Rate-aware Compression for NeRF-based Volumetric Video

Zhiyu Zhang[*]
Shanghai Jiao Tong University
Shanghai, China
zhiyu-zhang@sjtu.edu.cn

Guo Lu[*]
Shanghai Jiao Tong University
Shanghai, China
luguo2014@sjtu.edu.cn

Huanxiong Liang
Shanghai Jiao Tong University
Shanghai, China
huanxiong@sjtu.edu.cn

Zhengxue Cheng[†]
Shanghai Jiao Tong University
Shanghai, China
zxcheng@sjtu.edu.cn

Anni Tang
Shanghai Jiao Tong University
Shanghai, China
memory97@sjtu.edu.cn

Li Song[†]
Shanghai Jiao Tong University
Shanghai, China
song_li@sjtu.edu.cn

Abstract

The neural radiance fields (NeRF) have advanced the development of 3D volumetric video technology, but the large data volumes they involve pose significant challenges for storage and transmission. To address these problems, the existing solutions typically compress these NeRF representations after the training stage, leading to a separation between representation training and compression. In this paper, we try to directly learn a compact NeRF representation for volumetric video in the training stage based on the proposed rate-aware compression framework. Specifically, for volumetric video, we use a simple yet effective modeling strategy to reduce temporal redundancy for the NeRF representation. Then, during the training phase, an implicit entropy model is utilized to estimate the bitrate of the NeRF representation. This entropy model is then encoded into the bitstream to assist in the decoding of the NeRF representation. This approach enables precise bitrate estimation, thereby leading to a compact NeRF representation. Furthermore, we propose an adaptive quantization strategy and learn the optimal quantization step for the NeRF representations. Finally, the NeRF representation can be optimized by using the rate-distortion trade-off. Our proposed compression framework can be used for different representations and experimental results demonstrate that our approach significantly reduces the storage size with marginal distortion and achieves state-of-the-art rate-distortion performance for volumetric video on the HumanRF and ReRF datasets. Compared to the previous state-of-the-art method TeTriRF, we achieved an approximately -80% BD-rate on the HumanRF dataset and -60% BD-rate on the ReRF dataset.

CCS Concepts

• **Computing methodologies** → **Computer graphics**; **Computer vision**; **Virtual reality**; **Image compression**.

[*]Both authors contributed equally to this research.
[†]Corresponding authors.

Keywords

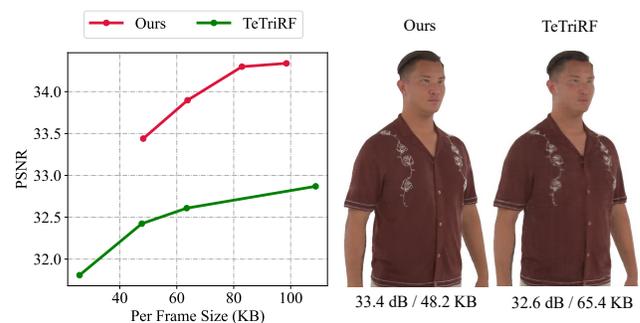Volumetric Video, NeRF, Compression, Rate Estimation

Figure 1: Comparison of compression performance with the state-of-the-art method, TeTriRF [52]. Compared to TeTriRF, our method achieves approximately 1 dB higher PSNR at a similar bitrate, and the BD-rate is -83%.

1 Introduction

3D volumetric video can provide immersive experience for viewers and exhibits a potential trend to be the next-generation video format. Previous works for volumetric videos reconstruction includes point cloud-based approaches [15] and depth-based approaches [3]. Recently, the popularized 3D representation known as Neural Radiance Fields (NeRF) [32], acclaiming for its photorealistic rendering capabilities from novel viewpoints, have attracted tremendous attention.

NeRF [32] utilizes Multilayer Perceptron (MLP) to model 3D scenes and employ volume rendering techniques to generate images from novel perspectives. To increase the training and rendering speeds, the mainstream works [6, 34, 44] incorporate explicit NeRF representations, such as grids [44], planes [6], and hash tables [34]. Furthermore, various methods [5, 10, 11, 28, 35, 36, 50]

have been proposed to extend NeRF of static scenes to dynamic scenes, i.e., volumetric videos. However, the majority of existing research primarily emphasizes enhancing the reconstruction quality of NeRF representations, while often overlooks the critical need for reducing the storage size and transmission bandwidth. This oversight presents significant challenges for practical applications, particularly in dynamic volumetric video scenarios.

To address these problems, several methods [14, 24, 25, 30, 41, 46] have been proposed to compress explicit NeRF representations for static scenes and dynamic scenes [48, 49, 52]. These methods utilize prediction, transformation, quantization, and entropy coding techniques, inspired from traditional video compression algorithms to compress the dynamic NeRF representations after the training stage, achieving significant compression rates. Among these approaches, ReRF [48] utilizes grid-based explicit representations to model dynamic scenes and devises a compression method akin to JPEG [47] to compress the representations. VideoRF [49] builds upon ReRF [48] by integrating traditional 2D video codec into the compression process. TeTriRF [52] employs a more compact tri-plane representation to model dynamic NeRF and utilizes an HEVC codec to compress the tri-plane representations. However, these methods concentrate exclusively on compressing dynamic NeRF representations post-training, neglecting the rate-distortion trade-off during the training phase. Therefore, they fail to adequately reduce spatial and temporal redundancy within the NeRF representations, thus their compression performance is far from optimal.

