Compressing Streamable Free-Viewpoint Videos to 0.1 MB per Frame

Luyang Tang[1,2], Jiayu Yang[2], Rui Peng[1,2], Yongqi Zhai[1,2], Shihe Shen[1], Ronggang Wang[1,2*]

[1]Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen Graduate School, Peking University, China
[2]Pengcheng Laboratory, China
tly926@stu.pku.edu.cn

Abstract

The success of 3D Gaussian Splatting (3DGS) in static scenes has inspired numerous attempts to construct Free-Viewpoint Videos (FVVs) of dynamic scenes from multi-view videos. Despite advancements in current techniques, simultaneously achieving photo-realistic view synthesis results, fast on-the-fly training, real-time rendering, and low storage costs remains a formidable problem. To address these challenges, we propose the first Gaussian-based streamable FVV intelligent compression framework named iFVC. Specifically, we utilize an anchor-based Gaussian representation to model the scene. To achieve on-the-fly training, we propose a Binary Transformation Cache (BTC) to model the dynamic changes between adjacent timesteps, which not only ensures compactness but also supports precise bit rate estimation. Furthermore, we carefully design a high-resolution transformation tri-plane assisted by a saliency grid as our BTC, allowing for accurate dynamic capture. The entire pipeline is regarded as a joint optimization of rate and distortion to achieve optimal compression performance. Experiments on widely used datasets demonstrate the state-of-the-art performance of our framework in both synthesis quality and efficiency, i.e., achieving per-frame training in 13 seconds with a storage cost of 0.1 MB and real-time rendering at 120 FPS.

Code — https://github.com/Pomelomm/iFVC

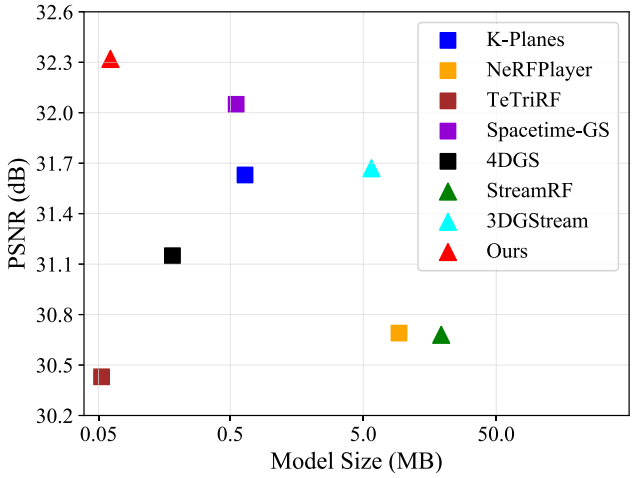

Figure 1: Rate-distortion performance compared with other methods on the N3DV dataset. □ represents offline training methods that require complete video sequences, and △ signifies online training on multi-view video streams. Our method achieves high-quality rendering with relatively low storage costs, while also supporting online training.

1 Introduction

Constructing Free-Viewpoints Videos (FVVs) from multi-view videos plays a critically important role in various applications such as virtual reality, telepresence, sports broadcasting and so on. It allows users to freely select their viewing angles of dynamic scenes, thus providing a unique and immersive exploration experience.

In recent years, Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) has demonstrated remarkable potential in novel view synthesis and promotes the development of FVV generation for dynamic scenes. However, despite these NeRF-based methods (Li et al. 2022a; Song et al. 2023) achieve photo-realistic synthesis results, they rely on costly sampling and evaluation at multiple points along each ray for volume rendering, which greatly limits rendering speed and hinders practical applications.

Recently, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has emerged as an alternative scene representation method due to its fast training, real-time rendering, and high-fidelity synthesis results. 3DGS leverages learnable Gaussians with various properties to represent 3D space and introduces an efficient rasterization-based scheme for rendering. However, it requires a substantial number of Gaussians to maintain high-quality rendering, which poses challenges for storage and transmission. Dynamic scenes contain information in both spatial and temporal domains, facing more severe storage challenges. Although some existing methods (Wu et al. 2024a; Li et al. 2024b) achieve compact dynamic scene representation by training a single model that combines all the frames, they necessitate complete video sequences for offline training and only support replaying dynamic scenes, thus unable to tackle online scenarios. A recent work 3DGStream (Sun et al. 2024) proposes a novel

*The corresponding author (rgwang@pkusz.edu.cn).

online training pipeline that employs a Neural Transformation Cache (NTC) to model the translations and rotations of Gaussians, achieving fast on-the-fly per-frame reconstruction and real-time rendering. However, it requires more than 2 GB to store the scene representation corresponding to a 300-frame multi-view video (about 10 seconds), posing significant challenges in the storage and transmission of long-duration dynamic scenes.

In this paper, we propose a novel **i**ntelligent **F**ree-Viewpoint **V**ideo **C**ompression (iFVC) framework, aiming to achieve high-fidelity synthesis, fast on-the-fly training, and real-time rendering with low storage costs. Specifically, our iFVC starts with an anchor-based Gaussian representation at timestep 0. For subsequent timestep t ($t > 0$), we introduce a Binary Transformation Cache (BTC) to model the transformations relative to the previous frame $t-1$ in a highly compact manner, fully exploiting the inter-frame similarities for fast online reconstruction. Furthermore, we carefully design a high-resolution transformation tri-plane assisted by a saliency 3D grid as our BTC, enhancing the capablity of binary grids to accurately capture dynamics. We formulate the per-frame training as a joint optimization of rate and distortion to minimize the entropy of scene representation while ensuring high reconstruction quality (see Fig. 1 for comparisons with the state-of-the-arts).

