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]Vision and AI Lab, IISc Bangalore [2]Samsung R & D Institute India - Bangalore [3]Brown University

Abstract

Neural Radiance Field (NeRF) approaches learn the underlying 3D representation of a scene and generate photo-realistic novel views with high fidelity. However, most proposed settings concentrate on modelling a single object or a single level of a scene. However, in the real world, we may capture a scene at multiple levels, resulting in a layered capture. For example, tourists usually capture a monument's exterior structure before capturing the inner structure. Modelling such scenes in 3D with seamless switching between levels can drastically improve immersive experiences. However, most existing techniques struggle in modelling such scenes. We propose Strata-NeRF*, a single neural radiance field that implicitly captures a scene with multiple levels.* Strata-NeRF *achieves this by conditioning the NeRFs on Vector Quantized (VQ) latent representations which allow sudden changes in scene structure. We evaluate the effectiveness of our approach in multi-layered synthetic dataset comprising diverse scenes and then further validate its generalization on the real-world RealEstate10K dataset. We find that* Strata-NeRF *effectively captures stratified scenes, minimizes artifacts, and synthesizes high-fidelity views compared to existing approaches.* https://ankitatiisc.github.io/Strata-NeRF/

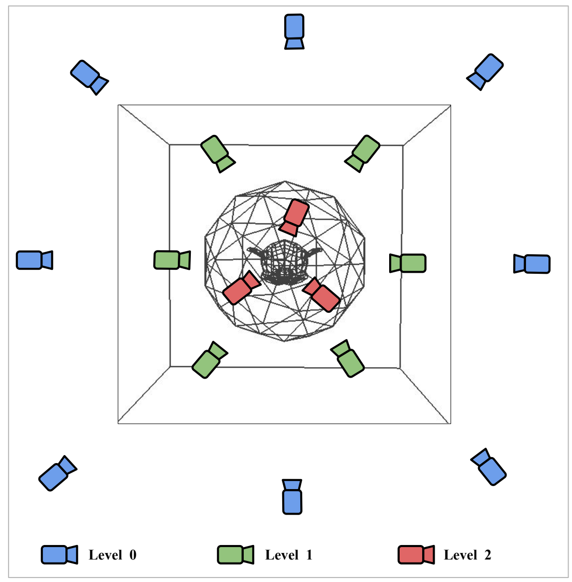

Figure 1. Top, wireframe view of a multi-layered stratified scene with three levels (monkey head inside sphere inside a cube). The camera colors indicate views of a specific level. *Strata-NeRF* enables high-quality reconstruction of such stratified scenes using a single neural network.

1. Introduction

Novel view synthesis is an ill-posed problem widely encountered in various areas such as augmented reality [24, 28], virtual reality [11], etc. A paradigm change for solving these kinds of problems was brought by the introduction of Neural Radiance Fields (NeRF) [34]. NeRFs are neural networks that take in the spatial coordinates and camera parameters as input and output the corresponding radiance field. Earlier version of NeRFs enable the generation of high-fidelity novel views for bounded scenes, significantly improving over existing techniques like Structure From Motion [47]. Further, the capability of NeRFs have been recently extended to model unbounded scenes by Mip-NeRF 360 [2]. This enabled NeRFs to model complex real-world scenes, where the scene content can exist at any distance from the camera.

However, similar to unboundedness in scenes, hierarchies in scenes are also natural. For example, images captured in a house can be categorized into images captured outside and inside across various rooms. Modelling such hierarchical scenes jointly for all levels through a NeRF could be particularly useful in cases of Virtual Reality applications. As it would not require switching to a different NeRF for each level, reducing memory requirement and latency in switching. Further, as the different hierarchies of a scene usually share texture and architectural commonalities, it could lead to effective knowledge sharing and reduce the requirement of training independent models. For tackling the above novel objective, we introduce a paradigm of scenes that can be deconstructed into several tiers, termed *"Stratified Scenes"*. A "stratified" scene has several levels or

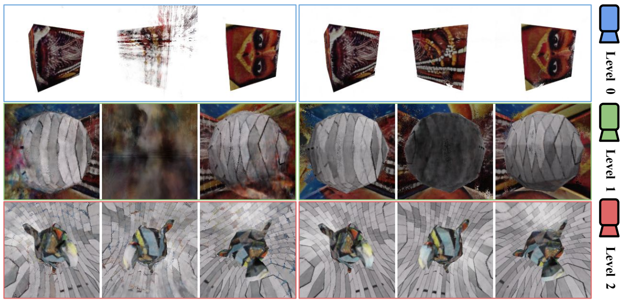

Figure 2. Novel views for stratified scene in Figure 1, from Mip-NeRF 360 [2] *(left)* and our method *"Strata-NeRF" (right)*. Existing methods struggle to capture stratified scenes with a single network while ours produces sharp results.

groupings of structure (Figure 1). In our work, we first propose a synthetic dataset of stratified scenes, i.e. scenes having multiple levels. This dataset comprises scenes from two categories: (i) Simpler geometry, such as spheres, cubes, or tetrahedron meshes, and (ii) Complex geometry, which closely emulates a real-world setup.

On such datasets, we find methods such as Mip-NeRF 360 perform well for each level of the hierarchy independently, but produce unsatisfactory results when images from all hierarchical levels are used together for training (Figure 3). This can be attributed to the continuous nature of NeRFs, which is unsuitable for modelling the sudden changes in scenes with shifts in hierarchical levels. Hence, in this work, we introduce *Strata-NeRF* that explicitly aims to model the hierarchies by conditioning [22, 38, 39, 66, 43] the NeRF on Vector Quantized (VQ) latents. The VQ latents enable the modelling of discontinuities and sudden changes in the scene, as they are discrete and less correlated with others [56]. In practice, the VQ conditioning is achieved by introducing two lightweight modules: the "Latent Generator" module that compresses the implicit information in encoded 3D positions to generate VQ latent code, which is directed through the "Latent Routing" module to condition various layers of radiance field. The additional parameters introduced through these modules are significantly less than training an independent NeRF model for each level, leading to a significant reduction in memory.

