Incremental Multi-Scene Modeling via Continual Neural Graphics Primitives

Prajwal Singh
singh_prajwal@iitgn.ac.in
Ashish Tiwari
ashish.tiwari@iitgn.ac.in
Gautam Vashishtha
gautam.pv@iitgn.ac.in
Shanmuganathan Raman
shanmuga@iitgn.ac.in

CVIG Lab
IIT Gandhinagar
Gujarat, India

Abstract

Neural radiance fields (NeRF) have revolutionized photorealistic rendering of novel views for 3D scenes. Despite their growing popularity and efficiency as 3D resources, NeRFs face scalability challenges due to the need for separate models per scene and the cumulative increase in training time for multiple scenes. The potential for incrementally encoding multiple 3D scenes into a single NeRF model remains largely unexplored. To address this, we introduce Continual-Neural Graphics Primitives (C-NGP), a novel continual learning framework that integrates multiple scenes incrementally into a single neural radiance field. Using a generative replay approach, C-NGP adapts to new scenes without requiring access to old data. We demonstrate that C-NGP can accommodate multiple scenes without increasing the parameter count, producing high-quality novel-view renderings on synthetic and real datasets. Notably, C-NGP models all 8 scenes from the Real-LLFF dataset together, with only a 2.2% drop in PSNR compared to vanilla NeRF, which models each scene independently. Further, C-NGP allows multiple style edits in the same network. The code implementation and dynamic visualizations can be accessed from here https://prajwalsingh.github.io/C-NGP/.

1 Introduction

Neural Radiance Fields (NeRF) synthesize photorealistic novel views by modeling 3D scenes using Multi-Layer Perceptrons (MLPs) and differentiable volumetric rendering. While powerful, NeRF and its variants typically require per-scene optimization, leading to high training and storage costs. Recent methods address speed [15, 34, 50], inference efficiency [11, 48], compression [7, 14, 17, 43, 46], scene editing [2, 52], and generalization to unseen scenes [8, 19, 51, 56, 58, 62]. However, learning multiple distinct scenes within a single NeRF model remains underexplored, with SCARF [53] being a recent attempt.

Training separate NeRFs for each scene is impractical for large-scale 3D collections due to resource demands. We propose a *continual learning* approach that incrementally

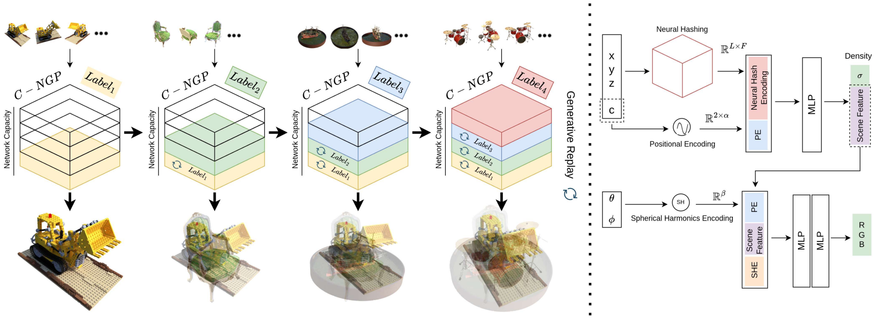

Figure 1: (Left) The proposed continual neural graphics primitives (C-NGP) framework to model multiple scenes. Every new scene is assigned a unique pseudo-label and trained sequentially while preserving the previously learned information via generative replay. Any specific scene can be rendered by conditioning C-NGP over the associated pseudo-label. (Right) Our modification to the Instant-NGP architecture [34] enables multi-scene modeling using pseudo labels.

encodes multiple scenes within a single NeRF while maintaining a fixed parameter count. Our framework learns new scenes sequentially without forgetting previous ones and without having to store past training data. Effective multi-scene modeling requires: *(a)* photorealistic rendering of all scenes, *(b)* integration of new scenes without access to old data, *(c)* prevention of catastrophic forgetting, and *(d)* constant model size. Moreover, continual learning in NeRFs is uniquely difficult. Unlike image classification, where an image seldom has different labels, NeRF requires scene-specific mappings since the same position and view direction across different scenes produce different outputs.

We introduce C-NGP - **C**ontinual **N**eural **G**raphics **P**rimitives that leverage fast training and efficient learning of Instant-NGP [34] by integrating scene conditioning into its hashing mechanism using pseudo-labels to accommodate multiple scenes into NGP's parameter space. This simple yet effective conditioning enables multi-scene optimization, allowing the model to learn new scenes while preserving high-fidelity rendering of previously encountered scenes. C-NGP uses generative-replay to accommodate new scenes without requiring access to training data from previously learned scenes. Thus, C-NGP is practical for real-world use, allowing multiple 3D scenes to be stored and shared in a single network checkpoint, where the new scenes can be added incrementally without creating separate models. Interestingly, even vanilla NeRF can be adapted to the proposed setting, but with far higher computational overhead (12 days of training time vs. 8 hours for C-NGP) and lower performance.