In this paper, we propose a rate-aware compression method tailored for NeRF-based volumetric video. Our approach estimates the bitrate of NeRF representations during its training stage and incorporates rate and distortion terms into the loss function, enabling end-to-end training. Therefore, we can obtain a compact NeRF representation with optimal rate-distortion performance. Specifically, first of all, targeting explicit NeRF grid representations, we propose inter prediction-based dynamic modeling technique by learning residual information based upon the previous frame, which effectively reduces the entropy of the NeRF representation. Second, we propose an adaptive quantization strategy with learnable quantization step to preserve more detailed information which contribute better reconstruction quality at different locations and scales. Third, in order to incorporate bitrate constraints into the training process and leverage rate-distortion loss, we introduce a tiny MLP-based implicit entropy model to estimate the rate. Given the potentially complex distribution of the explicit NeRF representation, we use temporal and spatial context for a more accurate rate estimation. Experimental results demonstrate that by applying our methodology to grid-based explicit representation [7], we can achieve state-of-the-art rate-distortion compression performance on the representative HumanRF [17] and ReRF [48] datasets. Compared with SOTA method TeTriRF [52], we achieve an approximate -80% BD-rate on HumanRF dataset and -60% BD-rate on ReRF dataset.

Our main contributions can be summarized as follows:

- We proposed a rate-aware compression framework for NeRF grid representations. Our pipeline introduces adaptive quantization strategy and spatial-temporal implicit entropy model to achieve joint rate-distortion optimization during training,

which greatly enhances the compression performance than post-training methods.

- Extensive experimental results on the benchmark datasets demonstrate the effectiveness of our approach. We can save more than 80% bitrate when compared with state-of-the-art dynamic NeRF compression approach at the same reconstruction quality.

2 Related Work

2.1 NeRF for Scene Representation

NeRF [32] enables the generation of images from arbitrary viewpoints through volume rendering, by employing a large-scale Multilayer Perceptron (MLP) to fit the colors and densities of sampled points in a scene. However, utilizing a large-scale MLP to represent the scene for real-time rendering is infeasible, prompting several works to propose combining explicit representations with smaller MLPs as a substitute for the pure implicit representation of the scene. This approach aims to reduce the computational complexity associated with large-scale MLPs. Some existing works have explored various approaches in this regard. For instance, [12] utilizes an octree, [44] employs a voxel grid, [34] implements hash tables, and [6] utilizes tensors.

Expanding NeRF [32] to dynamic scenes is not a trivial task, necessitating consideration of object movements within dynamic scenes. [9, 13, 53] model dynamic scenes by using time as an additional condition for the implicit MLP, yet these purely implicit modeling methods are not only inefficient but also perform poorly in motion-intensive settings. Alternatively, [26, 27, 35] incorporate motion modeling of the scene using a deformation field to predict displacements in dynamic scenes between each frame and a canonical frame. To expedite training and rendering, [10, 11] introduce explicit representations to accelerate the reconstruction of dynamic scenes. Essentially, these approaches utilize a single NeRF model to fit all frames within a dynamic scene, achieving commendable reconstruction results for short sequences. However, the reconstruction quality dramatically declines with longer sequences. Consequently, to ensure high-quality reconstructions in long-sequence settings, we adopt a frame-by-frame modeling approach for dynamic scenes in this paper.

2.2 NeRF Representations Compression

Although NeRF has achieved superior rendering quality and fast training speed via the explicit representations, it is at the cost of additional model size and storage cost. Recently, numerous works [14, 20, 24, 25, 30, 38, 41, 46, 57] have focused on compressing the explicit representation of NeRF in static scenes. [24] employs post-processing techniques to compress trained voxel grids through voxel pruning and vector quantization. [46] and [41] introduce compression-related operations during training, such as vector quantization [46] and binarization [41] to compress neural radiance fields, yet they do not consider rate-distortion optimization during training. [14, 20, 30] reduce the entropy of the NeRF representation by introducing rate loss during training, but their methods of rate estimation are relatively simplistic, and the compression performance is not optimal. NeRFCodec [25] compresses NeRF representations by adjusting encoder and decoder heads, building on

the reuse of 2D neural image codec, but this method is only applicable to plane-based NeRF representations. Moreover, these methods solely address compression in static scenes and do not account for temporal redundancy in dynamic scenes.

For dynamic NeRF compression, the recent works [16, 48, 49, 52, 56, 58] adopt frame-by-frame modeling, which enables high-quality reconstruction for long-duration sequences. These approaches also incorporate compression algorithms to ensure efficient storage utilization. ReRF [48] models the radiance field as a combination of motion fields and residual fields and employs traditional video compression pipelines to compress the representation of the radiance field. Guo *et al.* [16] adopts a vector quantization approach to eliminate the spatial-temporal redundancy of the radiance field. TeTriRF [52] directly models the radiance field as a tri-plane representation and applies a 2D video encoder to compress the representation. However, these methods do not consider the optimization of radiance field representation guided by rate-distortion loss functions. Our proposed method not only eliminates temporal redundancy but also leverages rate-distortion functions to optimize the representation of the radiance field, achieving optimal rate-distortion performance.

2.3 Image, Video and 3D Content Compression

To achieve efficient transmission, data compression has been investigated for other multimedia data, including image, video and 3D contents. This work also inspires from these previous approaches, thus we introduce them briefly.

In the field of image compression, various methods have been developed to formulates the rate-distortion optimization problem as an end-to-end training process to enhance compression efficiency utilizing neural networks [1, 8, 18]. Balle [1] proposed a hyperprior-based autoencoder to achieve better compression performance than traditional codecs, such as JPEG [47] and JPEG2000 [42]. Cheng [8] have employed discrete Gaussian mixture models to estimate the distribution of latents, achieving compression performance that exceeds the VVC-intra [4]. For video compression algorithms, traditional video coding standards [4, 45] utilize a hybrid coding framework that compresses video through intra prediction, inter prediction, transformation, quantization, and entropy coding. Neural network-based end-to-end video compression methods [19, 21–23, 29, 40] employ neural networks to replace modules within the hybrid coding framework. By using rate-distortion optimization to tune the entire network, the SOTA method have achieved compression performance surpassing the latest VVC standard [4].

