, which increases as the hash table size T decreases.

4.3. Memory replay through NeRF distillation

Catastrophic forgetting results from a misalignment between the current and cumulative training objectives. Replay-based approaches [26, 44, 47, 49, 35, 9] combat network forgetting by storing information from previous tasks either explicitly or implicitly through a generative model. Consider a NeRF with explicit feature embeddings trained on a set of tasks $\{\mathcal{I}_1, \ldots, \mathcal{I}_i\}$, with feature and MLP parameters characterized by (Φ_i, Θ_i). We can then treat (Φ_i, Θ_i) as a generator for 2D image data found in tasks $\{\mathcal{I}_1, \ldots, \mathcal{I}_i\}$. Let $\hat{\mathbf{R}}_i$ be the union of all ground truth rays in the first i tasks. The ground truth RGB value corresponding to a ray $\mathbf{r} \in \hat{\mathbf{R}}_i$ can then be approximated by $\hat{\mathbf{C}}(\mathbf{r}; \Phi_i, \Theta_i)$ following Eq. 2.

We approach continual learning in a self-distillation manner [21]. When training on the subsequent task $\mathcal{I}_{i+1}$, we no

Table 1. **Quantitative results: unconstrained setting.** PSNR of different continual learning methods. Every method is trained on each task until convergence, which differs by method. Approximate training time for all 10 tasks is listed next to each method. For each scene, we mark the best performing methods with gold ●, silver ● and bronze ● medals. Results marked with * are trained in a non-continual setting, where ground truth data from all tasks are available during scene optimization. These results serve as an upper bound for scenes trained in a continual setting. Our method consistently out-performs all baselines while taking significantly less time to converge.

	ScanNet			*Tanks & Temples*			*TUM RGB-D*	
Method	*0101*	*0146*	*0160*	*Truck*	*Caterpillar*	*Family*	*Desk 0*	*Desk 1*
NeRF-Incre (2 hours)	13.70	13.20	17.31	16.88 ●	15.36 ●	22.96 ●	13.05 ●	14.03
iNGP-Incre (10 min)	16.51 ●	16.64	19.98	13.49	14.55	21.15	12.70	14.65 ●
iNGP + EWC (10 min)	16.11	17.32 ●	20.16 ●	12.50	13.61	19.28	12.50	10.85
MEIL-NeRF (2 hours)	24.32 ●	26.82 ●	28.93 ●	22.74 ●	20.89 ●	26.57 ●	20.79 ●	19.80 ●
Ours (10 min)	25.72 ●	27.87 ●	30.28 ●	22.71 ●	22.51 ●	29.33 ●	20.65 ●	20.34 ●
NeRF* (2 hours)	26.15	28.48	30.88	24.80	23.14	29.33	22.35	20.88
iNGP* (10 min)	26.00	28.43	31.16	24.22	24.02	31.14	20.95	20.73

longer have access to ground truth image data from previous tasks. However, as explored in prior work [13], by saving network parameters (Φ_i, Θ_i) we effectively have access to pseudo ground truth values for all rays in $\hat{\mathbf{R}}_i$. We can then modify our training objective to minimize photometric loss for all rays in tasks $\{\mathcal{I}_1, \ldots, \mathcal{I}_{i+1}\}$, rather than just $\mathcal{I}_{i+1}$. The modified training objective is then given by

$$\mathcal{L}(\Phi, \Theta)_{i+1} = \sum_{\mathbf{r} \in \mathcal{I}_{i+1}} ||\hat{\mathbf{C}}(\mathbf{r}; \Phi, \Theta) - \mathbf{C}(\mathbf{r})||^2 + \sum_{\mathbf{r} \notin \mathcal{I}_{i+1}} ||\hat{\mathbf{C}}(\mathbf{r}; \Phi, \Theta) - \hat{\mathbf{C}}(\mathbf{r}; \Phi_i, \Theta_i))||^2. \tag{4}$$

During each task, we still sample rays uniformly over all previous and current tasks. However, for previous tasks where ground truth RGB values are no longer available, we instead query the frozen network to obtain a pseudo ground truth value. Figure 3 shows a visualization of the replay-based distillation method.

