Supplementary Material - Strata-NeRF : Neural Radiance Fields for Stratified Scenes

Ankit Dhiman[1,2] R Srinath[1] Harsh Rangwani[1] Rishubh Parihar[1]
Lokesh R Boregowda[2] Srinath Sridhar[3] R Venkatesh Babu[1]
[1]VAL Lab, IISc Bangalore [2]Samsung R & D Institute India,Bangalore [3]Brown University

Organization of Appendix

A. Introduction

We present additional results and other details related to our proposed method : Strata-NeRF. We elaborate on the proposed synthetic stratified dataset in Appendix B. We give the implemntation details in Appendix D. Then, we present additional ablation study and results in Appendix E. Please watch the video *"main-video-strata-NeRF.mp4"* in the supplementary material.

B. Synthetic Dataset Details

Figure 1 shows the representation of each level of each scene. Table 1 shows the level-wise split for each scene.

B.1. Cube-Sphere-Monkey

This dataset consists of simple geometric entities such as a cube, sphere and a monkey mesh provided in Blender [5]. Figure 1 illustrates the layout of this scene. *Cube* is at level 0, *Sphere* is at level 1 and *Monkey* is at the innermost level. The texture for *Cube* is an image generated from Stable Diffusion demo [6]. We sample camera poses from the curved surface of a hemisphere for the outer cube and from the curved surface of a sphere for the inner levels.

B.2. Coffee Shop

This dataset mimics an actual coffee shop setup inside another shopping complex. The outermost level consists of concrete walls. At level 1, i.e. when one enters the shopping complex, there is regular flooring and a concrete ceiling. Here, we also notice the exterior walls of our coffee shop. At level 2; i.e., inside the coffee shop; there is a layout with a counter, menu board and a table for visitors. All these scenes are composited with the help of Blender [5]. We sample camera poses from the curved surface of a hemisphere for all the levels.

B.3. Bhutanese House

A typical household setting inspired us to create this dataset. A typical residence features a table in the living room. In most cases, a decorative object is kept on the table. For the structure of the house, we choose a Bhutanese house model. The exterior of this structure is level 0. At level 1, i.e., inside the house, there are chairs, tables and other household items in the living room. At level 2, we have a glass bottle with a ship. We sample camera poses

(a) (b) (c) (d) (e)

Figure 1. (a) Cube-Sphere-Monkey, (b) Coffee Shop, (c) Bhutanese House, (d) Buddhist Temple and (e) Dragon In Pyramid. Representative images for each level.

Table 1. **train-val-test** level-wise split for each scene.

Scene	Split	Level 0	Level 1	Level 2
Cube-Sphere-Monkey	**train**	30	30	30
	val	30	30	30
	test	30	30	30
Coffee Shop	**train**	30	30	30
	val	15	15	15
	test	15	15	15
Bhutanese House	**train**	30	30	30
	val	15	15	15
	test	15	15	15
Buddhist Temple	**train**	30	20	20
	val	15	10	10
	test	15	10	10
Dragon In Pyramid	**train**	30	30	-
	val	15	15	-
	test	15	15	-

from the curved surface of a hemisphere. For level 2, we capture around the glass bottle on the circular table.

B.4. Dragon In Pyramid

This dataset captures a fantastical world filled with pyramids and dragons. We use a model of a *Mayan pyramid* as the outer structure. Inside the pyramid, we place a model of a dragon. Thus, this scene has two levels: 1.) the outer walls of the *Mayan pyramid* and 2.) the dragon residing inside the pyramid. All the camera poses are sampled from the curved surface of different hemispheres.

B.5. Buddhist Temple

This scene depicts an archaeological site or a typical monument location. We select a Buddhist temple to represent this scene. Two levels indicate the nearby rooms inside the structure in this context. Level 0 represents the outer structure of the monument, Levels 1 contains a bronze statue in the center of the monument, and Level 2 contains a Buddha statue mounted to the wall of one room.

C. Real Dataset

We evaluate our method on real-world scenes as well. We choose RealEstate10K [13] dataset, which contains camera poses corresponding to camera frames from video-clips exracted from Youtube videos. The camera poses are obtained by running SLAM and bundle adjustment algorithm over these large videos. To create a "stratified" scene from this dataset, first we cluster video clips belonging to same Youtube video using the video token provided in the ground-truth files. Then we extratcted camera frames and pose as per the timestamp information provided in the ground-truth files. The extracted camera pose for each video clip from a scene were already aligned with respect to a common coordinate system. We removed the video clips which had any dynamic motion within them. We extracted four scenes which are "Spanish Colonial Retreat in Scottsdale Arizona" [11], "139 Barton Avenue Toronto Ontario" [12] ,"31 Brian Dr Rochester NY" [2] and "7 Rutledge Ave Highland Mills" [7].

D. Implementation Details

Architecture Details. We provide architectural details of the "Latent Generator" and "Latent Router" networks in Figure 2 and 3 respectively.

Training. We use Adam [8] optimizer with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-6}$ and initial learning rate $= 0.002$. Further, the learning rate is log-linearly interpolated such that learning rate $= 0.00002$ at maximum steps. Additionaly, there are 512 warmup steps. Distortion loss proposed in Mip-NeRF 360 [1] is switched off for the blender datasets as proposed by the authors. We use one proposal MLP and one NeRF MLP. We weight the loss for "Latent Generator" with value $\lambda_2 = 0.1$.