3D content comes in various forms, such as NeRF, point clouds, and multi-view videos. For point cloud compression, MPEG has introduced G-PCC and V-PCC [39] to compress different types of point clouds. With the advancement of neural networks in image and video compression, numerous studies [37, 43] have explored neural network-based point cloud compression. [54] introduced 3D wavelet transform to compress volumetric images. [37] suggested enhancing the utilization of spatio-temporal information by leveraging voxelized information from neighboring nodes within an octree architecture, thereby further improving the efficiency of point cloud compression. For the compression of multi-view videos, MPEG has introduced MIV [3] standard, which encodes multi-view videos using 2D encoders, eliminates inter-view redundancies, and

utilizes novel view synthesis algorithms to synthesize new viewpoints. However, spatio-temporal redundancy reduction has not yet fully investigated for NeRF representations.

3 Preliminaries

NeRF [32] is a continuous 3D scene representation technique that learns a mapping fuction $g_\phi(\mathbf{x}, \mathbf{d}) : \mathbf{R}^d \rightarrow \mathbf{R}^c$, transforming the coordinates $\mathbf{x} = (x, y, z)$ of sampled points along a ray and the viewing direction $\mathbf{d} = (\theta, \phi)$ into color $\mathbf{c}$ and density σ:

$$(\mathbf{c}, \sigma) = g_\phi(\mathbf{x}, \mathbf{d}). \tag{1}$$

When rendering images from novel viewpoints, given a target camera extrinsic, the color $\hat{C}(\mathbf{r})$ of corresponding pixel can be obtained through volume rendering:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i,$$

$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_i), \quad \alpha_i = 1 - \exp(-\sigma_i \delta_i). \tag{2}$$

where T_i and α represents the transmittance and alpha value of i-th sampled point and δ_i denotes the distance between adjacent sampled points.

The original NeRF [32] employs a purely implicit MLP to approximate mapping function g. Subsequent works have introduced explicit representations (such as grids [7], planes [6], etc.) combined with a tiny rendering MLP to accelerate the training and rendering. It should be noted that our proposed compression method can be used for different explicit representations and we provide further analysis in our experimental part. Here, we adopt the DiF [7] as our default static representation in this paper and we extend it for the dynamic scenes.
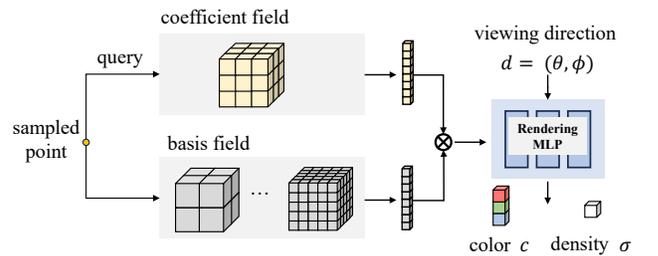

Figure 2: Demonstration of DiF representation.

DiF employs a grid-based explicit representation which decomposes the representation into a coefficient field and a basis field. As illustrated in Fig 2, the coefficient field comprised of a single-scale grid and the basis field consisting of six multi-scale grids. During rendering, features are initially queried from the coefficient grid and the basis grids through trilinear interpolation, followed by the fusion of features via the Hadamard product. Finally, a tiny rendering MLP maps the feature to color $\mathbf{c}$ and density σ.
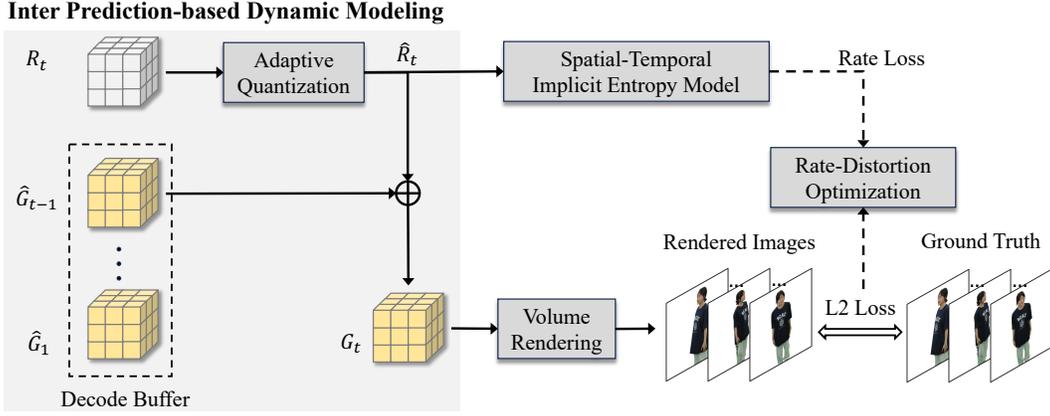
Inter Prediction-based Dynamic Modeling

Figure 3: The training pipeline of our proposed method. (a) Firstly, the reconstructed representation $\hat{G}_{t-1}$ from the previous frame is retrieved from the decode buffer, and based on this, the residual representation R_t is trained. (b) During training, adaptive quantization is utilized to enable representations at different scales to learn the optimal quantization step. Additionally, the spatio-temporal context implicit entropy model is used to estimate the bitrate of the explicit representation R_t. Ultimately, rate-distortion optimization is performed by integrating distortion loss and rate loss.

4 Methodology

Fig. 3 illustrates the training pipeline of our method. This includes a dynamic modeling method based on inter prediction, an adaptive quantization strategy, and bitrate estimation for the radiance field representation through a spatio-temporal implicit entropy model, combined with distortion loss for rate-distortion optimization.

4.1 Inter Prediction-Based Dynamic Modeling

Given that directly modeling entire dynamic scenes using NeRF representation like [5, 11] may result in poor performance for long sequences [48], we extend the current static NeRF representation DiF to dynamic scenes through a frame-by-frame inter prediction based modeling approach in the time dimension.

Specifically, the components required for volume rendering of a static scene include the explicit representation, occupancy grid, and the rendering MLP. These are also the components that need to be compressed and transmit. Tab 1 demonstrates the size of different components in the grid-based representation [7]. It is evident that independently modeling a NeRF for each frame without compression would require about 600 MBps (20MBx30fps) bandwidth for volumetric video, which is unacceptable. Given that the explicit representation occupies the majority of the model size, eliminating the inter-frame redundancy of the explicit representation could make dynamic NeRF more compact.

