How Far Can We Compress Instant-NGP-Based NeRF?

Yihang Chen [1, 2] Qianyi Wu [2] Mehrtash Harandi [2] Jianfei Cai [2]
[1]Shanghai Jiao Tong University [2]Monash University
yhchen.ee@sjtu.edu.cn, {qianyi.wu, mehrtash.harandi, jianfei.cai}@monash.edu

Abstract

In recent years, Neural Radiance Field (NeRF) has demonstrated remarkable capabilities in representing 3D scenes. To expedite the rendering process, learnable explicit representations have been introduced for combination with implicit NeRF representation, which however results in a large storage space requirement. In this paper, we introduce the Context-based NeRF Compression (CNC) framework, which leverages highly efficient context models to provide a storage-friendly NeRF representation. Specifically, we excavate both level-wise and dimension-wise context dependencies to enable probability prediction for information entropy reduction. Additionally, we exploit hash collision and occupancy grids as strong prior knowledge for better context modeling. To the best of our knowledge, we are the first to construct and exploit context models for NeRF compression. We achieve a size reduction of 100× and 70× with improved fidelity against the baseline Instant-NGP on Synthesic-NeRF and Tanks and Temples datasets, respectively. Additionally, we attain 86.7% and 82.3% storage size reduction against the SOTA NeRF compression method BiRF. Our code is available here: https://github.com/YihangChen-ee/CNC.

1. Introduction

High-quality photo-realistic rendering at novel viewpoints remains a pivotal challenge in both computer vision and computer graphics. Traditional explicit 3D representations, such as voxel grids [17, 25, 34, 37], have earned their place due to their efficiency across numerous applications. However, their discrete nature makes them susceptible to the limitations imposed by the Nyquist sampling theorem, often necessitating exponentially increased memory for capturing detailed nuances.

In the past few years, Neural Radiance Field (NeRF) [28] has emerged as a game-changer for novel view synthesis. NeRF defines both density and radiance at a 3D point as functions of the 3D coordinates. Its implicit representation, encapsulated within a Multi-Layer Perceptron (MLP), captures continuous signals of a 3D scene seamlessly. Leveraging frequency-based positional embeddings of 3D coordinates [28, 41, 49], NeRF has showcased superior novel view synthesis quality in comparison to traditional explicit 3D representations. While NeRF exhibits good characteristics in memory efficiency and image quality, its complex queries of the MLP slow down its rendering speed.

To boost NeRF's rendering speed, recent approaches have converged towards a hybrid representation, merging explicit voxelized feature encoding with implicit neural networks. This combination promises faster rendering without compromising on quality. These methods include varied data structures such as dense grids [7, 38–40], octrees [26, 50], sparse voxel grids [24], and hash tables [29]. Among them, Instant-NGP (INGP) [29] which

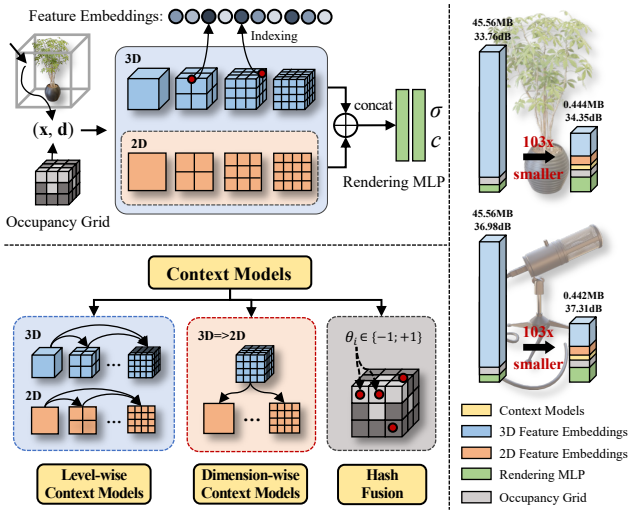

Figure 1. Motivation of our work. Instant-NGP represents 3D scenes using 3D hash feature embeddings along with a rendering MLP, which takes a non-negligible storage size with the embeddings accounting for over 99% of storage size (upper-left). To tackle this, we introduce context models to substantially compress feature embeddings, with the three key technical components (bottom-left). *Our approach achieves a size reduction of over 100× while simultaneously improving fidelity.*[1]

[1]The size of INGP is calculated under 16 levels with resolution from 16 to 2048. The feature vector dimension is 2 and represented with FP32.

introduces multi-resolution learnable hash embeddings is the most representative one. These hybrid strategies are fast becoming staples in modern NeRF architectures [1, 43]. Yet, with gains in rendering quality and speed, storage is becoming the new constraint. For example, with the occurrence of large-scale NeRFs [23, 42, 48], the total storage of their parameters restricts their accessibility and deployment. The storage challenge becomes even more pressing when further considering numerous 3D scenes.

This leads us to ponder: *Can we reduce the storage cost of modern NeRFs with hybrid representations such as Instant-NGP while maintaining high fidelity and rendering speed?* A few NeRF compression methods have been proposed to address this. The common idea is to follow the "Deep Compression"[11] concept, which relies on pruning and quantization techniques to squeeze the explicit feature encoding segment. For example, VQRF [20] pioneers the trimming of redundant voxel grids and employs vector quantization for parameter reduction. BiRF [36] goes a step further, using 1-bit binarization for feature embeddings compression. While these methods notably reduce storage needs, we advocate that the efficiency of voxel feature encoding can be further improved from a data compression perspective. Our core principle is to decrease the information uncertainty (entropy) of voxel feature encoding, which has been widely investigated in image and video compression but rarely explored in NeRF compression. By leveraging efficient entropy codecs like Arithmetic Coding (AE) [47], we aim to achieve a balance between minimizing storage cost and maintaining rendering quality and speed.