Our main contributions are: 1) C-NGP, the first fixed-size NeRF to continually encode multiple scenes, achieving only a 2.2% PSNR drop on Real-LLFF [32] compared to scene-specific vanilla NeRF; 2) training without access to previous data via generative replay; 3) an efficient scene-conditioning mechanism in Instant-NGP's hash grid; 4) support for multi-view consistent style editing across scenes; 5) competitive performance over continual learning baselines [6, 12, 37] and concurrent methods [53] in quality, training efficiency, and adaptability to new scenes.

2 Related Work

Single-Scene Neural Radiance Fields. NeRF encodes scene geometry and appearance but requires long per-scene optimization. Alternatives like Plenoxels [15] and Instant-NGP [34] improve efficiency using spherical harmonics and multi-resolution hash encoding. Other extensions enable faster inference [11, 48], scene editing [2, 52], or rendering from sparse images [26, 49, 57]. Another line of work aims to generalize NeRF to unseen scenes without per-scene training [6, 19, 51, 56, 58, 62]. These methods use convolutional feature extractors, 3D cost volumes, or attention modules [3, 36] to learn scene priors, enabling inference on new scenes. However, they often require fine-tuning for optimal quality and are not designed for incremental learning, suffering from catastrophic forgetting when adapting to multiple scenes in a shared parameter space. Unlike these approaches, our method focuses on continual learning to incrementally encode distinct scenes without forgetting, targeting compact 3D asset storage rather than generalization. Recently, 3D Gaussian Splatting (3DGS) methods [10, 21] have achieved real-time rendering for single scenes but do not support incremental multi-scene learning. However, these approaches continue to focus on single-scene optimization.

Continual Learning in NeRF. Continual learning enables models to learn new tasks sequentially without forgetting prior knowledge [41]. Techniques include parameter isolation [30], regularization [23, 28], and replay [40, 42]. In NeRF, applying continual learning is challenging due to scene-specific representations, as the same 3D point and view direction yield different densities and colors across scenes. Most existing methods focus on single-scene scenarios, addressing appearance or geometry changes over time [5, 12, 37, 59]. For example, MEIL-NeRF [12] and CLNeRF [5] reconstruct scenes from partial scans but are limited to single-scene forgetting. SLAM-based methods [9, 13, 44] assume past data availability for keyframe optimization, unsuitable for multi-scene settings without stored data.

Multi-Scene Continual Learning in NeRF. Encoding multiple distinct scenes in a single NeRF model is largely unexplored. A concurrent work, SCARF [53], uses a hyper-network to generate scene-specific weights, but this increases the parameter count with each new scene. In contrast, our C-NGP method leverages generative replay [42] to learn new scenes without storing past data, maintaining a fixed parameter count. By integrating scene conditioning into Instant-NGP's [34] hashing mechanism, C-NGP achieves high-fidelity rendering and fast convergence, outperforming continual learning baselines [5, 12] and competitive performance with SCARF [53] in multi-scene scenarios.

3 Method

We present C-NGP, a framework to incrementally model multiple scenes with the same set of parameters that is traditionally used for modeling a single scene in an NGP network. We begin with a brief overview of learning neural radiance fields and the multi-resolution hash encoding (MHE) technique. Next, we explore the conditional capabilities of Instant-NGP and describe how generative replay [42] can be integrated into conditional Instant-NGP. This extension enables incremental learning of new scenes while preserving information from previously encountered scenes.

3.1 Background

Neural Radiance Fields. Mildenhall *et al.* [33] introduced NeRF, a method for synthesizing novel views from a set of input images using a multi-layer perceptron (MLP) that maps a 5D input—3D spatial coordinates (x, y, z) and viewing direction (θ, ϕ)—to the density σ and color c of a point. Points are sampled along camera rays $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o}$ is the camera origin, $\mathbf{d}$ is the ray direction, and t is the sampled depth. Rendering is performed via volumetric integration [31], computing the color of a ray as $\hat{C}(r) = \sum_{i=1}^{N} w_i c_i$, where the weights $w_i = T_i \alpha_i$ depend on transmittance T_i and opacity $\alpha_i = (1 - e^{-\sigma_i \delta_i})$, with δ_i being the distance between adjacent samples. The model is trained by minimizing the squared error between predicted and ground-truth pixel colors: $\mathcal{L} = \sum_{r \in \mathcal{R}} ||\hat{C}(r) - C(r)||_2^2$, where $\mathcal{R}$ denotes the set of rays. Inputs to the MLP are encoded using sinusoidal positional encodings to better capture high-frequency scene details.

