CAGS: Color-Adaptive Volumetric Video Streaming with Dynamic 3D Gaussian Splatting

DAHENG YIN, Simon Fraser University, Canada and Jiangxing Intelligence Inc., China
YILI JIN, McGill University, Canada and Simon Fraser University, Canada
JIANXIN SHI, Nankai University, China and Simon Fraser University, Canada
ISAAC DING, Simon Fraser University, Canada
MIAO ZHANG, Simon Fraser University, Canada
FANGXIN WANG, The Chinese University of Hong Kong, Shenzhen, China
ZHAOWU HUANG, Fuzhou University, China and Southeast University, China
CONG ZHANG, Simon Fraser University, Canada and Jiangxing Intelligence Inc., China
JIANGCHUAN LIU*, Simon Fraser University, Canada
FANG DONG, Southeast University, China

Volumetric video (VV) streaming delivers truly immersive viewing experiences over the Internet, serving as a critical foundation for next-generation applications, including immersive telepresence in the metaverse, the surveillance of remote ecological systems, and robotic teleoperation for embodied AI, and beyond. Beyond immersive viewing, these applications turn VV streaming into a real-time interface to remote physical environments, imposing new system-level demands for photorealistic scene representation, low-latency interaction, and robust performance under heterogeneous network conditions. 3D Gaussian Splatting (3DGS) has been widely used for real-time photorealistic rendering, offering superior visual quality and rendering performance, but it faces challenges due to bandwidth consumption. Furthermore, as the foundation of adaptive VV streaming, existing Levels of Detail (LoD) methods based on density are not well-suited to Gaussian representations, leading to visible gaps and severe quality degradation. Recent studies have also explored attribute compression techniques to reduce bandwidth consumption. Our preliminary studies reveal that aggressive attribute compression primarily causes color distortion, which can be effectively corrected in the rendered image using a reference image. Motivated by these findings, we propose a novel Color-Adaptive scheme for adaptive VV streaming that uses vector quantization (VQ) to establish LoDs and correct color distortions with low-resolution reference images. We further present CAGS, an adaptive VV streaming system compatible with diverse Gaussian representations, which integrates the Color-Adaptive scheme by rendering reference images on the streaming server and performing color restoration on the client. Extensive experiments on our prototype system demonstrate that CAGS outperforms the existing adaptive streaming systems in PSNR by 5∼20 dB under fluctuating bandwidth, operates significantly faster than existing scalable Gaussian compression methods, and generalizes across different Gaussian representations. The code is available at https://github.com/yindaheng98/ColorAdaptiveGaussianSplatting.

CCS Concepts: • **Information systems** → **Multimedia streaming**; • **Networks** → *Application layer protocols.*

Additional Key Words and Phrases: Volumetric Video Streaming, 3D Gaussian Splatting, Color Restoration, Vector Quantization

ACM Reference Format:
Daheng Yin, Yili Jin, Jianxin Shi, Isaac Ding, Miao Zhang, Fangxin Wang, Zhaowu Huang, Cong Zhang, Jiangchuan Liu, and Fang Dong. 2026. CAGS: Color-Adaptive Volumetric Video Streaming with Dynamic 3D Gaussian Splatting. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3799902.3811058

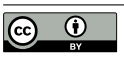
1 Introduction

Modern Internet infrastructure and real-time streaming technologies are reshaping online media, evolving it from delivering static images and videos toward live, interactive, photorealistic spatial experiences. Volumetric Video (VV) streaming is central to this evolution, which captures dynamic scenes as evolving 3D content, allowing interactive exploration from arbitrary viewpoints with six degrees of freedom (6DoF). Built on this infrastructure, emerging applications such as immersive communication, live digital twins, and embodied intelligence are turning VV streaming from an immersive viewing medium into a real-time interface to the physical world. This transition introduces new system-level demands: VV streaming must deliver high visual fidelity for accurate interpretation, minimal latency for responsive interaction, and robust performance across heterogeneous network conditions for reliable deployment.

Prior studies have established VV streaming as an effective 6DoF medium for real-world scenes, supporting more interactive experiences than conventional video in telepresence, collaboration, education, training, and immersive media [Guan et al. 2023; Han et al. 2020]. These capabilities naturally extend to a wide range of domains

*Corresponding authors.

Authors' Contact Information: Daheng Yin, Simon Fraser University, Burnaby, Canada and Jiangxing Intelligence Inc., Shenzhen, China, dya64@sfu.ca; Yili Jin, McGill University, Montreal, Canada and Simon Fraser University, Burnaby, Canada, yili.jin@mail.mcgill.ca; Jianxin Shi, Nankai University, Tianjin, China and Simon Fraser University, Burnaby, Canada, jxshi@nankai.edu.cn; Isaac Ding, Simon Fraser University, Burnaby, Canada, isaac_ding@sfu.ca; Miao Zhang, Simon Fraser University, Burnaby, Canada, mza94@sfu.ca; Fangxin Wang, The Chinese University of Hong Kong, Shenzhen, Shenzhen, China, wangfangxin@cuhk.edu.cn; Zhaowu Huang, Fuzhou University, Fuzhou, China and Southeast University, Nanjing, China, zwh@fzu.edu.cn; Cong Zhang, Simon Fraser University, Burnaby, Canada and Jiangxing Intelligence Inc., Shenzhen, China, congz@ieee.org; Jiangchuan Liu, Simon Fraser University, Burnaby, Canada, jcliu@cs.sfu.ca; Fang Dong, Southeast University, Nanjing, China, fdong@seu.edu.cn.



SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811058

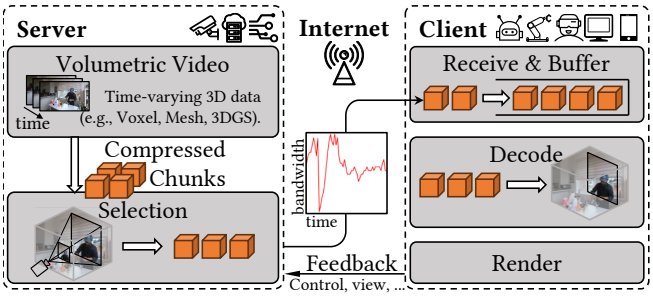


Fig. 1. Overview of a typical adaptive VV streaming pipeline. The server maintains time-varying 3D data, selects an appropriate quality-size version of each scene tile according to viewport and network feedback, and streams it to the client for decoding and rendering.

where remote perception, coordination, and action are critical, from healthcare [Gasques et al. 2021] and medical training [Rojas-Muñoz et al. 2019] to digital twins, robotics, autonomous systems [Tao et al. 2018; Zimmer et al. 2024], and environmental or industrial monitoring [de Koning et al. 2023; Hazeleger et al. 2024]. In this sense, VV streaming expands the Internet from a medium for transmitting images, videos, and messages into an infrastructure for delivering dynamic, interactive representations of the physical world.

To support such applications, VV streaming systems must deliver interactive 6DoF content with stable geometry for navigation, accurate visuals for interpretation, low latency for responsiveness, and consistent performance across varying network bandwidths. Adaptive VV streaming systems address these requirements by encoding scene tiles at multiple quality-size levels and dynamically selecting optimal versions based on viewport feedback, user interactions, or network status (Figure 1) [Zhang et al. 2022].

As a spatially decomposable and rasterization-friendly technique, 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] is widely used for real-time photorealistic 3D rendering. It represents scenes using semi-transparent ellipsoids ("Gaussians") with Spherical Harmonics (SH) for view-dependent photorealism, while enabling real-time rendering via rasterization. Despite these advantages, Gaussians carry complex attributes that produce large data volumes, posing significant challenges for streaming applications [Huang et al. 2026; Kim et al. 2025; Sun et al. 2025; Zhu et al. 2025]. Data reduction typically follows two orthogonal strategies: decreasing the number of Gaussians or compressing their attributes.

Existing systems typically support network adaptation through *density-based Level of Detail (LoD)* [Liu et al. 2023; Shi et al. 2025], which provides progressive scene representations with different numbers of primitives. Since each Gaussian captures fine-grained details within a volumetric region, removing Gaussians can lead to insufficient scene coverage and introduce structural gaps (Figure 2a).

Alternatively, attribute compression reduces data by compressing Gaussian attributes. A representative method is Vector Quantization (VQ), which maps high-dimensional Gaussian attributes to compact codebook indices and enables fast index-based decoding [Lee et al. 2024]. Nonetheless, existing attribute compression techniques primarily focus on static compression and lack the quality scalability required for adaptive VV streaming under fluctuating bandwidth.

(a) LoD 0 (~21 MB) (b) 40-bit VQ (~14 MB) (c) 100-bit VQ (~34 MB)

Fig. 2. Visual comparison of density-based LoD used in LTS [Sun et al. 2025] and attribute compression by Vector Quantization (VQ) (both compressed further by Draco [Google 2017]). Removing Gaussians causes structural gaps at low bitrates (a), while attribute compression better preserves structure with color distortion (b). Scenes are selected from the N3DV [Li et al. 2022b], Robo360 [Liang et al. 2023], AcinoSet [Joska et al. 2021] and MVSPS [Hein et al. 2025] datasets.