Implementation. Our implementation is based on Mip-NerF 360 [1] which uses JAX [3] framework.

D.1. Choice of Training Configuration File

The dataset described in Section B is created using Blender [5]. This dataset has white background for the level 0. Barron *et al.* [1] uses "*blender_256.gin*" file for

the blender scenes proposed in NeRF [9] which are small in size compared to our scenes. This configuration file does not work for the scenes we proposed in Appendix B. Hence, we use "*360.gin*" and alter the dataset type field in the configuration file.

Table 2 shows the quantitative comparison of the above mentioned configuration files on *Dragon In Pyramid* dataset. We observe that the "*360.gin*" configuration beats the "*blender_256.gin*" in all the levels. Figure 4 compares the qualitative results of these two configuration files. We notice that the novel views from "*blender_256.gin*" are inferior in quality compared to "*360.gin*" configuration. "*360.gin*" configuration has better performance because of the contract function proposed by Barron [1]. The contract function is defined as follows:

$$\text{contract}(x) = \begin{cases} x, & ||x|| \leq 1 \\ (2 - \frac{1}{||x||})\left(\frac{x}{||x||}\right), & \text{otherwise} \end{cases} \quad (1)$$

This contract function maps input coordinates onto a ball of radius 2. Effectively, a large range is bounded inside a radius of $2\,m$. This is the reason why "*360.gin*" configuration is better for large blender scenes. Hence, we use this configuration file for all the scenes other than "Cube-Sphere-Monkey".

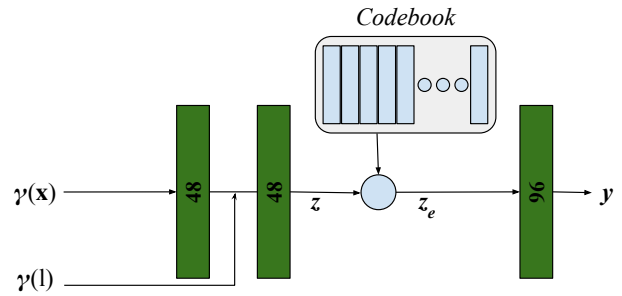


Figure 2. A diagram of "Latent Generator" network. This network takes position-encoded 3D point $\gamma(x)$ and position-encoded camera level $\gamma(l)$. This is passed through the encoder block to get z which is than matched to the nearest latent in the codebook to get z_e. z_e is passed through decoder block to reconstruct the position-encoded 3D point y.

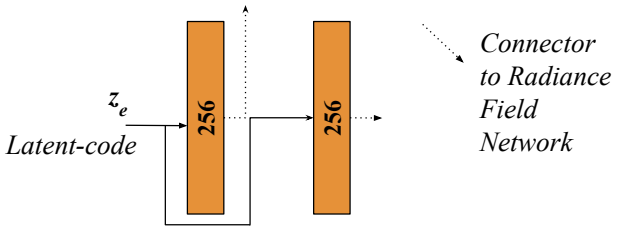


Figure 3. A diagram of "Latent Router" network. This network takes latent code z_e generated by the "Latent Generator" and connects it to the radiance field network after passing through linear layers.

Table 2. Performance on the *Dragon In Pyramid* dataset between two configuration files. We observe that "*360.gin*" works much better than the other configuration file.

Config	Level 0			Level 1			Total		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Blender	5.5654	0.3717	0.6252	22.9489	0.6320	0.5844	14.2571	0.5018	0.6048
360	**30.8758**	**0.9006**	**0.1367**	**24.3890**	**0.7054**	**0.5163**	**27.6324**	**0.8030**	**0.3265**

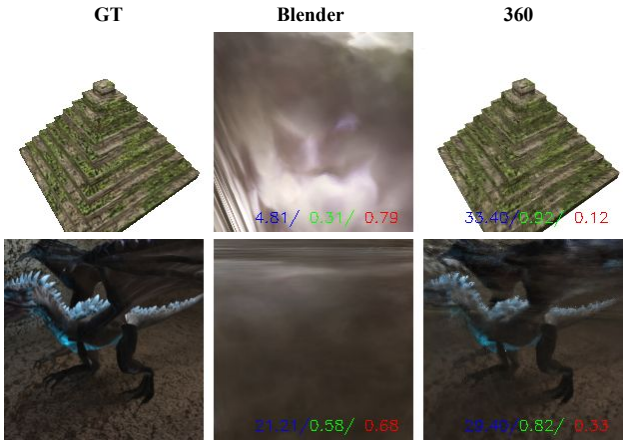


Figure 4. Qualitative comparison for different configuration files on *Dragon In Pyramid* scene. We observe that *360.gin* configuration generates better results. Metrics PSNR,SSIM and LPIPS are color-coded at the bottom of the result image

Table 3. No. of training parameters **(in millions)** for level-wise mip360 and our method with two different codebook sizes 1024 and 4096 for different number of levels.

Levels	Level-Wise mip360	Ours (1024 codebook)	Ours (4096 codebook)
1	0.835	0.924	1.071
3	2.506	0.924	1.071
4	3.341	0.924	1.071
5	4.176	0.924	1.071
6	5.011	0.924	1.071

Figure 5. Analysis on "7 Rutledge Ave" scene from RealEstate10K [13] dataset. We present visual results from two levels. Note how artifacts appear in results from mipNeRF-360 (all levels are trained jointly) whereas when mipNeRF-360 is used for each level separately (level-wise) we observe no artifacts.