The contributions of our work are summarized as follows:

- We propose iFVC, the first Gaussian-based compression framework for efficient FVV streaming. Our iFVC is optimized by a rate-distortion loss in an end-to-end manner, achieving optimal compression performance.
- We propose a highly compact Binary Transformation Cache (BTC) to model scene changes between adjacent timesteps. The binary representation is not only more storage-friendly but also enables precise bit rate estimation during training.
- Our BTC is carefully designed as a saliency-assisted transformation tri-plane, capable of capturing accurate dynamic changes to improve reconstruction quality.
- Extensive experiments demonstrate the superiority of iFVC in synthesis quality and model size, while also supporting on-the-fly training and real-time rendering.

2 Related Work

Novel View Synthesis for Static Scenes

Given a set of 2D images, Novel View Synthesis (NVS) aims to generate novel-view images from virtual camera poses. In recent years, Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) and its variants (Liu et al. 2020; Martin-Brualla et al. 2021; Sun, Sun, and Chen 2022; Pumarola et al. 2021) have achieved impressive rendering results via implicit scene representation. However, due to the computational bottleneck of volume rendering, these methods typically cannot support real-time applications. Recently, 3D Gaussian splatting (Kerbl et al. 2023) has emerged as a promising scene representation method for NVS, which utilizes a fast rasterization technique to synthesize views and achieves notable training and rendering speedups.

Novel View Synthesis for Dynamic Scenes

Constructing FVVs for dynamic scenes from multi-view videos remains a challenging problem due to temporal variations. Various methods (Li et al. 2022a; Song et al. 2023; Attal et al. 2023; Wang et al. 2023a; Yang et al. 2024; Du et al. 2021) have explored extending static NeRF representations to dynamic scenes, with similar efforts applied to 3DGS. Some methods (Yang et al. 2024; Li et al. 2024a; Wu et al. 2024a; Lin et al. 2024) define canonical 3D Gaussians and employ MLP networks, 4D neural voxels, or explicit residuals to model deformation fields. Other methods (Yan et al. 2024; Yang et al. 2023; Li et al. 2024b; Duan et al. 2024) lift 3D Gaussians into 4D space and temporally slice them to model the scene at each timestamp. Despite the methods above achieving remarkable rendering quality for dynamic scenes, they generally follow an offline modeling paradigm and lack streamable capabilities. To address these constraints, StreamRF (Li et al. 2022a) models dynamic scenes with an incremental learning framework, performing complete training at timestep 0 and tuning the model for subsequent frames online. NeRFPlayer (Song et al. 2023) introduces a sliding-window scheme on feature channels to support streamable rendering. ReRF (Wang et al. 2023b) optimizes a motion grid and a residual feature grid frame by frame to model the spatial-temporal feature space sequentially. Recently, 3DGStream (Sun et al. 2024) and HiCoM (Gao et al. 2024) propose efficient 3DGS-based pipelines for FVV streaming, enabling on-the-fly and high-quality reconstruction for dynamic scenes.

Neural Scene Representation Compression

Neural explicit representation methods improve training and inference efficiency but inevitably increase the storage and transmission costs of scenes. To address this issue, many NeRF-based and 3DGS-based techniques achieve considerable compression ratios through methods such as pruning (Deng and Tartaglione 2023; Niedermayr, Stumpfegger, and Westermann 2024), quantization (Li et al. 2023; Lee et al. 2024), decomposition (Chen et al. 2022), and entropy constraints (Chen et al. 2024b,a; Wang et al. 2024a,c). Among these, BiRF (Shin and Park 2024) introduces a 2D-3D hybrid binary feature grid, offering a storage-efficient representation for radiance fields. For dynamic scenes with spatial and temporal redundancies, Dynamic-codebook (Guo et al. 2024) compresses the six feature planes into a compact codebook. Streamable scene representations face more severe storage challenges, as their model size increases linearly with the number of frames. Therefore, it is crucial to develop effective compression techniques to efficiently store and transmit such data. ReRF (Wang et al. 2023b), VideoRF (Wang et al. 2024b), and TeTriRF (Wu et al. 2024b) combine traditional image/video coding techniques for feature grid compression. HPC (Zheng et al. 2024) proposes an end-to-end pipeline for progressive volumetric video encoding, streaming, and decoding. These coding frameworks for streamable scene representations are geared towards inward-facing scenes with foreground masks and a few objects. All are based on NeRF, falling short in training time and rendering speed.