In this paper, we propose a Context-based NeRF Compression (CNC) framework, a pioneering approach to create a storage-optimized NeRF model. Based on Instant-NGP [29] and its multi-resolution hash encoding, our model offers both rendering quality and efficiency. Our core proposition lies in the entropy minimization of explicit feature encoding using accurate context models. Specifically, we introduce a meticulously designed entropy estimation function for each resolution in feature embeddings, on the assumption of Bernoulli distribution. This is coupled with both level-wise and dimension-wise context models that combine different aspects of the hashing embeddings, see Fig. 1. We also leverage the hash collision and the occupancy grid from Instant-NGP to further ensure our context models' accuracy. In summary, the major contributions of this work are threefold:

1. To our knowledge, we are the first to propose to model the contexts of INGP's multi-resolution hashing feature embeddings to effectively reduce storage size while maintaining fidelity and speed simultaneously.
2. We design customized context models that effectively build not only multi-level but also cross-dimension dependencies for INGP hash embeddings. We also utilize hash collision and occupancy grid as strong prior knowledge to provide more accurate contexts.
3. Extensive experiments show that our CNC framework achieves a size reduction of over 100× and 70× while simultaneously improving fidelity, compared to the baseline INGP, on Synthetic-NeRF and Tanks and Temples datasets, respectively. Our approach significantly outperforms the SOTA NeRF compression algorithm, BiRF [36], with over 80% size reduction.

2. Related work

Neural radiance field: from implicit to explicit. In recent years, Neural Radiance Field (NeRF) [28] has significantly advanced the area of novel view synthesis by effectively reconstructing 3D radiance fields in a neural implicit way. Specifically, NeRF utilizes a coordinate-based implicit Multi-Layer Perceptron (MLP) to enable synthesis from arbitrary views. Nevertheless, due to the absence of scene-specific information in the input coordinates, the MLP is designed to be relatively complex to encompass all necessary information. Such complexity slows down the entire rendering process, resulting in days for training.

To expedite rendering, diverse data structures have been introduced as input to explicitly carry scene-specific information, to reduce or even eliminate the MLP to achieve much faster rendering. For example, Instant-NGP (INGP) [29], TensoRF [4] and K-Planes [8] employ learnable embeddings or voxels to represent 3D scenes, which significantly reduce the computational burden of the rendering MLP. Plenoxels [7] and DVGO [38] take this a step further by eliminating the entire implicit MLP and opting for a purely explicit representation of the whole 3D scene. However, one major downside of these explicit representations is the substantial parameter size, sometimes reaching hundreds of MBs [7, 38], which results in undesirably large storage costs, especially taking into account a vast number of 3D scenes. To address this issue, compression techniques are emerging for more compact NeRF representations. In this paper, we explore context models for the representative INGP-based structure and push NeRF compression to a new level.

Compression techniques: which is the most suitable? Before delving into NeRF compression, we would like to start with a glance at existing compression techniques. First and foremost, model compression stands as a significant category. Given that different model weights exert varying impacts on the final results, various approaches compress them based on weight significance via pruning [52], quantization [31], and low-rank approximation [14, 33]. Knowledge distillation [10] is another avenue in which student models are guided by teachers to create much more compact versions. With the evaluated

importance of parameters in NeRF models, some NeRF compression algorithms select the most representative ones to retain information using codebooks [20, 21] or gradients [6]. Among the existing NeRF compression algorithms, BiRF [36] achieves SOTA Rate-Distortion (RD) performance by utilizing quantization techniques to binarize hash embeddings of INGP-based NeRF.

Apart from leveraging weight importance, contextual dependencies among neighboring elements are another essential source for compression, which has been widely exploited as spatial relations in image compression [5, 12, 13], and as both spatial and temporal relations in video compression [18, 19, 35]. Some recent NeRF compression methods also exploit spatial relations by utilizing techniques such as rank-residual decomposition [44], wavelet decomposition [32], or probability models [9] to achieve better compression. However, all these approaches often overlook the unique structures of NeRFs, failing to fully extract contextual information. In contrast, our work discovers that the multi-level embeddings in INGP-based NeRFs exhibit highly organized structures, and introduces efficient context models to effectively model contextual relations at different levels and dimensions, which leads to remarkable improvement in rate-distortion (RD) performance.

3. Method

Our objective is to develop a *storage-friendly* NeRF with efficient rendering speed and high fidelity. Our approach builds upon Instant-NGP (INGP) [29]. As shown in the right of Fig. 1, the primary storage of INGP comes from explicit hash feature embeddings. To minimize the overall model size, we introduce a novel framework named Context-based NeRF Compression (CNC), comprising various modules as depicted in Fig. 2. The technical details are elaborated in the following subsections.

3.1. Preliminaries

Neural Radiance Field [28] renders a 3D scene through an implicit rendering MLP. This MLP, when provided with the input coordinate $\mathbf{x} \in \mathbb{R}^3$ and viewing direction $\mathbf{d} \in \mathbb{R}^2$, can generate density $\sigma(\mathbf{x})$ and color $\mathbf{c}(\mathbf{x}, \mathbf{d})$ for rendering. Given a ray $\mathbf{r}(v) = \mathbf{o} + v\mathbf{d}$ casting from the camera $\mathbf{o} \in \mathbb{R}^3$, the rendered pixel color $\hat{C}$ can be calculated by accumulating the density and color along the ray [27], *i.e.*:

$$\hat{C}(\mathbf{r}) = \int_{v_n}^{v_f} T(v)\sigma(\mathbf{r}(v))\mathbf{c}\big(\mathbf{r}(v), \mathbf{d}\big)dv, \qquad (1)$$

where $\sigma(\mathbf{r}(v))$ is the density at the sampled point and $T(v) = \exp\Big(-\int_{v_n}^{v}\sigma(\mathbf{r}(u))du\Big)$ measures the transmittance along the ray. To enhance the representation of high-frequency details, NeRF proposes to map the input coordinates with a frequency-based position encoding [28]. However, the extensive querying of the heavy MLP slows down the training and inference processes.