Multi-Resolution Hash Encoding. Müller *et al.* [34] introduced the concept of multi-resolution hash encoding (MHE) to enhance both the reconstruction accuracy and training efficiency of neural radiance fields (NeRFs) while maintaining minimal computational overhead. Unlike the traditional positional encoding used in NeRF, MHE employs a trainable multi-level 3D grid structure. The position of a sampled point is encoded by interpolating the features ($\mathbb{R}^F$) located at the vertices of the grid cell containing the point. These interpolated features are then fed into an MLP network, predicting both the density (σ) and the point's color (c).

The grid is organized into L levels, each containing a hash table of size $T \times F$, where T represents the number of features and F is the dimensionality of each feature. To efficiently map the grid coordinates, a spatial hash function $h(x)$, based on the approach of [47], is utilized:

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

Here, $\oplus$ denotes the bitwise XOR operation, and π_i are distinct large prime numbers.

3.2 Continual Neural Graphics Primitive (C-NGP)

This section introduces the C-NGP framework that incorporates scene-conditioning and continual learning in the Instant-NGP network [34].

Conditioning: To condition Instant-NGP, we use pseudo integer labels $C \in \{1, 2, 3, ...\}$ to differentiate scenes. Figure 1 illustrates the information flow. To get the Neural Hash Encoding (NHE) of coordinates, we combine the scene coordinates (x, y, z) with the pseudo label (C). Additionally, the pseudo labels undergo sinusoidal positional encoding (PE) [45], denoted as $\psi \in \mathbb{R}^{2\alpha}$, where $\alpha = 2$ represents the number of frequency components used. The neural hash encoding (NHE) and encoded pseudo labels (ψ) are concatenated and passed through a single-layer MLP to obtain density (σ) and scene features. The predicted scene feature is then used to compute the color for each input coordinate (x, y, z). Specifically, the scene feature is concatenated with the encoded viewing direction (SHE) and encoded pseudo label (ψ), forming the input to a two-layer MLP that predicts the final color values. The viewing direction is encoded using spherical harmonics encoding (SHE) [39], further refining the rendering process. We visualize how neural hashing helps segregate scenes within a shared representation space upon scene-conditioning in the supplementary material.

Continual learning: While conditioning helps non-conflicting scene representations, continual learning accommodates new, unseen scenes incrementally. Therefore, the proposed C-NGP involves training Instant-NGP (with scene conditioning) in a continual learning paradigm as follows. Each scene is encoded into conditional Instant-NGP using a pseudo-label. Before encoding a new scene, we first render all previously observed scenes using the camera parameters of the new scene. These rendered images are then added to the training set of the new scene, and the model parameters are updated accordingly. This strategy eliminates reliance on previous training datasets while ensuring high-fidelity modeling of all observed scenes. Rendering previously seen scenes with updated camera parameters is particularly effective for datasets like NeRF Synthetic 360°. For other scene types, we use the stored camera parameters for rendering.

3.3 Loss and Regularization

Floaters represent a significant challenge in radiance fields, typically manifesting as disconnected, dense spatial regions near the camera plane. Therefore, apart from the conventional rendering loss $\mathcal{L}_{mse}$, we include two additional regularizations to ensure stable C-NGP training.

Distortion. Following the work of [3, 55], we used distortion loss for compact point distribution on the camera ray:

$$\mathcal{L}_{dist} = \frac{1}{d(r)} \left(\sum_i w_i w_j \left| \frac{t_i + t_{i+1}}{2} - \frac{t_j + t_{j+1}}{2} \right| + \frac{1}{3} \sum_{i=1}^{N} w_i^2 (t_{i+1} - t_i) \right) \tag{2}$$

where, $d(r) = \frac{\sum_{i=1}^{N} w_i t_i}{\sum_{i=1}^{N} w_i}$ is depth along each ray. The first part of $\mathcal{L}_{dist}$ minimizes the weighted distance between all pairs of interval midpoints. The second part focuses on minimizing the weighted size of each interval. Jointly, the weights on the ray are encouraged to be compact by pulling distance intervals closer by consolidating each weight and minimizing the width of each interval [3].

Ray entropy. The entropy regularization calculates the entropy of the ray's distribution [4, 22] for each ray passing through the scene. This involves assessing the uncertainty or randomness in the predicted densities along the ray and avoiding the floaters in the rendered scene. The entropy regularization is described as per Equation 3

$$\mathcal{L}_{ent} = \left(-\sum_{i=1}^{N} p(r_i) log(p(r_i) \right) \tag{3}$$

Here, N is the number of sampled points on the ray r, and p_i is the opacity of each sampled point.