Table 1: Model Size of different components in DiF [7].

Explicit Repr.	Occupancy Grid	Rendering MLP	Total
19.21 MB	1.08 MB	0.17 MB	20.46MB

To simplify the description, we collectively refer to the coefficient grid and the basis grid as the feature grid G. To model dynamic scenes, we use a simple yet effective dynamic frame-by-frame modeling strategy. Here, we first divide the entire sequence into equally long groups, where the first frame of each group is an I-frame, modeled independently, and subsequent frames are P-frames, modeled only in terms of the residual relative to the previous frame. Specifically, when modeling a P-frame, the reconstruction grid of the previous frame $\hat{G}_{t-1}$ is retrieved from the decoded buffer, and we can learn the residual grid R_t for the current frame. Then the representation for current frame is expressed as:

$$G_t = \hat{G}_{t-1} + R_t. \tag{3}$$

Finally, $\hat{G}_t$ is stored into decode buffer for the reconstruction of the next frame. Furthermore, to ensure the temporal continuity and facilitate compression, we have applied an $\mathcal{L}1$ regularization to the magnitude of the residual grid, *i.e.,* $\mathcal{L}_{reg} = \|R_t\|_1$.

In addition, given the feature-level similarity of frames within the same group, we allow frames within a group to share the rendering MLP. This strategy significantly reduces bitrate consumption at a lower bitrate range without incurring substantial performance degradation.

4.2 Adaptive Quantization

In our proposed framework, the learned representation G_t (G_t for I-frame and R_t for P-frame) for the current frame should be quantized before the actual entropy coding. One straightforward approach is to use a uniform quantization step for all the items in G_t. However, the importance of different regions/scales in the learned representation G_t may vary and uniform quantization may be not the optimal solution.

In this paper, we have adopted an adaptive quantization training strategy to identify the optimal quantization step for NeRF representations. Taking DiF representation as an example, their explicit representations encompass grids at different scales. During training, we assign different quantization step parameters to grids of different

scales. These parameters are set as trainable and are continuously optimized throughout the training process. Then quantization is defined as $\hat{G}_t = \left\lfloor \frac{G_t}{q_t} \right\rceil$, where q_i is trainable quantization step.

Fig. 4 illustrates the distribution histograms of quantized explicit NeRF representations obtained through adaptive quantization training versus fixed quantization training. Fig. 4 demonstrates that the representations obtained from adaptive quantization training have a wider distribution range of [-150, 150] and a larger variance. On the other hand, the representations obtained from fixed quantization training are concentrated within the range of [-20, 20] and have a smaller variance. Additionally, the bitrates of the two cases are similar, indicating that adaptive quantization allows for more fine-grained quantization step adjustments. It does not indiscriminately remove high-frequency information but rather preserves useful high-frequency information. As a result, it can improve reconstruction quality while maintaining the same bitrate. We also present subjective result in the supplementary materials demonstrating the effects of adaptive quantization. The results using adaptive quantization are able to reconstruct more details.
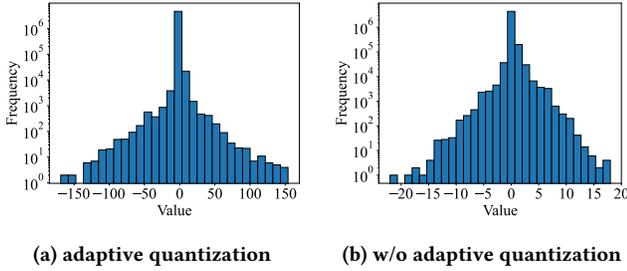

(a) adaptive quantization   (b) w/o adaptive quantization

Figure 4: Quantized explicit representation amplitude distribution histograms.

4.3 Spatial-Temporal Implicit Entropy Model

Considering that the NeRF representation is learned during training stage, it is plausible to incorporate a rate loss term into the loss function to guide the NeRF representation towards a lower compression bitrate direction. However, it is not feasible to obtain the actual bitrate of the NeRF representation during training stage because entropy coding is non-differentiable. Hence, we propose a spatial-temporal implicit entropy model for accurate rate estimation. Furthermore, unlike the learned entropy model in image compression [33], which is learned from large-scale training datasets, our proposed implicit entropy model is learned on-the-fly with the corresponding NeRF representation in the training stage.

Consequently, for each frame of the NeRF model, we opt to train a specific implicit entropy model, enabling us to accurately estimate the bitrate of the NeRF representation. Furthermore, we encode the implicit entropy model into the bitstream and the decoded entropy model will be used to decode NeRF representation at the decoder side. In particular, we employ the autoregressive entropy model to estimate the bitrate of the explicit NeRF representation. As shown in Fig. 5, the implicit entropy model consists of a two-layer shallow MLP network and incorporates both spatial and temporal context information to estimate the distribution of the NeRF representation.
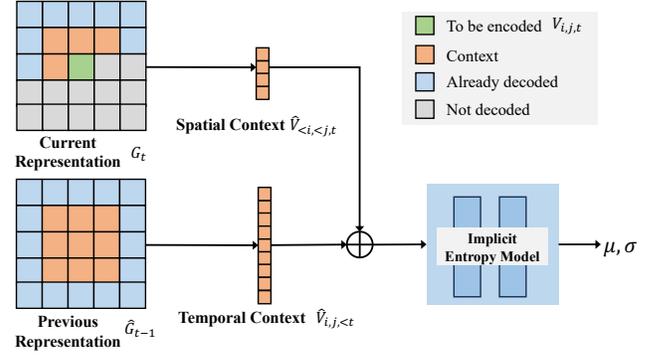

Figure 5: Illustration of spatial-temporal implicit entropy model. Utilizing the decoded spatial and temporal contexts to predict the distribution of the voxel to be encoded. Although we are actually searching for spatial context in a 3D space, for clarity, we use a 2D plane as an example in the illustration.