Instant-NGP [29]. To expedite the rendering process of NeRF, INGP [29] introduces the concept of multi-level feature embeddings as a novel approach to positional encoding, where deeper levels correspond to voxels with higher resolutions. This allows for the utilization of a more compact rendering MLP without compromising the quality. For a given 3D coordinate $\mathbf{x}$, it is situated within a voxel at each level. For each resolution level $l \in \{1, \dots, L\}$, the feature at $\mathbf{x}$ can be calculated by interpolating from the features on the vertex features in the surrounding voxel grid, *i.e.* $\mathbf{f}^l(\mathbf{x}) = \text{interp}(\mathbf{x}, \Theta)$, where $\Theta = \{\boldsymbol{\theta}_i^l = (\theta_i^{l,1}, \dots, \theta_i^{l,F}) \in \mathbb{R}^F | i = 1, \dots, T^l\}$ is the trainable feature embedding collection, F is the dimension of each feature vector $\boldsymbol{\theta}_i^l$, and T^l is the size of the feature embedding set Θ. For each level, when the resolution of the voxel grid exceeds a specified threshold, the vertex features will be acquired through a spatial hashing function [45] to query Θ for efficiency. The interpolated features from different levels are then concatenated together and fed into the size-reduced rendering MLP for reconstruction. Another technique that INGP employs to accelerate rendering is the occupancy grid, which skips the empty space by efficient ray sampling. More details can be found in [29]. Consequently, the total storage of INGP includes the feature embeddings, the occupancy grid and the rendering MLP, as shown in Fig. 1.

BiRF [36]. While the use of implicit feature embeddings significantly enhances rendering speed, it concurrently imposes a storage burden. The state-of-the-art method BiRF [36] introduces an innovative approach by binarizing θ in feature embeddings to $\{-1, +1\}$ using a sign function and backpropagating them through a straight-through estimator [2]. This quantization solution reduces the model size by a large margin. Additionally, BiRF shows that introducing extra tri-plane features can enhance reconstruction quality with a similar number of parameters. In this work, we follow their model design with hybrid 2D-3D feature embeddings for the radiance field and build our context models on top of that.

3.2. Compress Embeddings with Context Model

Without loss of generality, we omit the notation of resolution level l from $\boldsymbol{\theta}_i^l$ and assume the feature dimension F is 1 for simplicity, for which $\boldsymbol{\theta_i} = \theta_i$. The fundamental principle of our framework is to decrease the information uncertainty of θ_i. Inspired from the binarization concept of BiRF [36], we model each value θ_i to conform to a Bernoulli distribution, *i.e.* $\theta_i \in \{-1, +1\}$. This results in a differentiable bit consumption estimator, based on entropy,