5. Experiments

To highlight the effectiveness of our method in overcoming catastrophic forgetting, we compare our method against existing continual learning methods [25, 13]. We describe baseline methods in Section 5.1, datasets used in Section 5.2 and experimental settings in Section 5.3.

5.1. Baselines

NeRF and iNGP. We train NeRFs under the continual setting using frequency and multi-resolution hash encodings, referring to these baselines as *NeRF-Incre* and *iNGP-Incre* respectively. For our hash encoding experiments, we used a feature grid of $L = 16$ levels, a hash table size of $T = 2^{17}$, a feature dimension of $F = 2$ and grid resolutions ranging from 16 to 512. We also scale the original NeRF representation [39] to have 8 fully connected layers with 512 channels each, matching the total number of trainable parameters as the hash encoding models.

Elastic Weight Consolidation. Elastic Weight Consolidation (EWC) [25] is a form of feature regularization method for alleviating catastrophic forgetting. Let Φ_A be the set of hashed feature embeddings learned on task $\mathcal{I}_A$. Consider a subsequent task $\mathcal{I}_B$. EWC modifies the training objective to the following:

$$\mathcal{L}(\Phi) = \mathcal{L}_B(\Phi) + \frac{\lambda}{2} F(\Phi - \Phi_A)^2. \tag{5}$$

$\mathcal{L}_B$ represents the training objective on task $\mathcal{I}_B$ and F is an estimation of the diagonal of the Fischer information matrix given by the squared gradients of parameters Φ_A with respect to the training objective $\mathcal{L}_A$. Intuitively, Φ_A is recorded as a set of reference parameters. Deviation from these reference parameters are penalized, weighted on their importance relative to the training objective. We implement EWC on top of an iNGP backbone as a baseline method by fixing the trained network parameters after each training task as the reference parameters.

MEIL-NeRF. Recently, MEIL-NeRF [13] also proposed the use of memory replay through network distillation for alleviating catastrophic forgetting effects in NeRFs. However, MEIL-NeRF uses the original fully implicit NeRF representation as a backbone, which limits reconstruction quality and convergence speed. We include continual learning results following the general implementation of MEIL-NeRF. While MEIL-NeRF uses an additional ray generator network for sampling previous rays from previous tasks, this additional step leads to significant degradation in reconstruction results while providing minimal memory savings;

