TAC-4DGS: Trajectory-Aware Compression for Rate-Distortion Optimized 4D Gaussian Splatting

Stefan Schulz Fernando Edelstein Hannah Dröge Reinhard Klein Markus Plack
University of Bonn, Germany
{sschulz, edelstein, droege, rk, mplack}@cs.uni-bonn.de

Abstract

4D Gaussian Splatting (4DGS) enables real-time rendering of complex dynamic scenes with high fidelity, but its trajectory representation remains expensive to store. Recent rate-distortion optimized methods substantially reduce this cost, yet important dynamic components remain largely uncompressed. In particular, position wavelet approximation coefficients and rotation keyframes account for over 30% of the total bitstream. The key challenge is not simply to quantize these trajectory attributes, but to first put them into a form that can be quantized effectively. In their native form, these signals carry structure that is benign for representation but detrimental to codebook learning: wavelet approximation coefficient magnitudes vary by orders of magnitude across Gaussians, from near-static to fast-moving points, risking severe gradient imbalance during quantization-aware training if not explicitly conditioned, while absolute rotations and positions retain predictable temporal variation.

We present TAC-4DGS, a trajectory-aware compression method that reparameterizes these components before quantization. A data-adaptive logarithmic companding transform balances wavelet approximation coefficient scales and enables stable end-to-end codebook training, while inter-frame delta rotation coding and reference-frame position subtraction remove predictable structure before quantization. TAC-4DGS achieves 41.9% average bitstream reduction on N3V and 33.8% on Technicolor relative to state-of-the-art 4DGS compression, with PSNR differences of only -0.31 dB and $+0.40$ dB respectively. As a coding-level extension requiring no changes to the underlying rendering pipeline, TAC-4DGS achieves a 93.6–99.1% total size reduction relative to the uncompressed 4DGS representation.

The project page can be found here with accompanying code: https://stefanmschulz.github.io/TAC-4DGS_webpage

1. Introduction

Real-time rendering of dynamic 3D scenes has advanced rapidly starting with the introduction of 3D Gaussian Splatting (3DGS) [14] and its temporal extensions to 4D [16, 35, 37]. These methods represent scenes as collections of Gaussian primitives, each carrying position, rotation, and appearance attributes that evolve over time through explicit keyframe trajectories. While 4D Gaussian Splatting (4DGS) achieves real-time rendering at high visual fidelity, the storage cost of per-Gaussian motion trajectories remains a significant bottleneck: a single dynamic scene may require hundreds of megabytes [16], limiting streaming, mobile rendering, and archival applications.

Several recent methods address this challenge by redesigning the underlying scene representation for compactness [7, 13, 40], or by introducing rate-distortion optimized compression pipelines [9, 10, 15]. Among these, RD4DGS [15] achieves state-of-the-art compression by applying entropy-constrained vector quantization (ECVQ) [4] to static Gaussian attributes and Haar wavelet representations of position trajectories, training codebooks and entropy models jointly under a Lagrangian rate-distortion objective. However, a substantial fraction of the RD4DGS bitstream remains uncompressed: the wavelet approximation coefficients of position trajectories which capture the low-frequency shape of each Gaussian's motion are stored as raw floating-point arrays outside the entropy-coded pipeline, and rotation keyframes are quantized as independent absolute quaternions without exploiting temporal correlation. Together, these components constitute over 30% of the total bitstream.

We propose **TAC-4DGS** (Trajectory-Aware Compressed 4D Gaussian Splatting), which closes this gap by bringing the uncompressed trajectory data under rate-distortion-optimized entropy coding. The core mechanism is a dedicated ECVQ stage for the wavelet approximation coeffi-

cients, replacing raw storage with learned codebook quantization. Applying ECVQ to these coefficients is not straightforward: their values span orders of magnitude across Gaussians, from near-static points to fast-moving ones, and a naive quantization-aware training scheme risks gradient imbalances of orders of magnitude (up to $47\,000\times$ on common datasets, cf. Sec. 3.2) between typical and high-motion Gaussians, destabilizing codebook convergence. We address this with a data-adaptive logarithmic magnitude companding transform that normalizes the coefficient range before quantization, enabling stable convergence. Two additional techniques further improve reconstruction quality at negligible size overhead: inter-frame delta rotation coding, which concentrates the quaternion distribution around the identity for more efficient codebook utilization, and reference-frame position subtraction, which reduces the dynamic range of wavelet coefficients before the transform.

TAC-4DGS operates purely at *coding level*: it does not alter the 4DGS scene representation, the training procedure, or the rendering pipeline, and introduces zero overhead at render time. On the Neural 3D Video (N3V) dataset [19], TAC-4DGS achieves an average bitstream reduction of 41.9% across six scenes and four rate-distortion operating points, with an average PSNR change of only -0.31 dB relative to the state of the art.