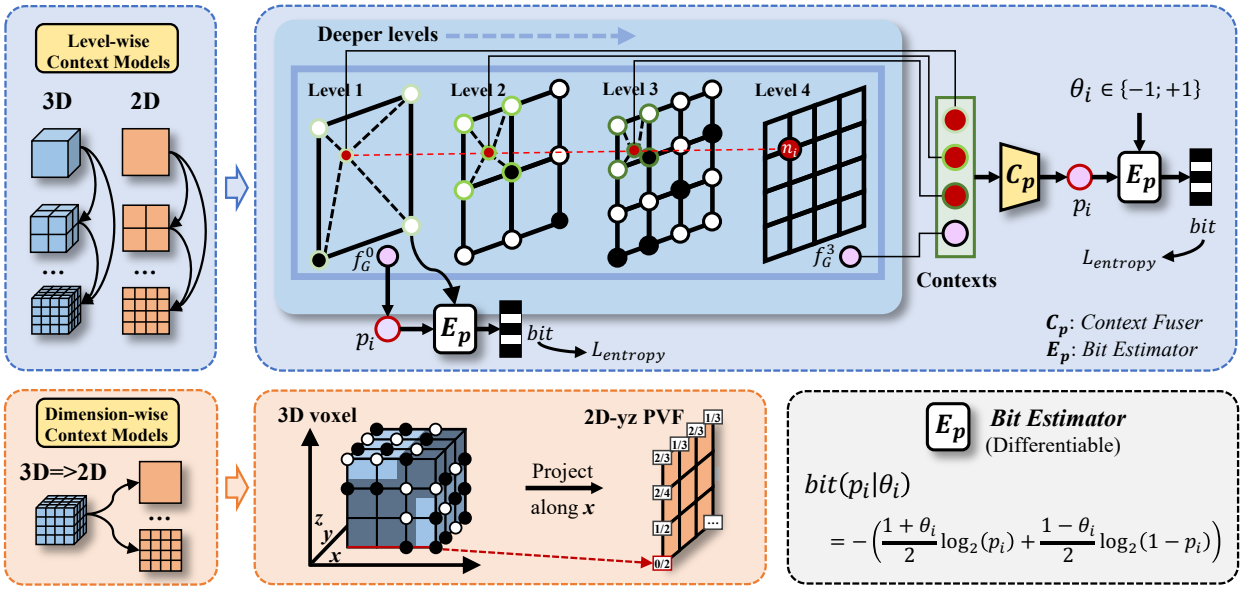

Figure 2. Overview of the proposed level-wise and dimension-wise context models. In the level-wise context model (dashed blue box), we first find the vertex n_i of the feature vector θ_i using hash function and then estimate its distribution probability p_i by a *Context Fuser* $\boldsymbol{C_p}$ with aggregated contexts from previously decoded levels. It's worth noting that while the illustration here is 2D, the same approach applies to 3D using trilinear interpolation. In the dimension-wise context models (dashed orange box), the last level of 3D voxel is projected onto 2D planes to obtain Projected Voxel Feature (PVF), which is then used for context interpolation. Deep-blue areas on the voxels indicate empty cells of the occupancy grid. At bottom-right (dashed black box), the formula of the entropy-based *Bit Estimator* $\boldsymbol{E_p}$ is provided, which is carefully designed to ensure a more efficient backward gradient.

for each θ_i with the probability $p_i = \mathbb{P}(\theta_i = +1) \in [0, 1]$:

$$bit(p_i|\theta_i) = -\left(\frac{1+\theta_i}{2}\log_2(p_i) + \frac{1-\theta_i}{2}\log_2(1-p_i)\right) = \begin{cases} -\log_2(p_i) & \theta_i = +1 \\ -\log_2(1-p_i) & \theta_i = -1 \end{cases} \tag{2}$$

A straightforward method to estimate p_i is to use the occurrence frequency $f_G = \frac{\#\{\theta_i|\theta_i=+1, \theta_i \in \Theta\}}{\#\{\theta_i|\theta_i \in \Theta\}}$, where $\#$ denotes the number counting, such that $p_i = f_G$ for $i = 1, \ldots, T$. However, we find this manner is suboptimal as f_G is not accurate for all the embeddings. Our key insight is that the spatial context in 3D space can enhance the precision of p_i estimation. For instance, if a point is empty in 3D space, we should spend fewer bits to store the corresponding features in Θ. This motivates us to introduce context models in the spatial domain when estimating p_i. Particularly, we propose two types of context models: level-wise and dimension-wise.

3.3. Level-Wise Context Models

The primary goal of the level-wise context models is to establish contextual dependencies among θ_is across different levels, with the expectation that more accurately predicted probability p_is lead to size reduction. Several critical issues need to be taken into consideration:

1. Contextual dependencies should obey causal processes. That is we can only utilize θ_is that have already been decoded as contexts to predict those yet to be decoded.
2. Context models themselves also consume storage space. This limitation prevents us from adopting arbitrarily large context models, even though having more parameters could enhance their prediction.
3. The order of contextual dependencies is of great importance. If more informative parts are decoded first, they can provide more context to others but at the cost of consuming more bits to store themselves.

In light of these considerations, we have designed our level-wise context models in a coarse-to-fine manner, as illustrated in the dashed blue box of Fig. 2 (upper). Consider an example of vertex n_i with associated feature θ_i at the current level $l = 4$, as shown in Fig. 2. Following the coarse-to-fine principle, the context of n_i depends on the interpolated features at the corresponding location from the previous L_c levels, where L_c is a preset constant (*e.g.*, $L_c = 3$ in Fig. 2). We also incorporate the occurrence frequency f_G of the current level as an auxiliary guidance for context modeling. All the context information is then concatenated and fed into a tiny 2-layer MLP named *Context Fuser* $\boldsymbol{C_p}$ to estimate the probability p_i at vertex n_i.

It is worth noting that if the number of previous layers is less than L_c, we set $L_c = l - 1$ (*i.e.*, using all the available previous layers for context). For level $l = 1$, we only utilize its occurrence frequency f_G^1 to estimate the bit consumption

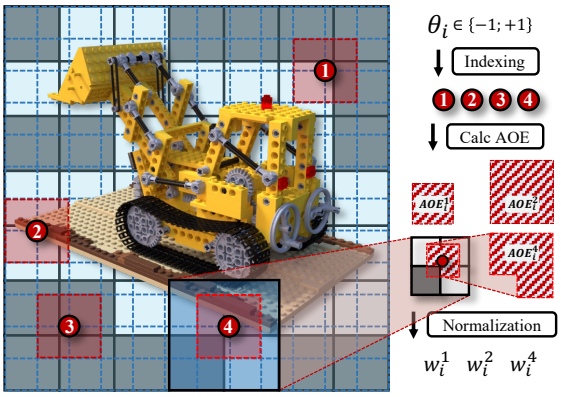

Figure 3. Illustration of Hash Fusion. In this toy example, the resolutions of the voxel and occupancy grids are 12 and 7, respectively. The weight of each hash collided vertex k of θ_i is normalized from its AOE, AOE_i^k, which measures the intersection area between the vertex grid (semitransparent dashed red square around the vertex) and occupied cells (light-colored).

because there is no previous layer.

3.4. Hash Fusion with Occupancy Grid

One important design to alleviate extensive storage at the finer resolution of the trainable embeddings Θ is adopting spatial hashing indexing [29, 45]. However, this introduces an issue of hash collision when building the context models. Here, we provide a solution to address it for regression of more accurate predictions, with the assistance of the occupancy grid. The occupancy grid plays a pivotal role in our approach, which partitions the entire 3D scene into grid cells and records occupancy conditions in binary format. Generally, only cells on the surfaces of the objects are occupied, while the rest are empty, resulting in the sparsity. This inherent sparsity makes the occupancy grid a spatial prior that greatly enhances our context modeling.