For evaluating the proposed *Strata-NeRF* we first test on the proposed synthetic *Stratified Scenes* dataset, where we find that *Strata-NeRF* learns the structure in scenes across all levels. In contrast, other baselines produce cloudy and sub-optimal novel views (Figure 2). Further, to test the generalizability of the proposed method on real-world scenes, we utilize the high-resolution RealEstate10K dataset. We find that *Strata-NeRF* significantly outperforms other baselines and produces high-fidelity novel views without artifacts compared to baselines. This is also observed quantitatively through improvement in metrics, where it establishes a new state-of-the-art. In summary,

- We first introduce the task of implicit representation for 3D stratified (hierarchial) scenes using a single radiance field network. For this, we introduce a novel synthetic dataset comprising of scenes ranging from simple to complex geometries.
- For implicit modelling of the stratified scenes, we propose *Strata-NeRF*, which conditions the radiance field based on discrete Vector-Quantized (VQ) latents to model the sudden changes in scenes due to change in hierarchical level (i.e. strata).
- *Strata-NeRF* significantly outperforms the baselines across the synthetic dataset and generalizes well on the real-world scene dataset of RealState10k.

2. Related Work

Generating photo-realistic novel views from densely sampled images is a classical problem. Earlier methods solved this issue using light-field-based interpolation techniques [10, 18, 27]. These techniques interpreted the input images as 2D slices of a 4D function - the light field. The only caveat in these methods is their overreliance on dense views. Another popular technique is Structure From Motion (SFM) which reconstructs 3D structure of a scene or an object by using a sequence of 2D images. We suggest readers to read survey papers [47, 37] to understand SFM methods in detail. Shum *et al*. [49] also provides an excellent review on traditional image based rendering techniques.

Neural Volume Reconstruction. NeRF [34] has shown remarkable results in encoding the 3D geometry of a scene implicitly using the multi-layer perceptron (MLP). Specifically, it trains an MLP, which takes 3D position and a viewing direction to predict colour and occupancy. Many papers have extended this idea to solve different scenarios such as dynamic scenes, low-light scenes, synthesis from fewer views, accelerating the performance etc. Mip-Nerf [1] mitigates the problem of aliasing when a novel view is generated at a different resolution. MVSNeRF [7] generalizes across all the scenes and optimizes the geometry and radiance field using only a few views. NerfingMVS [61] utilizes conventional SFM reconstruction and learning-based priors to predict the radiance field. UNISURF [36] combines implicit surface models and radiance fields to render both surface and volume rendering.

AR-NeRF [24] replaced pin-hole based camera ray-tracing with aperture camera based ray-tracing. DiVeR [62] uses a voxel based representation to learn the radiance field, Mip-NeRF 360 [2] improves view synthesis on the unbounded scenes and also proposed an online distillation scheme which significantly reduced the training and inference time. Neural Rays [31] solves the occlusion problem by predicting the visibility of the 3D points in their representation. Scene Representation Transformers [46] uses Vision Transformers [13] to infer latent representations to render the novel views. Further, many methods [30, 17, 44, 65, 51, 21, 58] have been proposed to improve the slow training and inference time for neural radiance field based methods. Despite many works, no work has focused on modelling the *stratified* scenes.

NeRF Extensions. Relighting discusses how to model different types of light and then using this model to relight a scene [32, 3, 50, 57, 20]. Breaking the myth that radiance field can only be used in small and bounded scenes, recent methods [52, 55, 45] have scaled it to large-scale city scenes. Another line of work focuses on modelling the dynamic scenes with presence of moving objects [38, 63, 29, 41, 14, 54, 16, 39] through NeRFs.

Neural Radiance Fields and Latents. Recently, a lot of methods have made use of the latents to bring generative capabilities to neural radiance fields. GRAF [48] uses disentangled shape and appearance latent codes to generalize on an object category. For viewpoint invariance, they used typical GAN based training. Pi-GAN [5] uses volumetric rendering equations for consistent 3D views in a generative framework. Pixel-NeRF [66] learns a scene prior to generalize across different scenes. GSN [12] decomposes the radiance field of a scene into local radiance fields by conditioning on a 2D grid of latent codes. Code-NeRF [22] learns the variation of object shapes and textures across by learning separate latent embeddings. LOLNeRF [43] uses a shared latent space which conditions a neural radinace field to model shape and appearance of a single class. PixNerF [4] extends Pi-GAN [5] and maps images to a latent manifold allowing object-centric novel views given a single image of an object. NeRF-W [33] optimizes latent codes to model the scene variations to produce temporally consistent novel view renderings. In contrast to these methods, we propose conditioning NeRF on learnable Vector Quantized latents.

Vector Quantized Variational Autoencoders (VQ-VAE) [56]: VQ-VAE uses vector quantization to represent a discrete latent ditribution. VQ-VAE has shown applications in Image Generation [42, 40], speech and audio processing [19, 59]. Further, it's extension like VQ-VAE2 [42] uses hierarchical latent space for high-quality generation.

3. Preliminaries

NeRF represents a scene as an implicit function $f : (X, d) \rightarrow (c, \sigma)$ which maps a 3D position $X = (x, y, z)$ and $d = (\theta, \phi)$ to a color $c = (r, g, b)$ and occupancy density σ. An MLP parametrizes this implicit function f. Before sending the inputs X and d through the network, a positional encoding is used to project them in a high dimensional space [53]. Finally, the volume rendering [23] procedure enables NeRF to represent scenes with photo-realistic rendering from novel camera viewpoints.

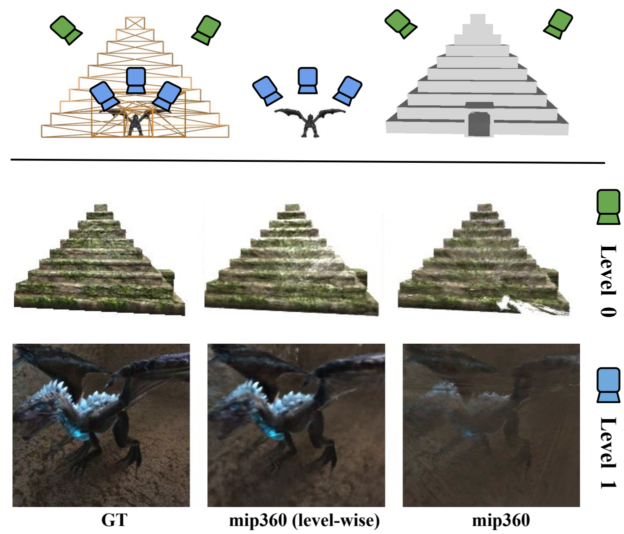

Figure 3. Analysis on "Dragon in pyramid" scene. The top row shows the layout of the levels in 3D scene. Observe that baseline works fine on the scenes when trained individually. Artefacts occur when the baseline is trained on views from the entire scene.