Contributions. In summary, our main contributions are:

- We identify that a substantial fraction of the RD4DGS bitstream consists of trajectory data that bypasses rate-distortion optimization, and close this gap by extending ECVQ to wavelet approximation coefficients with a data-adaptive logarithmic companding transform that resolves gradient instability during codebook training.
- We introduce inter-frame delta rotation coding and reference-frame position subtraction as lightweight quality recovery techniques that stabilize reconstruction on scenes with complex motion at high bitrates.
- We provide a comprehensive component analysis decomposing the bitstream savings by source and the quality impact by technique, evaluated on two multi-view video datasets across multiple rate-distortion operating points.

2. Related Work

2.1. 3D and 4D Gaussian Splatting

Neural Radiance Fields (NeRF) [24] laid the foundation for novel view synthesis via implicit neural representations, though their reliance on volume rendering incurs heavy computational overhead. To enable real-time rendering, 3D Gaussian Splatting (3DGS) [14] introduced explicit 3D Gaussian primitives combined with a hardware-friendly tile-based rasterizer. Extending 3DGS to dynamic scenes (4DGS) has largely bifurcated into two paradigms. The first paradigm employs implicit neural deformation fields or space-time feature grids (e.g., HexPlane voxels [35], 4D Rotor GS [5], and monocular deformation MLPs [36]) to query continuous Gaussian trajectories over time. The second research direction focuses on explicit keyframe dynamic representations, exemplified by Ex4DGS [16], which explicitly separates static background Gaussians from dynamic foreground trajectories. By providing discrete, structured motion trajectories, explicit representations eliminate the need for costly field queries and offer a highly streamable substrate that is particularly well-suited for rate-distortion compression frameworks.

2.2. Static Gaussian Splatting Compression

Compressing 3DGS representations has evolved rapidly through three primary technical primitives: parameter pruning, vector quantization (VQ), and learned entropy modeling. Early works established learnable masking and sensitivity-aware pruning to eliminate redundant or low-opacity Gaussians without degrading visual fidelity [6, 17, 25]. Subsequent methods integrated spatial structural anchors, such as Scaffold-GS [23], to group Gaussians locally and compress attributes via codebook-based VQ [3, 22]. More recently, advanced spatial context models, hyperpriors, context-adaptive triplanes, and hybrid entropy models [20, 33, 38] have been introduced to exploit spatial correlations for statistical entropy coding. While these static techniques effectively compress single-frame representations, they fail to leverage inter-frame dynamic correlations, leading to severe bandwidth redundancies when naively applied sequentially to dynamic video sequences.

2.3. Compressed Dynamic Representations

To address dynamic temporal redundancy, recent dynamic GS compression methods can be categorized into *representation-level* and *coding-level* approaches. Representation-level techniques alter the underlying scene representation to enforce spatial-temporal compactness. For example, anchor-driven dynamic architectures (ADC-GS) [11], hierarchical coherent motion grids (HiCoM) [7], sub-vector quantization (OMG4) [18], and feedforward dynamic models (D-FCGS [39], QUEEN [8]) constrain overall scene complexity.

Conversely, *coding-level* approaches maintain explicit 4DGS parameters while directly optimizing rate-distortion (RD) objective functions for storage and streaming. Frameworks such as MEGA [40], 4DGC [10], Light4GS [21], and TED-4DGS [9] leverage color decomposition, entropy-constrained deformation, implicit entropy models, or temporal activation masks to minimize bitrate. Positioned firmly within coding-level frameworks, RD4DGS [15] represents the current state-of-the-art baseline, optimizing explicit dynamic keyframe representations [16] end-to-end

via Entropy-Constrained Vector Quantization (ECVQ) and wavelet position coding. However, RD4DGS focuses predominantly on static properties and spatial position tracks, leaving fine-grained temporal trajectory variations and local rotational components uncompressed. In this work, we specifically target these uncompressed dynamic components, demonstrating that modeling inter-frame motion residuals yields substantial rate savings while preserving high-frequency dynamic detail.

2.4. Signal Conditioning for Learned Quantization

Quantizer performance depends critically on input signal statistics. In classical telephony, nonlinear companding before uniform quantization recovers 4–6 bits of effective dynamic range [12, 28], while video codecs such as H.264 [34] remove predictable temporal structure via residual coding before quantization. In learned compression, ECVQ [4] enables codebook learning under rate penalties, and the Straight-Through-Estimator [2] provides gradient pathways through discrete quantization, as adopted in VQ-VAE [31]. These frameworks assume well-conditioned inputs. When applied to 4DGS trajectories, this assumption fails: wavelet coefficients are extremely heavy-tailed and absolute quaternions waste codebook capacity on global scene layout. We address this by adapting classical conditioning principles to make trajectory signals amenable to learned ECVQ.