Complete loss. To train the C-NGP method, we used the combined loss formulation, as described in Equation 4.

$$\mathcal{L}_{total} = \mathcal{L}_{mse} + \lambda_{ent} \mathcal{L}_{ent} + \lambda_{dist} \mathcal{L}_{dist} \tag{4}$$

Here, λ_{ent} and λ_{dist} are the weights for the regularization. We keep these parameters for training the complete network as $\lambda_{ent} = 1e-3$ and $\lambda_{dist} = 1e-2$.

4 Experiments

In this section, we perform an exhaustive set of experiments to demonstrate the efficacy of the proposed framework. We start by discussing the datasets, evaluation metrics, and comparison baselines.

4.1 Datasets and Evaluation Metrics

We evaluate our model on widely used datasets: NeRF Synthetic 360° [33], Tanks and Temples [24], and Forward-Facing LLFF [32]. NeRF Synthetic 360° consists of eight diverse

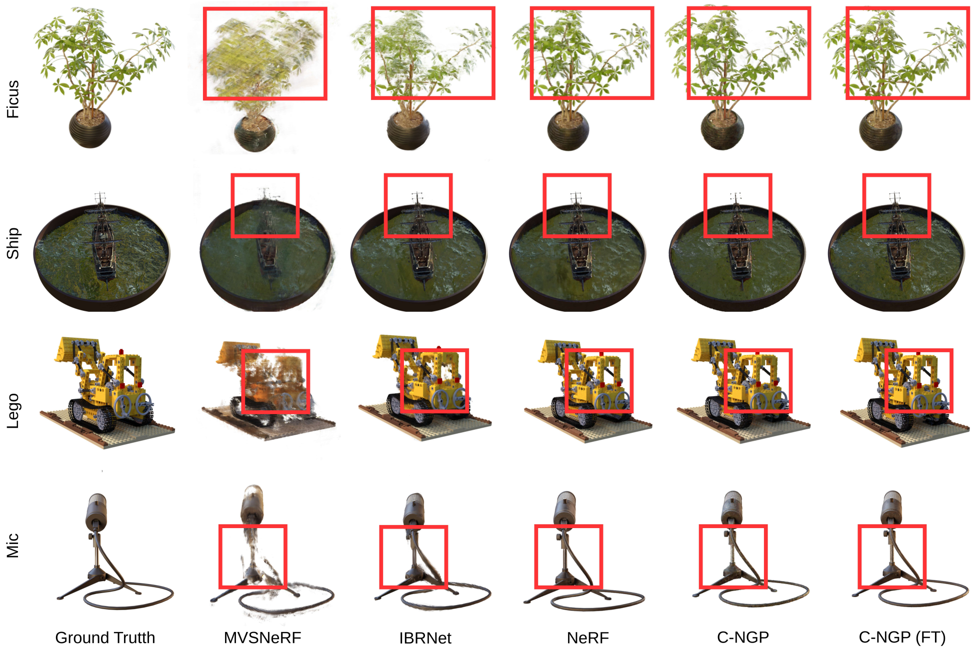

Figure 2: Qualitative Comparison on the Nerf Synthetic 360° dataset [33]. C-NGP is compared with MVSNeRF [6] and IBRNet [51] (both with per-scene fine-tuning), and vanilla NeRF [33] with per-scene optimization. C-NGP is observed to model scenes with minimal artifacts compared to those present in the other methods. C-NGP (FT) is fine-tuned for each test scene for five epochs after continual learning.

scenes, each with 100 training views and 200 test views at a resolution of 800×800. The images are rendered from either an upper hemisphere or a full hemisphere, and the camera parameters are not necessarily consistent across scenes. Forward-Facing LLFF contains eight real-world scenes captured with a handheld camera in a forward-facing manner. The number of images per scene varies from 20 to 62, with a resolution of 1008×756. Tanks and Temples feature five large-scale scenes with complex geometries and real-world objects, each captured at a resolution of 1920×1080. We use the masked version of this dataset for training and testing, following [29]. Additionally, we generate a dataset of 22 scenes using BlenderNeRF [38]. Each scene is created from freely available 3D object meshes and rendered with 100 training views and 100 test views from an upper hemisphere. This dataset is designed to stress test the continual learning setup of C-NGP and analyze the representative upper bound of network parameters. To ensure fair evaluation in a continual learning setup, all scenes in this dataset are rendered with the same camera parameters.

Metrics. We quantitatively evaluate the model performance Peak-Signal-to-Noise-Ratio (PSNR) [1], Structural Similarity Index (SSIM) [54], and Perceptual Score (LPIPS) [61].

4.2 Training Details

For training C-NGP, we used a batch size of $10,000$ rays and an initial learning rate of 2×10^{-3}. The model was trained for 30 epochs across synthetic and real datasets. C-NGP was conditioned on pseudo labels assigned to each scene to ensure scene-specific representations.