Volume Rendering. At the crux of NeRF lies the volume rendering equation. A ray $r(t) = o + td$ is cast from the camera center o through the pixel along direction d. The pixel's color value is estimated by integrating along the ray $r(t)$ as described in Eq. 1

$$c(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)\, dt \tag{1}$$

where transmittance $T(t) = exp(-\int_{t_n}^{t} \sigma(r(s))\, ds)$ is the probability that a ray passes unhindered from the near plane (t_n) to plane (t) and use this probability to integrate till far plane (t_f). In Mip-NeRF [1], a ray $r(t)$ is divided into intervals $T_i = [t_i, t_{i+1})$ which corresponds to a conical frustum. For each interval T_i, it computes the mean and variance (μ, Σ) and uses it for integrated position encoding as illustrated in Eq. 2.

$$\gamma(\mu, \Sigma) = \left\{ \begin{bmatrix} sin(2^l\mu)exp(-2^{2l-1}diag(\Sigma)) \\ cos(2^l\mu)exp(-2^{2l-1}diag(\Sigma)) \end{bmatrix} \right\}_0^{L-1} \tag{2}$$

This solves the aliasing issue in the original NeRF. Mip-NeRF 360 [2] proposed coarse-to-fine online distillation for proposal sampling, which efficiently reduces the training time as the proposed MLP only predicts density. They also proposed ray parametrization and regularisation techniques to alleviate hanging artifacts in unbounded scenes. *We'll refer Mip-NeRF 360 [2] as mip360 in all our discussions.* We choose mip360 [2] as the baseline for all our experiments.

Table 1. A quantitative comparison of mip360 (level-wise) and mip360 (all views) on "Dragon in pyramid" scene.

	Level 0			Level 1		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
mip360 (level-wise)	**31.5390**	**0.9181**	**0.1304**	**29.8560**	**0.8133**	**0.3484**
mip360	30.8847	0.9006	0.1367	24.3876	0.7055	0.5163

4. Motivation

The majority of real-world scenarios are stratified with multiple levels. For example, a commodity store has exterior and interior structures. This work addresses an essential question for such stratified scenes: *Can a single radiance field learn such hierarchical scenes?* This section introduces and discusses our observations on one such stratified scene: "Dragon in Pyramid", as illustrated in Figure 3. The outer structure of "Dragon in Pyramid" is a Mayan pyramid that has a dragon inside it. To validate our claim, we first train the baseline model on each level, i.e., on outer pyramid views and inner views (focusing dragon) independently. We refer to these separately trained models as *mip360 (level-wise)*. Then, we train a single *mip360* model using the outer and inner views for the scene. The term "level" in our work refers to each level in a stratified scene. In the scene depicted in Figure 3, level 0 denotes the pyramid's outer construction, while level 1 denotes the pyramid's interior structure, which contains a dragon.

Table 1 shows that the baseline model performs remarkably well when trained separately on each level. In comparison, the metric values for the baseline model trained jointly on both levels of stratified scene declines. PSNR at level 1 is $24.39\,dB$, a $5.47\,dB$ reduction compared to mip360 (level-wise). Similarly, performance in level 0 has declined, but less dramatically than in the inner level. This pattern is observed across all metrics. Furthermore, the qualitative results illustrated in Figure 3 backs up the quantitative study's findings. Figure 3 indicates that mip360 (level-wise) generates novel views on par with the ground truth. However, shown in Figure 3, the jointly trained model has white artifacts on the pyramid's outer structure and haziness in front of the dragon inside the pyramid. This demonstrates that current radiance field networks have issues while learning a 3D representation of a stratified scene. We perform a similar experiment for a RealEstate10K scene in Appendix **E.1** in the supplmentary material.

5. Method

This section describes our method : *Strata-NeRF* for stratified scenes. We generate latent codes with the latent generator described in Section 5.1. This latent code is fed into the radiance field architecture through the latent router, described in Section 5.2. Figure 4 depicts the overall architecture of Strata-NeRF. We adopt the base neural radiance field architecture proposed in mip360 [2].

5.1. Latent Generator

A latent space reflects the scene's "compressed" representation. It has been shown in various works that this space has rich properties. VQ-VAE [56] learns a codebook to model the discrete distribution of the latent space of a variational-autoencoder. The encoder's output is compared to all of the vectors in the codebook. The nearest vector is fed into the decoder as input. Since most data in the world is discrete, VQ based models have been highly successful in image generation [15], speech encoding [56], and other applications. In a stratified scene, the definition of level is also discrete. Hence, our method employs VQ-VAE as a latent generator because of their proven success in representing discrete distributions.

We use Integrated Positional Encoded (IPE) [2] $\gamma(\mathbf{x})$ as input to our latent generator. We encode $\gamma(\mathbf{x})$ and then search the codebook for the closest vector. After that, the closest vector from the codebook is used to condition the radiance field network. Specifically, $\gamma(\mathbf{x})$ is passed through a set of two hidden layers to generate an encoded input $\mathbf{z}$. The encoded latent code $\mathbf{z}$ is then passed through the quantizer bottleneck to determine the quantized latent code $\mathbf{z_e}$, where $\mathbf{z_e} \in E$; where $E \in R^{N\times D}$ is the codebook; N is the number of vectors in the codebook, and D is the dimension of the latent space. $\mathbf{z_e}$ is then supplied into the decoder network, which consists of two hidden layers, to yield $\mathbf{y}$ as the reconstructed output of $\gamma(\mathbf{x})$. The quantized latent $\mathbf{z_e}$ is also sent into the radiance field network through the "Latent Router" block. Loss for this variational autoencoder (VAE) block is defined as follows:

$$\mathcal{L}_{vq} = ||\gamma(\mathbf{x}) - \mathbf{y}||_2^2 + ||sg(\mathbf{z_e}) - \mathbf{z}||_2^2 + \beta||\mathbf{z_e} - sg(\mathbf{z})||_2^2 \quad (3)$$

The "Latent Generator" module based on VAE is jointly trained with the NeRF through backpropagation.

5.2. Latent Router

The Latent Router block is inspired by the CodeNeRF architecture [22], in which shape and texture latent codes are sent to the NeRF MLP through a residual connection. In our architecture, the quantized latent codes z_e that are generated in the "Latent Generator" block are input to the Radiance field after passing through an MLP layer in the Latent Router as shown in Figure 4.