3. TAC-4DGS: Trajectory-Aware Compression

We formulate dynamic Gaussian trajectory compression within the rate-distortion optimization, jointly considering position and rotation parameters together with the remaining Gaussian attributes. To make these trajectory signals amenable to learned vector quantization, we first reparameterize their native representations and condition their dynamic range before integrating them into quantization-aware optimization.

Concretely, our method combines trajectory reparameterization into relative position and rotation signals (Sec. 3.1), median-calibrated magnitude conditioning prior to quantization (Sec. 3.2), and quantization-aware ECVQ integrated into differentiable rasterization (Sec. 3.3). An overview of the resulting pipeline is shown in Fig. 1.

Dynamic Gaussian Representation. We represent each dynamic Gaussian $\mathcal{G}_i$ by its static attributes, i.e. scale $\mathbf{s}_i \in \mathbb{R}^3$, opacity $\alpha_i \in [0,1]$, and spherical harmonics coefficients $\mathbf{f}_i$, together with a temporal trajectory over K keyframes. The trajectory is given by per-keyframe positions $\{\boldsymbol{\mu}_i^{(k)}\}_{k=1}^K$, with $\boldsymbol{\mu}_i^{(k)} \in \mathbb{R}^3$, and orientations $\{\mathbf{q}_i^{(k)}\}_{k=1}^K$, represented as unit quaternions $\mathbf{q}_i^{(k)} \in \mathbb{H}$.

3.1. Reference-Centered Trajectory Coding

Absolute keyframe parameters are dominated by global scene layout rather than local motion. As a result, a codebook must cover the full range of scene positions and orientations, limiting its capacity to represent local trajectory variations. We therefore reparameterize trajectories relative to local reference states.

Inter-Frame Rotation Residuals. For each dynamic primitive i, we define the initial keyframe as an anchor orientation $\mathbf{q}_i^{\text{ref}} = \mathbf{q}_i^{(1)}$. Subsequent keyframe orientations are transformed into consecutive relative rotations using quaternion Hamilton composition:

$$\Delta\mathbf{q}_i^{(k)} = \left(\mathbf{q}_i^{(k-1)}\right)^{-1} \otimes \mathbf{q}_i^{(k)}, \quad k = 2, \ldots, K, \quad (1)$$

where $(\cdot)^{-1}$ represents the quaternion conjugate. We store only the vector part $\mathbf{v} = (x, y, z)$ of each delta quaternion. The scalar part is recovered as $w = \sqrt{1 - \|\mathbf{v}\|^2}$, with its sign tracked separately to maintain hemisphere consistency. Since keyframes are sampled densely relative to the underlying motion, inter-frame angular displacements are typically small. Consequently, the delta quaternions $\Delta\mathbf{q}_i^{(k)}$ concentrate around the identity quaternion $\mathbf{q}_{\text{id}} = (1, 0, 0, 0)$, yielding a zero-centered, low-entropy distribution.

Zero-Centered Wavelet Motion Decomposition. For spatial trajectories, global positional offsets are eliminated by subtracting the initial frame position,

$$\tilde{\boldsymbol{\mu}}_i^{(k)} = \boldsymbol{\mu}_i^{(k)} - \boldsymbol{\mu}_i^{(1)}, \quad k = 1, \ldots, K, \quad (2)$$

where the anchor coordinate $\boldsymbol{\mu}_i^{(1)}$ is retained once per primitive as a single float16 vector. The zero-centered displacement trajectory $\{\tilde{\boldsymbol{\mu}}_i^{(k)}\}_{k=1}^K$ is then decomposed along the temporal axis via a 1D Haar wavelet transform into approximation coefficients $\mathbf{c}_i^{\text{approx}} \in \mathbb{R}^{\frac{K}{2} \times 3}$ and high-frequency detail coefficients $\mathbf{c}_i^{\text{detail}} \in \mathbb{R}^{\frac{K}{2} \times 3}$. Owing to motion smoothness, high-frequency details carry negligible reconstruction energy and are therefore discarded, achieving an immediate $2\times$ parameter reduction prior to quantization.

3.2. Median-Calibrated Magnitude Companding

Even after relative delta transformation, trajectory coefficients exhibit extreme heavy-tailed distributions. A small minority of Gaussians undergoing rapid motion produce high-magnitude vectors, while the majority remain near-zero. Applying standard uniform vector quantization under a frequency-weighted rate-distortion loss causes codebooks to over-allocate entries to near-zero regions, leaving high-velocity dynamic regions under-represented.