Fig. 3 illustrates the details of the proposed hash fusion solution to address the hash collision issue. Particularly, suppose a feature vector θ_i corresponds to K vertices, denoted as $\{n_i^k | k = 1, \ldots, K\}$ (*e.g.*, $K = 4$ in Fig. 3). This implies for each θ_i, multiple probabilities $\{p_i^k | k = 1, \ldots, K\}$ will be estimated. We define the Area of Effect (AOE) of a vertex as the intersection between the surrounded voxel grid and the occupied cell to weigh the probability prediction, which effectively determines the significance of a vertex. For example, vertex n_i^3 in Fig. 3 has an AOE of 0, then it is invalid, and should not contribute to the calculation of p_i. The final weighted probability of p_i is expressed as

$$\begin{aligned} p_i &= \sum_{k=1}^{K} w_i^k \times p_i^k \\ w_i^k &= AOE_i^k / \sum_{j=1}^{K} AOE_i^j \end{aligned} \tag{3}$$

This not only enhances the training efficiency but also improves the context accuracy. Furthermore, if all associated vertices are invalid, then the corresponding θ_i is invalid and can directly be discarded to save storage space.

3.5. Dimension-Wise Context Models

Considering BiRF introduces hybrid 2D-3D feature embeddings to improve the reconstruction quality, besides modeling contextual dependencies at various levels, we also emphasize the importance of cross-dimensional relations. The main idea of dimension-wise context models is to leverage the inherent relationship between tri-plane features and voxel features. Through extensive experiments, we found that 2D tri-plane feature embeddings cannot provide sufficient contextual information to predict the probability of 3D voxelized feature embeddings, likely due to missing one dimension. Thus, we turn to a more natural approach, *i.e.*, estimating the probability of 2D plane embeddings from the 3D context. Specifically, we employ a dimension projection design, as illustrated in the dashed orange box of Fig. 2 (bottom-left). We first reconstruct the entire 3D voxel using the spatial hashing function. Then, we project this 3D voxel along three different axes and record the frequency of $+1$s along each axis direction to obtain 2D Projected Voxel Features (PVF). Here, we leverage the prior knowledge of valid 3D space by the occupancy grid during projection. If the AOE of a vertex is 0, then it will be omitted from the calculation during the projection. The PVF will serve as one additional "previous level" context to estimate the probability p_i for each 2D θ_i. Notably, PVFs can be obtained from three distinct 2D planes, *i.e.*, the xy, xz, and yz planes. In our work, we only utilize 3D feature embeddings that correspond to the largest resolution to generate PVFs, as they contain the most informative data.

Training loss. With the establishment of our context models, we can calculate the entropy loss $L_{entropy}$, which is defined as the sum of the bits associated with all *valid* feature vectors θs. The overall loss function then becomes

$$L = L_{mse} + \lambda L_{entropy}/M \tag{4}$$

where L_{mse} is the image reconstruction Mean Squared Error (MSE) loss and λ is a tradeoff factor to balance the two terms for variable bitrates. M is the number of θs in the embeddings, including both valid and invalid ones.

Decoding and rendering process. In the testing process, 3D embeddings are firstly decoded from shallow to deep levels using level-wise context models. Then the last 3D level is utilized to generate dimension-wise context for 2D embeddings. Finally, 2D embeddings are decoded in a coarse-to-fine order with the assistance of the dimension-wise context. It takes about 1 second for encoding/decoding. *It's worth noting that once the embeddings are decoded, all the rendering processes are the same as INGP, requiring no additional time.*

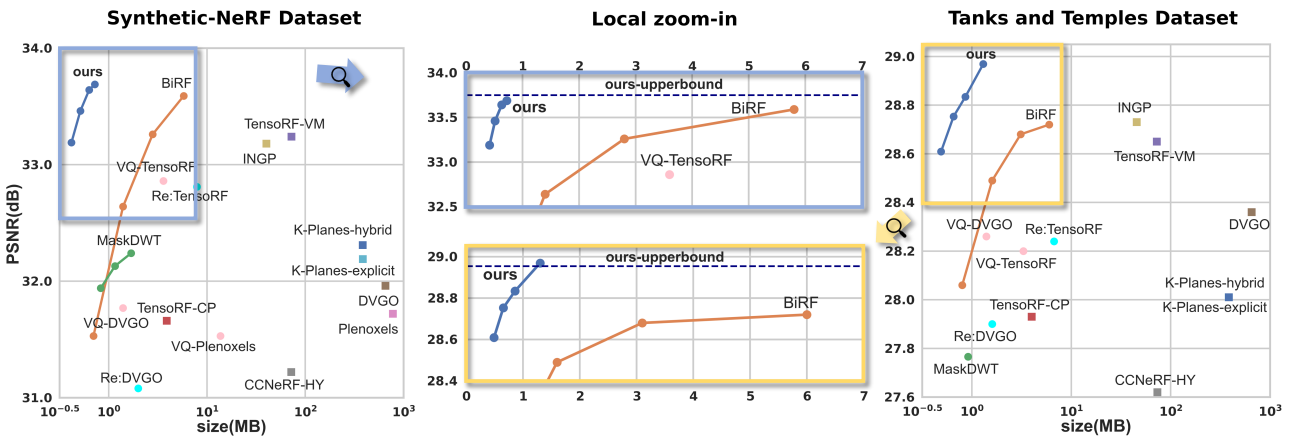

Figure 4. Performance overviews and detailed local zoom-in results of our proposed CNC and other methods. We apply log10 x-axis on the overviews for better visualization while linear x-axis on the zoom-in charts for better comparison. The more a curve goes upper-left, the better the rate-distortion (RD) performance is. Note that we achieve variable bitrates in our approach by changing λ from $0.7e-3$ to $8e-3$, while BiRF [36] achieves that by changing feature dimensions F from 1 to 8. The dashed line "ours-upperbound" represents the upper fidelity bound of our binary NeRF model (*i.e.*, $\lambda = 0$).