Here, to simplify the description, we assume the $\hat{G}_t$ is 2D representation and the voxel $V_{i,j,t}$ represents the corresponding item in the quantized representation $\hat{G}_t$, i.e., $V_{i,j,t} \in \hat{G}_t$, where (i, j) represents the spatial location. The essence of the autoregressive model is to predict the distribution of undecoded voxel using the already decoded voxel information. As shown in Fig 5, we model the distribution of undecoded voxels using proposed implicit entropy model. Assuming the current voxel to be encoded is $V_{i,j,t}$, we use the spatially adjacent and causally decoded voxels as the spatial context information, denoted as $\hat{V}_{<i,<j,t}$. At the same time, we take the voxel $\hat{V}_{|x-i|\leq 1,|y-j|\leq 1,t-1}$ from the previous frame as the temporal context information, denoted as $\hat{V}_{i,j,<t}$. As illustrated in Fig 5, after concatenating the temporal and spatial context information, we use a super-tiny MLP network to predict the current voxel's probability distribution:

$$\mu, \sigma = f_\psi \left(\hat{V}_{<i,<j,t} \oplus \hat{V}_{i,j,<t} \right). \tag{4}$$

$$p_{\hat{V}_{i,j,t}}(\hat{V}|\hat{V}_{<i,<j,t}, \hat{V}_{i,j,<t}) = \prod_i (\mathcal{N}(\mu, \sigma) * \mathcal{U}(-\tfrac{1}{2}, \tfrac{1}{2}))(\hat{V}_{i,j,t}). \tag{5}$$

where $\mathcal{N}(\mu, \sigma)$ denotes the Laplace distribution.

Then we can calculate the distribution's cumulative function $P_{cdf}(\cdot)$ using the predicted parameters μ and σ and estimate the bitrate of the current voxel:

$$\mathcal{L}_{rate} = \frac{1}{N} \sum_{V_{i,j,t} \in \hat{G}_t} -\log_2 \left[P_{cdf}\left(\tilde{V}_{i,j,t} + \frac{1}{2} \right) - P_{cdf}\left(\tilde{V}_{i,j,t} - \frac{1}{2} \right) \right].$$
$$\tag{6}$$

where N denotes the total number of voxels in the grids, $\tilde{V}_{i,j,t}$ denote the simulated quantized $V_{i,j,t}$.

Rate-Distortion Loss Function. Striking a balance between bitrate and distortion is pivotal to efficient NeRF representation compression. Thus, we estimate the bitrate of the NeRF representation and optimize the NeRF representation by using the rate-distortion

loss as follows,

$$\mathcal{L}_{total} = \sum_{\mathbf{r} \in \mathbf{R}} \|C(\mathbf{r}) - \hat{C}(\mathbf{r})\|^2 + \lambda(\mathcal{L}_{rate} + \alpha \mathcal{L}_{reg}). \quad (7)$$

where $C(\mathbf{r})$ is ground truth color, $\hat{C}(\mathbf{r})$ is the rendered color as shown in Equ. 2, and we use the L2 loss as the distortion loss. $\mathcal{L}_{reg}$ essentially serves as a rate constraint. By assigning a larger weight to $\mathcal{L}_{reg}$, the magnitude of the residual grid will be smaller, resulting in a lower bitrate. Therefore, we categorized $\mathcal{L}_{reg}$ and $\mathcal{L}_{rate}$ as bitrate-related loss. λ is a trade-off parameter for distortion and rate. To ensure stable training and control the trade-off between quality and bitrate solely through the λ parameter, we designed the weights of $\mathcal{L}_{rate}$ and $\mathcal{L}_{reg}$ to have a fixed linear relationship, specifically $\lambda_{reg} = \alpha * \lambda_{rate}$.

4.4 Overall Encoding and Decoding Pipeline

In summary, within the NeRF model, the components that need to be compressed and transmitted include the explicit representation, occupancy grid, rendering MLP, and the implicit entropy model. The encoding and decoding of the explicit representation rely on the implicit entropy model, while the other parts can be encoded and decoded independently.

Occupancy grid. The characteristic of the occupancy grid is that its elements are either 0 or 1. Thus, for compression, it is first flattened into a vector, grouped in sets of 8 to be packed into uint8 data, and then these uint8 data are compressed. Since a significant portion of the packed uint8 data consists of zeros, the LZMA dictionary encoding algorithm is employed for compression.

Rendering MLP and Implicit entropy model. Both the components are MLPs, with the parameters needing compression being the network's weights and biases. Given the small size of these parameters, we simply quantize them and then apply entropy encoding using a range coder [31].

Explicit representation. The encoding and decoding of the explicit representation rely on the distribution parameters estimated by the implicit entropy model. Hence, before encoding and decoding the explicit representation, the implicit entropy model must be encoded and decoded first. Based on the distribution parameters estimated by the implicit entropy model, we employ a range coder to encode and decode the explicit representation, following a spatio-temporal causal order.

Decoding pipeline. Fig. 6 illustrates the decoding pipeline of our compression framework. Initially, the occupancy grid, rendering MLP, and implicit entropy model are decoded from the bitstream. Subsequently, voxels in the explicit representation are decoded from the bitstream in causal order. Using the already decoded voxels as context, the distribution of undecoded voxels is predicted through an implicit entropy model. Based on the parameters of the distribution, voxels are decoded from the bitstream.