Table 4. A quantitative comparison of mip360 (level-wise) and mipNeRF-360 (all views) on "7 Rutledge Ave"

Methods	Lv 0	Lv 1	Lv 2	Lv 3	Lv 4	Lv 5	Lv 6	Total
mipNeRF-360 (x7)	**24.20**	22.42	**26.72**	24.78	22.73	**27.41**	24.78	24.25
mipNeRF-360	19.53	18.33	23.52	17.00	18.82	19.73	21.60	19.62

E. Additional Experiments

E.1. RealEstate10K [13] scene - Motivation Experiment

We presented motivation of our work on a synthetic scene "Dragon In Pyramid" in Section 4 in the main paper. We observed that no artifacts are observed if individual mipNeRF-360 is trained for each level (level-wise) separately. We performed a similar experiment on the RealEstate10k [13] scene and observed artifact-free novel views from level-wise mipNeRF-360. Similar to the observation for synthetic scenes, if all levels are trained combinedly we observe the artifacts in the rendered novel-views as shown in Fig 5. Further, PSNR values in Tab. 4 for level-wise mipNeRF-360, with 7 radiance fields (x7) are higher compared to a single mipNeRF-360 for all-levels. This further substantiates our claim that a single mipNeRF-360 network is not able to learn all the stratified levels.

Table 5. A quantitative comparison of InstantNGP [10] and TensoRF [4] on "7 Rutledge Ave"

Methods	Lv 0	Lv 1	Lv 2	Lv 3	Lv 4	Lv 5	Lv 6	Total
Instant-NGP	19.02	18.24	21,32	19.43	18.77	18.98	21.33	19.47
TensoRF	18.03	21.29	21.23	20.23	20.36	18.57	22.69	20.70
Ours	**22.84**	**25.14**	**24.83**	**25.67**	**25.15**	**23.10**	**26.75**	**25.04**

Table 6. Performance on the *Coffee Shop* dataset for different sizes of the vector codebook. **Best** results are marked in bold and Second-best results are underlined.

Size	Level 0			Level 1			Level 2			Total		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
512	24.4768	0.8605	0.2049	28.0758	0.8257	0.3632	**33.7944**	0.9306	**0.2003**	28.7824	0.8723	0.2561
1024	**26.4497**	**0.8803**	**0.1936**	**28.6387**	**0.8403**	**0.3449**	33.2695	0.9254	0.2243	**29.4526**	**0.8820**	0.2543
4096	25.3534	0.8729	0.1995	28.4341	0.8383	0.3539	33.6062	**0.9316**	0.2025	29.1312	0.8809	**0.2520**

E.2. Comparison with InstantNGP [10] and TensoRF [4]

Synthetic Scenes. We present qualitative comparison with InstantNGP [10] and TensoRF [4] in Fig. 6 and **??**. These methods work well in the outermost level. But suffer from artifacts because of the stratified scenes in the inner levels. We observe this pattern consistenly across all the synthetic scenes.

RealEsate10K [13] dataset Fig. 7 shows qualitative comparison on "7 Rutledge Ave" scene from RealEstate10K [13]. Our method generates novel-view without any artifact, whereas other methods have visible artifacts in the generated novel-views. Tab. 5 shows PSNR of the generated novel-views. Our method clearly outperforms InstantNGP [10] and TensoRF [4].

E.3. Comparison with level-wise radiance fields.

One trivial solution for the proposed stratified setting is training mip360 individually for multi-view images in each level. We show that with increase in no. of levels, no. of training parameters increases linearly. Consider a mip360 network with width 256 and depth 8. We present varaition of no. of training parameters in Table 3 for different number of levels. Our method's training parameter requirement doesnot increase linearly as it does in level-wise mip360.

For comparison, on "Spanish Colonial Retreat" scene, mipNerf-360 takes *5h 30m* to train, while our method, with a vector-codebook size of 1024, takes *6h 20m* for 150k iterations on a single NVIDIA RTX 3090 GPU.

E.4. Ablation on Vector-Codebook Size

We present more results on *Coffee Shop, Bhutanese House and Buddhist Temple* for the ablation : *Size of the vector-codebook in "Latent Generator"*. We tried with three sizes : 512, 1024 and 4096. Table 6 and 7 shows the quantitative results for the mentioned datasets. We observe that vector codebook of size 1024. gives us the overall best results.

