Incremental Multi-Scene Modeling via Continual Neural Graphics Primitives

Prajwal Singh Ashish Tiwari Gautam Vashishtha Shanmuganathan Raman

CVIG Lab , Indian Institute of Technology Gandhinagar, India

BMVC 2025

CVIG IIT Gandhinagar

PROBLEM

- Neural radiance fields (NeRF) have exhibited highly photorealistic rendering of novel views through per-scene optimization over a single 3D scene.
- NeRF's based method lacks scalability since a separate model needs to be stored for each scene, and the training time increases linearly with every newly added scene.
- In this work, we propose a novel framework, called C-NGP, to accommodate multiple scenes into the parameters of a single neural radiance field.
- We observe the framework is also inherently continual (via generative replay) with minimal, if not no, forgetting of the previously learned scenes.

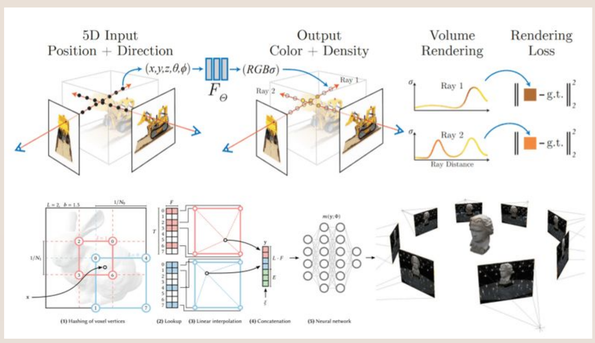


Fig.1: NeRF Pipeline[5]

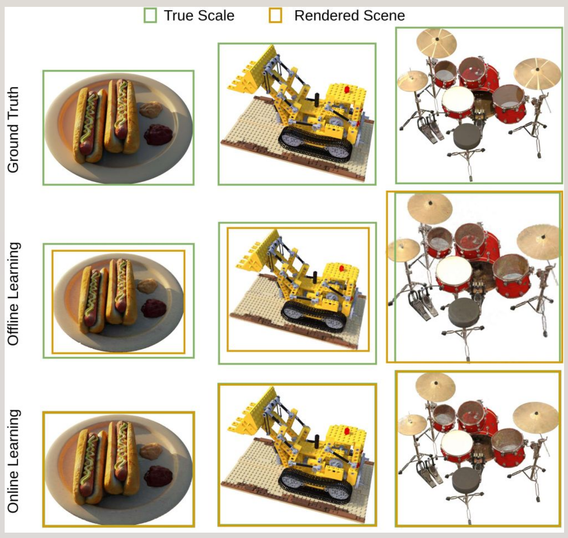
OVERVIEW

Algorithm 1: Learning new scenes using Conditional-cum-Continual NeRF (C-NGP)

Require: I_{new}: list of new scene images
Require: l_{new}: pseudo label for new scene
Require: L_{prev}: list of pseudo labels for previous scenes
Require: F_θ: NeRF network
Require: $C_{I_{new}}$: camera parameters of new scene
Require: k: number of images to render for previous scenes

/* Render previous scenes and store in I_{prev} */
Initialize $I_{prev} \leftarrow \{\}$
for $l^i_{prev} \in L_{prev}$ **do**
  $render\_image \leftarrow F_\theta(l^i_{prev}, C_{I_{new}}, k)$
  I_{prev}.append($render\_image$)
$T_{images} \leftarrow concat(I_{new}, I_{prev})$ // train images list
$T_{labels} \leftarrow concat(L_{new}, L_{prev})$ // train labels list
/* $TrainNeRF()$ train the F_θ on given scene list and labels */
$F^{updated}_\theta \leftarrow TrainNeRF(F_\theta, T_{images}, T_{labels}, C_{I_{new}})$
Output: Updated $F^{updated}_\theta$ on new scene

- Proposed algorithm for modeling multiple scene.



- **Fig.2**: Issue occur when only doing conditional modeling. It also illustrates the effect of online and offline sampling of training data.