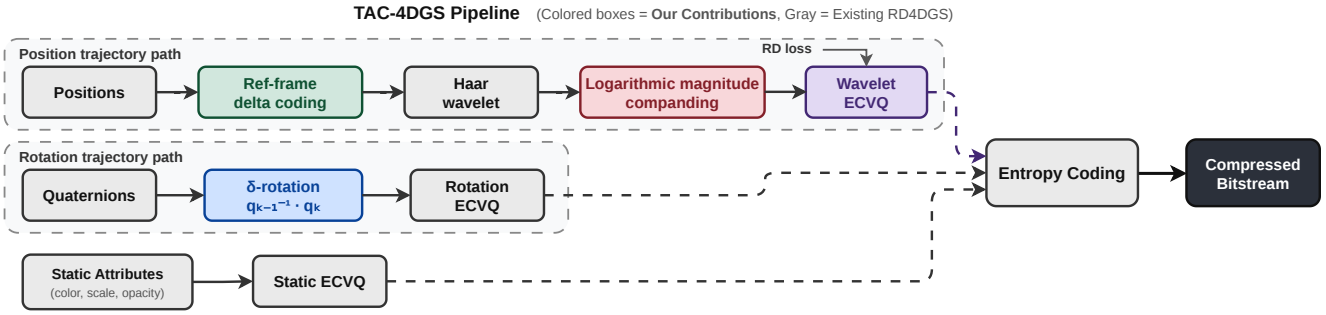


Figure 1. **Overview of the TAC-4DGS dynamic trajectory coding framework.** Continuous rotation keyframes and spatial position trajectories are parameterized into relative inter-frame deltas and reference-subtracted Haar wavelet approximation coefficients, then passed through a median-calibrated μ-law compander before joint Entropy-Constrained Vector Quantization (ECVQ). A straight-through estimator (STE), applied at the real-space rendered output rather than the quantized companded value, avoids the severe gradient imbalance that a naive placement would otherwise incur (Sec. 3.2).

More critically, a naive quantization-aware training scheme for these coefficients carries a second, distinct risk. Companding is the natural fix for the allocation problem above, but if the training gradient is allowed to back-propagate through the inverse-companding transform f_c^{-1} (Eq. (5)), e.g. via a straight-through estimator applied at the quantized, companded value rather than at the real-space rendering output, its local derivative grows exponentially with coefficient magnitude, since f_c^{-1} exactly inverts a logarithmic compression. We measure this directly: across the full RDO fine-tuning trajectory of `coffee_martini` at Level 1, the ratio between the largest and median per-Gaussian gradient magnitude under this construction stays consistently between $47\,038\times$ and $47\,701\times$ (Fig. 2), concentrated entirely on the small minority of high-motion Gaussians identified above (measured via automatic differentiation on f_c^{-1} directly, since the codeword lookup itself has no gradient to its input). Sec. 3.3 describes how our training procedure avoids ever exercising this path while still retaining companding's allocation benefit.

To stabilize training and maintain uniform relative quantization precision across varying motion scales, we introduce a non-linear vector companding operator $f_c(\cdot)$ prior to quantization.

Unbounded Magnitude Companding. For spatial position wavelet coefficients $\mathbf{v} \in \mathbb{R}^3$, the vector-valued logarithmic compander is defined as:

$$f_c(\mathbf{v}) = \frac{\mathbf{v}}{\|\mathbf{v}\|} \cdot \ln\left(1 + \frac{\|\mathbf{v}\|}{\rho}\right), \tag{3}$$

where we define $f_c(\mathbf{0}) = 0$ and ρ is a data-driven scale parameter set dynamically to the empirical median magnitude across all training vectors in the attribute domain:

$$\rho = \text{median}\left(\left\{\|\mathbf{v}_i\|\right\}_{i=1}^{N}\right). \tag{4}$$

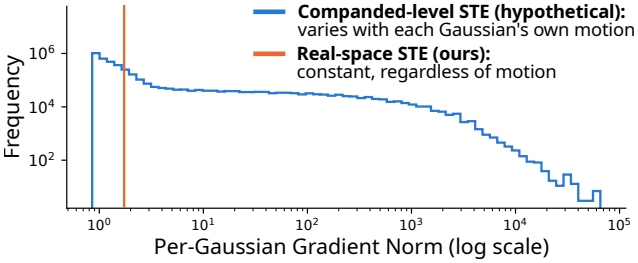


Figure 2. Per-Gaussian gradient sensitivity through f_c^{-1} on `coffee_martini` (Level 1). A companded-level STE (blue) varies with each Gaussian's own motion, reaching a consistent $\sim 47\,000\times$ max/median ratio, while our real-space STE (orange) yields constant unit sensitivity.