In this paper, we explore *scalable attribute compression* for adaptive, photorealistic VV streaming based on Gaussian representations. Our preliminary studies (Sec. 3) highlight that attribute compression largely avoids structural gaps but introduces color distortion as the dominant visual artifact. Accurate color representation is critical in streamed 3D scenes, as colors indicate object types and materials in robotics, liquid states and interactions in human contexts, tissue and instrument conditions in clinical settings, and species or environmental specifics in ecological observations (Figure 2b). Distorted color streams may obscure visual information required for accurate human interpretation and effective machine perception.

Encouragingly, our preliminary studies further reveal that such *color distortion caused by VQ can be effectively corrected in rendered images using a low-resolution reference image with accurate colors*. Motivated by this finding, we advocate a **Color Adaptation scheme** that *scalably compresses Gaussian attributes by VQ to establish LoDs, and restores colors using low-resolution reference images*. In streaming systems, the server renders these reference images from the high-fidelity Gaussians and streams them to clients to enable color restoration.

Implementing this concept in practical streaming systems introduces three challenges. First, adaptive streaming requires scalable

VQ rather than traditional flat VQ. We therefore design **Scalable Vector Quantization (SVQ)** that organizes Gaussian attributes into a base layer and multiple enhancement layers, establishing LoDs with different levels of quantization error. Second, latency constraints require the streaming server to predict the client's viewport and render reference images in advance. Prediction errors inevitably result in mismatches between reference images and actual client viewports, severely degrading color restoration quality. We address this with **Post-Render Perspective Alignment (PRPA)**, which realigns reference images on the client side using locally rendered depth. Third, prediction errors may leave visible regions uncovered by reference images, creating gaps during PRPA. We introduce an **Adaptive Field of View** strategy that uses server-side LSTM-based prediction to dynamically adjust the reference field of view (FoV), balancing visible-region coverage and reference quality.

Based on these designs, we introduce **Color-Adaptive Gaussian Streaming (CAGS)**, an adaptive VV streaming system compatible with Gaussian representations organized as differential volumetric frames (Sec. 4). This design facilitates integration with diverse Gaussian representations and can benefit from ongoing advances in 3D/4D Gaussian compression. The contributions of this paper are:

- We analyze the interaction between VQ and color restoration, revealing an opportunity to build adaptive VV streaming around a novel Color Adaptation scheme.
- We identify key system challenges of bringing Color Adaptation into adaptive VV streaming, and address them with SVQ, PRPA, and Adaptive FoV.
- We implement and evaluate a CAGS prototype on volumetric video datasets and real-world network traces, demonstrating 5∼20 dB PSNR gains over existing LoD methods and broad generalizability across Gaussian representations.

2 Related Work

2.1 3D Gaussian Splatting for Volumetric Video Streaming

As the most popular representation for VV streaming systems [Guan et al. 2023; Liu et al. 2023; Wang et al. 2024b; Zhang et al. 2024], point clouds require high-density data and significant bandwidth (300–800 Mbps) to ensure good visual quality [Han et al. 2020] due to their discrete nature [Han et al. 2020] and lack view-dependent effects that are critical for photorealism (e.g., specular highlights) [Wegen et al. 2024]. While Neural Radiance Fields (NeRF) [Mildenhall et al. 2021] offer superior fidelity with view-dependent effects, they suffer from severe rendering latency [Shi et al. 2024; Yin et al. 2024]. Alternatively, 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] enables real-time photorealistic rendering [Lee et al. 2024] and has been extended to dynamic scenes [Li et al. 2024; Luiten et al. 2024; Yan et al. 2024b]. Despite its advantages, the large data footprint of high-dimensional Gaussian attributes [Papantonakis et al. 2024] poses significant challenges for bandwidth-constrained streaming.

Vector Quantization (VQ) has become popular in recent Gaussian compression and streaming systems [Girish et al. 2024; Li et al. 2025; Wang et al. 2024a], enhancing compression efficiency through advanced codebook designs [Wang et al. 2024a; Xu et al. 2024a] and clustering techniques [Xie et al. 2025; Xu et al. 2024b]. Critically, most existing VQ-based methods prioritize compression efficiency for storage, overlooking the scalability indispensable for adaptive streaming [Guan et al. 2023; Han et al. 2020]. While other approaches explore scalable compression for 3DGS [Chen et al. 2025b; Liu et al. 2024], they typically involve complex structures that introduce high decoding latency (Table 1), thus hindering real-time streaming capabilities. To address this gap, we propose a scalable VQ framework that retains the fast decoding speed of VQ while enabling bitrate adaptation for robust VV streaming.

2.2 Level of Detail for Adaptive Streaming

Level of Detail (LoD) has long been used to reduce rendering cost [Cui et al. 2024; Kerbl et al. 2024; Yan et al. 2024a]. In adaptive streaming, LoD also facilitates bitrate adaptation by allowing clients to switch between quality levels according to real-time network conditions. To maximize visual quality without causing playback stalls, existing adaptive VV streaming systems typically combine visibility-aware transmission [Hladky et al. 2019; Zhu et al. 2025] with LoD selection [Han et al. 2020; Liu et al. 2023; Sun et al. 2025], prioritizing content that is visible or likely to become visible. For Gaussian-based representations, most LoD designs are density-based. Representative approaches include progressive training [Kerbl et al. 2024; Lu et al. 2024; Shi et al. 2025] and importance evaluation based on geometric properties [Fan et al. 2024; Papantonakis et al. 2024]. However, since each Gaussian occupies considerable space, low-density layers lack sufficient Gaussians to preserve fine details, resulting in significant quality degradation at lower LoDs. Alternatively, recent scalable neural codecs [Chen et al. 2025a; Liu et al. 2024] offer LoD-like quality-size scalability, but their complex decoding pipelines incur substantial latency (Table 1), limiting their suitability for real-time interaction. Other latency-resilient systems [Hladky et al. 2022; Lu and Rowe 2025] approximate views using geometry proxies, relying on server-side rendering, which increases computational load proportionally to client display resolution. In contrast, our method keeps Gaussian rendering on the client and uses only a fixed low-resolution reference image for color restoration, decoupling server-side rendering cost from the final display resolution.

2.3 Image Restoration

Learning-based image restoration has been widely studied for tasks such as super-resolution [Luo et al. 2022] and color restoration [Bozic et al. 2024]. While related techniques have been combined with neural scene representations to accelerate rendering [Huang et al. 2023; Wang et al. 2022], their potential for mitigating compression artifacts in 3DGS-based VV streaming remains largely unexplored. In this paper, we explore example-based color restoration [Cong et al. 2024; Zhang et al. 2019] to address the color distortion caused by VQ on Gaussians for VV streaming systems.

3 Measurement and Motivation

To motivate our proposed Color Adaptation scheme, we conduct preliminary experiments to evaluate the effectiveness of example-based color restoration in addressing color distortions introduced by attribute compression.

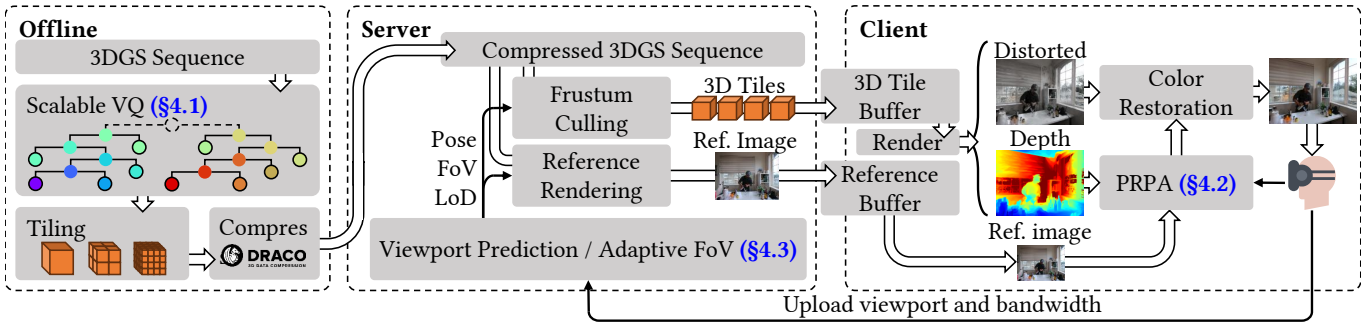


Fig. 3. Overview of CAGS. The server predicts the viewport, selects tiles and LoDs, renders a low-resolution reference image from the highest-quality layer of the compressed 3DGS, and streams it with Gaussian data. The client renders the tiles, aligns the reference image using PRPA, and restores colors for display.

3.1 Experimental Setup

We select three videos from the N3DV dataset [Li et al. 2022b] and train a 3DGS scene on the first frame of each video. We render each scene from interpolated training viewports at two resolutions: 1600×1200 as ground truth and 400×300 as reference images. We compress each 3DGS scene using KMeans VQ at five quality levels, followed by lossless compression with Draco [Google 2017]. After decompression and dequantization, we render the dequantized scenes from the same viewports to obtain color-distorted images. We adapt a lightweight SRResNet [Ledig et al. 2017] for restoration by modifying its input layer to support three settings: 1) color restoration from the distorted image, 2) super-resolution from the reference image, and 3) example-based restoration from both images.