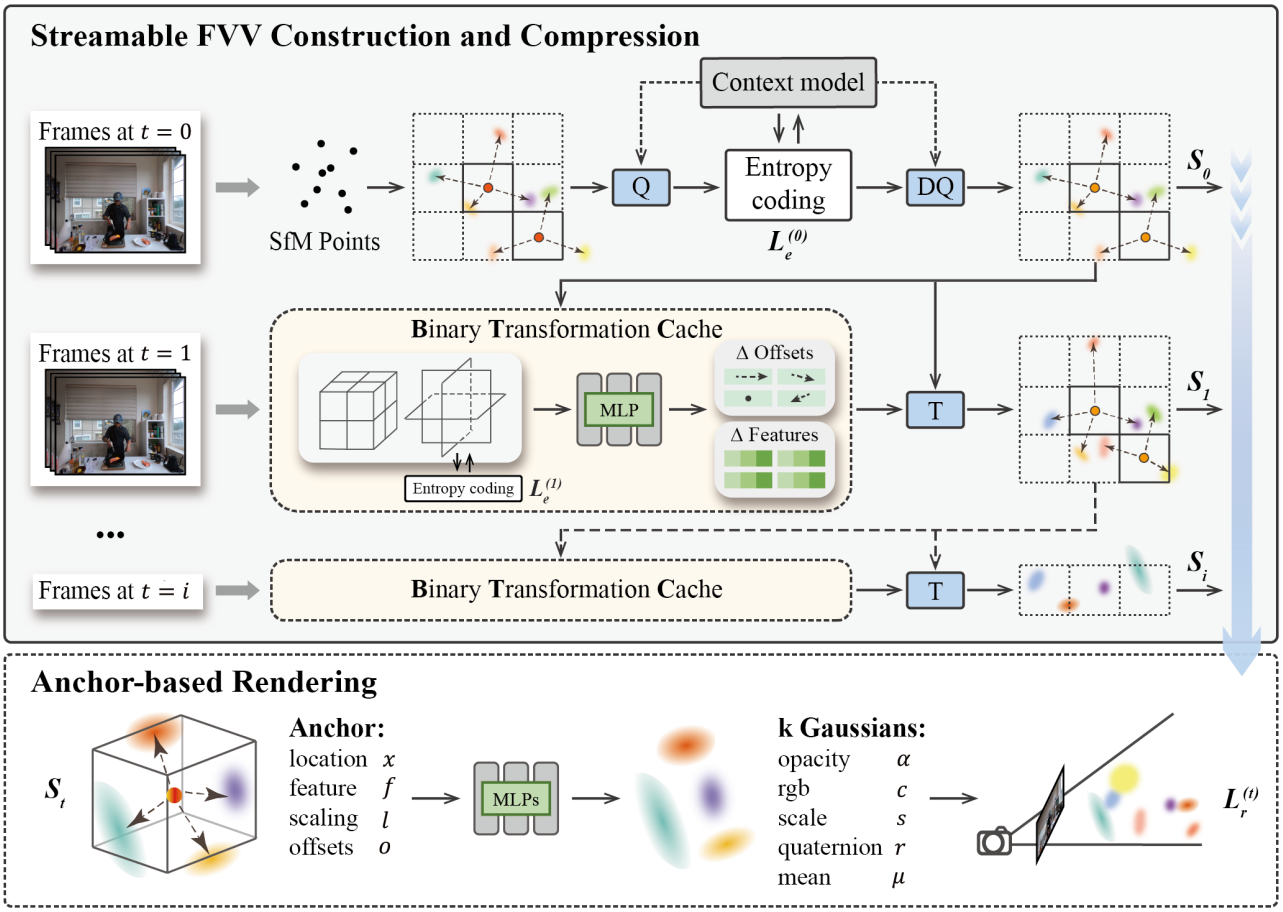

Figure 2: Overview of iFVC. To enable on-the-fly learning of dynamic scenes, we perform a frame-by-frame training pipeline to construct FVV streams. We utilize an anchor-based Gaussian representation at timestep 0. For the subsequent timestep i, we introduce a Binary Transformation Cache to model the translations and feature residuals relative to the previous timestep $i-1$ in a highly compact manner. The entire pipeline is formulated as a joint optimization of rate and distortion, aiming to achieve photo-realistic rendering quality at low storage costs. "Q", "DQ", and "T" denote the quantization, dequantization, and transformation operations, respectively. "Entropy coding" includes arithmetic encoding and decoding.

3 Preliminary

3.1 3D Gaussian Splatting

3DGS (Kerbl et al. 2023) explicitly models 3D space with anisotropic Gaussians and uses a fast differentiable rasterizer for image rendering. Starting from sparse points produced by Structure-from-Motion (SfM) (Schonberger and Frahm 2016), each Gaussian is defined by a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ and location (mean) $\mu \in \mathbb{R}^3$:

$$G(\mathbf{x}) = exp(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)),\ \Sigma = RSS^TR^T, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$ represents the coordinates of a 3D point and Σ is calculated by a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a scaling matrix $S \in \mathbb{R}^{3\times3}$ to ensure its positive semi-definite characteristics. During rendering, Gaussians are splatted to 2D and sorted in depth order. Through α-composed blending, we can get the pixel color $C \in \mathbb{R}^3$:

$$C = \sum_{i\in N} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

where $\alpha \in \mathbb{R}$ is the Gaussian's opacity after 2D projection, $c \in \mathbb{R}^3$ represents the view-dependent color, and N is the sorted points overlapping the pixel.

3.2 Anchor-based Gaussian Splatting

Scaffold-GS (Lu et al. 2024) introduces a structured representation method, utilizing anchors to cluster neighboring similar Gaussians for compact modeling and high-quality rendering. Each anchor consists of a location $x \in \mathbb{R}^3$, a local feature vector $f \in \mathbb{R}^D$, an anchor scaling factor $l \in \mathbb{R}^3$ and k Gaussian offsets $o \in \mathbb{R}^{k\times3}$. During rendering, k neural Gaussians are spawned from an anchor x. Their locations are given by:

$$\{\mu^i\}_{i=0}^{k-1} = x + \{o^i\}_{i=0}^{k-1} \cdot l. \tag{3}$$

Their properties are predicted through MLPs F:

$$\{c^i, r^i, s^i, \alpha^i\}_{i=0}^{k-1} = F(f, \sigma_c, \overrightarrow{d_c}), \tag{4}$$

where $c^i \in \mathbb{R}^3$, $r^i \in \mathbb{R}^4$, $s^i \in \mathbb{R}^3$, and $\alpha^i \in \mathbb{R}$ represent the color, rotation quaternion, scale, and opacity of the i-th Gaussian, respectively. Additionally, σ_c is the relative distance between the anchor and the camera, and $\overrightarrow{d_c}$ indicates the viewing direction. Gaussians with opacity α greater than τ will be selected to synthesize the viewpoint image.