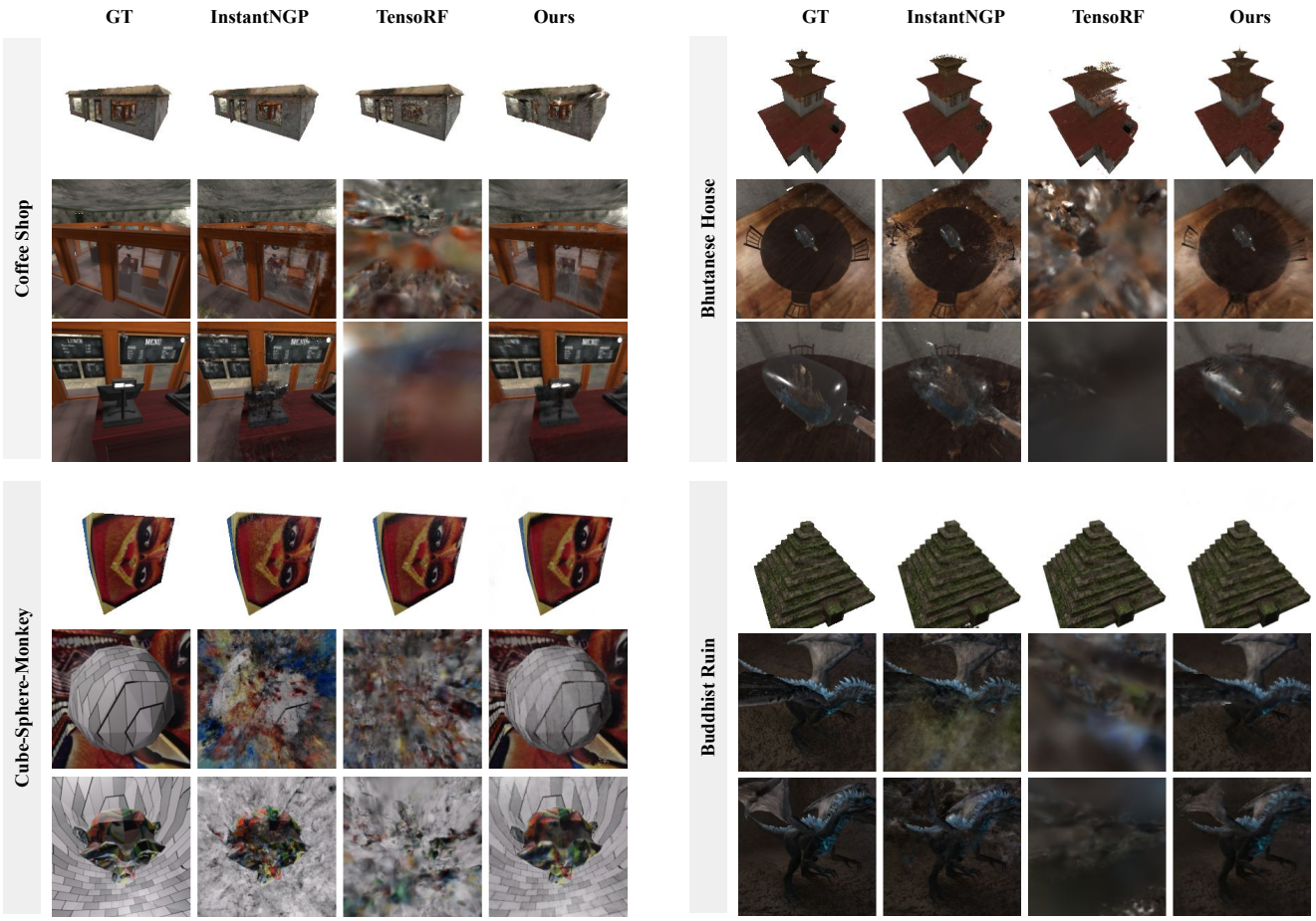


Figure 6. Qualitative Comparison on synthetic dataset for InstantNGP [10] and TensoRF [4]

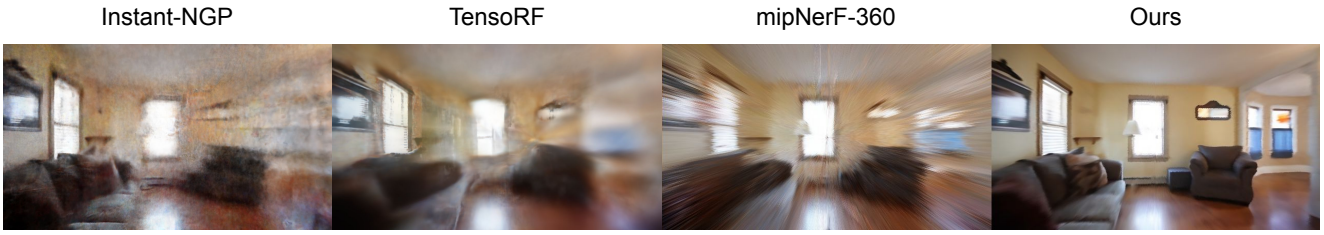


Figure 7. Qualitative Comparison on "7 Rutledge Ave" scene from RealEstate10K [13] dataset. The novel-view generated from our method is better than InstantNGP [10], TensoRF [4] and mipNeRF-360 [1]

Table 7. Performance on the *Buddhist Temple* dataset for different sizes of the vector codebook. **Best** results are marked in bold and Second-best results are underlined.

Size	Level 0			Level 1			Level 2			Total		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
512	27.3121	0.8881	0.1861	25.3407	0.7619	0.362	25.4983	0.7476	0.3691	26.2306	0.8119	0.2886
1024	**27.5529**	**0.8935**	**0.1775**	**27.3453**	**0.7894**	**0.3240**	**25.5956**	**0.7717**	**0.3456**	**26.9343**	**0.8289**	**0.2674**
4096	20.9017	0.8075	0.2680	27.0011	0.7856	0.3340	23.4656	0.7189	0.3853	23.3769	0.7759	0.3204

E.5. Architectural Design Choices.