3.2 Measurement Insights

Figure 4 shows that example-based color restoration consistently achieves better visual quality than single-image color restoration and super-resolution. The results indicate that aggressive VQ mainly damages color while preserving much of the scene structure. Hence, example-based color restoration leverages preserved structural details to achieve more accurate and visually appealing results. Moreover, when severe distortion makes the distorted image less trustworthy, example-based restoration relies more on the reference image and resembles super-resolution results. Thus, super-resolution defines the lower bound of example-based restoration. These findings highlight the potential benefit of integrating VQ and example-based color restoration into adaptive VV streaming systems.

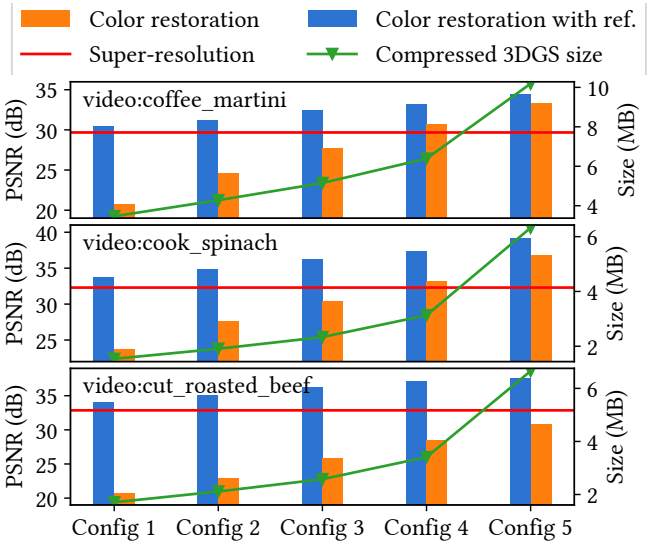


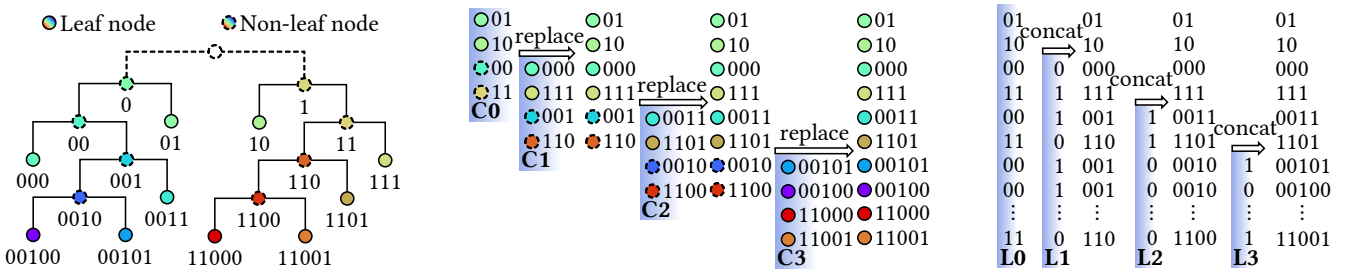
Fig. 4. Comparison of compressed frame size and PSNR for super-resolution, single-image color restoration, and example-based color restoration. Configs 1–5 correspond to KMeans VQ settings where scales are quantized to 8/10/12/14/16 bits; rotation and SH are quantized to 4/7/10/13/16 bits; and opacity is fixed at 4 bits.

4 System Design

Figure 3 provides an overview of the CAGS pipeline. CAGS is compatible with Gaussian representations organized as differential frames, where each frame stores only the Gaussians that differ from the previous frame. In the offline phase, CAGS establishes LoDs by **Scalable Vector Quantization** (Sec. 4.1). Each quantized frame is then tiled and compressed with Draco [Google 2017]. In the online streaming phase, the server predicts the client viewport from the uploaded viewport history and determines an **Adaptive Field of View** (Sec. 4.3). Based on the available bandwidth and predicted viewport, the server then renders low-resolution reference images and selects suitable tiles and LoDs. Low-resolution reference images are encoded as a video stream and streamed together with the selected tiles. For each frame, the client renders the received tiles to obtain a color-distorted image and a depth map, aligns the reference image using **Post-Render Perspective Alignment** (Sec. 4.2), and finally applies a lightweight network to restore colors.

4.1 Scalable Vector Quantization

While KMeans VQ effectively compresses Gaussian attributes [Fan et al. 2024; Lee et al. 2024], its flat clustering structure lacks the scalability required for adaptive streaming. Hierarchical vector quantization has been studied extensively for organizing codebooks across quality levels [Gersho and Shoham 1984]. A representative method is Agglomerative Hierarchical Clustering (AHC), which builds a cluster tree by repeatedly merging the closest clusters. Unfortunately, AHC involves computing pairwise distances in each iteration, which is computationally prohibitive for Gaussian representations that typically contain >100k Gaussians per frame. To bridge this gap, we propose Scalable Vector Quantization (SVQ) that integrates the efficiency of KMeans clustering with the hierarchical structure of AHC, with an index assignment strategy to build a scalable codebook.



(a) Cluster Tree (b) Structure of Codebook (c) Structure of Quantized Data

Fig. 5. Illustration of Scalable Vector Quantization (L0: the base layer quantized data; L1–L3: enhancement layers quantized data; C0–C3: corresponding codebook layers; colors indicate cluster centers; each row in (c) represents a Gaussian).

4.1.1 Hierarchical Codebook Construction. SVQ first runs KMeans on a random subset of Gaussians, then iteratively merges these clusters into a binary tree, similar to AHC. To preserve fidelity, we define a cluster distance metric based on the quantization error introduced by merging two clusters $d(C_1, C_2)$:

$$d(C_1, C_2) = \text{mean}(\{c - \text{mean}(C_1 \cup C_2) | c \in C_1 \cup C_2\})$$

At each iteration, SVQ merges the pair of clusters with the smallest distance, progressively forming a binary tree (Figure 5a), until the number of clusters matches the target codebook size. By starting from KMeans clusters rather than individual Gaussians, SVQ significantly reduces computational complexity compared to AHC.

4.1.2 Scalable Indexing. After building the tree, SVQ assigns indices to clusters to create a scalable codebook. Inspired by Huffman coding, indices are assigned based on cluster positions in the binary tree (Figure 5a). This design provides inherent scalability: truncating lower-order bits naturally maps to its parent cluster at a coarser LoD (Figure 5b). Consequently, the codebook is organized into a base layer to store higher-order index bits and subsequent enhancement layers to store lower-order bits (Figure 5b).

4.1.3 Dequantization. Dequantization involves only concatenating the received bits (Figure 5c) and indexing into the codebook (Figure 5b), and is therefore computationally efficient without becoming a performance bottleneck (Table 2).

4.2 Post-Render Perspective Alignment

To support real-time color restoration at high resolutions, CAGS relies on lightweight models such as SRResNet [Ledig et al. 2017]. Our evaluation shows that restoration quality depends on accurate alignment between the reference and distorted images (Sec. 5.4).

Unfortunately, misalignment is inevitable in server-side rendering. Due to inherent latency on the Internet (typically 40–90 ms), uploading the client viewport and waiting for the corresponding reference image for every frame would exceed the latency budget for interactive VV streaming (33.3 ms per frame at 30 FPS) [Gül et al. 2022]. Han et al. [Han et al. 2020] have shown that practical systems must predict future viewports to satisfy this constraint. Accordingly, the CAGS server renders low-resolution reference images in advance from the highest-quality level of the compressed Gaussian sequence. The prediction errors inevitably introduce misalignment between the reference and distorted images.

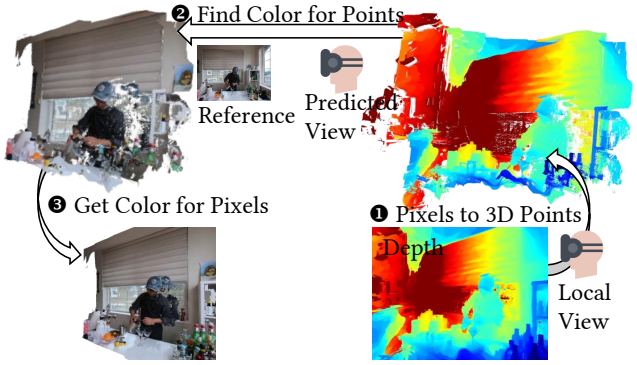


Fig. 6. PRPA aligns the reference image in three steps: ❶ unprojects client-view pixels using the depth map, ❷ reprojects them into the reference view, and ❸ samples the corresponding colors to produce an aligned image.

To mitigate this issue, we propose Post-Render Perspective Alignment (PRPA). PRPA takes the server-rendered reference image, its rendering viewport, and the client-side depth map produced by the 3DGS rasterizer. It aligns the reference image to the actual client viewport before color restoration.