C-NGP FRAMEWORK

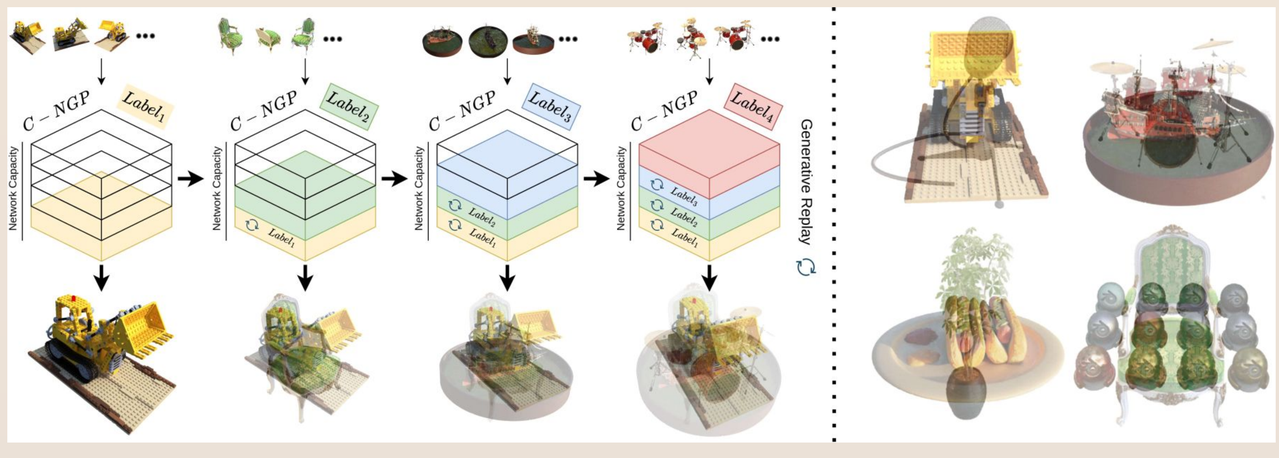


Fig.3: Conditional-cum-Continual learning framework for modeling multiple scenes in a single neural radiance field. Each new scene is assigned a pseudo label and trained continually while preserving the information of previously learned scenes via generative replay.

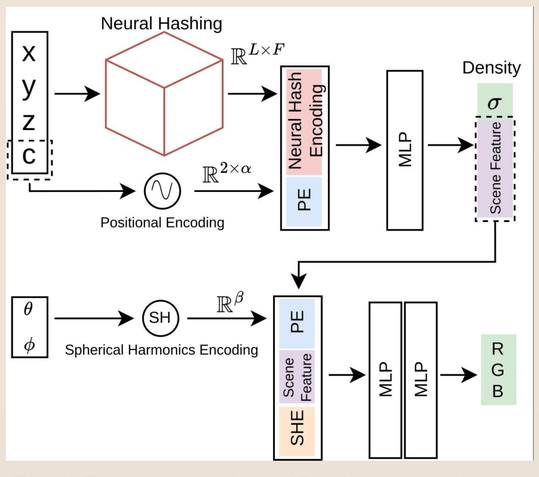


Fig.4: The proposed modification to the Instant NGP architecture[2] incorporates information from multiple scenes using pseudo labels.