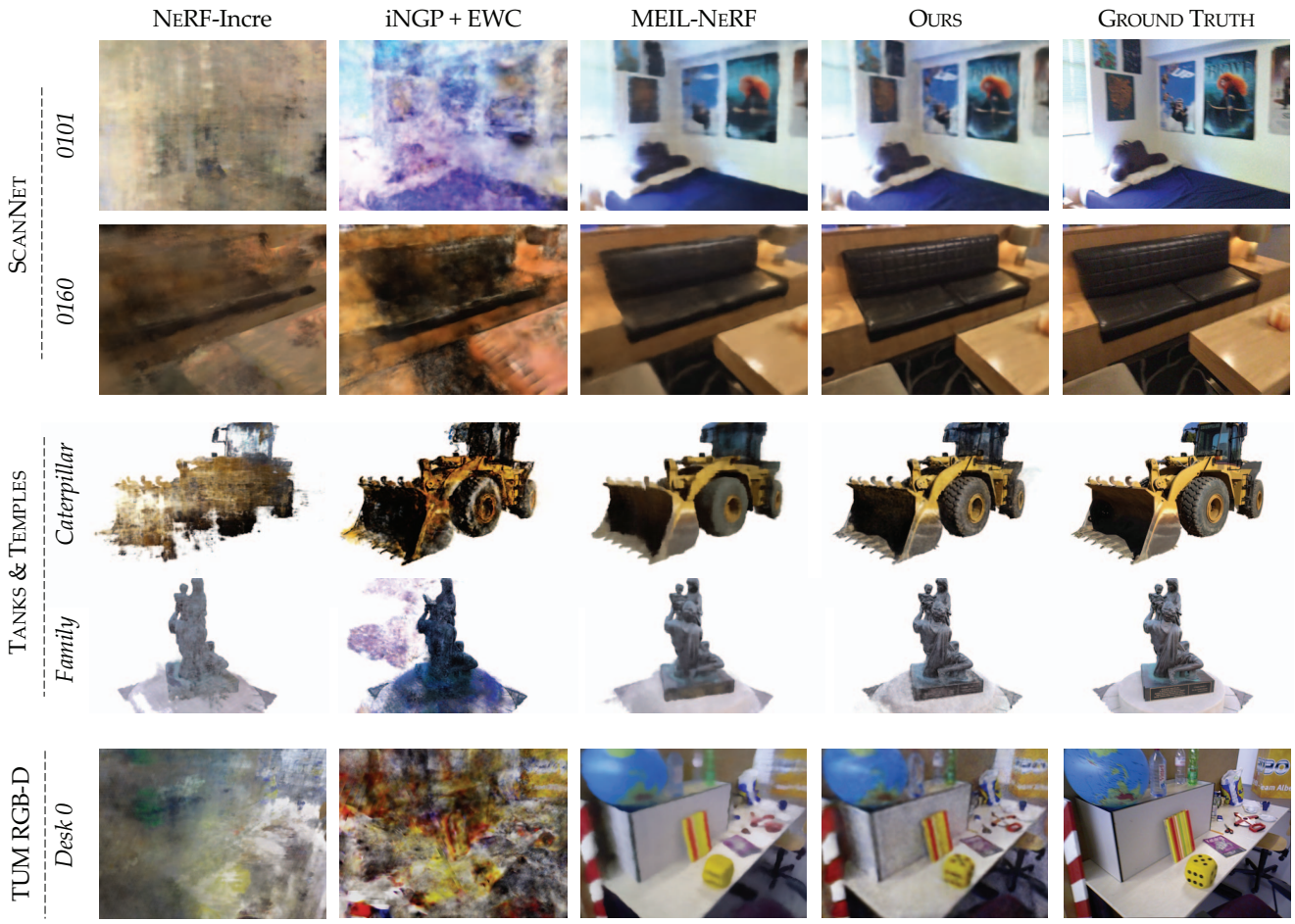

Figure 4. **Qualitative results: unconstrained setting.** We show reconstructed views from a previously supervised (forgotten) task across different methods. Our method consistently outperforms all other baselines in visual quality. NeRF trained in a continual setting suffers from catastrophic forgetting, as illustrated by poor early-task reconstruction results. Parameter regularization through EWC aids in alleviating forgetting effects, however, reconstruction results still suffer from severe visual artefacts. MEIL-NeRF adopts a similar replay approach as our method, using a frozen copy of the scene representation as guidance when training future tasks. However, the fully implicit representation in MEIL-NeRF forgets high-frequency detail from earlier tasks. In contrast, our method is able to retain high-frequency details for earlier tasks through the use of explicit features.

we therefore omit this step and sample ground truth rays instead. MEIL-NeRF also explores using Charbonnier penalty function, we consider changes to the penalty function a tangential area of exploration, and choose to train both our method and MEIL-NeRF using the loss function detailed in Equation 4.

5.2. Datasets

We compare methods on the task of continual scene fitting using the Tanks & Temples [27], ScanNet [14] and TUM RGB-D datasets [52]. Data for each scene is represented by a trajectory of ground truth camera poses and corresponding RGB images, with each trajectory containing 100–300 images depending on scene. We emulate the setting of continual learning by partitioning each trajectory into 10 temporally sequential tasks.