5.3. Training Strata-NeRF

For training Strata-NeRF, we utilize the losses suggested by mip360 [2] as we use a similar radiance field design. $\mathcal{L}_{recon}(c(r,t), c^*(r))$ denotes the reconstruction loss between the estimated colour along a ray and the actual colour value. $\mathcal{L}_{dist}(s, w)$ is the distortion loss where s is the normalized ray distances and w is the weight vector. Note that

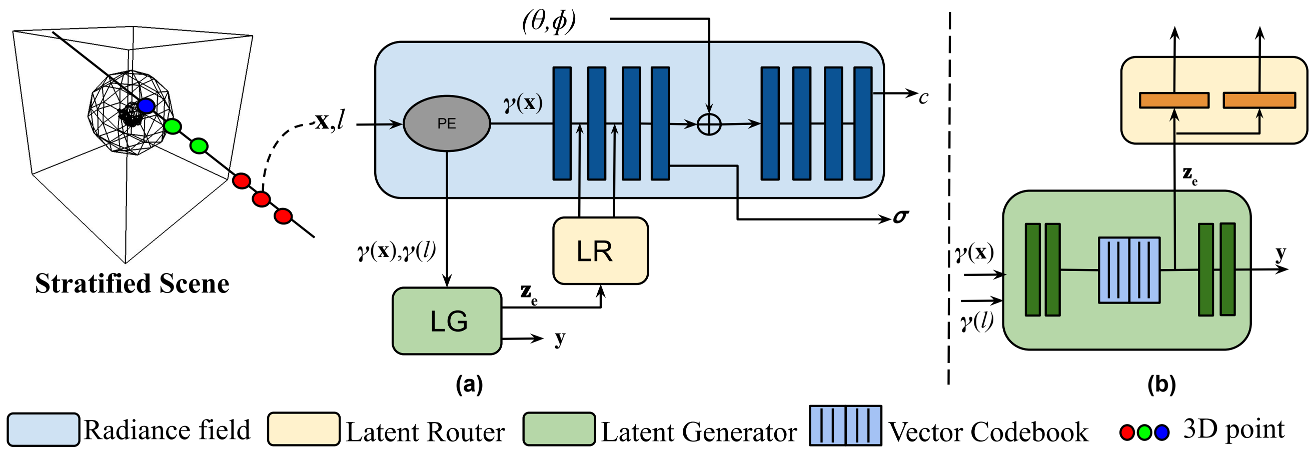

Figure 4. For each $3D$ point along the projected ray, we generate a latent code using our "Latent generator" module. The generated latent code is routed to the MLP using "Latent Router". Vector Codebooks learn the discrete distribution of positionally encoded $3D$ points. (a) Our model's end-to-end architecture; (b) components of the "Latent Generator" and "Latent Router" blocks.

Table 2. Characteristic Comparison of the proposed methods

Method	Discrete Representation	Photometric Losses	VAE loss
NeRF [34]	✗	✓	✗
mip360 [2]	✗	✓	✗
Plenoxel[64]	✓	✓	✗
Instant-NGP[35]	✓	✓	✗
TensoRF[6]	✓	✓	✗
Ours	✓	✓	✓

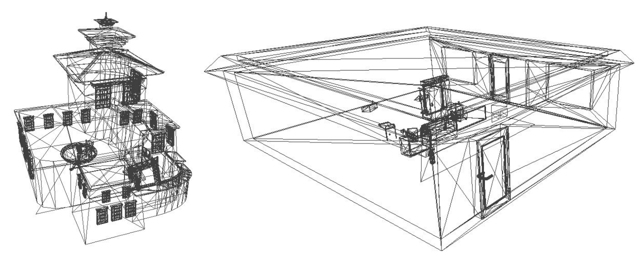

Figure 5. Skeleton mesh of the stratified scenes : Bhutanese House and Coffee Shop. More details are in the supplementary material.

we don't alter anything in the proposal MLP. More details are provided in mip360 [2]. The total loss for Strata-NeRF is given as:

$$\mathcal{L}_{total} = \mathcal{L}_{recon}(c(r,t), c^*(r)) + \lambda_1 \mathcal{L}_{dist}(s,w) + \lambda_2 \mathcal{L}_{vq} \tag{4}$$

We use $\lambda_1 = 0.01$, $\lambda_2 = 0.1$ and $\beta = 1.0$ across all our experiments, as they work robustly [2] for *Strata-NeRF*.

6. Experiments

We discuss implementation details in Section 6.1. Section 6.2 discusses the dataset used for evaluating our method with other baselines. In Section 6.3, we present quantiative and qualitative comparison with the baseline methods. Additionally, we discuss the ablations for the proposed method.

6.1. Implementation Details

Our method builds on mip360 [2] as the base radiance field. We use a latent generator network which consists of an encoder-decoder architecture and a vector-codebook. The encoder has two linear layers of hidden size 48, and the decoder has one linear layer of hidden size 96. The output dimension of our decoder matches the output from Integrated Positional Encoding (IPE) block. The size of our codebook is 1024, and the dimension of each vector in the codebook is 48. We condition the neural radiance field through the latent generated after the quantization step in the latent generator. We use a Latent routing module consisting of two linear layers of hidden-size 256. As illustrated in Figure 4, the output of the linear layer in the routing module conditions the first two layers of the radiance field network. We employ the losses outlined in Section 5. On each scene, we train our approach for $150k$ iterations. We use Adam [25] optimizer with a learning rate of $1e^{-6}$. Further details are provided in supplementary material.

6.2. Evaluation Dataset

Most of the radiance field methods evaluate their results on the synthetic (Blender) and real-world (LLFF) datasets proposed in NeRF [34]. These scenes either include a solitary object on a white background or a frontal view of a natural scene. According to our description of stratified scenes, these datasets has only one level. Even large-scale reconstruction datasets like TanksandTemples [26] are not representative of our setting as they only have views either inside or outside of the structure. Similarly, Scannet [9] a dataset for real-world interior scenes, lacks the characteristics of

Table 3. Quantitative evaluation on test-set against baselines discussed in Section 6.1. Each column is depicts the **best** and second best.