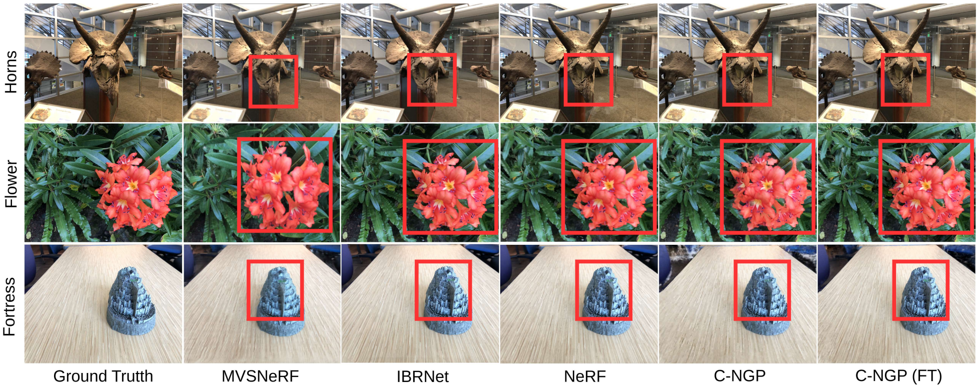

Figure 3: Qualitative Comparison on the Real Forward Facing (LLFF) dataset [32]. We compare C-NGP with MVSNeRF [6] and IBRNet [51] (both with per-scene fine-tuning), and vanilla NeRF [33] with per-scene. C-NGP (FT) is fine-tuned for each test scene for five epochs after continual learning.

Method	Tanks and Temples	
	PSNR ↑	SSIM ↑
EWC [23] + NeRF	15.64	0.420
PackNet [30] + NeRF	16.71	0.547
MEIL-NeRF [19]	17.98	0.580
CLNeRF [42]	21.30	0.640
SCARF [53]	26.78	**0.89**
C-NGP (ours)	**26.93**	0.87

Method	NeRF Synthetic 360			Real Forward-Facing		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SCARF [53]	**30.94**	0.94	-	**26.44**	**0.80**	-
C-NGP (ours)	29.40	**0.94**	0.09	22.12	0.66	0.37

Datasets	Offline Sampling			Online Sampling		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF Synthetic 360°	25.211	0.86	0.151	29.40	0.93	0.09
Blender Synthetic	37.735	0.98	0.022	37.734	0.98	0.022

Table 1: (Left) Quantitative comparison of C-NGP on the Tanks and Temple dataset [24] with baselines that combine either conventional continual learning methods with NeRF or introduce continual learning for single scene optimization along with a concurrent work. (Right-Top) Quantitative comparison of C-NGP on NeRF Synthetic [33] and Real LLFF [32] with concurrent work. (Right-Bottom) Analyzing the quantitative performance on training with (online) and without (offline) generative replay while learning C-NGP. The enhanced performance with online sampling is due to the same camera parameters across all the scenes.

The hash table size for Instant-NGP was set to $T = 2^{19}$, with a feature size of $F = 4$. For the NeRF Synthetic 360° dataset, we applied the generative replay method [42], where previously observed scenes were re-rendered using the camera parameters of the new scene. For real-world datasets like LLFF [32] and Tanks and Temples [24], which lack full 360° coverage, we only store the camera parameters of each scene. This allowed us to generate novel views while avoiding artifacts such as floaters or ghosting, less problematic issues in synthetic datasets with complete 360° coverage.

4.3 Experimental Evaluation

Choice of Baselines. Since no existing work incrementally models multiple scenes in a single radiance field, we compare our approach with the most closely related baselines that align, at least partially, with our experimental setup [6, 19, 51, 53]. We have shown the quantitative analysis results in the supplementary. Among them, SCARF [53] is the closest and most concurrent to our work. It handles multiple scenes, however, by increasing the number of parameters through a global parameter generator, unlike us. We compare the quantitative

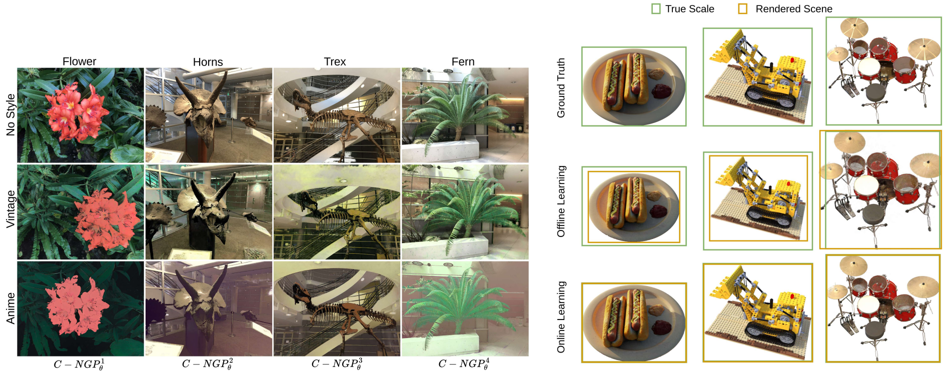

Figure 4: (Left) The figure illustrates how the proposed C-NGP method stores multiple edit styles within a single network. Each scene is trained separately, represented by θ^i, and stores different edit styles for respective scenes. (Right) Effect of the difference in camera parameters across scenes in the offline setting.

