InsertNeRF: Instilling Generalizability into NeRF with HyperNet Modules

Halle B #255

Yanqi Bao[1] , Tianyu Ding[2] , Jing Huo[1*], Wenbin Li[1], Yuxin Li[1], Yang Gao[1]

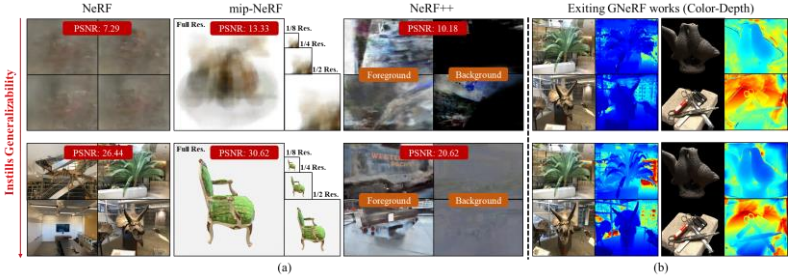
1 State Key Laboratory for Novel Software Technology, Nanjing University, China

2 Microsoft Corporation, Redmond, USA

ICLR International Conference On Learning Representations

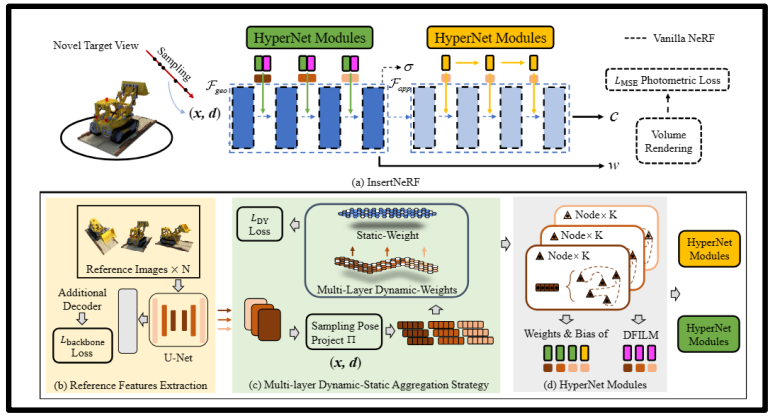
Motivation

How to directly **INS**till g**E**ne**R**alizabili**T**y into **NeRF? (InsertNeRF)**



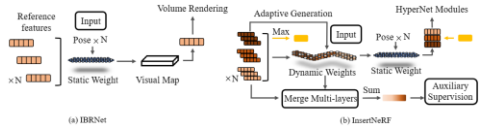
- **InsertNeRF** can insert generalization to NeRF's derivative works, such as mip-NeRF, NeRF++.
- **InsertNeRF** has forsaken time-consuming components such as transformers or cost volumes, which are mentioned in existing works.

Pipeline



InsertNeRF, a novel paradigm that inserts multiple **plug-and-play HyperNet modules** into the **NeRF-like framework**, endowing NeRF-like systems with instilled generalizability.

HyperNet Modules



InsertNeRF proposes a **Multi-Layer Dynamic-static aggregation strategy**, which models the **views-occlusion** and **globally completes information** based on the multi-view relationships.

Sampling-aware Filter:
Based on a Graph Structure

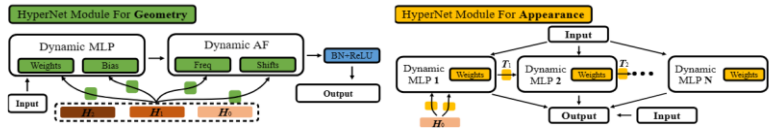
$$\boldsymbol{H}_l = (\boldsymbol{I} - \boldsymbol{A}_l)\,\boldsymbol{F}_{\text{view}}\boldsymbol{W}_l^a$$

Dynamic MLP:

$$\text{Weight}_{\boldsymbol{H}_l} \times \boldsymbol{F}_{\text{input}} + \text{Bias}_{\boldsymbol{H}_l}$$

Dynamic Activation Function:
Based on DFiLM:

$$\boldsymbol{F}_{\text{output}} = \text{Shift}_{\boldsymbol{H}_l}(\text{Freq}_{\boldsymbol{H}_l}(\text{Weight}_{\boldsymbol{H}_l} \times \boldsymbol{F}_{\text{input}} + \text{Bias}_{\boldsymbol{H}_l}))$$



(a) HyperNet module in $\mathcal{F}_{geo}$ (b) HyperNet module in $\mathcal{F}_{app}$

Two different types of HyperNet Modules for **Geometry and Appearance**

Quantitative Experiments

Quantitative experiments on different settings:

Table 1: Comparisons of InsertNeRF against SOTA methods with Setting I.