	Cube-Sphere-Monkey											
	Level 0			**Level 1**			**Level2**			**Total**		
	PSNR ↑	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
Nerf [34]	28.3314	0.9383	0.1034	18.1806	0.4976	0.4981	22.1178	0.5995	0.3825	22.8766	0.6784	0.3280
mip360 [2]	28.3149	0.9298	0.1156	19.0443	0.5343	0.4930	24.9136	0.7326	0.3245	24.0909	0.7322	0.3110
Plenoxels [64]	25.3547	0.9169	0.1238	13.1148	0.3320	0.6895	21.5568	0.6523	0.3803	20.0087	0.6337	0.3979
Instant-NGP [35]	28.2104	0.9168	0.1123	14.3648	0.1830	0.7216	17.6914	0.2744	0.5997	20.0889	0.4581	0.4779
TensoRF [6]	**32.0077**	**0.9532**	**0.0692**	13.7487	0.1537	0.7106	13.0075	0.2496	0.6886	19.5880	0.4521	0.4894
Ours	26.9335	0.9298	0.1255	**25.7088**	**0.7738**	**0.2959**	**26.1912**	**0.8172**	**0.2549**	**26.2778**	**0.8403**	**0.2254**

	Bhutanese House											
	Level 0			**Level 1**			**Level 2**			**Total**		
	PSNR ↑	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
Nerf [34]	11.4478	0.6917	0.3711	17.1209	0.5886	0.7078	18.3918	0.6952	0.6591	15.6535	0.6585	0.5793
mip360 [2]	26.6240	0.9002	0.2062	24.5946	0.7296	0.4739	29.4225	**0.8577**	0.4156	26.8804	0.8291	0.3652
Plenoxels [64]	15.2205	0.7752	0.3052	13.0386	0.4670	0.6703	19.3050	0.5819	0.5886	15.8547	0.6080	0.5214
Instant-NGP [35]	23.9791	**0.9217**	0.1500	24.7316	0.7009	0.4237	27.6617	0.8136	**0.3786**	25.4575	0.8121	**0.3174**
TensoRF [6]	13.8880	0.7607	0.3142	17.0244	0.4856	0.6421	16.8170	0.6306	0.6332	15.9098	0.6256	0.5298
Ours	**27.6842**	0.9046	0.2045	**24.9180**	**0.7371**	**0.4616**	**29.4646**	0.8575	0.4172	**27.3556**	**0.8331**	0.3611

Table 4. Quantitative evaluation on test-set against baselines discussed in Section 6.1. Each column is depicts the **best** and second best.

	Coffee Shop											
	Level 0			**Level 1**			**Level2**			**Total**		
	PSNR ↑	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
Nerf [34]	06.7446	0.6197	0.4698	16.1398	0.4915	0.7982	12.8889	0.4213	0.8158	11.9244	0.5108	0.6946
mip360 [2]	26.2073	0.8825	**0.1867**	27.0500	0.8086	0.3785	**34.2023**	**0.9362**	**0.1950**	29.1532	0.8757	0.2534
Plenoxels [64]	19.3204	0.7968	0.2579	12.3871	0.4044	0.6904	22.4325	0.6856	0.4585	18.0467	0.6289	0.4689
Instant-NGP [35]	29.9425	0.9324	0.0992	28.1040	0.8193	0.3452	29.6574	0.8680	0.2621	29.2347	0.8732	**0.2355**
TensoRF [6]	**33.0337**	**0.9435**	**0.0692**	19.3115	0.5331	0.6580	21.1852	0.7169	0.4594	24.5102	0.7312	0.3955
Ours	26.4499	0.8802	0.1939	**28.6392**	**0.8403**	**0.3450**	33.2692	0.9254	0.2243	**29.4528**	**0.8819**	0.2544

	Dragon In Pyramid											
	Level 0			**Level 1**			**Level 2**			**Total**		
	PSNR ↑	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
Nerf [34]	14.6405	0.6595	0.3800	20.8368	0.6052	0.6856	-	-	-	17.7386	0.6323	0.5328
mip360 [2]	30.8758	0.9006	0.1367	24.3890	0.7054	0.5163	-	-	-	27.6324	0.8030	0.3265
Plenoxels [64]	13.0667	0.6247	0.4217	14.5126	0.3572	0.6498	-	-	-	13.7896	0.4910	05358
Instant-NGP [35]	23.9054	0.9010	0.0949	24.7389	0.6594	0.4664	-	-	-	24.3222	0.7802	0.2807
TensoRF [6]	**35.3015**	**0.9632**	**0.0414**	19.5573	0.5221	0.6809	-	-	-	27.4294	0.7427	0.3611
Ours	29.4773	0.8700	0.1699	**26.1722**	**0.7489**	**0.4573**	-	-	-	**27.8248**	**0.8095**	**0.3136**

a stratified dataset. Because of the direct unavailability of stratified scenes, we built our own dataset that replicates the intended "stratified" scenario. We create a synthetic scene dataset using a mesh-editing software Blender [8] and real scene dataset by altering RealEstate10K dataset which was proposed for the camera localization task.

The proposed synthetic dataset has two important variations based on: (a) the number of stratified levels and (b) the geometric complexity. We classify based on the geometry's complexity as follows: (a) *Simple Scenes*: Stratified scenes using geometric components such as the sphere, cube, and so on; and (b) *Complex Scenes*: Stratified scenes that mimic real-world scenes. For Simple Scenes, we leverage models and textures provided by Blender [8]. We utilized publicly available graphical models and composited them to create a real-world configuration for Complex scenes. For example, to design the *"Coffee shop"* scene, we selected a building structure for the outer level and walls and glasses for the intermediate level structure. For the core level, we composited elements such as a cash register, coffee cups, and so on to simulate a real-world coffee-shop scene. To avoid photo-metric changes, we use fixed illumination. For each stratified level, the camera settings : field of vision and focal length are fixed. Each scene is rendered at 200 × 200 resolution . The camera viewpoint are sampled evenly from the curved surface of a hemisphere and then randomly divided into train, validation, and test sets. Inner objects in *Simple Scenes* are rendered from the surface of a sphere. Figure 5 depicts the proposed dataset's skeletal meshes. Further information on dataset is present in Appendix **B** in the supplementary material.