performance with SCARF based on their reported results in the paper across different datasets, and omit the qualitative comparison due to the unavailability of their code.

Quantitative. We compare the proposed paradigm with continual learning frameworks like Elastic Weight Consolidation (EWC) and PacketNet [30] combined with NeRF, and single-scene continual NeRF methods like MEIL-NeRF [12] and CLNeRF [42]. As shown in Table 1 (left), on the Tanks and Temples [24], C-NGP achieves the highest PSNR and SSIM comparable to SCARF (0.02 units lower) while avoiding additional parameters per scene. On Nerf Synthetic 360° [33] (see Table 1 (right)), C-NGP shows competitive performance with SCARF. However, over scenes from the LLFF dataset [32], C-NGP does not achieve the best scores due to the inherent struggle of Instant-NGP over the unbounded scenes. To support the claim, more empirical results are detailed in the supplementary material.

Qualitative. We provide a qualitative comparison of C-NGP with MVSNeRF [6], IBRNet [51], and vanilla NeRF [33] in Figure 2 and 3, over NeRF synthetic and Real LLFF datasets, respectively. Given its multi-scene modeling capability, we expect C-NGP's performance to at least closely match per-scene optimized vanilla NeRF. Our results show that C-NGP achieves the best rendering quality on the NeRF Synthetic dataset and closely matches vanilla NeRF (which performs better than Instant-NGP) on the Real Forward-Facing dataset [32]. Additionally, we evaluate the visual fidelity of different scenes rendered by C-NGP across the NeRF Synthetic [33], Real Forward-Facing [32], and Tanks and Temples [24] in the supplementary.

While C-NGP shows strong performance on synthetic datasets and Tanks & Temples, its performance on unbounded LLFF scenes is relatively lower. We attribute this partly to the Instant-NGP backbone, which is known to be less effective in such settings [3, 27]. Nonetheless, C-NGP shows better performance than Instant-NGP after fine-tuning while retaining efficiency and scalability.

4.4 Additional Results and Analysis

Style Editing. To show the application of the proposed C-NGP framework, we adapt it for style editing in 3D scenes [20, 25, 35]. The current methods are only able to do single style transfer to any NeRF model at a time [16, 18], and for each new style, they have to retrain

Method	Current Scene [**L**]				Current Scene [L → **C**]				Current Scene [L → C → **S**]				Current Scene [L → C → S → **D**]			
	Lego	Chair	Ship	Drums	Lego	Chair	Ship	Drums	Lego	Chair	Ship	Drums	Lego	Chair	Ship	Drums
Offline Sampling	35.62	x	x	x	34.59	34.63	x	x	34.19	34.06	28.53	x	14.41	18.12	28.06	25.02
Online Sampling	35.62	x	x	x	34.81	34.82	x	x	34.24	34.35	28.68	x	33.39	33.89	28.29	25.20

Table 2: Analysing progressive degradation in the rendering quality with *offline* and *online sampling* of scenes from the NeRF Synthetic 360° dataset [33]. The cell highlighted in red color shows high degradation in the PSNR value under offline sampling.

	Single Dataset		Mixed Dataset	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Groups 1 (8 Scenes)	35.889	0.978	29.40	0.94
Groups 2 (12 Scenes)	33.989	0.971	28.295	0.915
Groups 3 (16 Scenes)	32.448	0.964	27.190	0.906
Groups 4 (22 Scenes)	30.472	0.954	25.744	0.896

Experiments	PSNR ↑	SSIM ↑	LPIPS ↓
XYZ ∣ C + $SE(\theta, \phi)$	24.108	0.812	0.201
XYZ + $\psi(C)$ + $SE(\theta, \phi)$	12.966	0.813	0.460
XYZ ∣ C + $\psi(C)$ + $SE(\theta, \phi)$	24.492	0.824	0.184
XYZ ∣ C + $\psi(C)$ + $SE(\theta, \phi)$ + $\psi(C)$	**25.927**	**0.880**	**0.139**