The parameter ρ is evaluated once at codebook initialization and stored in the bitstream header. This transformation maps the median coefficient magnitude to $f_c(\rho) = \ln(2) \approx 0.693$, logarithmically compressing extreme high-energy outliers while maintaining quasi-linear behavior for small displacements ($\|\mathbf{v}\| \ll \rho$).

Following dequantization, the original spatial scale is restored via the exact analytical inverse $f_c^{-1}(\mathbf{v}')$:

$$f_c^{-1}(\mathbf{v}') = \frac{\mathbf{v}'}{\|\mathbf{v}'\|} \cdot \rho\left(e^{\|\mathbf{v}'\|} - 1\right). \tag{5}$$

Norm-Preserving Companding for Quaternions. For rotation deltas, the vector part $\mathbf{v} = (x, y, z)$ must satisfy $\|\mathbf{v}\| \leq 1$ to allow exact scalar reconstruction $w = \sqrt{1 - \|\mathbf{v}\|^2}$. We enforce this boundary in compressed space using a normalized compander:

$$f_{c,\text{bound}}(\mathbf{v}) = \frac{\mathbf{v}}{\|\mathbf{v}\|} \cdot \frac{\ln\left(1 + \frac{\|\mathbf{v}\|}{\rho}\right)}{\ln\left(1 + \frac{1}{\rho}\right)}, \tag{6}$$

which strictly bounds companded magnitudes within $[0, 1]$, preventing non-unit quaternion errors during rendering.

3.3. Trajectory Quantization-Aware Training

Companded trajectory vectors $\mathbf{v}' = f_c(\mathbf{v})$ are quantized using an Entropy-Constrained Vector Quantizer (ECVQ) [4]. Each attribute maintains a codebook $\mathcal{C} = \{\mathbf{e}_j\}_{j=1}^{M}$ and a categorical probability distribution $p(j)$ parameterized via soft-max logits.

During the forward pass, vectors are mapped to codebook entries j^* via joint rate-distortion distance minimization:

$$j^* = \arg\min_{j\in\{1,\dots,M\}} \|\mathbf{v}' - \mathbf{e}_j\|_2^2 + \lambda^{-1}\log_2\frac{1}{p(j)}, \quad (7)$$

where λ governs the rate-distortion trade-off. Codebook indices are losslessly encoded via arithmetic coding according to $p(j)$.

Quantization-Aware Rasterization. To eliminate train-test distribution shifts, quantized trajectory parameters are routed directly into the differentiable rasterizer during training via Straight-Through Estimation (STE). Specifically, position parameters used during rasterization are substituted with quantized spatial inputs $\hat{\boldsymbol{\mu}}_{\text{quant}}$:

$$\hat{\boldsymbol{\mu}}_{\text{render}} = \text{sg}\left(\hat{\boldsymbol{\mu}}_{\text{quant}} - \boldsymbol{\mu}\right) + \boldsymbol{\mu}, \quad (8)$$

where $\text{sg}(\cdot)$ is the stop-gradient operator. The forward pass computes image losses on quantized renders, while backward gradients update continuous trajectory parameters $\boldsymbol{\mu}$, ensuring stable convergence.

4. Experiments

We evaluate TAC-4DGS across multiple dynamic-scene benchmarks and rate-distortion operating points, and analyze the contribution of each proposed component through ablations.

4.1. Experimental Setup

Datasets. We evaluate our proposed framework on two commonly used dynamic 3D scene benchmarks: Neural 3D Video (N3V) Dataset [19], and Technicolor Multi-View Dataset [26]. *N3V* is a real-world multi-view dynamic benchmark comprising six synchronized sequences. Following standard protocol, each sequence contains 300 frames evaluated with keyframe interval $N = 36$. *Technicolor Multi-View Dataset* is a multi-camera dynamic benchmark featuring complex human and non-human motion and lighting. Following the baseline RD4DGS evaluation protocol, we train and evaluate on 50 continuous frames per sequence using $N = 22$ as specified in the config files.

Baselines and Operating Points. We compare against state-of-the-art dynamic representation frameworks, including NeRF-based methods (NeRFPlayer [29], HyperReel [1], K-Planes [27], MixVoxels [32]) and 3D/4D Gaussian Splatting models (4DGS [35], 4DGaussians [37], 3DGStream [30], Ex4DGS [16]). Further explanations and comparison models can be found in the Supplemental with a discussion on how comparable these methods are.

Our primary rate-distortion baseline is RD4DGS [15]. To evaluate across bit-budget regimes, we evaluate across four standard rate-distortion operating points (Levels 1, 2, 4, 6), ranging from aggressive low-rate compression (Level 1) to high-fidelity regimes (Level 6). All TAC-4DGS models are fine-tuned from identical pretrained base 4DGS checkpoints to isolate the quantitative impact of our unified trajectory coding pipeline.