The proposed method consists of Latent Generator (LG) and Latent Router(LR) as shown in Figure 4 in the main paper. Latent Generator(LG) and Latent Router(LR) are described in Section 5.1 and 5.2 respectively in the main paper. To further motivate this choice of the architecture, we discuss the following design choices for the proposed method:

1. Disabling the second router in **LR**: **D1**
2. Disabling the first router in **LR**: **D2**
3. removing **LR** and directly concatenating the generated embedding to the input positional encoding : **D3**

Table 8. Ablation studies on the key design choices for the proposed method. **D1**: Disable second router in LR, **D2**: Disable first router in LR, **D3**: Remove LR and directly concatenate generated embedding with the positional encoding and **D4**: Replace VQ-VAE with VAE in LG. Acronyms D1, D2, D3, D4 are explained in more detail in Appendix E.5

	D1	D2	D3	D4	Ours
Synthetic	26.04	27.34	27.41	26.96	**28.25**
RealEstate10K	23.79	24.24	23.79	20.99	**24.75**

Table 9. **Quantitative Comparison on "7 Rutledge Ave"**

Ours-Ind.	21.03	23.54	24.15	23.85	22.83	22.64	25.41	23.53
Ours	22.84	**25.14**	24.83	**25.67**	**25.15**	23.10	**26.75**	**25.04**

4. Replacing the VQ-VAE block with the VAE block in **LG** : **D4**

We present overall results for synthetic and RealEstate10K scenes in Tab. 8. We conclude that using two parallel dense layers is better than an individual dense layer in **LR**. Further, we observe that how using Latent Router is better than directly concatenating the generated embedding with the input positional embedding. Similarly, the VAE version of our method underperforms the discrete VQ-VAE used in our method.

E.6. Why shared codebooks are important?

We provide another ablation by creating independent code-book vectors for different levels : "Ours-Ind.". In our method, codebooks are shared between level which yield better results. This is natural as walls, etc. are shared between levels in the scene.

E.7. Experiments on the standard novel-view synthesis dataset.

We train the "garden" scene from the mipNeRF-360 dataset by treating it as a single-level scene. We achieved a PSNR of 26.40 on the test dataset, while mipNeRF-360 reports a PSNR of 26.98. We achieve an average PSNR of 33.21 across all NeRF-synthetic scenes, while mipNeRF-360 achieves 33.09. Our proposed method performs comparably on these datasets, despite being designed for stratified scenes.

E.8. Number of Views

We present here another ablation which evaluates the effect of increasing number of views for a scene. Table 10 shows quantitative results on *Dragon In Pyramid* scene by increasing number of views 2× and 3×. Note that 2× views mean that train, validation and test views will be doubled. We observe that as number of views are increased, overall metrics improves in both mip360 [1] and our method. Further, we compare qualitative performance of our method with mip360 [1] with increased number of views in Figure 8 and 9. We observe that quality of depth map is much better in our method. Also, generated novel views from our method has less artefacts.

Table 10. Performance on the *Dragon In Pyramid* dataset for different number of views in the dataset. **Best** results are marked in bold.

		Level 0			Level 1			Total		
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
1x Views	mip360	**30.8758**	**0.9006**	**0.1367**	24.3890	0.7054	0.5163	27.6324	0.8030	0.3265
	Ours	29.4773	0.8700	0.1699	**26.1722**	**0.7489**	**0.4573**	**27.8248**	**0.8095**	**0.3136**
2x Views	mip360	**29.5127**	**0.8436**	**0.1830**	26.2172	0.7245	0.4627	27.8650	0.7841	0.3228
	Ours	29.1104	0.8099	0.2176	**27.4282**	**0.7661**	**0.4244**	**28.2693**	**0.7880**	**0.3210**
3x Views	mip360	**31.1511**	**0.8764**	**0.1715**	26.5231	0.7239	0.4638	28.8371	0.8001	0.3176
	Ours	30.5436	0.8461	0.1882	**27.4354**	**0.7693**	**0.4385**	**28.9895**	**0.8077**	**0.3134**

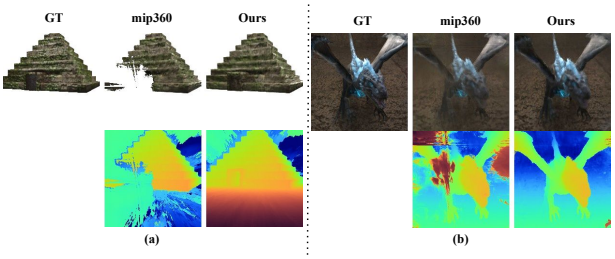


Figure 8. Qualitative Results for 2× views on *Dragon In Pyramid* scene. Observe that our results have less artefacts and much smoother depth maps.

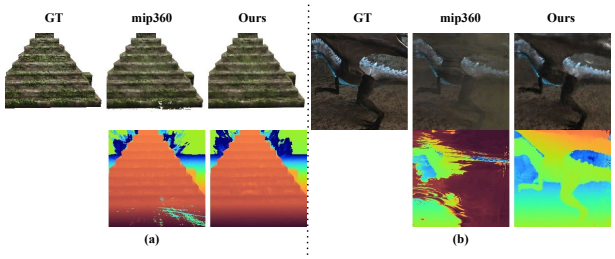


Figure 9. Qualitative Results for 3× views on *Dragon In Pyramid* scene. Observe that our results have less artefacts.