Methods	NeRF Synthetic PSNR↑	NeRF Synthetic SSIM↑	NeRF Synthetic LPIPS↓	LLFF PSNR↑	LLFF SSIM↑	LLFF LPIPS↓	DTU PSNR↑	DTU SSIM↑	DTU LPIPS↓
PixelNeRF (CVPR2021)	22.65	0.808	0.202	18.66	0.588	0.463	19.40	0.463	0.447
MVSNeRF (ICCV2021)	25.15	0.853	0.159	21.18	0.691	0.301	23.83	0.723	0.286
IBRNet (CVPR2021)	26.73	0.908	0.101	25.17	0.813	0.200	25.76	0.861	0.173
ContraNeRF (CVPR2023)	-	-	-	25.44	0.842	0.178	27.69	0.904	0.129
GeoNeRF† (CVPR2022)	28.33	0.938	0.087	25.44	0.839	0.180	-	-	-
WaveNeRF† (ICCV2023)	26.12	0.918	0.113	24.28	0.794	0.212	-	-	-
NeuRay(CVPR2022)	28.92	0.920	0.096	25.85	0.832	0.190	28.30	0.907	0.130
InsertNeRF (**Ours**)	**30.35**	**0.938**	**0.065**	**26.44**	**0.844**	**0.169**	**29.75**	**0.925**	**0.077**

Table 2: Comparisons and ablations with Setting II.

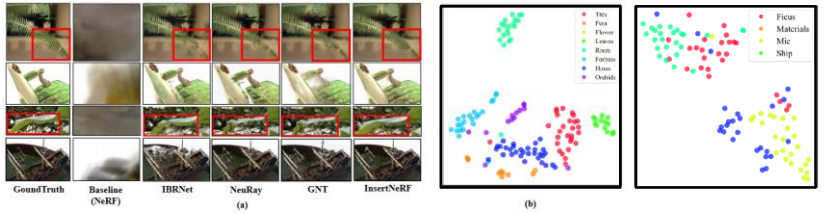
Methods	NeRF Synthetic PSNR↑	NeRF Synthetic SSIM↑	NeRF Synthetic LPIPS↓	LLFF PSNR↑	LLFF SSIM↑	LLFF LPIPS↓
GNT (ICLR2023)	27.29	**0.937**	0.056	25.59	0.858	0.128
Baseline (NeRF)	7.29	0.512	0.690	11.46	0.328	0.582
NeRF with HyperNetwork	25.86	0.902	0.081	24.25	0.793	0.177
InsertNeRF w/o MLDS	25.12	0.896	0.098	24.41	0.814	0.156
InsertNeRF (**Ours**)	**27.57**	0.936	**0.056**	**25.68**	**0.861**	**0.126**

Table 3: Results with sparse inputs.

Methods	3-view PSNR↑	3-view SSIM↑	3-view LPIPS↓
DietNeRF (ICCV 2021)	14.94	0.370	0.496
RegNeRF (CVPR 2022)	19.08	0.587	0.336
GeCoNeRF (ICML 2023)	18.77	0.596	0.338
FreeNeRF (CVPR 2023)	**19.63**	0.612	**0.308**
InsertNeRF (w/o retrain)	19.41	**0.618**	0.330

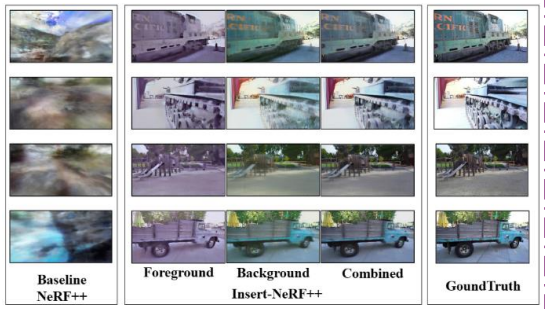
Qualitative Experiments

Vanilla NeRF and t-SNE plot of the scene-specific representation:



Insert Mip-NeRF:

Insert NeRF++:

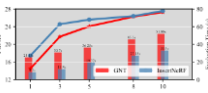


Ablations

HyperNet Modules ablations:

Methods	PSNR↑	LLFF SSIM↑	LLFF LPIPS↓
w/o D-MLP	23.33	0.774	0.198
w/o Sampling Filter	24.67	0.815	0.158
w/o DFiLM	25.04	0.832	0.152
w/o original MLP	25.44	0.848	0.131
InsertNeRF (**Ours**)	**25.68**	**0.861**	**0.126**

Efficiency:



MLDS aggregation strategy ablations:

Static-Weight	Dynamic-Weight	Auxiliary-Supervision	Multi-Layers	Single-Layer	LLFF PSNR↑	LLFF SSIM↑	LLFF LPIPS↓
✓			✓		24.88	0.827	0.154
✓	✓		✓		25.55	0.851	0.128
	✓	✓	✓		25.53	0.850	0.131
✓	✓	✓		✓	25.15	0.838	0.139
✓	✓	✓	✓		**25.68**	**0.861**	**0.126**

Conclusion

Key: **plug-and-play HyperNet modules,** endowing NeRF-like systems with instilled generalizability.

Future: We hope that such representations can find more applications in generalized representation and 3D generation.

Code(soon) *Paper*

[1] 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