5 Experiments

5.1 Experimental Setup

Datasets. We utilized two datasets: HumanRF [17] and ReRF [48], to validate the effectiveness of our method. For the HumanRF
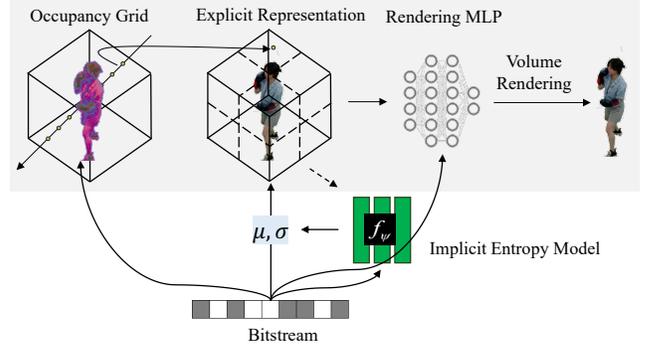

Figure 6: Decoding pipeline.

dataset [17], we selected 100 viewpoints with the same camera intrinsics. The image resolution was approximately 1020x750. Among these viewpoints, we designated viewpoints 15 and 25 as the test viewpoints, while the remaining 98 viewpoints were used for training. Regarding the ReRF dataset [17], it comprised approximately 75 viewpoints with an image resolution of 1920x1080. We chose viewpoints 6 and 39 as the test viewpoints following [52], while the rest of the viewpoints were employed for training.

Evaluation Metrics. To assess the compression performance of our method on the experimental datasets, we employ Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [51], and Learned Perceptual Image Patch Similarity (LPIPS) [55] as quality evaluation metrics. Additionally, we utilize KB per frame as the bitrate evaluation metric. Furthermore, for an overall comparison of compression performance, we utilize the BD-rate metric [2].

Implementation Details. In the experiment, we set the group size to 20, which means that an I-frame is inserted every 20 frames within the dynamic scene. For the rate-distortion loss, we defined four different λ values($7e-4$, $1e-3$, $2e-3$, and $5e-3$) to achieve different compression ratios. We set α to 10 for the regularization loss. For the training details of NeRF, we refer to the default configuration of DiF [7].

5.2 Comparison Results

In our experiments, our primary benchmarks are the state-of-the-art methods TeTriRF [52] and ReRF [48]. Fig. 7 and Fig. 8 illustrate the rate-distortion curves on the HumanRF [17] and ReRF [48] datasets respectively. The figures clearly show that our method achieves the best rate-distortion performance on all metrics—PSNR, SSIM, and LPIPS—both in the training and testing viewpoints. Due to the substantial disparity in the bitrate range between our method and ReRF [48], we only calculate the BD-rate in comparison with TeTriRF [52]. When computing the BD-rate, PSNR is selected as the quality metric. On the HumanRF dataset, our method achieves a BD-rate reduction of **-84.52%** for training viewpoints and **-85.49%** for testing viewpoints. On the ReRF dataset, our method attains a BD-rate reduction of **-60.58%** for training viewpoints and **-76.60%** for testing viewpoints.

Table 2: The quantitative results on the HumanRF and ReRF datasets. Bold data indicate the best performance, while <u>underlined</u> data indicate the second best.

| | HumanRF Dataset | | | | | | | ReRF Dataset | | | | | | |
| | training views | | | testing views | | | Size(KB) | training views | | | testing views | | | Size (KB) |
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)		PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	
ReRF	33.60	0.960	0.111	32.27	0.950	0.128	221.86	36.08	0.972	0.046	30.69	0.961	0.055	200.25
TeTriRF	34.84	0.965	0.096	32.49	0.954	0.115	94.14	37.65	0.977	0.039	32.21	0.968	0.045	122.15
Ours(Low)	<u>36.50</u>	<u>0.969</u>	<u>0.090</u>	<u>33.87</u>	<u>0.961</u>	<u>0.107</u>	**55.72**	<u>38.41</u>	<u>0.980</u>	<u>0.035</u>	<u>32.47</u>	<u>0.970</u>	<u>0.043</u>	**67.02**
Ours(High)	**37.40**	**0.973**	**0.079**	**34.30**	**0.965**	**0.095**	<u>80.63</u>	**39.28**	**0.984**	**0.031**	**32.70**	**0.973**	**0.039**	<u>116.64</u>
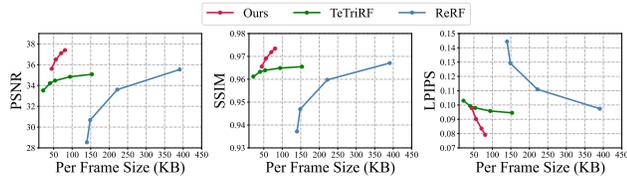
(a) Rate-distortion performance for the training views.
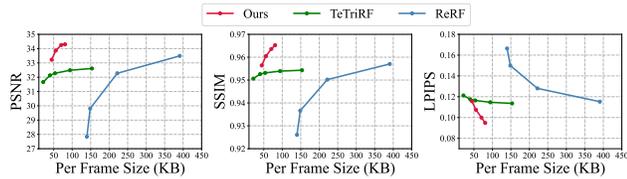
(b) Rate-distortion performance for the testing views.

Figure 7: The rate-distortion performance comparison results on HumanRF dataset. The BD-rate of our method compared to TeTriRF is as follows: in the training viewpoints, the BD-rate is -84.52%, and in the testing viewpoints, the BD-rate is -85.49%.
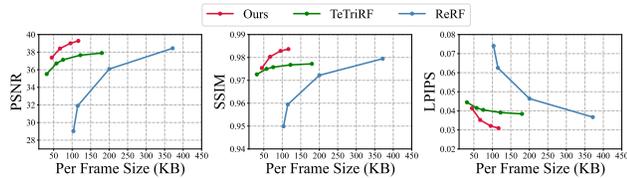
(a) Rate-distortion performance for the training views.
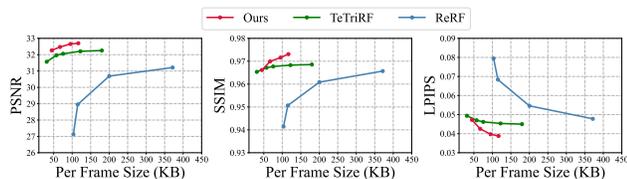
(b) Rate-distortion performance for the testing views.

Figure 8: The rate-distortion performance comparison results on ReRF dataset. The BD-rate of our method compared to TeTriRF is as follows: in the training viewpoints, the BD-rate is -60.58%, and in the testing viewpoints, the BD-rate is -76.60%.