4 Methods

4.1 Overview

Given a multi-view video stream, our framework performs on-the-fly FVV construction and compression in a frame-by-frame manner, as depicted in Fig. 2. We utilize an anchor-based Gaussian representation (Lu et al. 2024) as our base model. Firstly, we train a complete scene representation at timestep 0. For subsequent timesteps, a highly compact Binary Transformation Cache is employed to model dynamic changes between adjacent frames. The outputs of BTC are used to transform the previous representation S_{t-1} to the current timestep t $(t > 0)$, effectively leveraging inter-frame similarities to facilitate fast on-the-fly training. We only need to store BTC for the current timestep, greatly reducing temporal redundancy. To achieve an optimal rate-distortion performance, we apply an entropy loss to estimate the bit consumption of BTC during training. The entire training process is regarded as a joint optimization of model size (rate) and reconstruction quality (distortion).

4.2 Dynamic Anchors

In our framework, the FVV is represented by a set of dynamic anchors along timesteps, with BTC used to model the property changes of Gaussians in these anchors. However, we found that directly extending the anchor-based representation to dynamic scenes is non-trivial. As shown in formula (3), the Gaussian location μ is related to the anchor location x, scaling factor l, and offsets o. Simultaneously learning changes $\{\Delta x, \Delta l, \Delta o\}$ to model Gaussian translations may lead to suboptimal solutions, as there are numerous combinations that can yield the same result. For convenience, we assume that only offsets o and local feature vector f change over time, while anchor location x and scaling factor l remain constant. So each dynamic anchor has the following parameters:

- an anchor location x (consistent over all timesteps),
- an anchor scaling factor l (consistent over all timesteps),
- k Gaussian offsets o_t for each timestep t,
- a local feature vector f_t for each timestep t.

Our BTC at timestep t takes x as input and outputs the Gaussian translations Δo and feature residuals Δf relative to the previous timestep. Then we can obtain the transformed anchors for t:

$$\begin{aligned} o_t &= o_{t-1} + \Delta o, \\ f_t &= f_{t-1} + \Delta f, \end{aligned} \tag{5}$$

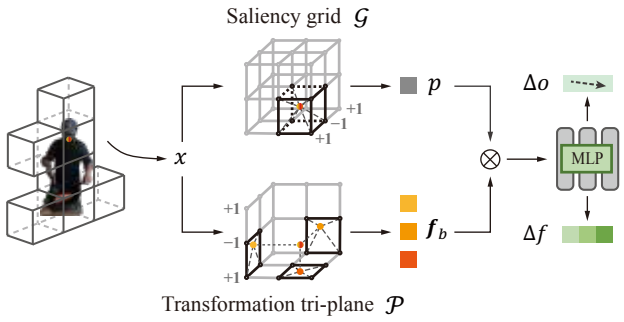

Figure 3: Illustration of BTC. It consists of a high-resolution transformation tri-plane and a saliency grid. The shallow fully-fused MLP predicts the translations Δo and residuals Δf from the interpolated features.

where Δf, as a high-dimensional feature vector, can provide flexible property changes of Gaussians in the time domain, including color, opacity, rotation, and scale. Furthermore, the number of activated Gaussians can be adjusted dynamically by opacity, allowing the model to handle emerging objects.

4.3 Binary Transformation Cache

Motivation of the Binary Representation. A simple idea for on-the-fly training is to directly finetune the previous representation S_{t-1} $(t > 0)$. However, it suffers from slow learning speed and high storage costs. 3DGStream (Sun et al. 2024) pioneers the use of multi-resolution hash encoding and a fully-fused MLP derived from INGP (Müller et al. 2022) to predict the translations and rotations of Gaussians between adjacent frames, achieving fast training and high-quality reconstruction. Although hash tables help reduce the storage requirements for encoding the entire scene to some extent, they still have a non-negligible model size. To this end, we seek a more compact structure to accurately model the transformations in the temporal domain. Motivated by the success of BiRF (Shin and Park 2024) in static scenes, we propose our highly compact Binary Transformation Cache (BTC) to model the translations and feature residuals in the anchor. On the one hand, the binary representation discards the expensive floating-point data, resulting in a great reduction in storage overhead. On the other hand, its parameters are in a format of either $+1$ or -1, enabling easy and precise bit rate estimation, thus enabling the joint optimization of rate and distortion during training.

Structure Design. Obviously, while binarization greatly reduces storage requirements, it also diminishes the representational capacity of features. To address this, a common design (Shin and Park 2024; Chen et al. 2024a) for binary grids in static scenes is a hybrid 2D-3D structure, leveraging two types of features to enhance the expressive power. However, this structure is not suitable for modeling transformations in dynamic scenes (32.29 dB drops to 31.89 dB in Tab. 4). Since the dynamics are typically concentrated in certain areas, a large portion of low-resolution 3D grids struggle to capture detailed changes. This not only impairs the

(a) (b)

Figure 4: Visualization of (a) translations and (b) saliency weights.

representation of high-resolution features but also leads to inefficient use of storage. To attain a good trade-off between effectiveness and compactness, we design a high-resolution transformation tri-plane for precisely and accurately modeling the changes in numerous anchors. Moreover, an additional one-channel saliency grid is introduced to mitigate hash collisions at high-resolution planes, further enhancing the reconstruction quality of our BTC in a compact manner.