RealEstate10K dataset. We extracted four scenes *"Spanish Colonial Retreat in Scottsdale Arizona", "139 Barton Avenue Toronto Ontario" , "31 Brian Dr Rochester NY"*

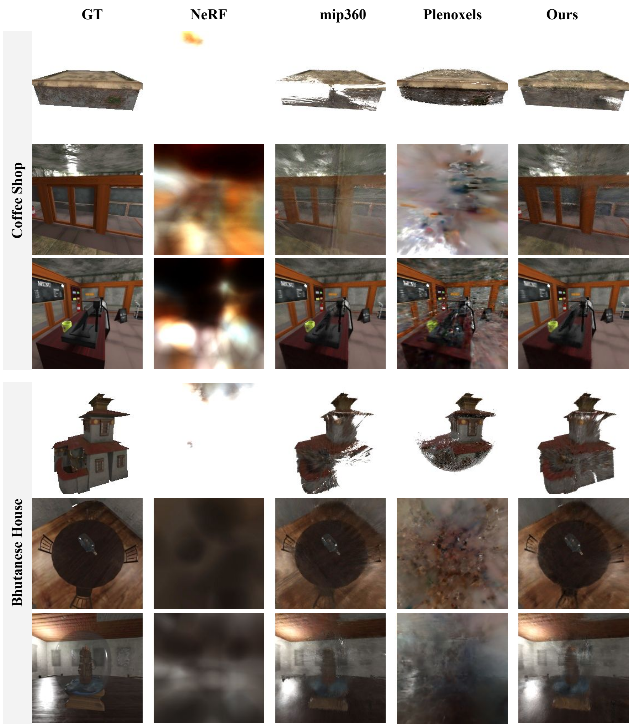

Figure 6. (From top to bottom) Qualitative results on the proposed synthetic datasets (Figure 5). Each row represents a novel view from a level of the stratified scene. The ground-truth (GT) is shown in Column 1. Compared to baselines (Column 2-4), our method's (Column 5) renderings are more consistent to GT.

and "7 Rutledge Ave Highland Mills" from RealEstate10K dataset. We manually inspected and removed regions which had dynamic components in them. More details about converting RealEstaet10k dataset for our stratified setting is provided in Appendix **C** the supplementary material.

6.3. Evaluation

We present quantitative and qualitative analysis of *Strata-NeRF* on the datasets described in Section 6.2.

Baselines. We compare our model with NeRF [34] , mip360 [2], Instant-NGP [35], TensoRF [6] and Plenoxels [64]. We chose Plenoxels [64] for comparison because it uses sparse-voxel representation which already discretizes the continuous 3D space, which can be useful in stratified scenes. It is worth noting that the sizes of the synthetic scenes in our dataset differ. As a consequence, the authors' recommended configuration file did not produce the optimal results. As a result, we modified the configuration files for unbounded scenes released by the creators of mip360 [2] to improve performance. For Instant-NGP [35], TensoRF [6] and Plenoxels [64], we change the hyperparameters like bound and scale as suggested in the official implementatinos. More information is in Appendix **D** in the supplementary material. Table 2 provides an overview

Table 5. Quantitative comparison of our model and baseline on ***"139 Barton Avenue"*** scene of RealEstate10K dataset.

	Metrics	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
mip360[2]	**PSNR ↑**	18.086	16.496	24.459	20.862	17.479	10.999
	SSIM ↑	0.618	0.595	0.771	0.702	0.584	0.409
Ours	**PSNR ↑**	**23.164**	**21.665**	**25.236**	**24.156**	**22.879**	**25.409**
	SSIM ↑	**0.826**	**0.757**	**0.789**	**0.791**	**0.753**	**0.782**

Table 6. Quantitative comparison of our model and mip360 baseline on Six Layer Scene.

Dataset	**Levels**	**mip360 [2]**	**Ours**	**mip360 [2]**	**Ours**
Spanish Colonial Retreat	5	20.106	**22.514**	0.622	**0.685**
31 Brian Dr Rochester	4	23.273	**28.026**	0.715	**0.835**
139 Barton Avenue	6	18.991	**23.433**	0.642	**0.780**
7 Rutledge Ave	7	19.621	**25.040**	0.566	**0.791**

of baselines.

Quantitative Results. Table 3 & 4 shows the average PSNR, SSIM [60] and LPIPS [67] for each stratified level in unseen test views. We find that our method surpasses other methods across all metrics most of the time. The baseline mip360 [2] works fine for the exterior structure but fails for the inner layers in the "Cube-Sphere-Monkey" scene. *Strata-NeRF*, on the other hand, offers superior metrics at all stratified levels. The baseline models do well in the outer scene but perform sub-optimally in the inner levels, especially in level 1. These outcomes demonstrate that our method outperforms the baseline models significantly.

Table 6 shows the summary of average PSNR and SSIM for all the levels in a scene for RealEstate10K dataset. In this case, we only compare our method with mip360 as it is the best performing one among others on the synthetic dataset. We observe that our method outperforms the baseline method in all scenarios. Further, we present level-wise result for a specific scene in Table 5. We observe that for real datasets *with increase in number of levels, the magnitude of performance improvement increases*, which demonstrates the effectiveness of the proposed approach. Further, we also compare Instant-NGP [35] and TensoRF [6] on a RealEstate10K scene in Appendix **E.2** in the supplementary material.

Qualitative Results. Figure 6 & 8 depicts the qualitative results for the synthetic dataset scenes described in Section 6.2. We observe that NeRF [34] performs poorly regardless in majority of scenarios. The generated novel views for "Coffee Shop" are poor. It only works well in level 0 of "Cube-Sphere-Monkey" dataset. mip360 [2] outperforms NeRF but falters in level 1. Furthermore, in level 0 of the "Cube-Sphere-Monkey" dataset, mip360 only generates a white patch with no visible structure. For RealEstate10K dataset, it can be observed in Figure 7 that mip360 generates blurry results compared to our approach. Further, we find that our approach generates consistent and structurally salient novel views throughout all levels and scenes. We show qualitative results for Instant-NGP [35] and Ten-