Table 3: The comparison results of our method applied on TensoRF [6] and the baseline applied on TensoRF.

Ours (TensoRF)				Baseline (TensoRF)			
PSNR	SSIM	LPIPS	Size (KB)	PSNR	SSIM	LPIPS	Size(KB)
33.39	0.966	0.093	58.07	33.65	0.970	0.081	1745.43
32.30	0.963	0.099	47.14	32.45	0.962	0.097	1254.91
31.02	0.954	0.116	39.59	30.84	0.948	0.117	999.48
29.24	0.943	0.132	30.82	27.03	0.905	0.154	671.51

Tab. 2 shows the detailed quantitative results on HumanRF dataset and ReRF dataset, where Ours(Low) and Ours(High) respectively represent the results of our method under different λ configurations. The table indicates that both our method and TeTriRF can achieve good reconstruction quality at lower bitrate range, while ReRF requires significantly higher bitrate for similar quality level. Moreover, Ours(Low) achieves a bitrate substantially lower than TeTriRF with comparable reconstruction quality. And Ours(High) offers much higher reconstruction quality at a bitrate similar to TeTriRF, further demonstrating the rate-distortion performance advantage of our method.

Fig. 9 displays qualitative comparison results on the HumanRF's *actor07* sequence and the ReRF's *kpop* sequence. The figure illustrates that our method can reconstruct more details at a lower bitrate, such as the beard in *actor07* and the clothing logo in *kpop*, which are lost in the reconstructions by TeTriRF and ReRF. This demonstrates that our method also provides a superior subjective experience.

5.3 Evaluation on Plane Representaion

Our method is universally applicable to a large amount of explicit representation. To verify the effectiveness of our method, we further evaluated its performance on the plane-based representation TensoRF [6]. The comparison baseline is: modeling each frame in dynamic scenes with an independent TensoRF model, followed by quantization and entropy encoding of TensoRF after modeling is complete.

Tab. 3 shows the comparison results between our method and the baseline method at various compression ratios. It is evident that our method achieves significantly lower bitrates than the baseline method, without a noticeable decline in reconstruction quality. This demonstrates that our method is not only suitable for DiF [7] but also applicable to TensoRF [6], proving its effectiveness across different explicit representations.

Figure 9: Qualitative comparisons of *actor07* sequence from HumanRF [17] and *kpop* sequence from ReRF [48]. PSNR and size results are given at lower-left.

5.4 Ablation Studies

In the ablation studies, we aim to verify the efficacy of inter prediction-based dynamic modeling, adaptive quantization, and rate-distortion joint optimization we proposed. We conduct ablation studies on the *actor02* sequence from the HumanRF [17] dataset. Based on the full method, we disable dynamic modeling and adaptive quantization separately during training. Tab. 4 shows the bitrates for different ablation experiments at the same quality level (PSNR = 34 dB). The definition of "Baseline" is similar to Sec. 5.3, where each frame of the dynamic scene is modeled using an independent DiF model, followed by quantization and entropy encoding of the representation after modeling is complete.

Table 4: At the same quality level (PSNR = 34 dB), the bitrates for different ablation experiments. "w/o Dynamic" signifies the disabling of dynamic modeling, and "w/o Adaptive" indicates the disabling of adaptive quantization.

Baseline	w/o Dynamic	w/o Adaptive	Full
1193.29 KB	137.45 KB	80.31 KB	66.61 KB

From Tab. 4, it is apparent that, based on the "Full" method, either disabling dynamic modeling or adaptive quantization results in an increase in bitrate, indicating the effectiveness of these two modules. Comparing the "Baseline" with the other three experiments, it is evident that the bitrates of the other three experiments are significantly lower than that of the "Baseline", demonstrating the efficacy of our proposed rate-distortion joint optimization.

Moreover, we analyze the average bit consumption for I-frame and P-frame under different λ configurations, and the result is illustrated in Tab. 5. It is evident that the bit consumption of P-frames is significantly lower than that of I-frames, which validates

Table 5: Analysis of average bit consumption for I-frame and P-frame.

	$\lambda = 0.0007$	$\lambda = 0.001$	$\lambda = 0.002$	$\lambda = 0.005$
I-frame (KB)	272.28	207.69	161.42	116.25
P-frame (KB)	94.05	79.67	61.28	46.48

the effectiveness of our dynamic modeling method in removing inter-frame redundancy and reducing the bitrate of P-frames.

6 Conclusion

In this paper, we present a dynamic modeling and rate-distortion optimization framework tailored for explicit NeRF representation. By incorporating a rate-aware training strategy into the NeRF training process, our approach demonstrates enhanced compression efficiency. Utilizing an implicit entropy model for accurate rate estimation of NeRF representations, coupled with adaptive quantization, we further amplify the compression efficacy. Experimental results reveal that our method secures the highest compression efficiency across two benchmark datasets. The NeRF-based modeling and compression methodology we propose is capable of reducing the data volume of a single frame of radiance field to below 100KB, markedly advancing the transmission of volumetric videos.

References

[1] J. Ballé, Valero Laparra, and Eero P. Simoncelli. 2018. End-to-End Optimized Image Compression. In *Proceedings of Intl. Conf. on Learning Representations (ICLR)*.

[2] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-curves. *ITU SG16 Doc. VCEG-M33* (2001).

[3] Jill M Boyce, Renaud Doré, Adrian Dziembowski, Julien Fleureau, Joel Jung, Bart Kroon, Basel Salahieh, Vinod Kumar Malamal Vadakital, and Lu Yu. 2021. MPEG immersive video coding standard. *Proc. IEEE* 109, 9 (2021), 1521–1536.

[4] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. 2021. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 10 (2021), 3736–3764.

[5] Ang Cao and Justin Johnson. 2023. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 130–141.

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*. Springer, 333–350.

[7] Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. 2023. Dictionary fields: Learning a neural basis decomposition. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–12.