E.9. Out of Distribution Views

The training set's views are uniformly sampled from the curved surface of a hemisphere with the camera's $z - axis$ always pointing towards the subject. Out-of-distribution (OOD) is any new view that does not lie on this hemisphere and whose $z - axis$ is not necessarily aligned with the subject. We investigated the quality of novel view synthesis for OOD views. We apply a random rotation and translation to the camera pose in the test set to produce OOD camera poses. A random translation value is sampled uniformly between $(10cm, 10cm)$, which is then used to translate the camera position along its $z - axis$. We randomly choose the rotation axis and angle from $(-45°, 45°)$ for random rotation and change the current pose with this transformation. Figure 10 shows the novel views and their corresponding depth maps. The depth map shows that our technique

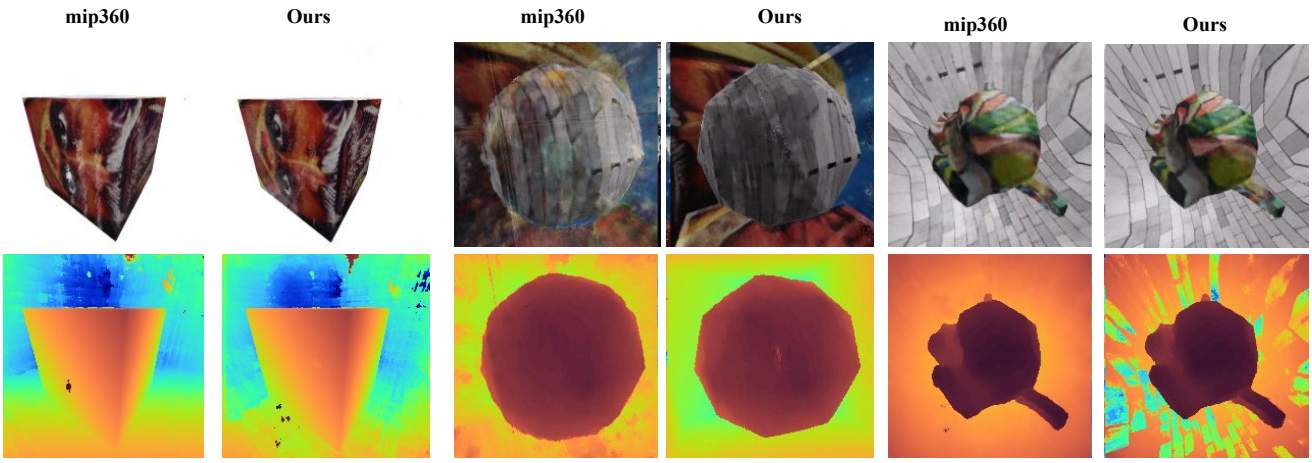


Figure 10. Qualitative comparison on OOD views. (**Top Row**) Generated novel views. (**Bottom Row**) Corresponding depth map. Check the quality of depth maps in inner levels for our method.

regularises the 3D geometry significantly better than other methods. Furthermore, the depth map quality is substantially better, which aids our method in producing non-blurry results.

E.10. Additional Results

We provide more results for the Out Of Distribution views in Figure 11. Further, we provide a sequence of generated novel views for *Cube-Sphere-Monkey* in Figure 12 and a sequence of depth maps for the *Buddhist Temple* in Figure 13. There are distinct artefacts in column one and three in Figure 11(a), column one in 11(b) and column three in 11(b). We compare the generated depth maps in Figure 11 and Figure 13. We observe that the depth maps from our method are smooth and have less artefacts than Mip-NeRF 360 [1]. Notice the collapse in floor of the *Buddhist Temple* scene in Figure 13. From these results, it's clear that the generated novel views from our method has less artefacts and better 3D representation of such stratified scenes.

E.11. Impact of Image-Resolution on training.

On 800×800 resolution for "Cube-Sphere-Monkey" scene, mipNeRF-360 achieves an overall PSNR of 23.17 and our method achieves 26.41. This is similar to behavior observed on low-resolution (200×200) and high-resolution RealEstate10K.

References

[1] 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. 2, 3, 5, 6, 7, 8, 10

[2] birdhousemediatv. 139 barton avenue, toronto, ontario. 2

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: composable transformations of python+ numpy programs. *Version 0.2*, 5:14–24, 2018. 2