4.2.1 Reference Image Alignment. As shown in Figure 6, PRPA reprojects pixels from the client-side depth map to the server-rendered reference and samples the corresponding colors to produce an aligned image. Note that PRPA fundamentally differs from Depth Image Based Rendering (DIBR) [Fehn 2004; Wu et al. 2023]. DIBR warps a reference image using its *own* depth to a target view. In contrast, PRPA aligns the reference image using the target depth.

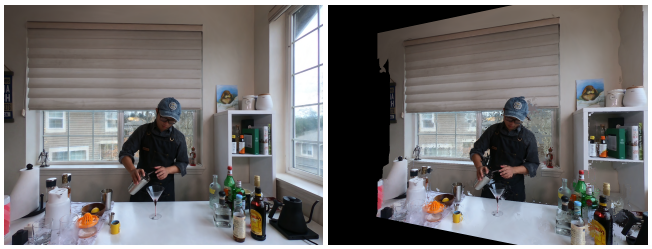
4.2.2 Error Erosion on Occluded Regions. Naive alignment may map pixels to occluded regions in the reference view, causing visual artifacts (Figure 7b). PRPA reduces these artifacts through error erosion. During reprojection into the reference view, PRPA records the projected depth of each pixel. When multiple pixels map to the same reference pixel, only the pixel with the smallest depth is treated as visible, while the others are marked as occluded. PRPA then iteratively replaces each occluded pixel with the average color of its non-occluded neighbors, yielding a cleaner aligned reference image for color restoration (Figure 7e).

Implemented with optimized GPU matrix operations, PRPA introduces negligible overhead that does not bottleneck real-time applications (Table 2). Pseudocode is provided in supplementary.

(a) Reference image (b) Aligned reference image

(c) Before erosion (d) Erosion Step 3 (e) Erosion Step 30

Fig. 7. Illustration of alignment errors and error erosion in PRPA. (b) Aligned reference image before error erosion, with red circles highlighting occlusion artifacts. (c)-(e) Iterative error erosion reduces these artifacts.



(a) Small FoV ref. image (b) Aligned small FoV ref. image

(c) Large FoV ref. image (d) Aligned large FoV ref. image

Fig. 8. Visualization of PRPA results with different reference FoVs. A small FoV (a) misses boundary content (b), whereas a large FoV (c) preserves coverage but reduces pixel density in the aligned reference image (d).

4.3 Adaptive Field of View

Viewport prediction errors may cause the reference image to cover only part of the client viewport, leaving missing regions after PRPA (Figure 8b) and degrading restoration quality. While enlarging the reference FoV improves coverage (Figure 8c), it reduces pixel density in target regions (Figure 8d), which also hurts quality.

To balance coverage and density, we propose an Adaptive FoV strategy driven by a lightweight LSTM model. We choose LSTM because it is efficient and effective for sequential, latency-sensitive prediction. Though complex backbones like Transformers could achieve similar accuracy, they usually require additional optimization for real-time use. Notably, our strategy is decoupled from the viewport prediction, providing a plug-and-play solution compatible with various prediction methods [Han et al. 2020; Liu et al. 2023]. In our evaluation, we use autoregression for viewport prediction.

4.3.1 LSTM-based Adaptive FoV Model. The LSTM predicts a scaling factor $s_i = (s_i^x, s_i^y)$ for frame i, which adjusts the reference FoV (F^x, F^y) relative to the fixed client viewport FoV (F_0^x, F_0^y), such that $F^x = (1 + s_i^x)F_0^x$ and $F^y = (1 + s_i^y)F_0^y$. The model takes the client viewport (rotation q_i and position p_i), their temporal changes, and historical states as inputs. Formally, the predicted scaling factor $\hat{s}_i$ and hidden state h_i are computed as:

$$(\hat{s}_i, h_i) = \text{LSTM}(q_i, q_i - q_{i-1}, p_i, p_i - p_{i-1}, s_{i-1}^a, h_{i-1})$$

where s_{i-1}^a denotes the approximate ground-truth FoV scaling factor from the previous frame (Sec. 4.3.2).

In a real streaming system, viewport predictions typically span multiple future frames [Liu et al. 2023], where actual client viewports are unavailable. In this case, the model uses predicted viewports as q_i and p_i, and uses the previously predicted FoV scaling factor $\hat{s}_{i-1}$ as s_{i-1}^a. To limit error accumulation, we refresh the LSTM hidden state whenever actual client viewport updates arrive.

4.3.2 Fast Approximate Ground-truth FoV. Computing the exact ground-truth FoV requires projecting all pixels from the reference image into the client viewport and then finding the smallest FoV that covers them, which is computationally prohibitive. Instead, we approximate the ground-truth FoV by projecting only the four corner pixels of the reference image into the client viewport at a fixed depth (e.g., 10 m). Given that viewport prediction errors are usually small over short intervals [Han et al. 2020], this approximation can greatly reduce computation while maintaining sufficient accuracy. Since the server cannot access the actual client viewport in real-time, this approximate FoV is only used for offline supervision and to update s_i^a once client viewport updates arrive. The FoV for reference rendering is predicted by the Adaptive FoV model.

4.3.3 LSTM Model Training. We train the LSTM model offline using collected viewport datasets. First, we run the viewport prediction on ground-truth viewports to obtain predicted viewports. Then we compute the ground-truth FoV scaling factors s_i and their approximations s_i^a using the method above. The training loss $\mathcal{L}$ is computed between $\hat{s}_i$ and s_i over all N frames:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} |\hat{s}_i - s_i|$$

5 Evaluation

5.1 Prototype Implementation

We implemented a CAGS prototype to evaluate its effectiveness, following prior VV streaming practice [Sun et al. 2025; Wang et al. 2024a]. We prepare volumetric videos using TrackerSplat [Yin et al. 2025] and apply importance-based pruning [Papantonakis et al. 2024; Zhou et al. 2024] to remove redundant Gaussians. To enable adaptive streaming, we build a linear LoD hierarchy by interleaving

SVQ layers according to the empirically measured visual impact of Gaussian attributes, prioritizing high-impact layers at lower LoDs. We tile Gaussians via Morton sorting [Jiang et al. 2025] to balance the number of Gaussians per tile. Spatial data and base LoD are compressed with Draco [Google 2017], while enhancement layers and the codebook are compressed with Gzip. During streaming, the server applies a bandwidth-aware adaptation strategy that prioritizes tiles covering visible Gaussians (identified via reference rendering) and progressively raises their LoDs until reaching the bandwidth limit. Our prototype is deployed on consumer-grade GPUs (RTX 3080) for both server and client, streaming at 30 FPS while rendering at 60 FPS on the client side. Additional engineering details and underlying rationales are provided in the supplementary.

5.2 Evaluation Setup

5.2.1 Dataset. We evaluate CAGS on four datasets with diverse motion patterns, scene scales, and capture resolutions: Neural 3D Video Synthesis (N3DV) [Li et al. 2022b], ST-NeRF [Zhang et al. 2021], Meeting Room [Li et al. 2022a], and Dynamic 3DGS [Luiten et al. 2024]. We adapt UnityGS [Pranckevičius 2023] and develop an application to record viewport traces using a Meta Quest 3. The traces used to train color restoration and FoV prediction are collected independently from those used for evaluation.

5.2.2 Metrics. We render the original uncompressed Gaussians at the corresponding viewports as ground truth. We report PSNR, SSIM, and LPIPS [Zhang et al. 2018].

5.2.3 Network Settings. We evaluate both fixed and dynamic bandwidth settings. For fixed bandwidth, we set the bandwidth limit (Sec. 5.1) to 30, 60, 90, 120, and 150 Mbps. For dynamic bandwidth, we use a representative segment from 5Gophers [Narayanan et al. 2020], as shown in Figure 9.

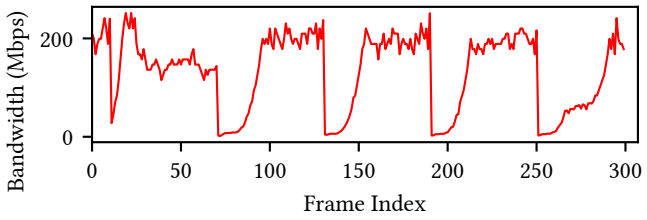


Fig. 9. Throughput of the selected network trace from the 5Gophers dataset (lines 3249 to 3549, 1.47–251.7 Mbps, average 133.7 Mbps).

5.2.4 Baselines. We compare CAGS with two state-of-the-art Gaussian-based VV streaming systems and two restoration variants, specifically focusing on methods that are compatible with the real-time client loop required for interactive streaming.

LTS-F: LTS [Sun et al. 2025] is an adaptive VV streaming system based on density-based LoD. We use its released LoD component and implement its corresponding network adaptation strategy as the baseline LTS-F. For consistency with Sec. 5.1, we encode the sequence into independently decodable Groups of Frames (GoF) and select the LoD for each frame during streaming.