Table 3: (Left) Analysing upper bound on the number of scenes C-NGP can accommodate with reasonable loss in rendering quality across single and mixed datasets. Single: 22 scenes from BlenderNeRF dataset. Mixed: 14 scenes from BlenderNeRF + 8 scenes from NeRF Synthetic 360° dataset. (Right) Ablation on the configuration choice for conditioning Instant-NGP network [34] - the backbone of C-NGP.

the NeRF model. The incremental nature of C-NGP also allows for accommodating multiple different styles within a scene. For each scene, we train a separate C-NGP network θ^1, θ^2, θ^3, θ^4 and for each θ^i, we first learn the same scene for K times incrementally using C-NGP and to transfer the style, we only fine-tune the MLP layer responsible for color (RGB). As shown in Figure 4 (left), we learn two different styles for each scene, i.e., each θ^i stores three different variations of the same scene, where the first is the original scene itself, and the next two are the edited scene styles which are vintage and anime. To generate consistent multi-view style scenes for training, we follow the work by Fujiwara *et al.* [16], which uses fully fused attention and depth-conditioned ControlNet [60].

Sampling Strategy. To study the effect of how well C-NGP performs when in a continual paradigm with and without using the generative replay method (using stored images of previous scenes). Here, we define *online sampling* and *offline sampling* for the case with and without generative replay, respectively. As shown in Table 1 (bottom-right), the performance with *online* and *offline* sampling is nearly the same over the BlenderNeRF dataset, in fact, better than with *offline sampling* over the NeRF Synthetic dataset [33]. We attribute such performance trends to the nature of camera parameters. Figure 4 (right) shows such misalignments due to differences in camera parameters that eventually cause a reduction in PSNR, leading to reduced performance in the offline sampling. To further emphasize this, Table 2 exclusively evaluates the rendering quality of scenes from the NeRF Synthetic dataset with these sampling methods.

Upper bound. Table 3 (left) attempts to demonstrate the number of scenes that can be accommodated within a given set of parameters with an acceptable loss in the rendering quality of the previously learned scenes. While we do not claim to avoid forgetting completely, most importantly, we succeed in slowing it down to the extent that we can reasonably model around ~ 20 different scenes in a single neural radiance field.

Design choices. We evaluated different strategies for concatenating pseudo labels and their sinusoidal positional encodings with scene coordinates and viewing directions as summarized in Table 3 (right). In our work, we use the configuration with the highest PSNR. More architectural design choices are detailed in the supplementary.

Supplementary. Due to space constraints, we elaborate: (a) quantitative comparison with other methods, (b) time analysis, (c) shared feature space visualization, and (d) additional discussion in the supplementary material. We urge the readers to kindly refer to the supplementary material as well for a holistic understanding of this work. These additional studies complement the main results and ensure that our evaluation is comprehensive, addressing performance, efficiency, and representation quality.

5 Conclusion

We introduced C-NGP, a continual learning framework for incremental multi-scene modeling within a single neural radiance field. The proposed method accommodates new scenes while reasonably preserving the previously learned information via scene conditioning and generative replay. Extensive empirical studies on synthetic and real datasets demonstrate that our method achieves strong performance in multi-scene representation learning and outperforms existing methods over training and inference speed, offering real-time rendering capabilities. C-NGP offers practical value for scalable 3D repositories, enabling multiple scenes and even style edits to be stored and shared in a single lightweight checkpoint, avoiding the overhead of per-scene models. While at this stage, the method incurs an acceptable loss in rendering quality while modeling multiple scenes, its performance in highly diverse scene settings and attempt towards zero forgetting warrants further exploration.

6 Acknowledgment

This work was supported by the Prime Minister Research Fellowship awarded to Prajwal Singh and Ashish Tiwari, and by the Jibaben Patel Chair in Artificial Intelligence held by Shanmuganathan Raman.

References

[1] Implementation of a modified cvsd coder. *International Journal of Electronics*, 62(3): 473–479, 1987. doi: 10.1080/00207218708920998.

[2] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20919–20929, 2023.

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[4] Matteo Bonotto, Luigi Sarrocco, Daniele Evangelista, Marco Imperoli, and Alberto Pretto. Combinerf: A combination of regularization techniques for few-shot neural radiance field view synthesis. *arXiv preprint arXiv:2403.14412*, 2024.

[5] Zhipeng Cai and Matthias Müller. Clnerf: Continual learning meets nerf. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23185–23194, 2023.

[6] 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. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021.

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022.

[8] Jianchuan Chen, Wentao Yi, Liqian Ma, Xu Jia, and Huchuan Lu. Gm-nerf: Learning generalizable model-based neural radiance fields from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20648–20658, 2023.

[9] Yue Chen, Xingyu Chen, Xuan Wang, Qi Zhang, Yu Guo, Ying Shan, and Fei Wang. Local-to-global registration for bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8264–8273, 2023.

[10] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024.