Fig. 3 shows the structure design of BTC. Given an anchor at location x, we interpolate the transformation tri-plane $\mathcal{P}$ and saliency grid $\mathcal{G}$ as follows:

$$\begin{aligned} f_b &= interp(x, \mathcal{P}), \\ p &= Sigmoid(interp(x, \mathcal{G})), \end{aligned} \tag{6}$$

where $Sigmoid$ operator is used to ensure $0 < p < 1$. Then a light-weight MLP is used to predict the transformation for the anchor x:

$$\Delta o, \Delta f = \text{MLP}(p f_b), \tag{7}$$

where $\Delta o \in \mathbb{R}^{k\times 3}$, $\Delta f \in \mathbb{R}^{D}$ represent the property changes of k Gaussians. For two anchors x_1, x_2 with hash collisions in $\mathcal{P}$, their saliency weights p_1, p_2 enable different inputs to MLP, thus mitigating negative effects of collisions. We visualize Δo and p in Fig. 4. It shows that our BTC successfully models hand movements, while the saliency grid $\mathcal{G}$ tends to assign more attention to important regions in dynamic scenes, such as moving objects and object boundaries.

4.4 Rate-Distortion Optimization

To achieve the optimal rate-distortion performance at timestep t $(t > 0)$, we calculate the weighted sum of the rendering loss $\mathcal{L}_r^{(t)}$ and the entropy loss $\mathcal{L}_e^{(t)}$ as our supervision signals:

$$\mathcal{L}^{(t)} = \mathcal{L}_r^{(t)} + \lambda_t \mathcal{L}_e^{(t)}. \tag{8}$$

The rendering loss is used to ensure the reconstruction quality, which is calculated by:

$$\mathcal{L}_r^{(t)} = \mathcal{L}_1 + \lambda_{SSIM} \mathcal{L}_{SSIM}. \tag{9}$$

For entropy loss, we count the occurrence frequency h_f of the symbol "+1" in our transformation tri-plane $\mathcal{P}$ to estimate its bit consumption (Chen et al. 2024b):

$$\mathcal{L}_e^{(t)} = M_+ \times (-\log_2(h_f)) + M_- \times (-\log_2(1-h_f)), \tag{10}$$

where M_+, M_- represent the total counts of "+1" and "−1", respectively.

At timestep 0, we also introduce a context model (Chen et al. 2024a) for compression to estimate the quantization steps and probability distributions of anchor attributes.

5 Experiments

5.1 Experimental Setup

We conduct experiments on three real-world dynamic scene datasets as follows:

- **Neural 3D Video (N3DV)** (Li et al. 2022b) contains 6 dynamic scenes captured by a multi-view system of 21 cameras at a resolution of 2704×2028. Each multi-view video consists of 300 frames. Following prior works (Sun et al. 2024), we downsample the original videos by a factor of two for training and testing.
- **Meet Room** (Li et al. 2022a) contains 3 dynamic scenes captured by 13 cameras at a resolution of 1280×720. Each multi-view video also consists of 300 frames. Same as the experiment setting of 3DGStream (Sun et al. 2024) and StreamRF (Li et al. 2022a), we utilize 12 views for training and reserve 1 for testing.
- **VRU Basketball Game** contains 2 dynamic scenes captured by 34 cameras at a resolution of 1920×1080. Each multi-view video contains 250 frames. We utilize 30 views for training and reserve 4 for testing. The dataset captures real basketball games, featuring many moving basketball players with fast and large-scale movements, posing challenges for dynamic modeling.

The whole framework starts with sparse points from SfM at timestep 0. To obtain a high-quality and compact initial representation, we train for 15K steps on the N3DV and MeetRoom datasets, and for 30K steps on the VRU dataset. For subsequent frames t $(t > 0)$, our transformation tri-plane consists of 4-level 2D embeddings, whose resolutions range from 512 to 4096 and feature dimension is 4. The size of our one-channel saliency grid is $514 \times 514 \times 514$. We implement our transformation tri-plane and saliency grid using binary hash encoding to reduce the storage cost. The maximum hash table size is 2^{15}. We train our BTC for 300 iterations and control the storage size of each frame by adjusting the weight parameter λ_t in the loss function (set to 0.004 by default).

5.2 Comparisons

We compare the proposed method with a range of representative methods. The experimental results of these offline and online training methods are derived from their original papers. Tab. 1 and Tab. 2 present the averaged metrics over the whole 300 frames for each scene in the N3DV dataset and MeetRoom datasets, respectively. Offline training methods use a single model to represent a group of frames, which inherently offers a storage advantage over online methods. As shown in Tab. 1, iFVC not only enables online training but also achieves higher rendering quality with comparable or even smaller model size compared to offline methods. When compared to the state-of-the-art online training method, 3DGStream, our framework slightly increases training and testing time but greatly reduces storage requirements (by 86× on N3DV and 45× on MeetRoom) and improves synthesis quality. For challenging scenes with large-scale and complex motions in the VRU dataset, iFVC also demonstrates higher rendering quality at lower storage costs (as