5.3. Experimental settings

We evaluate our method in two separate settings: an unconstrained setting where each method is trained on every task until convergence, and a constrained setting where each task is trained on a fixed time budget. The unconstrained setting aims at testing the upper-bound performance of each method, while the constrained setting mimics a real-time continual scene reconstruction setting. Each model is trained on a single RTX 6000 GPU, with a ray batch size of 1024. For the unconstrained settings, we trained methods using a hash encoding for 1 minute per task and methods built on fully implicit NeRFs for 10 minutes per task.

5.4. Results

Unconstrained setting. We show quantitative results of each method for the unconstrained setting in Table 1. Methods are evaluated using peak signal-to-noise ratio (PSNR),

	ScanNet			*Tanks & Temples*			*TUM RGB-D*	
Method	*0101*	*0146*	*0160*	*Truck*	*Caterpillar*	*Family*	*TUM 1*	*TUM 2*
Ours (1 s)	19.61	22.18	23.84	19.19	17.24	23.24	15.05	16.65
Ours (5 s)	24.10 ●	26.13 ●	28.37 ●	21.93 ●	20.59 ●	26.29 ●	19.35 ●	19.02 ●
Ours (30 s)	25.54 ●	27.84 ●	30.45 ●	23.97 ●	22.62 ●	29.21 ●	21.09 ●	20.42 ●
MEIL–NeRF (30 s)	18.85	21.41	22.72	18.11	16.93	21.78	16.05	15.96
MEIL–NeRF (1 min)	20.65	22.76	24.39	19.38	18.41	23.19	17.44	16.19
MEIL-NeRF (10 min)	24.32 ●	26.82 ●	28.93 ●	22.74 ●	20.89 ●	26.57 ●	20.79 ●	19.80 ●

Table 2. **Quantitative results: time constrained.** We show reconstruction PSNR for our method and MEIL-NeRF trained on a fixed time limit per task. Our method converges to better results at a much faster rate. Our method trained for only 5s per task outperforms MEIL-NeRF trained for 1 min per task and is competitive with MEIL-NeRF trained for 10 min per task. Given its rapid convergence, our method uniquely enables real-time continual scene reconstruction.

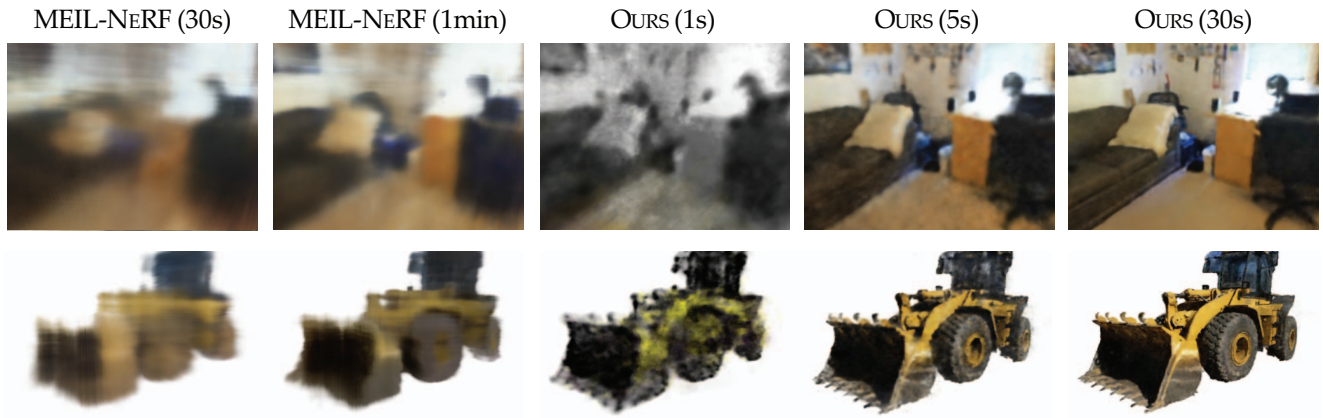

Figure 5. **Qualitative results: time constrained.** We show reconstructed views from an earlier supervised (forgotten) task for our method and MEIL-NeRF trained for fixed times per task. Our method consistently outperforms MEIL-NeRF given equal time budget. With only 5s per task, our method already reconstructs the scene with reasonable fidelity, illustrating that our method is well-suited for real-time continual scene fitting.