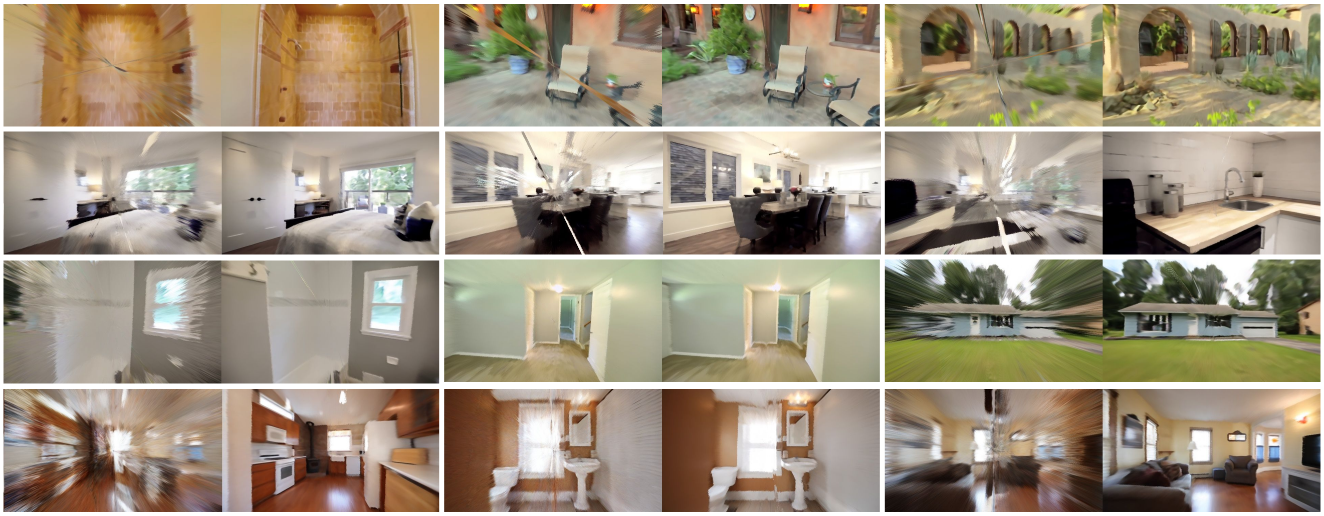

Figure 7. Qualitative comparison on Scenes from RealEstate10K dataset between mip360 (left image) and our method *Strata-NeRF* (right image) in a pair. Each row represents a scene in RealEstate10K and each pair represents a level in that scene. Our method outperforms and produce good quality novel views compared to mip360.

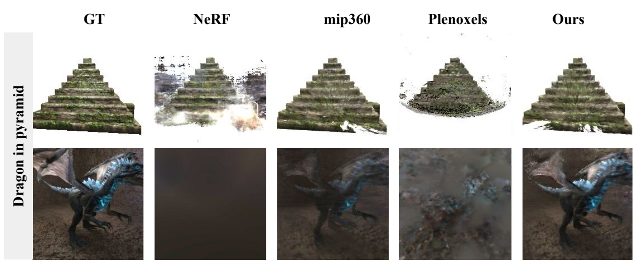

Figure 8. (From top to bottom) Qualitative results on the proposed synthetic datasets. Each row represents a novel view from each level of the stratified scene. The ground-truth view is shown in Column 1. Compared to prior works (Column 2-4) our method's (Column 5) renderings are more similar to the ground-truth.

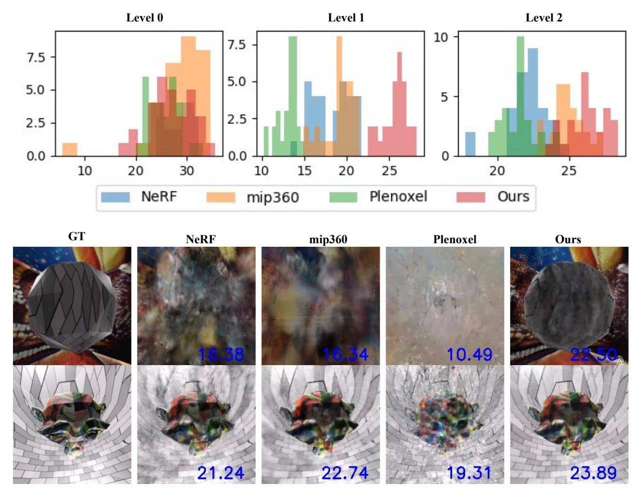

Figure 9. (Top Row) Comparison of histogram plots for the test-set for PSNR on **"Cube-Sphere-Monkey"**. Note how distribution of our our method is always towards the right compared to other methods. $x - axis$ denote metric value and $y - axis$ denotes the frequency. A qualitative comparison of our method's worst-case PSNR results. PSNR is present at the bottom of the result image.

soRF [6] in Appendix **E.2** in the supplementary material.

Worst Case Analysis. When comparing different methods, average metrics are often insufficient to determine which method is superior to the others. As we have observed in Figure 9 that the baseline method fails on some of test images, hence we also compare the methods in worst care scenarios. The worst-case analysis describes a method's worst performance on the dataset. The worst case analysis is particularly useful to detect the shortcomings of the methods. We present analysis in two categories: (a) histogram distribution for each metric on the test set, and (b) qualitative comparison of the worst-case scenario for our method on PSNR metric.

Figure 9 compares PSNR histogram plots on test-set views for the **"Cube-Sphere-Monkey"** scene. We can see that the mip360 approach performs poorly on PSNR and ranks low on practically all stratification levels. This supports our argument that the mip360 approach produces artifacts in such stratified scenes. For our method, the PSNR distributions are on the right. This implies that the novel views on test-set from our method will not be having serious artifacts in most cases, demonstrating its reliability.

Images in Figure 9 depict the qualitative results for the worst-case PSNR instances. All methods perform well in level 0. Hence, we are discussing interior levels which are level 1 and level 2. Other approaches fail in the worst-case scenario for our method at level 1. The outputs from NeRF, mip360 and Plenoxel are visually impaired. At level 2, our method has less blur compared to other approaches. These

Figure 10. Novel-views from different levels of 'Real Estate Video Tour 7 Rutledge Ave Highland Mills NY 10930 Orange County NY' scene in Real Estate 10K dataset. The two rows are from two-different view-points.

Table 7. Quantitative comparison of our model and baseline on Synthetic Six Layer Scene.