[11] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16569–16578, 2023.

[12] Jaeyoung Chung, Kanggeon Lee, Sungyong Baik, and Kyoung Mu Lee. Meil-nerf: Memory-efficient incremental learning of neural radiance fields. *arXiv preprint arXiv:2212.08328*, 2022.

[13] Tianchen Deng, Guole Shen, Tong Qin, Jianyu Wang, Wentao Zhao, Jingchuan Wang, Danwei Wang, and Weidong Chen. Plgslam: Progressive neural scene representation with local to global bundle adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19657–19666, 2024.

[14] Shuangkang Fang, Weixin Xu, Heng Wang, Yi Yang, Yufeng Wang, and Shuchang Zhou. One is all: Bridging the gap between neural radiance fields architectures with progressive volume distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 597–605, 2023.

[15] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

[16] Haruo Fujiwara, Yusuke Mukuta, and Tatsuya Harada. Style-nerf2nerf: 3d style transfer from style-aligned multi-view images. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–10, 2024.

[17] Cameron Gordon, Shin-Fang Chng, Lachlan MacDonald, and Simon Lucey. On quantizing implicit neural representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 341–350, 2023.

[18] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023.

[19] Hao He, Yixun Liang, Shishi Xiao, Jierun Chen, and Yingcong Chen. Cp-nerf: Conditionally parameterized neural radiance fields for cross-scene novel view synthesis. In *Computer Graphics Forum*, volume 42, page e14940. Wiley Online Library, 2023.

[20] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13869–13878, 2021.

[21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 2023. doi: 10.1145/3592433.

[22] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022.

[23] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[24] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[25] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, volume 40, pages 29–43. Wiley Online Library, 2021.

[26] SeokYeong Lee, JunYong Choi, Seungryong Kim, Ig-Jae Kim, and Junghyun Cho. Few-shot neural radiance fields under unconstrained illumination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2938–2946, 2024.

[27] Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: Efficient sampling accelerates nerfs. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 18537–18546, 2023.

[28] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[29] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

[30] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[31] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[32] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019.

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[35] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. Snerf: stylized neural implicit representations for 3d scenes. *arXiv preprint arXiv:2207.02363*, 2022.

[36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[37] Ryan Po, Zhengyang Dong, Alexander W Bergman, and Gordon Wetzstein. Instant continual learning of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3334–3344, 2023.

[38] Maxime Raafat. GitHub - maximeraafat/BlenderNeRF: Easy NeRF synthetic dataset creation within Blender — github.com. https://github.com/maximeraafat/BlenderNeRF. [Accessed 19-05-2024].

[39] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, 2001.

[40] Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1320–1328, 2017.

[41] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[42] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[43] Seungjoo Shin and Jaesik Park. Binary radiance fields. *Advances in neural information processing systems*, 36, 2024.

[44] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6229–6238, 2021.

[45] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

[46] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. Compressible-composable nerf via rank-residual decomposition. *Advances in Neural Information Processing Systems*, 35:14798–14809, 2022.

[47] Matthias Teschner, Bruno Heidelberger, Matthias Müller, Danat Pomerantes, and Markus H Gross. Optimized spatial hashing for collision detection of deformable objects. In *Vmv*, volume 3, pages 47–54, 2003.

[48] Ziyu Wan, Christian Richardt, Aljaž Božič, Chao Li, Vijay Rengarajan, Seonghyeon Nam, Xiaoyu Xiang, Tuotuo Li, Bo Zhu, Rakesh Ranjan, et al. Learning neural duplex radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8307–8316, 2023.

[49] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9065–9076, 2023.

[50] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[51] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021.

[52] Yuze Wang, Junyi Wang, Yansong Qu, and Yue Qi. Rip-nerf: learning rotation-invariant point-based neural radiance field for fine-grained editing and compositing. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*, pages 125–134, 2023.

[53] Yuze Wang, Junyi Wang, Chen Wang, Wantong Duan, Yongtang Bao, and Yue Qi. Scarf: Scalable continual learning framework for memory-efficient multiple neural radiance fields. *arXiv preprint arXiv:2409.04482*, 2024.

[54] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[55] Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4180–4189, 2023.

[56] Hao Yang, Lanqing Hong, Aoxue Li, Tianyang Hu, Zhenguo Li, Gim Hee Lee, and Liwei Wang. Contranerf: Generalizable neural radiance fields for synthetic-to-real novel view synthesis via contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16508–16517, 2023.

[57] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8254–8263, 2023.

[58] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[59] Letian Zhang, Ming Li, Chen Chen, and Jie Xu. Il-nerf: Incremental learning for neural radiance fields with camera pose alignment. *arXiv preprint arXiv:2312.05748*, 2023.

[60] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[62] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5449–5458, 2022.