averaged over all images in the test trajectory. We also provide quantitative results of the fully implicit NeRF and hash-encoding representations trained in a non-continual setting. These results serve as an upper bound for their continual learning counterparts. Quantitatively, our method consistently outperforms all baselines in reconstruction quality. While performance of MEIL-NeRF comes close to our method for certain scenes, our method takes significantly less time to train due to the convergence properties of the hash encoding representation. Results from our method also come very close to the theoretical upper bound set by the results obtained from non-continual training, further illustrating the effectiveness of our method.

Figure 4 shows qualitative results from the unconstrained setting. Naively training NeRF under the continual setting leads to catastrophic forgetting, as earlier views contain heavy artefacts. Parameter regularization through EWC helps alleviate forgetting for certain scenes, however, reconstruction quality is still limited. MEIL-NeRF produces visually pleasing results, but reconstruction of earlier views lack high-frequency details. In contrast, our method is able to retain these high frequency details, as the underlying multi-resolution hash encoding stores high-frequency features explicitly, allowing high frequency details to be retained during training.

Time-constrained setting. We evaluate our method against MEIL-NeRF in the time-constrained setting. We trained both methods on each task for a fixed period of time and show reconstruction PSNR averaged over all views along the test trajectory in Table 2. Our method trained on 30 seconds per task out performs MEIL-NeRF, even when trained for 10 minutes per task. More importantly, our method trained for just 5 seconds produces results comparable to MEIL-NeRF at convergence. Qualitative results in Figure 5 show that our method provides reasonable scene reconstruction quality at much shorter training times, illustrating that our method is uniquely suited for the task of real-

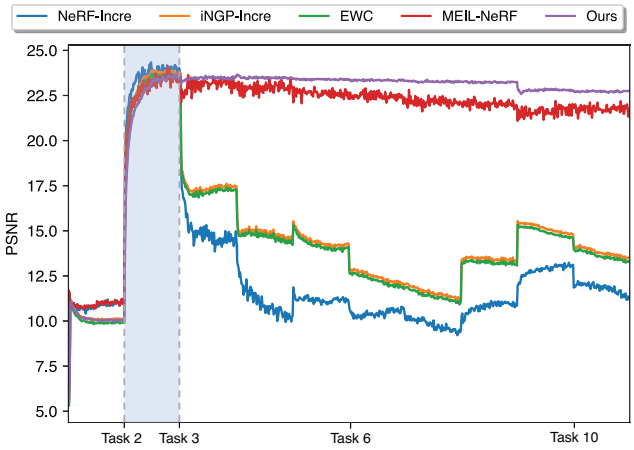

Figure 6. **Reconstruction quality of early-supervised tasks.** Reconstruction PSNR of task 2 over the course of training. Our method successfully alleviates degradation effects from catastrophic forgetting, and consistently outperforms all other baselines.

time continual scene fitting.

5.5. Ablation study

Degradation of early tasks. We evaluate reconstruction PSNR of the second task (of ten total) over the course of training using different methods in Figure 6. Conventional methods naively trained in the continual setting (*NeRF-Incre & iNGP-Incre*) experience severe degredation due to catasatrophic forgetting. MEIL-NeRF succesfully alleviates forgetting effects through self-distillation, however, forgetting effects are still observed after training for many tasks. In contrast, our method is able to maintain high PSNR for previous tasks even after training for many tasks.