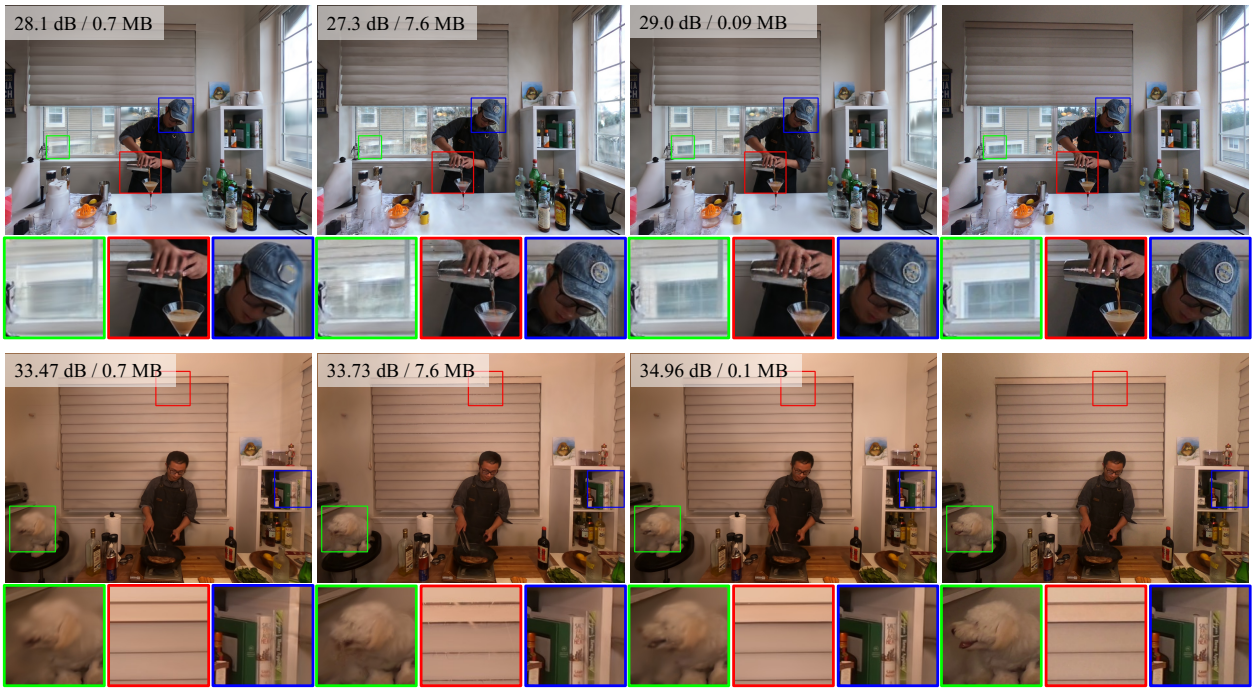

Figure 5: Qualitative comparisons on the *coffee martini* and *sear steak* scene in the N3DV dataset. For a fair comparison, we retrain their codes with the same initial sparse points as ours.

Methods	PSNR↑ (dB)	Storage↓ (MB)	Train↓ (mins)	Render↑ (FPS)
Offline training				
K-Planes	31.63	1.0	0.8	0.15
HyperReel	31.10	1.2	1.8	2.00
MixVoxels	30.80	1.7	0.3	16.7
TeTriRF	30.43	**0.06**	0.65	4.2
NeRFPlayer	30.69	17.1	1.2	0.05
4DGS	31.15	0.3	**0.13**	30
4DRotor-GS	31.62	-	0.2	**277**
Gaussian-Flow	32.00	-	0.14	-
Realtime-4DGS	32.01	-	-	114
Spacetime-GS	32.05	0.7	0.8	140
SaRO-GS	32.15	-	-	40
Online training				
StreamRF	30.68	31.4	0.23	8.3
3DGStream	31.67	7.8	0.20	215
iFVC	**32.32**	0.09	0.22	126

Table 1: Quantitative comparisons on the N3DV dataset. Highlights are **best** and second best.

shown in Tab. 3). In general, the proposed iFVC achieves high-quality synthesis, low storage costs, and supports on-the-fly training and real-time rendering, primarily due to: (1) the well-designed anchor-based coding framework, which

Methods	PSNR↑ (dB)	Storage↓ (MB)	Train↓ (mins)	Render↑ (FPS)
StreamRF	26.72	9.0	0.17	10
3DGStream	30.79	4.1	**0.10**	**288**
iFVC	**32.38**	**0.09**	0.14	157

Table 2: Quantitative comparisons on the MeetRoom dataset.

enables joint optimization of rate and distortion; and (2) the compact and effective BTC design, which facilitates accurate dynamic capture.

We provide qualitative comparisons of the state-of-the-art online method 3DGStream and offline method Spacetime-GS in Fig. 5. The proposed iFVC achieves competitive rendering quality in both dynamic and static regions at lower storage costs. Compared to 3DGStream, our method effectively handles emerging objects (e.g., the coffee in *Coffee Martini* scene) by learning feature residuals without adding extra points or incurring additional storage overhead. For more detailed experimental results, please refer to our supplementary material.

5.3 Ablation Study

Design of Binary Transformation Cache. To validate the effectiveness of our BTC, we explore different structures for modeling dynamic changes as Tab. 4 shows, including (a)

Methods	PSNR↑ (dB)	Storage↓ (MB)	Train↓ (mins)
3DGStream	26.20	8.12	**0.22**
iFVC (500 iterations)	**29.55**	**0.16**	0.59

Table 3: Quantitative comparisons on the VRU dataset.