	Metrics	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
mip360 [2]	**PSNR** ↑	22.215	16.183	15.084	12.012	21.813	21.539
	SSIM ↑	0.777	0.442	0.510	0.344	0.817	0.647
Ours	**PSNR** ↑	**23.889**	**21.449**	**21.456**	**24.095**	**28.283**	**21.898**
	SSIM ↑	**0.833**	**0.681**	**0.685**	**0.722**	**0.883**	**0.686**

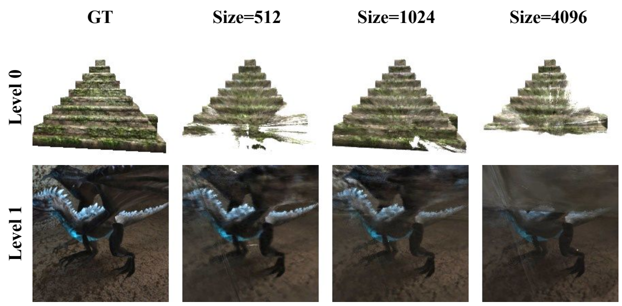

Figure 11. Comparisons of different codebook size on **"Dragon in Pyramid"** scene for different vector-codebook sizes. Note at size=1024 we achieve the best results with less artifacts.

findings demonstrate that our method is better suited to represent stratified scenes than others.

Ablation Studies. To analyse our proposed method, we present an ablation on the size of the vector codebook in our latent generator. Table 8 shows the ablation for the size of the vector codebook on the "Coffee Shop" dataset. We trialed with codebook sizes of 512, 1024 and 4096. We found that size 1024 provides optimal performance. As shown in Figure 11, increasing the codebook size induces haziness in the generated novel views, while decreasing the size creates white artifacts in level 0. As a result, we fix the size 1024 for all of our synthetic experiments. Whereas for RealEstate10K dataset we find that codebook size of 4096 produces the optimal tradeoff of results across levels, as it contains more number of levels and details.We further discuss the key architectural design choices for Latent Generator and Latent Router modules in Appendix **E.5**.

No. of levels: To further test the efficacy of our method on higher number of levels, we created a *"Simple Geometry"* scene consisting of primitive geometry shapes like cube and spheres. More details are in the supplementary

Table 8. Quantitative results on **"Cube-Sphere-Monkey"** scene for ablation on size of the vector codebook in Latent Generator.

Size	**PSNR** ↑		**SSIM** ↑		**LPIPS** ↓	
	Level 0	**Level 1**	**Level 0**	**Level 1**	**Level 0**	**Level 1**
512	**29.5458**	26.3497	**0.8743**	0.7395	0.1675	**0.4899**
1024	29.4834	26.1715	0.8701	**0.7489**	**0.1367**	0.5163
4096	28.4609	**27.8274**	0.8628	0.7342	0.1776	0.5027

material. Table 7 displays the results for both the baseline and our approach across a six levels stratified scene. The average PSNR/SSIM for the mip360 baseline is **15.35 / 0.487**, while our method achieved PSNR/SSIM of **23.54 / 0.754** which improves PSNR and SSIM by **53.35 %** and **54.83 %** respectively. This shows that our method performs better on increasing number of levels when compared with the baseline method. These observations also hold true for scenes in the RealEstate10K dataset as shown in Table 5.

7. Conclusion

In this work, we focus on the problem of modelling the 3D representation of a stratified and hierarchical scene, implicitly through a single neural field. For this, we propose *Strata-NeRF*, which models scenes with stratified structures by introducing a VQ-VAE-based latent generator to implicitly learn the distribution of latent space of input 3D locations and condition the neural radiance field with the latent code generated from this distribution. We also introduce a new synthetic dataset with stratified-level scenes and use it to analyse various existing approaches. Through quantitative, qualitative, and worst-case analysis on this dataset, we show that *Strata-NeRF* has a more stable 3D representation than the other methods. Further, the improvements due to *Strata-NeRF* also generalize to real-world RealEstate10K dataset, where it outperforms baselines by a significant margin establishing a new state-of-the-art. We believe designing a new volume rendering equation for modelling complex stratified scenes is a good direction for future work.

Acknowledgement. This work was supported by Samsung R&D Institute India, Bangalore, PMRF and Kotak IISc AI-ML Centre (KIAC). Srinath Sridhar was partly supported by NSF grant CNS-2038897

References

[1] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[2] 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 (CVPR)*, 2022. 1, 2, 3, 4, 5, 6, 7, 9

[3] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020. 3

[4] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision (ECCV)*. Springer, 2022. 5, 6, 7, 8

[7] 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 (ICCV)*, 2021. 2

[8] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 6

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 5

[10] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, volume 31. Wiley Online Library, 2012. 2

[11] Nianchen Deng, Zhenyi He, Jiannan Ye, Budmonde Duinkharjav, Praneeth Chakravarthula, Xubo Yang, and Qi Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(11), 2022. 1

[12] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W Taylor, and Joshua M Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[14] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV) (ICCV)*. IEEE Computer Society, 2021. 3

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[16] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[17] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[18] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[19] Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, and Theophane Weber. Temporal difference variational auto-encoder. *arXiv preprint arXiv:1806.03107*, 2018. 3

[20] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[21] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. Efficientnerf efficient neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[22] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 4

[23] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3), 1984. 3

[24] Takuhiro Kaneko. Ar-nerf: Unsupervised learning of depth and defocus effects from natural images with aperture rendering neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[26] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 5

[27] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[28] Chaojian Li, Sixu Li, Yang Zhao, Wenbo Zhu, and Yingyan Lin. Rt-nerf: Real-time on-device neural radiance fields towards immersive ar/vr rendering. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, 2022. 1

[29] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[30] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. 3

[31] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[32] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[33] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (CVPR)*, June 2021. 3

[34] 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 (ECCV)*. Springer, 2020. 1, 2, 5, 6, 7

[35] 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), 2022. 5, 6, 7

[36] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[37] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26, 2017. 2

[38] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[39] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 2, 3

[40] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[41] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[42] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 3

[43] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[44] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[45] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[46] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[47] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[48] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. 3

[49] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067. SPIE, 2000. 2

[50] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[51] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), 2022. 3

[52] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[53] 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 (NeurIPS)*, 33, 2020. 3

[54] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Nonrigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[55] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[56] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 2, 3, 4

[57] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (CVPR)*. IEEE, 2022. 3

[58] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[59] Xin Wang, Shinji Takaki, Junichi Yamagishi, Simon King, and Keiichi Tokuda. A vector quantized variational autoencoder (vq-vae) autoregressive neural f_0 model for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, 2019. 3

[60] 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), 2004. 7

[61] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[62] Liwen Wu, Jae Yong Lee, Anand Bhattad, Yu-Xiong Wang, and David Forsyth. Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[63] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[64] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021. 5, 6, 7

[65] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[66] 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 (CVPR)*, 2021. 2, 3

[67] 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 (CVPR)*, 2018. 7