5.6. Applications: autonomous vehicle data

Our method is well-suited for scenarios such as autonomous vehicle captures and drone footage, where data is sequentially acquired and an updated 3D scene representation should be immediately available. To illustrate this, we train our method on data obtained from the Waymo open dataset [54]. A single trajectory in the Waymo dataset consists of a video stream from 5 calibrated cameras mounted at the top of the vehicle. Similar to the experimental settings described in Section 5, we split each trajectory into 10 temporally sequential tasks. We show qualitative results using our method and iNGP-Incre in Figure 7, training each task for 30 seconds for a total of 5 minutes. Our method recovers meaningful geometry and reconstructs earlier views with much higher quality.

6. Discussion

Limitations and future work. Our method relies on ground truth camera poses to perform scene fitting. Although prior works have explored simultaneous optimization of camera poses and scene parameters for NeRFs, they

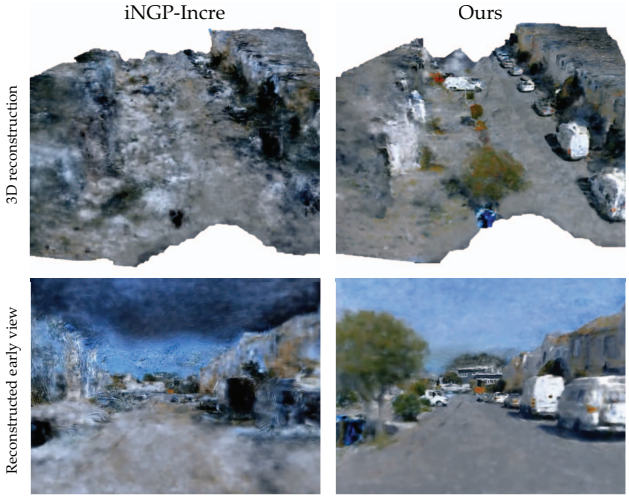

Figure 7. **Qualitative results on Waymo open dataset.** Our method recovers earlier training views at higher quality than training NeRFs naively in a continual learning setting.

either rely on good initializations [32] or specific constraints on the distribution of camera poses [30]. Simultaneous estimation of camera poses in the setting of continual learning for NeRFs has also yet to be explored. It may be fruitful to explore this direction further in relation to methods for SLAM [3].

We chose to use multi-resolution hash encodings [40] to leverage its fast convergence properties and explicitly defined features to combat forgetting. Alternate representations, such as triplanes [8] and TensoRF [10] can also be explored as potential substitutes, potentially further increasing robustness to catastrophic forgetting through more structured encodings.

Our method uses a frozen version of the scene representation network trained on previous tasks as a pseudo ground truth oracle. Querying the network for pseudo ground truth values requires volume rendering through the scene, adding computational overhead to the training process. A potential direction of exploration is to find other forms of compression, such as 2D coordinate networks [50], to act as the pseudo ground truth oracle. Additionally, if any single oracle network is not of sufficient quality, this will continue to affect downstream training on subsequent tasks.

Conclusion. In this work, we aim to extend the practical viability of NeRFs, specifically in the continual setting, where training data is sequentially captured and a 3D representation needs to be immediately available. By combining multi-resolution hash encodings and replay methods through network distillation, our approach alleviates the effects of catastrophic forgetting observed in the continual learning of NeRFs. While previous approaches struggle with quality and speed, our method is able to produce visually compelling reconstruction of earlier tasks while being an order of magnitude faster than existing methods.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. *ArXiv*, abs/1711.09601, 2017.

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7120–7129, 2016.

[3] Tim Bailey and Hugh F. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics & Automation Magazine*, 13:108–117, 2006.

[4] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5844, 2021.

[5] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2021.

[6] Alexander W. Bergman, Petr Kellnhofer, and Gordon Wetzstein. Fast training of neural lumigraph representations using meta learning. In *NeurIPS*, 2021.

[7] Fabio Bruno, Stefano Bruno, G. De Sensi, Maria Laura Luchi, Stefania Mancuso, and Maurizio Muzzupappa. From 3d reconstruction to virtual reality: A complete methodology for digital archaeological exhibition. *Journal of Cultural Heritage*, 11:42–49, 2010.