[4] 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. 1, 4, 5

[5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 1, 2

[6] Hugging Face. *Stable Diffusion Demo*. 1

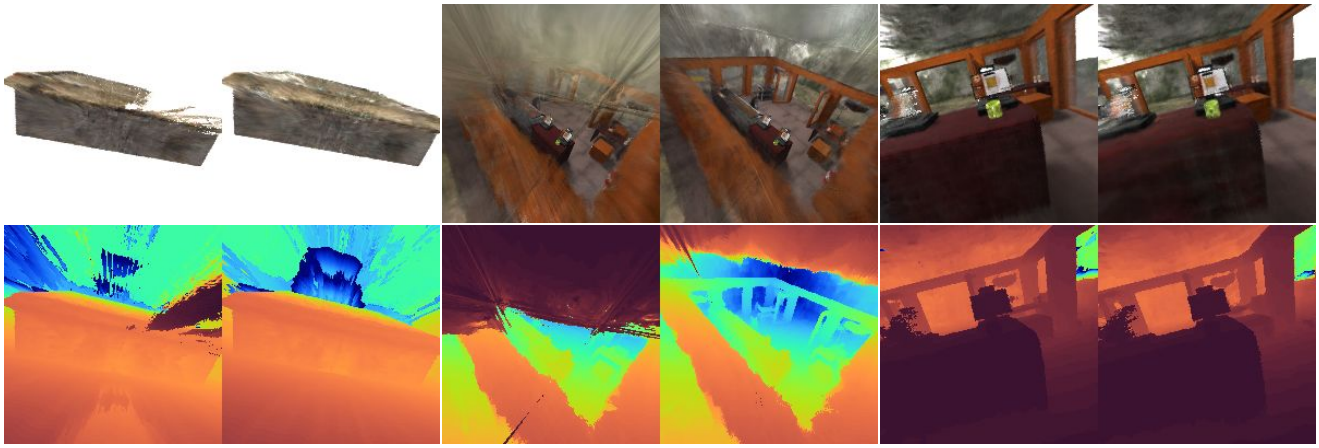
[7] HomeTourVision. Real estate video tour — 7 rutledge ave, highland mills, ny 10930 — orange county, ny. 2

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

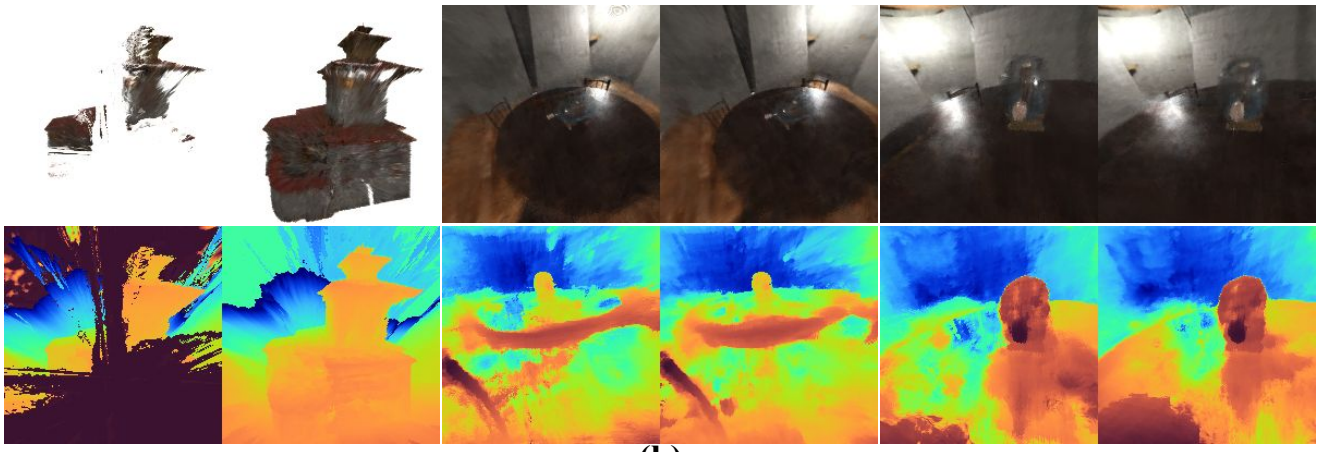
[9] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 3

[10] 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. 1, 4, 5

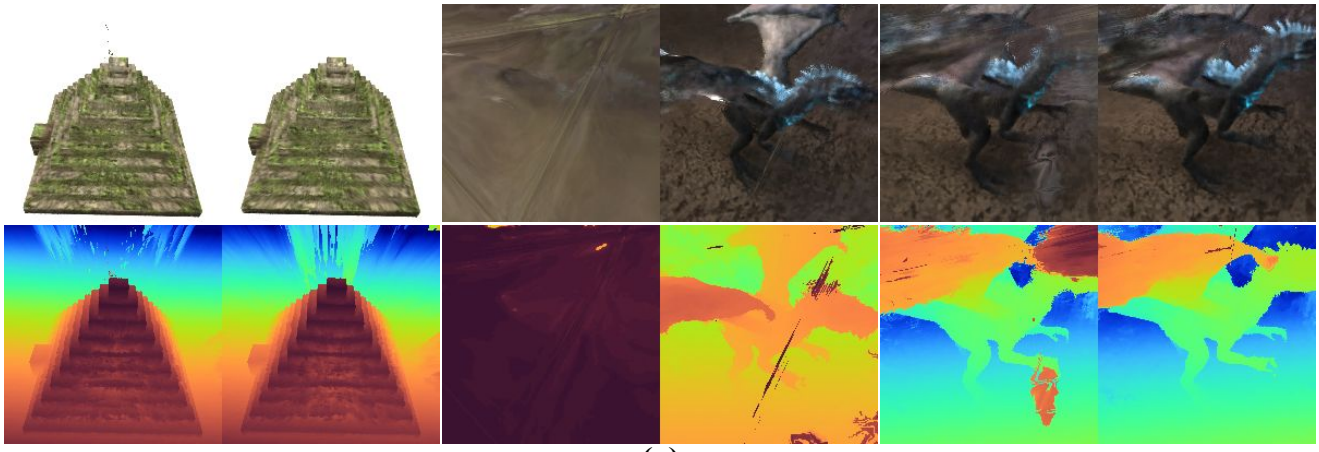
[11] Sotheby's International Realty. Spanish colonial retreat in scottsdale, arizona. 2



(a)



(b)



(c)

Figure 11. Out of distribution views for (a) Coffee Shop, (b) Bhutanese House and (c) Dragon In Pyramid Scene. **Odd** columns are results from Mip-NeRF 360 [1] and **even** columns are results from our method. We observe that generated novel views from our method has less artefacts and better depth maps. Check the clarity in claws of dragon in last column of (c).