Evaluation Metrics. Novel view synthesis quality is evaluated using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). Bitstream size is reported as total compressed disk storage in Megabytes (MB). Relative compression savings are computed as $\Delta S = (1 - S_{\text{ours}}/S_{\text{base}}) \times 100\%$, and quality shifts as $\Delta\text{PSNR} = \text{PSNR}_{\text{ours}} - \text{PSNR}_{\text{base}}$. The reported size is using MB with MB = bytes$/1024^2$.

Implementation Details. TAC-4DGS is implemented within PyTorch on top of the RD4DGS codebase. Quantization codebook sizes are configured to $M = 8192$ for scale, rotation deltas, and DC color; $M = 4096$ for higher-order spherical harmonics; and $M = 1024$ for position wavelet approximation coefficients. Codebook vectors use dimension $d = 3$.

Training follows a two-stage schedule: 30,000 iterations of unconstrained 4DGS learning followed by 10,000 rate-distortion optimization (RDO) fine-tuning iterations. Scale factors μ are computed as the empirical median vector norm after the initial stage and fixed during RDO. All models were trained on a single NVIDIA RTX 4090 GPU.

4.2. Compression Performance

N3V Benchmark. Tab. 1 summarizes quantitative performance on the Neural 3D Video dataset. Across all six dynamic scenes, TAC-4DGS achieves substantial bitstream reductions over the state-of-the-art RD4DGS baseline. At Level 1 (maximum compression), TAC-4DGS reduces average bitstream size from 1.59 MB to 0.99 MB (-37.8%) with negligible fidelity change (-0.17 dB PSNR). At Level 6, TAC-4DGS reduces bitstream size from 13.86 MB to 7.40 MB (-46.6%) which is also a 93.6% total size reduction over the uncompressed 4DGS representation.

Across all evaluated operating levels and sequences, bitstream savings average **41.9%** (ranging between 36.6% and 47.8%). Crucially, savings increase monotonically at higher rate points (Level 6: -46.6% vs Level 1: -37.8%), demonstrating that parameterizing and quantizing trajectory parameters eliminates redundant bitstream scaling as model capacity grows.

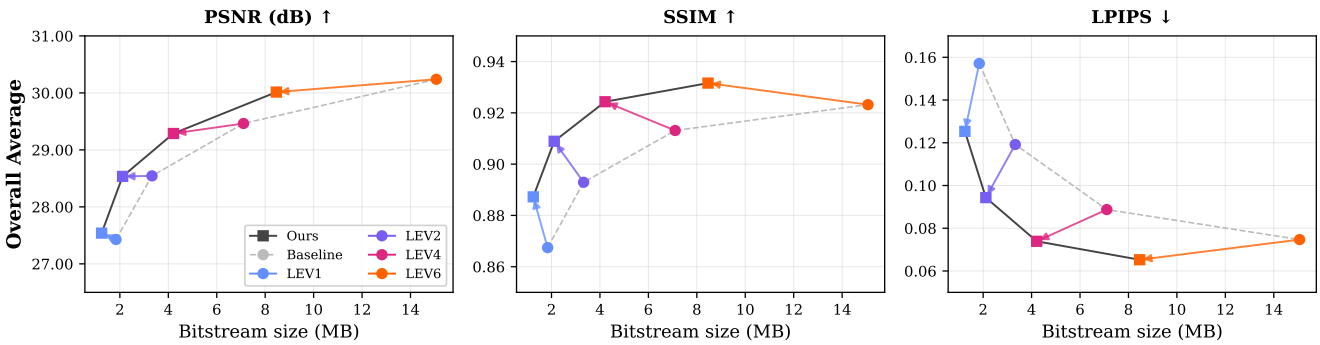


Figure 3. **Averaged Rate-Distortion Curves across N3V [19] and Technicolor [26].** TAC-4DGS consistently shifts the RD curves toward lower bitrates across all operating points. Despite a small PSNR decrease at most rates, it achieves consistently better SSIM and LPIPS, with substantial gains at several operating points, demonstrating a favorable trade-off between bitrate and perceptual reconstruction quality.