[8] Eric Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, S. Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16102–16112, 2021.

[9] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *ArXiv*, abs/1812.00420, 2018.

[10] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, 2022.

[11] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14104–14113, 2021.

[12] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, and Huchuan Lu. Animatable neural radiance fields from monocular rgb video. *ArXiv*, abs/2106.13629, 2021.

[13] Jaeyoung Chung, Kang-Ho Lee, Sungyong Baik, and Kyoung Mu Lee. Meil-nerf: Memory-efficient incremental learning of neural radiance fields. *ArXiv*, abs/2212.08328, 2022.

[14] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017.

[15] Hugh F. Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics & Automation Magazine*, 13:99–110, 2006.

[16] Chrisantha Fernando, Dylan S. Banarse, Charles Blundell, Yori Zwols, David R Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *ArXiv*, abs/1701.08734, 2017.

[17] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3:128–135, 1999.

[18] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8645–8654, 2020.

[19] Zekun Hao, Arun Mallya, Serge J. Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14052–14062, 2021.

[20] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul E. Debevec. Baking neural radiance fields for real-time view synthesis. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021.

[21] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

[22] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-nerf: Using a neural radiance field to grasp transparent objects. In *Conference on Robot Learning*, 2021.

[23] Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *ArXiv*, abs/1704.05519, 2017.

[24] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan P. Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4285–4295, 2021.

[25] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2016.

[26] Georg S. W. Klein and David William Murray. Parallel tracking and mapping for small ar workspaces. *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[27] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples. *ACM Transactions on Graphics (TOG)*, 36:1 – 13, 2017.

[28] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38:199–218, 1999.

[29] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ale*vs*. Leonardis, Gregory G. Slabaugh, and

Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3366–3385, 2019.

[30] Axel Levy, Mark J. Matthews, Matan Sela, Gordon Wetzstein, and Dmitry Lagun. Melon: Nerf with unposed images using equivalence class estimation. *ArXiv*, abs/2303.08096, 2023.

[31] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947, 2016.

[32] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5721–5731, 2021.

[33] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *ArXiv*, abs/2007.11571, 2020.

[34] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason M. Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40:1 – 13, 2021.

[35] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, 2017.

[36] Xinzhu Ma, Zhihui Wang, Haojie Li, Wanli Ouyang, and Pengbo Zhang. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6850–6859, 2019.

[37] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2017.

[38] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7206–7215, 2020.

[39] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf. *Communications of the ACM*, 65:99 – 106, 2021.

[40] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41:1 – 15, 2022.

[41] Raul Mur-Artal, José M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31:1147–1163, 2015.

[42] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5569–5579, 2021.

[43] Sida Peng, Junting Dong, Qianqian Wang, Shang-Wei Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14294–14303, 2021.

[44] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, 2016.

[45] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14315–14325, 2021.

[46] Anthony V. Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connect. Sci.*, 7:123–146, 1995.

[47] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *Neural Information Processing Systems*, 2018.

[48] Antoni Rosinol, John J. Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *ArXiv*, abs/2210.13641, 2022.

[49] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NIPS*, 2017.

[50] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *ArXiv*, abs/2006.09661, 2020.

[51] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

[52] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[53] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6209–6218, 2021.

[54] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott M. Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2451, 2019.

[55] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2845–2854, 2020.

[56] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *ArXiv*, abs/2006.10739, 2020.

[57] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020.

[58] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, volume 41, pages 703–735. Wiley Online Library, 2022.

[59] A. Triki, Rahaf Aljundi, Matthew B. Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1329–1337, 2017.

[60] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd E. Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490, 2021.

[61] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4688–4697, 2021.

[62] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 41, 2021.

[63] Ju Xu and Zhanxing Zhu. Reinforced continual learning. *ArXiv*, abs/1805.12369, 2018.

[64] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2853–2863, 2021.

[65] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5491–5500, 2021.

[66] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5732–5741, 2021.

[67] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4576–4585, 2020.

[68] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12776–12786, 2021.