Methods	PSNR (dB)	Storage (MB)
(a) NTC	32.29	7.60
(b) HBG	31.89	0.11
(c) BTP	32.24	0.12
(d) SA-BTP (**BTC**)	**32.32**	**0.09**

Table 4: Ablation study of BTC on the N3DV dataset.

Transformation	PSNR (dB)	Storage (MB)
(a) $\Delta f, \Delta o$	32.32	0.09
(b) $\Delta f, \Delta o, \Delta x$	32.03	0.20
(c) $\Delta f, \Delta o, \Delta x, \Delta l$	31.76	0.21

Table 5: Ablation study of our dynamic anchor design on the N3DV dataset.

Resolutions	PSNR (dB)	Storage (MB)
128, 256, 512, 1024	32.09	0.077
256, 512, 1024, 2048	32.25	0.087
512, 1024, 2048, 4096	32.32	0.092
1024, 2048, 4096, 8192	32.36	0.098

Table 6: Ablation study of tri-plane resolution on the N3DV dataset.

Neural Transformation Cache from 3DGStream (NTC), (b) Hybrid 2D-3D Binary Grids from BiRF (HBG), (c) Binary Transformation tri-Plane in our framework (BTP), and (d) Saliency-Assisted Transformation tri-Plane (SA-BTP). For a fair comparison, we model subsequent frames based on the same initial representation S_0 and report the average metrics over the entire videos. Experiments (a) and (b) show that the commonly used hybrid 2D-3D binary grid reduces model size but leads to an unacceptable performance drop. In contrast, the high-resolution tri-plane design allows for more accurate modeling of dynamic changes, thus improving reconstruction quality (experiments (b) and (c)). Experiments (c) and (d) show that an extra saliency grid not only enhances reconstruction quality but also reduces model size. The main reason is that the saliency grid improves the representation capability of the transformation tri-plane by mitigating the negative effects of hash collisions, enabling more compact modeling. In Fig. 6, we present the per-frame results on the *coffee martini* scene. Our BTC achieves comparable per-frame reconstruction quality to NTC with an 86× compression ratio, further demonstrating the effectiveness of our design.

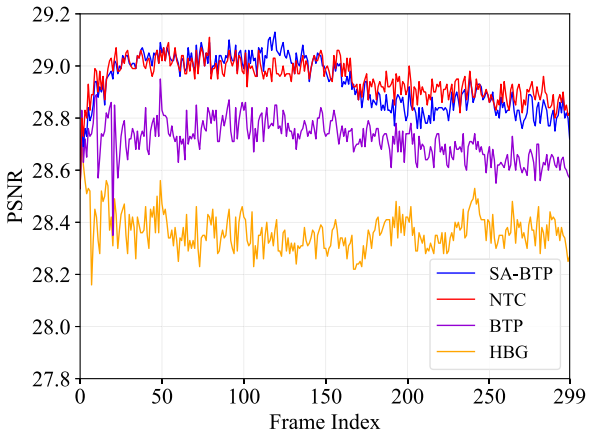

Figure 6: Per-frame results on the *coffee martini* scene.

Design of Dynamic Anchors. We conduct experiments on the N3DV dataset to evaluate our dynamic anchor design. Experiment (a) in Tab. 5, which fixes anchor location x and scaling factor l, achieves superior performance in rendering quality and model size. As formula (3) shows, the Gaussian location μ is determined by x, l and o. Simultaneously learning changes of these parameters to model Gaussian motions is prone to falling into suboptimal solutions. This could explain the lower reconstruction quality of experiments (b) and (c). In addition, the dynamic attributes that need to be modeled in our BTC increase, leading to a larger model size in (b) and (c).

Resolution of Tri-plane $\mathcal{P}$. As shown in Tab. 6, we evaluate the performance of our framework on the N3DV dataset under transformation tri-planes at different resolutions. Higher-resolution planes allow for more detailed modeling of dynamic areas, enhancing rendering quality but increasing storage requirements. However, as the resolution increases, the improvements in rendering quality gradually diminish and eventually reach a bottleneck, primarily due to higher resolutions being more prone to hash collisions, which negatively impact the representation capability.

6 Discussions and Limitations

In this paper, we propose iFVC, the first end-to-end Gaussian-based compression framework for FVV streaming, simultaneously achieving fast on-the-fly training, high-fidelity synthesis, and real-time rendering at low storage costs. Despite the superior performance, iFVC still has some limitations, which will be explored in our future research. Similar to other streamable reconstruction methods, the quality of the initial representation is crucial to iFVC. Additionally, to ensure training efficiency, we limit the number of iterations, which may result in poor fitting of large-scale motions and cause error accumulation. Moreover, how to further improve the reconstruction and compression performance of dynamic scenes with large-scale and complex motions deserves further investigation.

Acknowledgments

This work is financially supported by Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (Grant No. 2024B1212010006), National Natural Science Foundation of China U21B2012, Shenzhen Science and Technology Program-Shenzhen Cultivation of Excellent Scientific and Technological Innovation Talents project (Grant No. RCJC20200714114435057), this work is also financially supported for Outstanding Talents Training Fund in Shenzhen.

References

Attal, B.; Huang, J.-B.; Richardt, C.; Zollhoefer, M.; Kopf, J.; O'Toole, M.; and Kim, C. 2023. Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16610–16620.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, 333–350. Springer.

Chen, Y.; Wu, Q.; Cai, J.; Harandi, M.; and Lin, W. 2024a. HAC: Hash-grid Assisted Context for 3D Gaussian Splatting Compression. *arXiv preprint arXiv:2403.14530*.