[12] Bayer Video Tours. 31 brian dr, rochester, ny presented by bayer video tours. 2

[13] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning

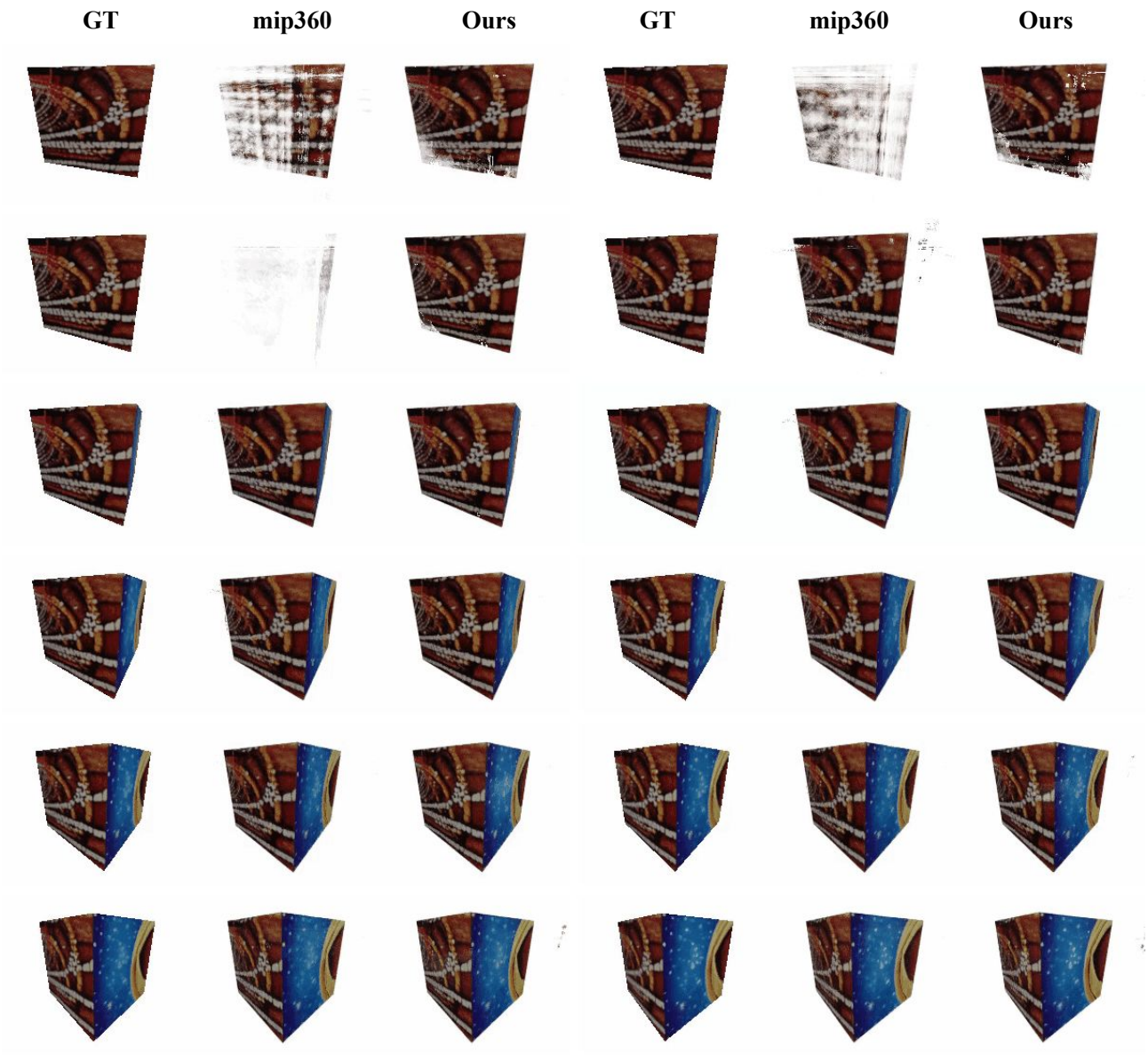


Figure 12. Sequence of generated novel views for Level 0 of *Cube-Sphere-Monkey* scene. Please note that sequence is represented in zig-zag pattern. The generated novel views from our method has less artefacts. **Please check the video provided in the supplementary material to appreciate our results better.**

view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 1, 2, 4, 5

Figure 13. Sequence of depth maps of generated novel views for Level 1 of *Buddhist Temple* scene. Please note that sequence is represented in zig-zag pattern. We observe that there is a collapse in the floor region for output from mip360 [1] output. Whereas, our method generates smooth depth maps. **Please check the video provided in the supplementary material to appreciate our results better.**