V^3-A: V^3 [Wang et al. 2024a] is a VV streaming system based on VQ (hash grids trained with entropy loss). It does not support LoD and thus lacks a built-in network adaptation strategy. For the V^3-A baseline, we integrate its released code as a compression module into our pipeline (including PRPA and adaptive FoV for fairness), and implement network adaptation in Sec. 5.1, except that already-transmitted tiles are resent at higher quality until the bandwidth limit is reached.

SR align ref.: Uses SRResNet [Ledig et al. 2017] for super-resolution on the reference image aligned by PRPA.

CR w/o ref.: Performs color restoration (Sec. 3.1) directly on the distorted image without a reference.

5.3 Evaluation Results

5.3.1 Evaluation of Scalable Vector Quantization. Before evaluating the full system, we benchmark SVQ against state-of-the-art scalable compression methods: SPZ [Niantic Labs 2025], CompGS [Liu et al. 2024], HAC [Chen et al. 2025a], and HAC++ [Chen et al. 2025b]. As shown in Table 1, SVQ achieves comparable rate-distortion performance to these methods and outperforms them in decoding latency, which is uniquely suited for real-time streaming.

5.3.2 Evaluation under Fixed Bandwidth. Figure 10 reports the average PSNR and SSIM under fixed bandwidth constraints. CAGS achieves higher PSNR than the baselines in most cases, indicating better color fidelity. LTS-F is limited by density-based LoD, which provides a weaker rate–quality trade-off and therefore selects lower-quality LoDs under the same bandwidth. V^3-A lacks scalability and causes redundant transmissions, resulting in lower quality.

In some cases such as the "walking" (Figure 10c), LTS-F achieves higher SSIM at high bandwidth. Analysis shows this sequence contains fewer Gaussians, allowing density-based LoD to approach uncompressed quality when bandwidth is sufficient, while CAGS and V^3-A remain bounded by lossy VQ compression.

5.3.3 Evaluation under Fluctuating Bandwidth. Figure 11 reports per-frame PSNR/SSIM under the fluctuating 5G trace. CAGS achieves higher and more stable quality over time, demonstrating robustness to bandwidth variations. Similar to fixed bandwidth results, the "walking" sequence favors the baseline due to its fewer Gaussians.

Figure 11 also reveals occasional quality drops (e.g., frame 32 of "basketball" and frame 240 of "discussion"). These drops are caused by rapid head movements, which increase viewport prediction errors and lead to either large missing regions in the PRPA output or overly large FoV predictions. While noticeable in measurements, these degradations have minimal perceptual impact because: 1) they are short-lived, since humans cannot move rapidly for long, and once movement stabilizes, prediction errors drop and quality quickly recovers; and 2) rapid movements naturally limit human visual perception, making such temporary degradations less noticeable.

5.3.4 Visualization Results. Figure 12 shows a representative example of color restoration. The color distortion in Figure 12a is effectively corrected in Figure 12c, demonstrating the effectiveness of our restoration method.

5.3.5 System Performance. We profile key components on both server and client. Table 2 summarizes the profiling results, demonstrating that CAGS supports real-time streaming and rendering.

Table 1. Performance and quality comparison of our SVQ method (with 66-bit and 76-bit initialization) against state-of-the-art **scalable** compression methods at the highest quality level. Best and second-best results are highlighted in **bold** and underline, respectively. Size is in MB and decoding time ("dec.t") in seconds. Full results including SSIM and LPIPS are provided in the supplementary material.

Datasets	**Neural3DV**			**Meet.Room**			**Dyn.3DGS**		
Methods	psnr	size	dec.t	psnr	size	dec.t	psnr	size	dec.t
SPZ low	23.3	3.33	0.076	21.8	2.54	0.062	21.1	2.26	0.052
SPZ high	23.1	4.53	0.075	21.8	3.41	0.057	21.0	3.32	0.057
CompGS	24.9	22.20	6.213	25.0	5.58	0.799	19.1	4.27	2.772
HAC	22.2	25.69	24.407	25.2	8.69	9.135	20.5	2.84	1.571
HAC++	21.5	17.98	51.708	25.2	6.78	14.706	20.9	**2.00**	6.169
SVQ 66bit	24.5	**2.00**	**0.015**	26.0	**1.53**	**0.018**	**25.3**	2.25	**0.028**
SVQ 76bit	**25.1**	2.14	0.015	**26.4**	1.69	0.019	23.3	2.42	0.029

5.4 Ablation Studies

We conduct two ablation studies to quantify the contributions of key components: 1) **w/o PRPA**: feeds the misaligned reference image directly to restoration. 2) **w/o Adaptive FoV**: fixes the reference rendering FoV to 10% larger than the client viewport.

Figures 10 and 11 show that removing PRPA leads to a significant quality drop, confirming that accurate reference–target alignment is essential for effective restoration. Adaptive FoV also provides meaningful improvements, indicating that sufficient boundary coverage is important for reducing the impact of viewport prediction errors. These components effectively handle viewport prediction errors: PRPA aligns reference images rendered from predicted viewpoints with the actual viewport, enabling accurate color restoration. Adaptive FoV expands the reference FoV according to head motion, ensuring complete coverage. Together, they ensure the quality stability observed in Sec. 5.3.3.

5.5 Generality to Other Representations

CAGS is designed for Gaussian representations where each frame stores the Gaussians that differ from the previous frame. To validate its generality, we integrate CAGS with three representative methods for preparing volumetric videos: Dynamic 3DGS [Luiten et al. 2024], 4DGS [Wu et al. 2024] and HiCoM [Gao et al. 2024], and evaluate them under the same bandwidth trace using the baselines (SR align ref., and CR w/o ref.) and ablations (w/o PRPA, w/o Adaptive FoV).

Figures 13 and 14 show that CAGS consistently improves quality across different preparation methods. These results suggest that CAGS can extend beyond the tested methods, potentially supporting a wide range of 3D/4D Gaussian representations and benefiting from ongoing evolutions in 3D/4D Gaussian compression.

6 Limitation and Future Work

CAGS has certain limitations and leaves room for improvement.

Error erosion reduces color distortion in occluded regions but still leaves artifacts in the PRPA output. To ensure real-time performance, we did not complicate our restoration model to specifically handle these artifacts. Although our lightweight restoration model can learn to suppress many of them during training, small artifacts may still remain (can be observed in our provided video results). Future work can explore artifact handling without sacrificing speed.

Table 2. Performance of system components.

	Component	**Time**
Encoding (Offline)	SVQ Codebook (per-video)	36.8 s
	SVQ & Draco Encoding	370 ms
Server	Viewport Prediction	1 ms
	Dynamic FoV	1 ms
	Rendering (400x300)	1.7 ms
Client Decoding	Draco Decoding	9 ms
	SVQ Decoding	2.58 ms
Client Rendering (1600x1200)	Render Distorted+Depth	9.32 ms
	PRPA	2.11 ms
	Color Restoration	6.5 ms
Client Rendering (1600x1200, RTX 4090)	Render Distorted+Depth	1.21 ms
	PRPA	1 ms
	Color Restoration	3.3 ms

Although our VQ design is developed for Gaussian representations, VQ itself is more general and can also be applied to other 3D representations. Prior works have shown its effectiveness for NeRFs [Zhong et al. 2024] and also report color distortion as a side effect [Takikawa et al. 2022], suggesting that such distortion may be a common issue inherent to VQ across different 3D representations. This highlights the broader potential of the Color Adaptation scheme beyond Gaussian-based streaming: it could serve as a general solution for streaming 3D content across diverse scene representations. Future work can explore integrating the color-adaptive scheme with a wider range of 3D representations.

7 Conclusion

In this paper, we present a Color Adaptation scheme for volumetric video streaming based on dynamic 3D Gaussian Splatting. We first identify limitations of existing LoD methods for Gaussian representations. We reveal that vector quantization primarily causes color distortion, which can be effectively corrected using reference images. We also identify the challenges of implementing Color Adaptation in real systems and address them with the CAGS system. Extensive evaluation of our prototype demonstrates the effectiveness and efficiency of our design.

References

Vukasin Bozic, Abdelaziz Djelouah, Yang Zhang, Radu Timofte, Markus Gross, and Christopher Schroers. 2024. Versatile Vision Foundation Model for Image and Video Colorization. In *ACM SIGGRAPH 2024 Conference Papers (SIGGRAPH '24)*. 1–11.

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. 2025a. HAC: Hash-Grid Assisted Context for 3D Gaussian Splatting Compression. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). 422–438.

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. 2025b. HAC++: Towards 100X Compression of 3D Gaussian Splatting. doi:10.48550/arXiv.2501.12255

Xiaoyan Cong, Yue Wu, Qifeng Chen, and Chenyang Lei. 2024. Automatic Controllable Colorization via Imagination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2609–2619.

Jiadi Cui, Junming Cao, Fuqiang Zhao, Zhipeng He, Yifan Chen, Yuhui Zhong, Lan Xu, Yujiao Shi, Yingliang Zhang, and Jingyi Yu. 2024. LetsGo: Large-Scale Garage Modeling and Rendering via LiDAR-Assisted Gaussian Primitives. In *SIGGRAPH Asia 2024 Conference Papers (SA '24)*.