Chen, Y.; Wu, Q.; Harandi, M.; and Cai, J. 2024b. How Far Can We Compress Instant-NGP-Based NeRF? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20321–20330.

Deng, C. L.; and Tartaglione, E. 2023. Compressing explicit voxel grid representations: fast nerfs become also small. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1236–1245.

Du, Y.; Zhang, Y.; Yu, H.-X.; Tenenbaum, J. B.; and Wu, J. 2021. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 14304–14314. IEEE Computer Society.

Duan, Y.; Wei, F.; Dai, Q.; He, Y.; Chen, W.; and Chen, B. 2024. 4D-Rotor Gaussian Splatting: Towards Efficient Novel View Synthesis for Dynamic Scenes. In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Gao, Q.; Meng, J.; Wen, C.; Chen, J.; and Zhang, J. 2024. HiCoM: Hierarchical Coherent Motion for Dynamic Streamable Scenes with 3D Gaussian Splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Guo, H.; Peng, S.; Yan, Y.; Mou, L.; Shen, Y.; Bao, H.; and Zhou, X. 2024. Compact neural volumetric video representations with dynamic codebooks. *Advances in Neural Information Processing Systems*, 36.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Lee, J. C.; Rho, D.; Sun, X.; Ko, J. H.; and Park, E. 2024. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21719–21728.

Li, D.; Huang, S.-S.; Lu, Z.; Duan, X.; and Huang, H. 2024a. ST-4DGS: Spatial-Temporally Consistent 4D Gaussian Splatting for Efficient Dynamic Scene Rendering. In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Li, L.; Shen, Z.; Wang, Z.; Shen, L.; and Bo, L. 2023. Compressing volumetric radiance fields to 1 mb. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4222–4231.

Li, L.; Shen, Z.; Wang, Z.; Shen, L.; and Tan, P. 2022a. Streaming radiance fields for 3d video synthesis. *Advances in Neural Information Processing Systems*, 35: 13485–13498.

Li, T.; Slavcheva, M.; Zollhoefer, M.; Green, S.; Lassner, C.; Kim, C.; Schmidt, T.; Lovegrove, S.; Goesele, M.; Newcombe, R.; et al. 2022b. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5521–5531.

Li, Z.; Chen, Z.; Li, Z.; and Xu, Y. 2024b. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8508–8520.

Lin, Y.; Dai, Z.; Zhu, S.; and Yao, Y. 2024. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21136–21145.

Liu, L.; Gu, J.; Zaw Lin, K.; Chua, T.-S.; and Theobalt, C. 2020. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33: 15651–15663.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20654–20664.

Martin-Brualla, R.; Radwan, N.; Sajjadi, M. S.; Barron, J. T.; Dosovitskiy, A.; and Duckworth, D. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7210–7219.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Niedermayr, S.; Stumpfegger, J.; and Westermann, R. 2024. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10349–10358.

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10318–10327.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Shin, S.; and Park, J. 2024. Binary radiance fields. *Advances in neural information processing systems*, 36.

Song, L.; Chen, A.; Li, Z.; Chen, Z.; Chen, L.; Yuan, J.; Xu, Y.; and Geiger, A. 2023. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5): 2732–2742.

Sun, C.; Sun, M.; and Chen, H.-T. 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.

Sun, J.; Jiao, H.; Li, G.; Zhang, Z.; Zhao, L.; and Xing, W. 2024. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20675–20685.

Wang, F.; Tan, S.; Li, X.; Tian, Z.; Song, Y.; and Liu, H. 2023a. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19706–19716.

Wang, H.; Zhu, H.; He, T.; Feng, R.; Deng, J.; Bian, J.; and Chen, Z. 2024a. End-to-End Rate-Distortion Optimized 3D Gaussian Representation. *arXiv preprint arXiv:2406.01597*.

Wang, L.; Hu, Q.; He, Q.; Wang, Z.; Yu, J.; Tuytelaars, T.; Xu, L.; and Wu, M. 2023b. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 76–87.

Wang, L.; Yao, K.; Guo, C.; Zhang, Z.; Hu, Q.; Yu, J.; Xu, L.; and Wu, M. 2024b. VideoRF: Rendering Dynamic Radiance Fields as 2D Feature Video Streams. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 470–481.

Wang, Y.; Li, Z.; Guo, L.; Yang, W.; Kot, A. C.; and Wen, B. 2024c. ContextGS: Compact 3D Gaussian Splatting with Anchor Level Context Model. *arXiv preprint arXiv:2405.20721*.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2024a. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20310–20320.

Wu, M.; Wang, Z.; Kouros, G.; and Tuytelaars, T. 2024b. TeTriRF: Temporal Tri-Plane Radiance Fields for Efficient Free-Viewpoint Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6487–6496.

Yan, J.; Peng, R.; Tang, L.; and Wang, R. 2024. 4D Gaussian Splatting with Scale-aware Residual Field and Adaptive Optimization for Real-time Rendering of Temporally Complex Dynamic Scenes. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 7871–7880.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20331–20341.

Yang, Z.; Yang, H.; Pan, Z.; Zhu, X.; and Zhang, L. 2023. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*.

Zheng, Z.; Zhong, H.; Hu, Q.; Zhang, X.; Song, L.; Zhang, Y.; and Wang, Y. 2024. HPC: Hierarchical Progressive Coding Framework for Volumetric Video. *arXiv preprint arXiv:2407.09026*.