4. Experiments

In this section, we first describe the implementation details, then perform comparisons with previous methods on two benchmark datasets, and finally analyze different components of our CNC framework via extensive ablations.

4.1. Implementation Details

Our model is implemented based on NerfAcc [22] under PyTorch framework [30] and is trained using a single NVIDIA RTX 3090 GPU. We use Adam optimizer [15] with an initial learning rate of 0.01 and train for 20000 iterations. For 3D embeddings, it contains 12 levels with resolutions from 16 to 512. For 2D embeddings, the resolutions range from 128 to 1024 with 4 levels. The numbers of maximum feature vectors are set to 2^{19} and 2^{17} per level for 3D and 2D, respectively. The resolution of the occupancy grid is 128. We set the feature vector dimension F as 8, and the number of context levels L_c as 3. The structure of the rendering MLP is the same as [29] but with a width of 160. During training, we vary λ in Eq. 3 from $0.7e-3$ to $8e-3$ to obtain different bitrates. More details can be found in Sec. A of the supplementary.

4.2. Performance Evaluation

Baselines. We mainly compared our method with the recent NeRF compression approaches. Among them, BiRF [36] and MaskDWT [32] minimize NeRF model size during training, while VQRF [20] and Re:NeRF [6] are post-training compression algorithms. We also compared several major variants of NeRF to see their storage cost, including DVGO [38], Plenoxels [7], TensoRF [4], CCNeRF [44], INGP [29] and K-Planes [8].

Datasets. Experiments are conducted on a synthetic dataset *Synthetic-NeRF* [28] and a real-world large-scale dataset *Tanks and Temples* [16]. We follow the setting as BiRF [36].

Metrics. Besides the conventional PSNR versus size results, we also employ BD-rate [3] to assess the Rate-Distortion (RD) performance of these approaches, which measures the relative size change under the same fidelity quality. *A reduced BD-rate signifies decreased bit consumption for the same quality*.

Results. We report the quantitative and qualitative results in Fig. 4 and Fig. 6, respectively. For more fidelity metrics (SSIM [46] and LPIPS [51]) and visual comparisons, please refer to Tab. B-C and Fig. A-B of the the supplementary. Our proposed CNC achieves a significant RD performance advantage over others. *Compared to the SOTA* (i.e., *BiRF), our proposed CNC achieves 86.7% and 82.3% BD-rate reduction on the two datasets.* For Synthetic-NeRF dataset, our CNC closely approaches the upper fidelity bound while maintaining a much smaller size, showcasing the effectiveness of CNC. For the Tanks and Temples dataset, our CNC even surpasses the upper-bound. We conjecture that, to some extent, the entropy constraint from the context models serves as regularization to prevent overfitting.

Bitstreams. Our bitstream comprises four components: 3D and 2D feature embeddings, the rendering MLP, context models and the occupancy grid. Their average sizes are 0.220MB, 0.148MB, 0.011MB and 0.039MB in Synthetic-NeRF dataset with $\lambda = 4e-3$. They are encoded/stored as follows. Feature embeddings are entropy encoded by Arithmetic Coding (AE) [47] with probabilities predicted by context models. The rendering MLP parameters are quantized from the original 32 bits to 13 bits, which only causes a slight performance drop of less than 0.02 dB in PSNR while saving up to 0.216 MB. Context models are preserved in float32 to maintain

2D context	3D context	Dimension	BD-rate(all/emb)
✓	✓	✓	0%/0%
✓	✓	✗	+5.7%/+9.2%
✓	✗	✗	+30.8%/+54.3%
✗	✗	✗	+43.7%/+78.8%

Table 1. Ablation study on context models on Synthetic-NeRF dataset. Compared to CNC, disabling context models results in undesirable increases in BD-rate. "BD-rate (all/emb)" denotes the relative size changes in terms of the total model size or the size of the embeddings only.

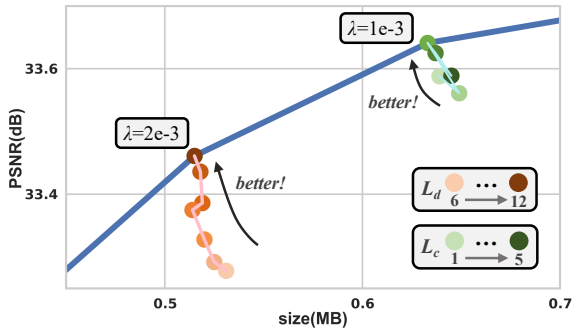

Figure 5. Orange points represent ablation studies on L_d from 6 to 12, with $\lambda = 2e-3$. Green points represent ablation studies on L_c from 1 to 5, with $\lambda = 1e-3$. Best results are obtained at $L_d = 12$ and $L_c = 3$. Experiments are on Synthetic-NeRF dataset.

prediction accuracy. The occupancy grid is binary and can be compressed by AE [47].

4.3. Ablation Study

We contemplate what the optimal design is for context models. To address this question, we first deactivate certain context models to observe the extent of performance drop. Then, we delve into the detailed effect of inter-level dependencies in level-wise context models. Regarding the hash fusion module, we explore the crucial function of AOE, which can address the hash collision issue.