Koen de Koning, Jeroen Broekhuijsen, Ingolf Kühn, Otso Ovaskainen, Franziska Taubert, Dag Endresen, Dmitry Schigel, and Volker Grimm. 2023. Digital twins: dynamic

model-data fusion for ecology. *Trends in ecology & evolution* 38, 10 (2023), 916–926.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. 2024. LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction and 200+ FPS. *Advances in Neural Information Processing Systems* 37 (2024), 140138–140158.

Christoph Fehn. 2004. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In *Stereoscopic Displays and Virtual Reality Systems XI*, Vol. 5291. International Society for Optics and Photonics, SPIE, 93 – 104.

Qiankun Gao, Jiarui Meng, Chengxiang Wen, Jie Chen, and Jian Zhang. 2024. HiCoM: Hierarchical Coherent Motion for Dynamic Streamable Scenes with 3D Gaussian Splatting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Danilo Gasques, Janet G Johnson, Tommy Sharkey, Yuanyuan Feng, Ru Wang, Zhuoqun Robin Xu, Enrique Zavala, Yifei Zhang, Wanze Xie, Xinming Zhang, et al. 2021. Artemis: A collaborative mixed-reality system for immersive surgical telementoring. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–14.

A. Gersho and Y. Shoham. 1984. Hierarchical vector quantization of speech with dynamic codebook allocation. In *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 9. 416–419.

Sharath Girish, Tianye Li, Amrita Mazumdar, Abhinav Shrivastava, David Luebke, and Shalini De Mello. 2024. QUEEN: QUantized Efficient ENcoding of Dynamic Gaussians for Streaming Free-viewpoint Videos. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Google. 2017. Draco 3D Graphics Compression. https://github.com/google/draco.

Yongjie Guan, Xueyu Hou, Nan Wu, Bo Han, and Tao Han. 2023. MetaStream: Live Volumetric Content Capture, Creation, Delivery, and Rendering in Real Time. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. Number 29. 1–15.

Serhan Gül, Cornelius Hellge, and Peter Eisert. 2022. Latency Compensation Through Image Warping For Remote Rendering-Based Volumetric Video Streaming. In *2022 IEEE International Conference on Image Processing (ICIP)*. 2026–2030.

Bo Han, Yu Liu, and Feng Qian. 2020. ViVo: Visibility-Aware Mobile Volumetric Video Streaming. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–13.

W Hazeleger, JPM Aerts, Peter Bauer, MFP Bierkens, Gustau Camps-Valls, MM Dekker, FJ Doblas-Reyes, Veronika Eyring, C Finkenauer, Arthur Grundner, et al. 2024. Digital twins of the Earth with and for humans. *Communications earth & environment* 5, 1 (2024), 463.

Jonas Hein, Nicola Cavalcanti, Daniel Suter, Lukas Zingg, Fabio Carrillo, Lilian Calvet, Mazda Farshad, Nassir Navab, Marc Pollefeys, and Philipp Fürnstahl. 2025. Next-generation surgical navigation: Marker-less multi-view 6DoF pose estimation of surgical instruments. *Medical Image Analysis* 103 (2025), 103613.

Jozef Hladky, Hans-Peter Seidel, and Markus Steinberger. 2019. The camera offset space: real-time potentially visible set computations for streaming rendering. *ACM Trans. Graph.* 38, 6 (2019), 14 pages.

Jozef Hladky, Michael Stengel, Nicholas Vining, Bernhard Kerbl, Hans-Peter Seidel, and Markus Steinberger. 2022. QuadStream: A Quad-Based Scene Streaming Architecture for Novel Viewpoint Reconstruction. *ACM Trans. Graph.* 41, 6 (2022), 13 pages.

Xudong Huang, Wei Li, Jie Hu, Hanting Chen, and Yunhe Wang. 2023. RefSR-NeRF: Towards High Fidelity and Super Resolution View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8244–8253.

Zhaohui Huang, Cong Zhang, Jianxin Shi, Xiaoyi Fan, Laizhong Cui, and Jiangchuan Liu. 2026. ACPGS: Towards Bandwidth-Efficient Delivery of 3D Gaussian Splatting. In *Proceedings of the 36th Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '26)*. 127–133.

Yuheng Jiang, Chengcheng Guo, Yize Wu, Yu Hong, Shengkun Zhu, Zhehao Shen, Yingliang Zhang, Shaohui Jiao, Zhuo Su, Lan Xu, Marc Habermann, and Christian Theobalt. 2025. Topology-Aware Optimization of Gaussian Primitives for Human-Centric Volumetric Videos. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*. 1–12.

Daniel Joska, Liam Clark, Naoya Muramatsu, Ricardo Jericevich, Fred Nicolls, Alexander Mathis, Mackenzie W. Mathis, and Amir Patel. 2021. AcinoSet: A 3D Pose Estimation Dataset and Baseline Models for Cheetahs in the Wild. arXiv:2103.13282 [cs.CV]

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023), 139:1–139:14.

Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. 2024. A Hierarchical 3D Gaussian Representation for Real-Time Rendering of Very Large Datasets. *ACM Transactions on Graphics* 44, 3 (2024).

Gunjoong Kim, Seonghoon Park, Jeho Lee, Chanyoung Jung, Hyungchol Jun, and Hojung Cha. 2025. Vega: Fully Immersive Mobile Volumetric Video Streaming with 3D Gaussian Splatting. In *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*. 1106–1120.

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4681–4690.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. 2024. Compact 3D Gaussian Representation for Radiance Field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21719–21728.

Hao Li, Sicheng Li, Xiang Gao, Abudouaihati Batuer, Lu Yu, and Yiyi Liao. 2025. GIFStream: 4D Gaussian-based Immersive Video with Feature Stream. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jiyang Li, Lechao Cheng, Zhangye Wang, Tingting Mu, and Jingxuan He. 2024. Loopy-Gaussian: Creating 3D Cinemagraph with Multi-view Images via Eulerian Motion Field. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*. 476–485.

Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. 2022a. Streaming Radiance Fields for 3D Video Synthesis. *Advances in Neural Information Processing Systems* 35 (2022), 13485–13498.

Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. 2022b. Neural 3D Video Synthesis From Multi-View Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5521–5531.

Litian Liang, Liuyu Bian, Caiwei Xiao, Jialin Zhang, Linghao Chen, Isabella Liu, Fanbo Xiang, Zhiao Huang, and Hao Su. 2023. Robo360: a 3D omnispective multi-material robotic manipulation dataset. *arXiv preprint arXiv:2312.06686* (2023).

Junhua Liu, Boxiang Zhu, Fangxin Wang, Yili Jin, Wenyi Zhang, Zihan Xu, and Shuguang Cui. 2023. CaV3: Cache-assisted Viewport Adaptive Volumetric Video Streaming. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. 173–183.

Xiangrui Liu, Xinju Wu, Pingping Zhang, Shiqi Wang, Zhu Li, and Sam Kwong. 2024. CompGS: Efficient 3D Scene Representation via Compressed Gaussian Splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*. 2936–2944.

Edward Lu and Anthony Rowe. 2025. QUASAR: Quad-based Adaptive Streaming And Rendering. *ACM Trans. Graph.* 44, 4 (2025), 18 pages.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. 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.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. 2024. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. In *3DV*.

Zhengxiong Luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. 2022. Learning the Degradation Distribution for Blind Image Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6063–6072.

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.

Arvind Narayanan, Eman Ramadan, Jason Carpenter, Qingxu Liu, Yu Liu, Feng Qian, and Zhi-Li Zhang. 2020. A First Look at Commercial 5G Performance on Smartphones. In *Proceedings of The Web Conference 2020 (WWW '20)*. 894–905.

Niantic Labs. 2025. spz: File Format for 3D Gaussian Splats. https://github.com/nianticlabs/spz.

Panagiotis Papantonakis, Georgios Kopanas, Bernhard Kerbl, Alexandre Lanvin, and George Drettakis. 2024. Reducing the Memory Footprint of 3D Gaussian Splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 7, 1 (2024), 16:1–16:17.

Aras Pranckevičius. 2023. Gaussian Splatting playground in Unity. https://github.com/aras-p/UnityGaussianSplatting.

Edgar Rojas-Muñoz, Maria Eugenia Cabrera, Daniel Andersen, Voicu Popescu, Sherri Marley, Brian Mullis, Ben Zarzaur, and Juan Wachs. 2019. Surgical telementoring without encumbrance: a comparative study of see-through augmented reality-based approaches. *Annals of surgery* 270, 2 (2019), 384–389.

Jianxin Shi, Miao Zhang, Linfeng Shen, Jiangchuan Liu, Yuan Zhang, Lingjun Pu, and Jingdong Xu. 2024. Towards Full-scene Volumetric Video Streaming via Spatially Layered Representation and NeRF Generation. In *Proceedings of the 34th Edition of the Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '24)*. 22–28.

Yuang Shi, Géraldine Morin, Simone Gasparini, and Wei Tsang Ooi. 2025. LapisGS: Layered Progressive 3D Gaussian Splatting for Adaptive Streaming. In *International Conference on 3D Vision 2025*.