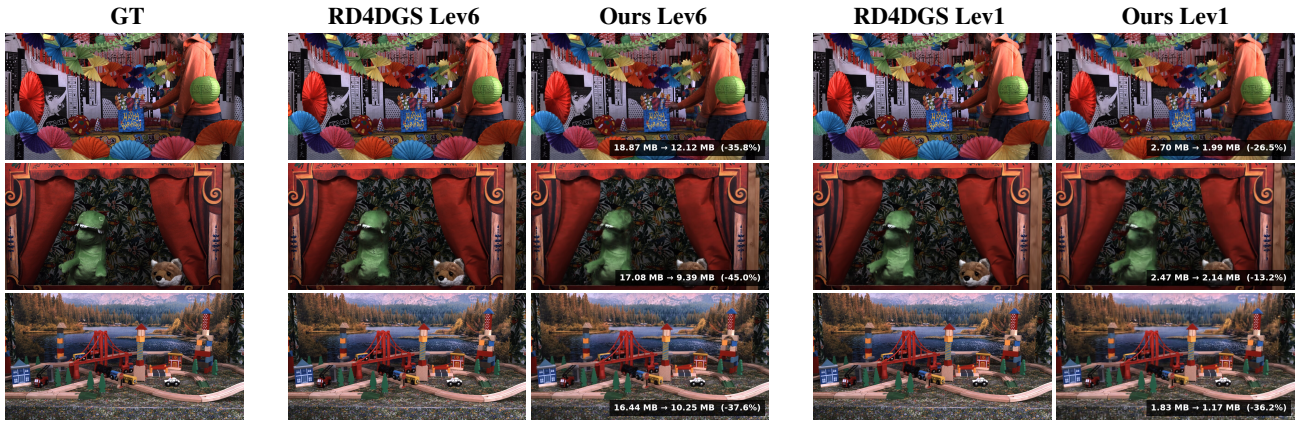


Figure 4. **Qualitative results on a subset of scenes of the Technicolor [26] dataset.** We compare ground truth against RD4DGS and TAC-4DGS at low- and high-rate operating points (Levels 1 and 6), with the corresponding bitrate savings reported for each setting

Technicolor Benchmark. Tab. 2 presents multi-scene metrics on the Technicolor dynamic dataset across 50-frame test sequences. TAC-4DGS achieves an average bitstream compression saving of **33.8%** (reducing storage from 8.27 MB to 5.22 MB) while simultaneously improving reconstruction fidelity across all quality metrics (+0.40 dB PSNR, +0.0467 SSIM, and −0.0759 LPIPS on average). The quality improvement, most pronounced on Train (+1.70 dB at Level 1), suggests that compressing trajectory data provides an implicit regularization effect by constraining model capacity and reducing overfitting to training views, consistent with prior observations [15]. On high-rate configurations (Level 6), TAC-4DGS reduces total storage by **39.4%** (17.46 MB → 10.59 MB) while maintaining high visual fidelity (30.82 dB PSNR). On sparse or fast trajectories, such as `Painter`, unconditioned RDO can fail to optimize the rate-distortion trade-off, whereas TAC-4DGS remains stable.

Rendering Speed and Decoding Overhead. Because TAC-4DGS operates purely on parameter representations prior to rendering, it does not alter the underlying 4DGS volumetric rasterization pipeline. Decoded Gaussian primitives are fully compatible with standard rasterization CUDA kernels. Consequently, rendering frame rates remain identical to the baseline (165.2 FPS at Level 1, 104.1 FPS at Level 6 on an NVIDIA RTX 3090). Additional decoding overhead comprising inverse logarithmic companding and quaternion delta composition requires under 0.5 second per scene during file initialization as added latency.

Interactive 4DGS Results. We provide easily viewable results in the suppl. material for qualitative inspection.

4.3. Ablation Study & Component Analysis

We separately analyze (i) the sources of bitstream savings and (ii) the effect of the proposed signal-conditioning operations on reconstruction quality.

Bitstream Decomposition. Tab. 3 decomposes the net 42.1% bitstream reduction relative to RD4DGS. The primary source of compression is ECVQ of the previously uncompressed temporal wavelet approximation coefficients, which accounts for 31.1% of the total bitstream reduction.



Figure 5. **Qualitative results on scences of the N3V [19] dataset.** TAC-4DGS achieves substantial bitrate savings over RD4DGS at the cost of a modest reduction in visual fidelity, illustrating the rate-distortion trade-off across Levels 1 and 6.

Table 1. **Quantitative novel view synthesis benchmark on Neural 3D Video (N3V) dataset [19].** PSNR (dB), overall model size (MB), and rendering frame rate (FPS) evaluated across six dynamic sequences. FPS were measured on a NVIDIA 3090 GPU for fair comparison. We take all but the RD4DGS [15] numbers from the original papers/the RD4DGS paper and retrain RD4DGS and ours only.