MODELLING MULTIPLE SCENE

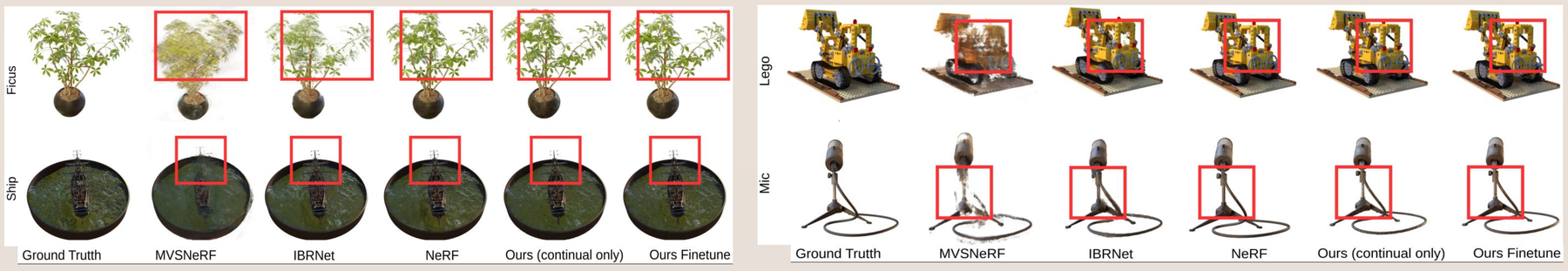


Fig.5: Qualitative demonstration of the quality of rendered images across different views through C-NGP over the scenes from the NeRF Synthetic 360 dataset[1].

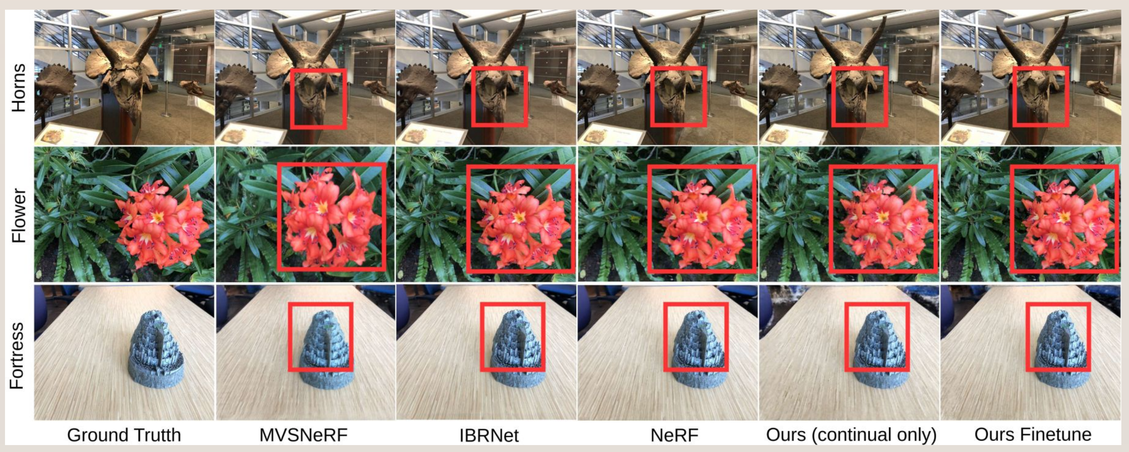


Fig.6: Qualitative demonstration of the quality of rendered images across different views through C-NGP over the scenes from the Real Forward-Facing dataset[1].

Fig.7: Qualitative demonstration of the quality of rendered images across different views through C-NGP over the scenes from the Tanks and Temples dataset[3].

Method	Tanks and Temples	
	PSNR ↑	SSIM ↑
EWC [23] + NeRF	15.64	0.420
PackNet [30] + NeRF	16.71	0.547
MEIL-NeRF [12]	17.98	0.580
CLNeRF [12]	21.30	0.640
SCARF [53]	26.78	**0.89**
C-NGP (ours)	**26.93**	0.87

Tab.1: Quantitative comparison on Tanks and Temple dataset[3] across different baseline methods.

Tab.2: (Top) Quantitative comparison on NeRF Synthetic 360 dataset[1] and Real Forward-Facing dataset[1] with baseline method. (Bottom) Analyzing the quantitative performance on training with (online) and without (offline) generative replay while learning C-NGP.