[8] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. 2020. Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[9] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. 2021. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 14304–14314.

[10] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. 2022. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

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

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

[13] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. 2021. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5712–5721.

[14] Sharath Girish, Abhinav Shrivastava, and Kamal Gupta. 2023. SHACIRA: Scalable HAsh-grid Compression for Implicit Neural Representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 17513–17524.

[15] Danillo Graziosi, Ohji Nakagami, Shinroku Kuma, Alexandre Zaghetto, Teruhiko Suzuki, and Ali Tabatabai. 2020. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Transactions on Signal and Information Processing* 9 (2020), e13.

[16] Haoyu Guo, Sida Peng, Yunzhi Yan, Linzhan Mou, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 2024. Compact neural volumetric video representations with dynamic codebooks. *Advances in Neural Information Processing Systems* 36 (2024).

[17] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. 2023. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–12.

[18] Atefeh Khoshkhahtinat, Ali Zafari, Piyush Mehta, and Nasser Nasrabadi. 2024. Laplacian-guided Entropy Model in Neural Codec with Blur-dissipated Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[19] Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont. 2024. C3: High-performance and low-complexity neural compression from a single image or video. (2024).

[20] Soonbin Lee, Fangwen Shu, Yago Sanchez, Thomas Schierl, and Cornelius Hellge. 2023. ECRF: Entropy-Constrained Neural Radiance Fields Compression with Frequency Domain Optimization. *arXiv preprint arXiv:2311.14208* (2023).

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

[22] Jiahao Li, Bin Li, and Yan Lu. 2023. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 22616–22626.

[23] Jiahao Li, Bin Li, and Yan Lu. 2024. Neural Video Compression with Feature Modulatio. (2024).

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

[25] Sicheng Li, Hao Li, Yiyi Liao, and Lu Yu. 2024. NeRFCodec: Neural Feature Compression Meets Neural Radiance Fields for Memory-efficient Scene Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[26] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. 2022. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5521–5531.

[27] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2021. 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*. 6498–6508.

[28] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2022. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *Advances in Neural Information Processing Systems* 35 (2022), 36762–36775.

[29] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. 2019. DVC: An End-to-end Deep Video Compression Framework. (2019).

[30] Omnia Mahmoud, Théo Ladune, and Matthieu Gendrin. 2023. CAwa-NeRF: Instant Learning of Compression-Aware NeRF Features. *arXiv preprint arXiv:2310.14695* (2023).

[31] G Nigel N Martin. 1979. Range encoding: an algorithm for removing redundancy from a digitised message. In *Proc. Institution of Electronic and Radio Engineers International Conference on Video and Data Recording*, Vol. 2.

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[33] David Minnen, Johannes Ballé, and George D Toderici. 2018. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems* 31 (2018).

[34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–15.

[35] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5865–5874.

[36] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10318–10327.

[37] Zizheng Que, Guo Lu, and Dong Xu. 2021. Voxelcontext-net: An octree based framework for point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6042–6051.

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

[39] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al. 2018. Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2018), 133–148.

[40] Xihua Sheng, Li Li, Dong Liu, and Houqiang Li. 2024. VNVC: A Versatile Neural Video Coding Framework for Efficient Human-Machine Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[41] Seungjoo Shin and Jaesik Park. 2023. Binary Radiance Fields. *arXiv preprint arXiv:2306.07581* (2023).

[42] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. 2001. The JPEG 2000 still image compression standard. *IEEE Signal processing magazine* 18, 5 (2001), 36–58.

[43] Rui Song, Chunyang Fu, Shan Liu, and Ge Li. 2023. Efficient Hierarchical Entropy Model for Learned Point Cloud Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[44] Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5459–5469.

[45] Vivienne Sze, Madhukar Budagavi, and Gary J Sullivan. 2014. High efficiency video coding (HEVC). In *Integrated circuit and systems, algorithms and architectures*. Vol. 39. Springer, 40.

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

[47] Gregory K Wallace. 1991. The JPEG still picture compression standard. *Commun. ACM* 34, 4 (1991), 30–44.

[48] Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. 2023. Neural Residual Radiance Fields for Streamably Free-Viewpoint Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 76–87.

[49] Liao Wang, Kaixin Yao, Chengcheng Guo, Zhirui Zhang, Qiang Hu, Jingyi Yu, Lan Xu, and Minye Wu. 2023. VideoRF: Rendering Dynamic Radiance Fields as 2D Feature Video Streams. *arXiv preprint arXiv:2312.01407* (2023).

[50] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. 2022. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13524–13534.

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

[52] Minye Wu, Zehao Wang, Georgios Kouros, and Tinne Tuytelaars. 2023. TeTriRF: Temporal Tri-Plane Radiance Fields for Efficient Free-Viewpoint Video. *arXiv preprint arXiv:2312.06713* (2023).

[53] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. 2021. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9421–9431.

[54] Dongmei Xue, Haichuan Ma, Li Li, Dong Liu, and Zhiwei Xiong. 2022. aiWave: Volumetric image compression with 3-D trained affine wavelet-like transform.

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

[56] Zhiyu Zhang, Guo Lu, Huanxiong Liang, Anni Tang, Qiang Hu, and Li Song. 2024. Efficient Dynamic-NeRF Based Volumetric Video Coding with Rate Distortion Optimization. *arXiv preprint arXiv:2402.01380* (2024).

[57] Zhiyu Zhang, Anni Tang, Chen Zhu, Guo Lu, Rong Xie, and Li Song. 2023. High-Fidelity Free-View Talking Head Synthesis for Low-Bandwidth Video Conference. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 1–5.

[58] Zihan Zheng, Houqiang Zhong, Qiang Hu, Xiaoyun Zhang, Li Song, Ya Zhang, and Yanfeng Wang. 2024. JointRF: End-to-End Joint Optimization for Dynamic Neural Radiance Field Representation and Compression. *arXiv preprint arXiv:2405.14452* (2024).

IEEE Transactions on Medical Imaging 42, 3 (2022), 606–618.