Yuan-Chun Sun, Yuang Shi, Cheng-Tse Lee, Mufeng Zhu, Wei Tsang Ooi, Yao Liu, Chun-Ying Huang, and Cheng-Hsin Hsu. 2025. LTS: A DASH Streaming System for Dynamic Multi-Layer 3D Gaussian Splatting Scenes. In *Proceedings of the 16th ACM Multimedia Systems Conference (MMSys '25)*. 136–147.

Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. 2022. Variable Bitrate Neural Fields. In *Special Interest*

Group on Computer Graphics and Interactive Techniques Conference Proceedings. 1–9.

Fei Tao, He Zhang, Ang Liu, and Andrew YC Nee. 2018. Digital twin in industry: State-of-the-art. *IEEE Transactions on industrial informatics* 15, 4 (2018), 2405–2415.

Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. 2022. NeRF-SR: High Quality Neural Radiance Fields Using Supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*. 6445–6454.

Penghao Wang, Zhirui Zhang, Liao Wang, Kaixin Yao, Siyuan Xie, Jingyi Yu, Minye Wu, and Lan Xu. 2024a. V^3: Viewing Volumetric Videos on Mobiles via Streamable 2D Dynamic Gaussians. In *SIGGRAPH Asia 2024 Conference Papers*.

Yizong Wang, Dong Zhao, Huanhuan Zhang, Teng Gao, Zixuan Guo, Chenghao Huang, and Huadong Ma. 2024b. Bandwidth-Efficient Mobile Volumetric Video Streaming by Exploiting Inter-Frame Correlation. *IEEE Transactions on Mobile Computing* 23, 10 (2024), 9410–9423.

Ole Wegen, Willy Scheibel, Matthias Trapp, Rico Richter, and Jurgen Dollner. 2024. A Survey on Non-photorealistic Rendering Approaches for Point Cloud Visualization. *IEEE Transactions on Visualization and Computer Graphics* (2024), 1–20.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 2024. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20310–20320.

Jiangkai Wu, Yu Guan, Qi Mao, Yong Cui, Zongming Guo, and Xinggong Zhang. 2023. ZGaming: Zero-Latency 3D Cloud Gaming by Image Prediction. In *Proceedings of the ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23)*. 710–723.

Shuzhao Xie, Jiahang Liu, Weixiang Zhang, Shijia Ge, Sicheng Pan, Chen Tang, Yunpeng Bai, Cong Zhang, Xiaoyi Fan, and Zhi Wang. 2025. SizeGS: Size-aware Compression of 3D Gaussian Splatting via Mixed Integer Programming. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*. Association for Computing Machinery, New York, NY, USA, 8214–8223. doi:10.1145/3746027.3755370

Jiawei Xu, Zexin Fan, Jian Yang, and Jij Xie. 2024a. Grid4D: 4D Decomposed Hash Encoding for High-Fidelity Dynamic Gaussian Splatting. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NIPS '24, Vol. 37)*. 123787–123811.

Zhen Xu, Yinghao Xu, Zhiyuan Yu, Sida Peng, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. 2024b. Representing Long Volumetric Video with Temporal Gaussian Hierarchy. *ACM Trans. Graph.* 43, 6 (2024), 171:1–171:18.

Jinbo Yan, Rui Peng, Luyang Tang, and Ronggang Wang. 2024b. 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 (MM '24)*. 7871–7880.

Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. 2024a. Multi-Scale 3D Gaussian Splatting for Anti-Aliased Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20923–20931.

Daheng Yin, Isaac Ding, Yili Jin, Jianxin Shi, and Jiangchuan Liu. 2025. TrackerSplat: Exploiting Point Tracking for Fast and Robust Dynamic 3D Gaussians Reconstruction. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*. 1–11.

Daheng Yin, Jianxin Shi, Miao Zhang, Zhaowu Huang, Jiangchuan Liu, and Fang Dong. 2024. FSVFG: Towards Immersive Full-Scene Volumetric Video Streaming with Adaptive Feature Grid. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*. 11089–11098.

Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. 2022. YuZu: Neural-Enhanced Volumetric Video Streaming. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. 137–154.

Anlan Zhang, Chendong Wang, Yuming Hu, Ahmad Hassan, Zejun Zhang, Bo Han, Feng Qian, and Shichang Xu. 2024. Habitus: Boosting Mobile Immersive Content Delivery through Full-body Pose Tracking and Multipath Networking. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 1677–1695.

Bo Zhang, Mingming He, Jing Liao, Pedro V. Sander, Lu Yuan, Amine Bermak, and Dong Chen. 2019. Deep Exemplar-Based Video Colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8052–8061.

Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. 2021. Editable Free-Viewpoint Video Using a Layered Neural Representation. *ACM Transactions on Graphics* 40, 4 (2021), 149:1–149:18.

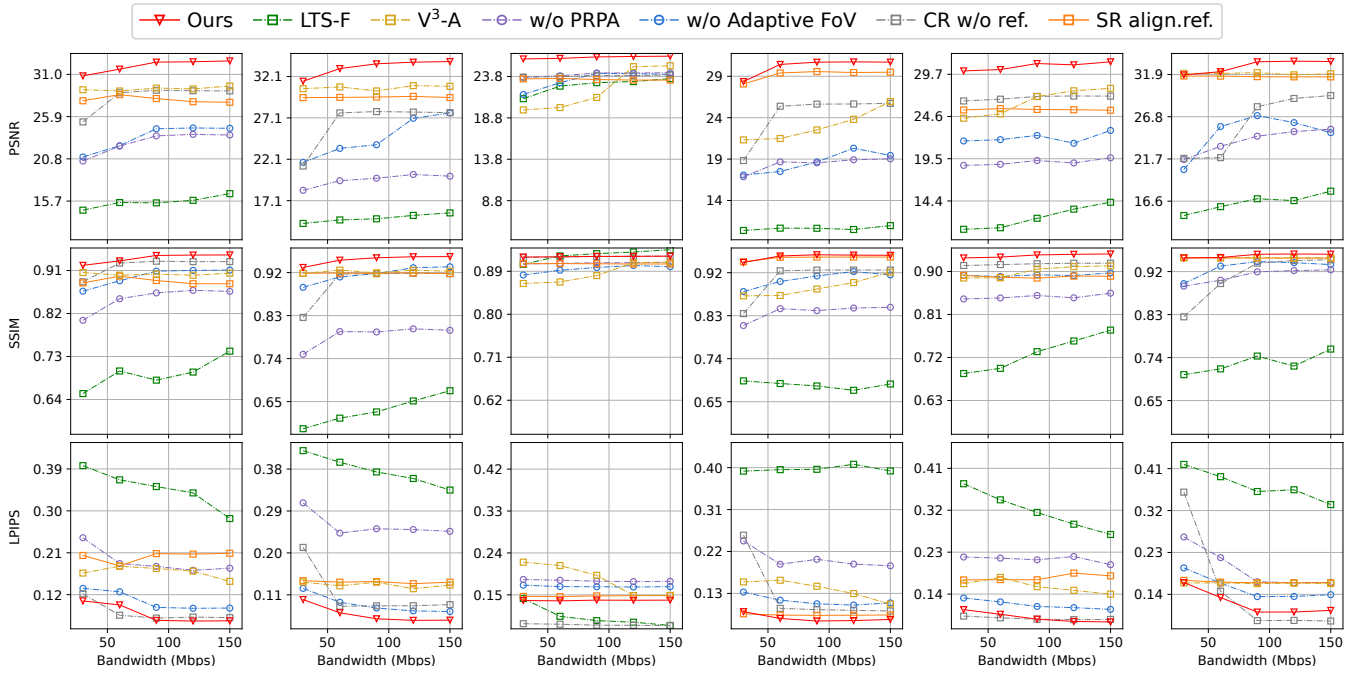
Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Hongliang Zhong, Jingbo Zhang, and Jing Liao. 2024. VQ-NeRF: Neural Reflectance Decomposition and Editing With Vector Quantization. *IEEE Transactions on Visualization and Computer Graphics* 30, 9 (2024), 6247–6260.

Zhi Zhou, Junke Zhu, and Zhangjin Huang. 2024. Gaussian Splatting with Neural Basis Extension. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*. 6043–6052.