Which context model is the most useful? First of all, we evaluate the capabilities of context models by disabling level-wise (3D and 2D embeddings) or dimension-wise context models. When context models are disabled, occurrence frequency f_G^l is utilized to estimate all vector θs in embeddings for each level l. Note that f_G^l is updated with the training. The corresponding results are shown in Tab. 1. It can be observed that a lack of either context model leads to a significant BD-rate increase. 3D context models contribute more than 2D ones since they occupy more storage space and are more sparse, thus having more potential for compression. Even though context models themselves introduce extra bits, the savings they bring in feature embeddings are remarkable, thanks to the accurate prediction of the probabilities.

Ablation items	BD-rate(all/emb)
Fine-to-coarse level-wise contexts	+33.8%/+53.0%
No discarding of θs	+43.7%/+78.8%
Proper discarding of θs	+32.6%/+59.1%
Over discarding of θs	N/A

Table 2. Ablation study on context dependencies and hash fusion on Synthetic-NeRF dataset.

How to design the level-wise context model? We now delve into the contribution of context models from each level. Specifically, we gradually replace the context model with f_G^l from deeper to shallower layers, where we use L_d to indicate the level starting from which context models are disabled. Experimental results are shown in Fig. 5 by orange points. We observe that as L_d becomes smaller, RD performance decreases. This suggests that a single f_G^l is inadequate to predict the feature distribution for each level l. In contrast, our context models exhibit greater capability in context aggregation. Experiments on context levels L_c are also conducted, as shown in green points in Fig. 5. Increasing L_c does not always lead to improved performance, as a distant level may provide limited information but introduce additional complexity.

Which contextual order is suitable? We investigate the order of fine-to-coarse in level-wise context models in Tab. 2. It can be seen that the coarse-to-fine context models performs much better than the reverse one. This suggests a coarse-to-fine flow aligns better with the information restoration behavior for a multi-resolution structure.

To which extent should invalid vectors be discarded in hash fusion? Lastly, we conduct experiments to assess the effectiveness of hash fusion, for which a key function is to discard invalid feature vectors using AOEs to save storage space. To demonstrate its effectiveness, we vary the extent of discarding to observe the impact on RD performance. For ablation purposes, we disable both level-wise and dimension-wise context models and only use the frequency f_G^l to estimate probabilities for each level l. The results are presented in Tab. 2. Initially, *no discarding of θs*: we do not discard any of the feature vector θs and retain all of them, leading to significant storage waste on invalid vectors. This setting is the same as the last line of Tab. 1. Moving one step further, *proper discarding of θs*: we apply f_G^ls only to valid θs at each level l and encode them, whose validity is judged by AOEs. This approach aligns with our current methodology. Finally, *over discarding of θs*: we alter the criterion by simply determining the validity of θ based on whether it is located in an occupied cell, rather than using AOE. However, this may cause over-discarding, where vertices necessary for rendering might be undecodable. This leads to a significant degradation in fidelity to an extremely low level (approximately 27.2 dB in PSNR under

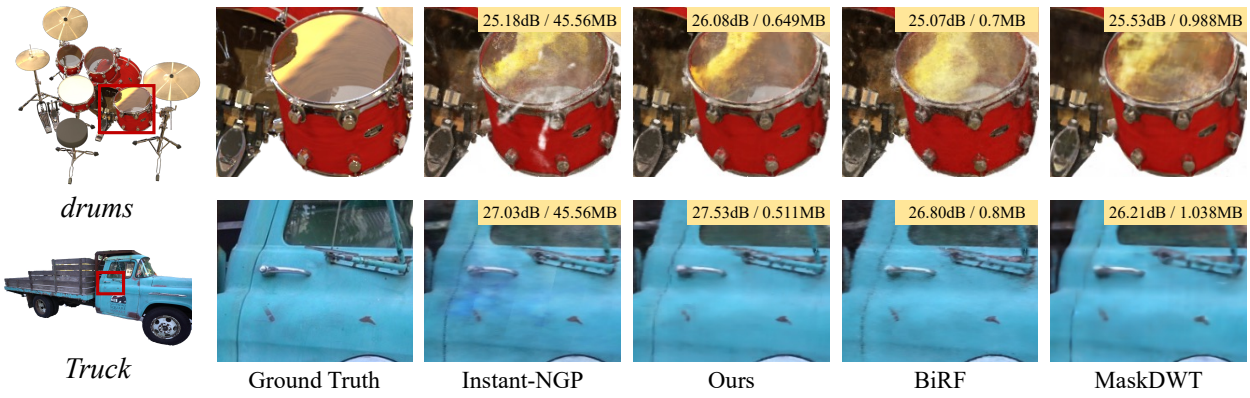

Figure 6. Qualitative quality comparisons of *drums* in Synthetic-NeRF dataset and *Truck* in Tanks and Temples dataset. We mainly compare recent NeRF compression approaches, along with our base model Instant-NGP. While some compression algorithms can achieve a low size of 1MB, they significantly sacrifice reconstruction fidelity. Our approach exhibits the best visual quality at the low size. Quantitative results of PSNR/size are shown in the upper right.

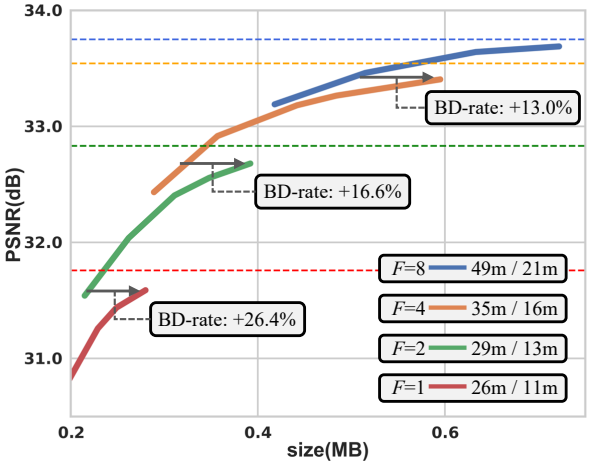

Figure 7. Fidelity upper-bound influences the RD performance. Dashed lines indicate the fidelity upperbounds at different hyperparameter settings of F. We also report the training time of our model with/without context models (bottom-right).