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

STYLE EDITING & REPRESENTATION SPACE

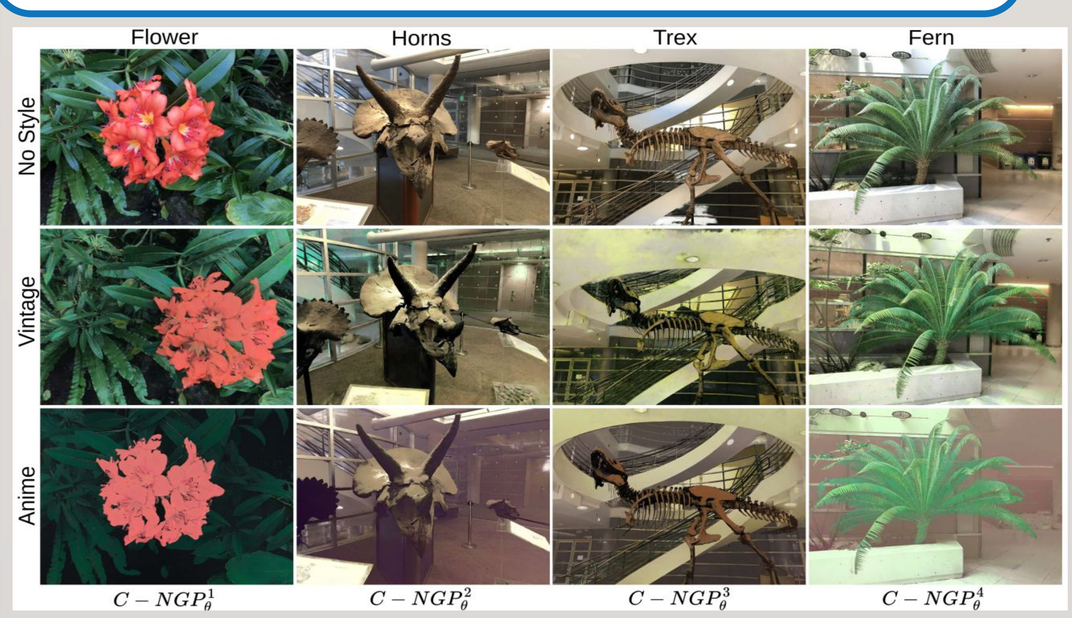


Fig.8: The figure illustrates the style editing[4] capabilities of C-NGP, for each scene we train a new C-NGP and each style is encoded in its equivalent scene network using the proposed approach.

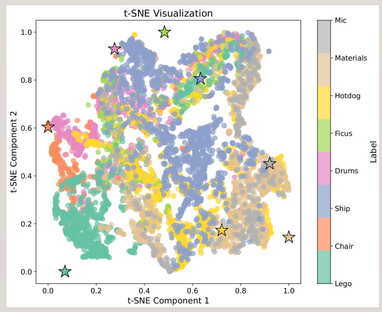
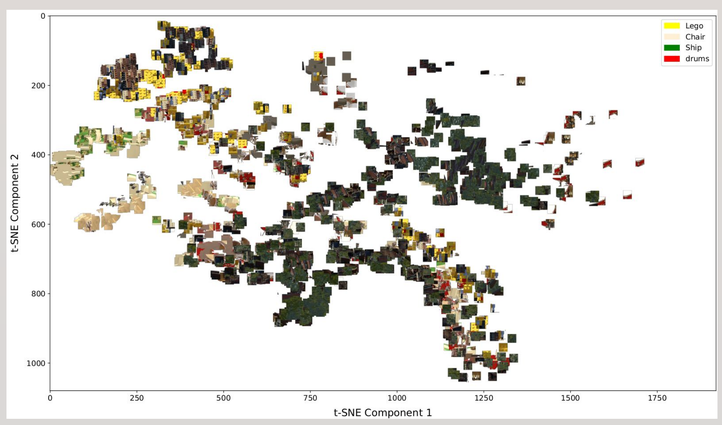


Fig.9-10: The figure illustrates the representation space of C-NGP, visualized through a t-SNE map over scenes from NeRF Synthetic 360 dataset.



REFERENCES

[1] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2020). Nerf: representing scenes as neural radiance fields for view synthesis (2020). arXiv preprint arXiv:2003.08934.
[2] Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG), 41(4), 1-15.
[3] Knapitsch, A., Park, J., Zhou, Q. Y., & Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG), 36(4), 1-13.
[4] Fujiwara, H., Mukuta, Y., & Harada, T. (2024, December). Style-nerf2nerf: 3d style transfer from style-aligned multi-view images. In SIGGRAPH Asia 2024 Conference Papers (pp. 1-10).
[5] https://theaisummer.com/nerf/