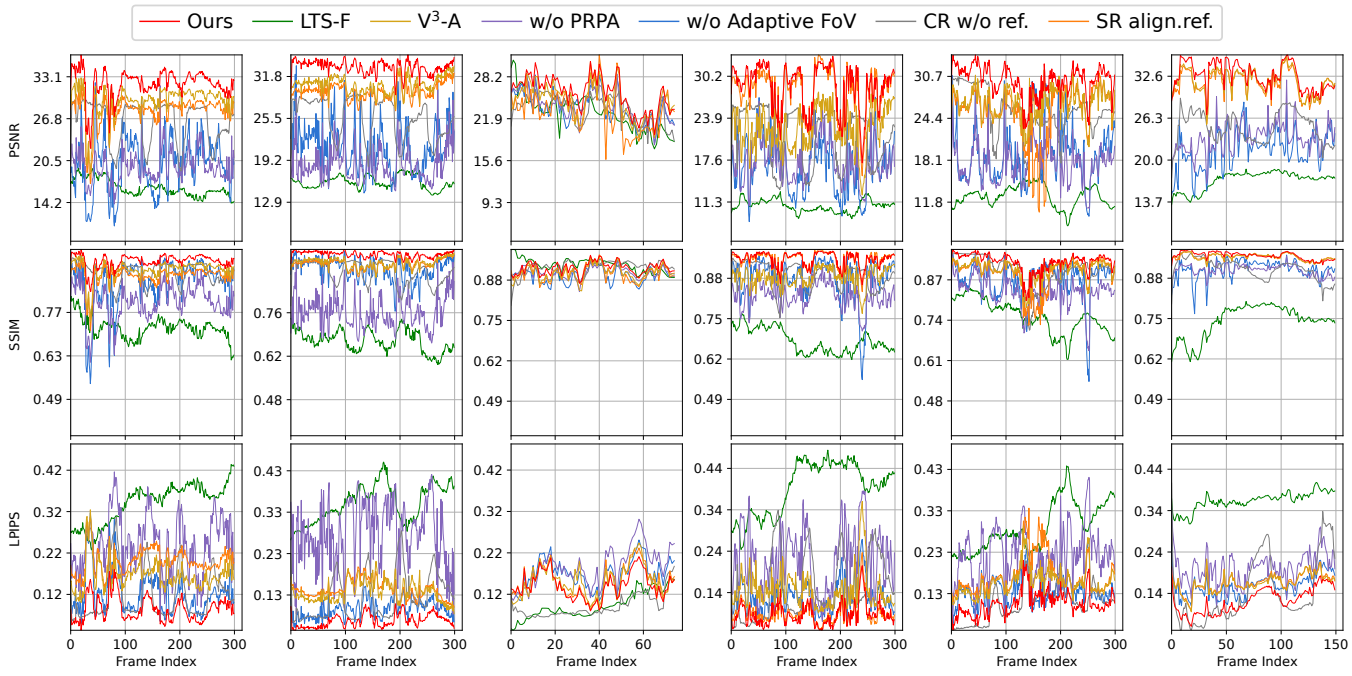
Mufeng Zhu, Mingju Liu, Cunxi Yu, Cheng-Hsin Hsu, and Yao Liu. 2025. SGSS: Streaming 6-DoF Navigation of Gaussian Splat Scenes. In *Proceedings of the 16th ACM Multimedia Systems Conference (MMSys '25)*. 46–56.

Walter Zimmer, Gerhard Arya Wardana, Suren Sritharan, Xingcheng Zhou, Rui Song, and Alois C Knoll. 2024. Tumtraf v2x cooperative perception dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 22668–22677.



(a) "cook spinach" (b) "cut roasted beef" (c) "walking" (d) "discussion" (e) "stepin" (f) "basketball"

Fig. 10. Comparison of visual quality under fixed bandwidth. The y-axis ranges vary across subplots while maintaining equal scale spans for each metric across videos. Results for all videos are included in the supplementary material.

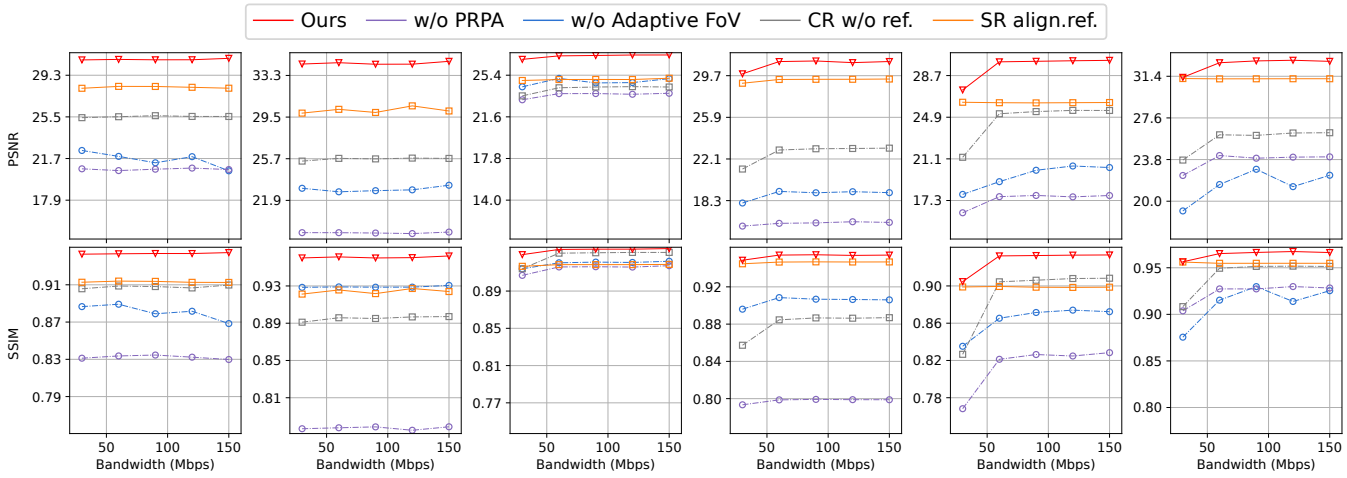


(a) "cook spinach" (b) "cut roasted beef" (c) "walking" (d) "discussion" (e) "stepin" (f) "basketball"

Fig. 11. Comparison of visual quality under fluctuating bandwidth. The y-axis ranges vary across subplots while maintaining equal scale spans for each metric across videos. Results for all videos are included in the supplementary material.

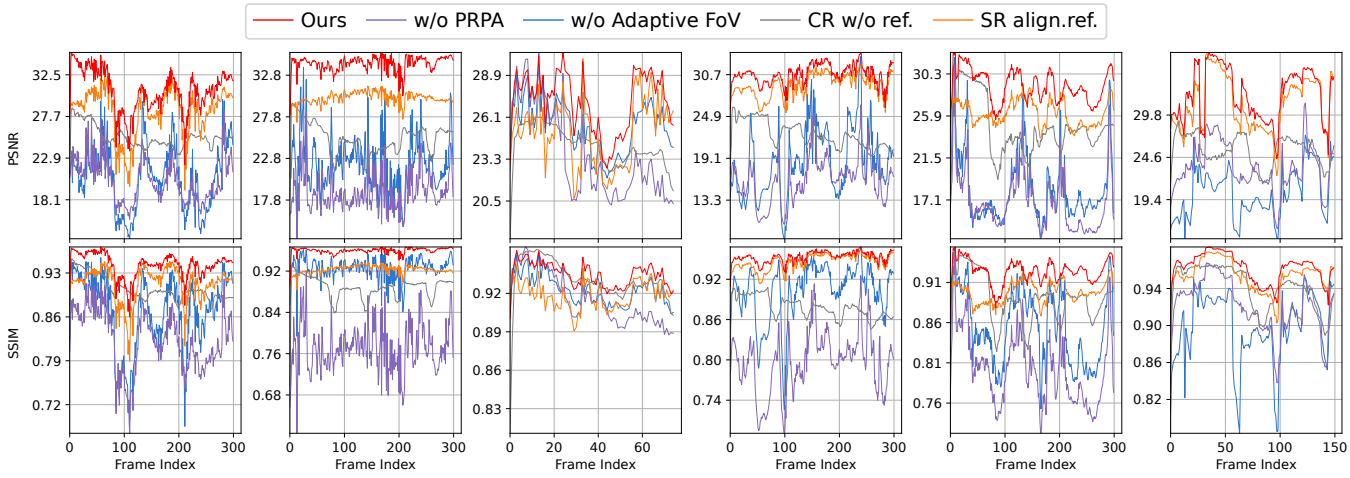
(a) Color-distorted (1600x1200) (b) Reference image (400x300) (c) Color-restored (1600x1200) (d) Ground truth (1600x1200)

Fig. 12. Visual results of the frame 64 in "coffee martini" under 30Mbps. Additional visual and video results are included in supplementary materials.



(a) "cook spinach" (b) "cut roasted beef" (c) "walking" (d) "discussion" (e) "stepin" (f) "basketball"

Fig. 13. Comparison of visual quality under fixed bandwidth on volumetric videos prepared by HiCoM [Gao et al. 2024]. The y-axis ranges vary across subplots while maintaining equal scale spans for each metric across videos. Results for all videos (including volumetric videos prepared by Dynamic 3DGS [Luiten et al. 2024], 4DGS [Wu et al. 2024] and HiCoM [Gao et al. 2024]) under fixed bandwidth conditions are included in the supplementary material.



(a) "cook spinach" (b) "cut roasted beef" (c) "walking" (d) "discussion" (e) "stepin" (f) "basketball"

Fig. 14. Comparison of visual quality under fluctuating bandwidth on volumetric videos prepared by HiCoM [Gao et al. 2024]. The y-axis ranges vary across subplots while maintaining equal scale spans for each metric across videos. Results for all videos (including volumetric videos prepared by Dynamic 3DGS [Luiten et al. 2024], 4DGS [Wu et al. 2024] and HiCoM [Gao et al. 2024]) under fluctuating bandwidth conditions are included in the supplementary material.