$\lambda = 0.7e-3$), resulting in no intersection on the y-axis, making BD-rate incalculable (N/A).

4.4. Fidelity Upper-Bound Influences Performance

In this subsection, we delve into a fundamental difference between NeRF compression and other data formats (such as image compression). To be specific, the ground-truth image for image compression is always available, which theoretically allows for perfect fidelity if no entropy constraint is applied. However, this is not the case for NeRF compression. The ground truth 3D scene is not known in advance, and the upper-bound of the fidelity is fundamentally determined by the capability of the reconstruction algorithm. In our case, it is the CNC model without the entropy constraint, *i.e.*, $\lambda = 0$. Fig. 7 shows our fidelity upperbounds under different feature dimensions F, ranging from 1 to 8. We can see that larger feature dimensions result in higher fidelity upperbounds and better RD performance. This is because a larger feature dimension allows more room for context models to eliminate redundancy and perform compression. However, a higher upper-bound also leads to increased training and rendering time, and compression becomes more challenging when approaching the upper-bound.

5. Conclusion

In this paper, we have proposed a Context-based NeRF Compression (CNC) framework, where context models are carefully designed to eliminate the redundancy of binarized embeddings. Hash collision and occupancy grid are also fully exploited to further improve prediction accuracy. Experimental results on two benchmark datasets have demonstrated that our CNC can significantly compress multi-resolution Instant-NGP-based NeRFs and achieve SOTA performance. The success of NeRF compression on static scenes provides a solid proof of concept for more advanced and space-taking applications such as dynamic or large-scale NeRFs.

Limitation. The main drawback of our approach is the slowdown in training time, resulting in about 1.3× longer training duration over the one without context models. However, this limitation can be mitigated by: 1) reducing fidelity upper-bound; 2) adjusting context models; 3) improving the code to execute context models and the rendering MLP concurrently.

Acknowledgement

The paper is supported in part by The National Natural Science Foundation of China (No. U21B2013).

MH is supported by funding from The Australian Research Council Discovery Program DP230101176.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 2

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3

[3] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *ITU SG16 Doc. VCEG-M33*, 2001. 6

[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. 2, 6

[5] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948, 2020. 3

[6] Chenxi Lola Deng and Enzo Tartaglione. Compressing explicit voxel grid representations: fast nerfs become also small. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1236–1245, 2023. 3, 6

[7] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 1, 2, 6

[8] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 2, 6

[9] Sharath Girish, Abhinav Shrivastava, and Kamal Gupta. Shacira: Scalable hash-grid compression for implicit neural representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17513–17524, 2023. 3

[10] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021. 2

[11] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*, 2016. 2

[12] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021. 3

[13] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5718–5727, 2022. 3

[14] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *Proceedings of the British Machine Vision Conference 2014*. British Machine Vision Association, 2014. 2

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015. 6

[16] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 6

[17] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38:199–218, 2000. 1

[18] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021. 3

[19] Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1503–1511, 2022. 3

[20] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Liefeng Bo. Compressing volumetric radiance fields to 1 mb. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4222–4231, 2023. 2, 3, 6

[21] Lingzhi Li, Zhongshu Wang, Zhen Shen, Li Shen, and Ping Tan. Compact real-time radiance fields with neural codebook. In *ICME*, 2023. 3

[22] Ruilong Li, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: A general nerf acceleration toolbox. *arXiv preprint arXiv:2210.04847*, 2022. 6

[23] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3205–3215, 2023. 2

[24] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 1

[25] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4):65:1–65:14, 2019. 1

[26] Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics (TOG)*, 40 (4):1–13, 2021. 1

[27] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 3

[28] 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. 1, 2, 3, 6

[29] 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, 2, 3, 5, 6, 4

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

[31] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations*, 2018. 2

[32] Daniel Rho, Byeonghyeon Lee, Seungtae Nam, Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Masked wavelet representation for compact neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20680–20690, 2023. 3, 6, 4

[33] Roberto Rigamonti, Amos Sironi, Vincent Lepetit, and Pascal Fua. Learning separable filters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2754–2761, 2013. 2

[34] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35:151–173, 1999. 1

[35] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*, 2022. 3

[36] Seungjoo Shin and Jaesik Park. Binary radiance fields. *Advances in neural information processing systems*, 2023. 2, 3, 6, 4

[37] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 1

[38] 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*, pages 5459–5469, 2022. 1, 2, 6

[39] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021.

[40] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1

[41] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 1

[42] 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*, pages 8248–8258, 2022. 2

[43] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 2

[44] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. Compressible-composable nerf via rank-residual decomposition. *Advances in Neural Information Processing Systems*, 35:14798–14809, 2022. 3, 6

[45] Matthias Teschner, Bruno Heidelberger, Matthias Müller, Danat Pomerantes, and Markus H Gross. Optimized spatial hashing for collision detection of deformable objects. In *Vmv*, pages 47–54, 2003. 3, 5

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[47] Ian H Witten, Radford M Neal, and John G Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987. 2, 6, 7

[48] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European conference on computer vision*, pages 106–122. Springer, 2022. 2

[49] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8254–8263, 2023. 1

[50] 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*, pages 5752–5761, 2021. 1

[51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[52] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. In *International Conference on Learning Representations (ICLR)*, Vancouver, CANADA, 2018. 2