	Model	PSNR (dB)↑							MB↓	FPS↑
		Coffee Martini	Cook Spinach	Cut Roasted Beef	Flame Salmon	Flame Steak	Sear Steak	Average	Size (Avg)	FPS
NeRF-based	NeRFPlayer [29]	31.53	30.56	29.35	31.65	31.93	29.13	30.69	5130	0.05
	HyperReel [1]	28.37	32.30	32.92	28.26	32.20	32.57	31.10	360	2.0
	K-Planes [27]	29.99	32.60	31.82	30.44	32.38	32.52	31.63	311	0.3
	MixVoxels-L [32]	29.63	32.25	32.40	29.81	31.83	32.10	31.34	500	37.7
	MixVoxels-X [32]	30.39	32.31	32.63	30.60	32.10	32.33	31.73	500	4.6
3D/4DGS-based	4DGS [35]	26.51	32.11	31.74	26.93	31.44	32.42	30.19	6057	72.0
	4DGaussians [37]	26.69	31.89	25.88	27.54	28.07	31.73	28.63	34	146.6
	3DGStream [30]	27.75	33.31	33.21	28.42	34.30	33.01	31.67	1200	–
	Ex4DGS [16]	28.79	33.23	33.73	29.29	33.91	33.69	32.11	115	72.3
	RD4DGS `Level 1` [15]	26.11	28.37	27.92	25.12	28.40	28.73	27.44	1.59	165.2
	RD4DGS `Level 6` [15]	27.97	30.87	30.55	27.23	31.42	32.28	30.05	13.86	104.1
	Ours `Level 1`	25.84	28.05	27.75	25.28	27.81	28.88	27.27	**0.99** (-37.8%)	165.2
	Ours `Level 6`	27.74	30.33	30.09	27.03	30.59	31.87	29.61	**7.40** (-46.6%)	104.1

Lossless compression of non-differentiable trajectory metadata provides an additional 12.1% reduction.

In contrast, the three signal-conditioning operations introduce only a small bitstream overhead: reference-frame position delta coding (+0.3%), delta rotation coding (+0.4%), and median-calibrated companding (+0.4%), for a combined overhead of +1.1%. Their primary purpose is therefore not to reduce the bitrate directly, but to condition the trajectory signals for more effective vector quantization and to preserve reconstruction quality. This separation highlights that the majority of the compression gain comes from bringing previously uncompressed trajectory components into the ECVQ pipeline, while the proposed signal transformations make these components more amenable to quantization.

Quality Impact of Signal Conditioning. Table 4 evaluates the effect of the proposed signal-conditioning operations at low and high rates. We report the average ΔPSNR relative to the uncompressed RD4DGS baseline over `coffee_martini` and `sear_steak`. Since

Table 2. **Quantitative rate-distortion evaluation on Technicolor dataset [26].** Average multi-view synthesis quality across 3 scenes evaluated on 50 continuous test frames. TAC-4DGS reduces storage by 24.5–39.4% across operating points, with larger savings at higher rates while improving all reported quality metrics.

Lvl	Size (MB)↓		Save	PSNR↑	SSIM↑	LPIPS↓
	Base	Ours	(%)	Ours (Δ)	Ours (Δ)	Ours (Δ)
L1	2.33	**1.76**	**24.5**	28.08 (+0.68)	.8637 (+.0613)	.1192 (-.1042)
L2	4.25	**2.91**	**31.5**	29.24 (+0.30)	.9014 (+.0507)	.0762 (-.0862)
L4	9.02	**5.63**	**37.6**	30.50 (+0.39)	.9279 (+.0402)	.0501 (-.0620)
L6	17.46	**10.59**	**39.4**	30.82 (+0.22)	.9344 (+.0110)	.0437 (-.0510)
Avg.	8.27	**5.22**	**33.8**	**+0.40**	**+.0467**	**-.0759**

Table 3. **Bitstream decomposition of TAC-4DGS.** Percentage changes show the contribution of each component to the average bitstream reduction relative to RD4DGS. Negative values denote savings, whereas positive values denote additional overhead.

Component	Bitstream Change (%)
Compression Sources	
Wavelet Approximation ECVQ	−31.1
Lossless Metadata Compression	−12.1
Signal Conditioning Overhead	
Reference-Frame Position Delta	+0.3
Delta Rotation Coding	+0.4
Median-Calibrated Companding	+0.4
TAC-4DGS	**−42.1**

all configurations have bitstream sizes within 1% of one another, the differences in reconstruction quality primarily reflect the effectiveness of the trajectory representations rather than differences in bitrate. At the high-rate operating point (Level 6), unconditioned ECVQ exhibits a substantial quality degradation on the more challenging `sear_steak` sequence, reaching −2.28 dB relative to RD4DGS. The full TAC-4DGS representation reduces this degradation to −0.41 dB, corresponding to a 1.87 dB improvement over unconditioned ECVQ. The individual ablations further show the importance of the proposed transformations. Removing position delta coding reduces the `sear_steak` Level 6 result to −2.51 dB, while removing delta rotation coding and companding produces degradations of −0.71 dB and −0.52 dB, respectively.

Overall, the results indicate that reference-frame position deltas, delta rotation coding, and median-calibrated companding provide complementary benefits for trajectory quantization. Their bitrate overhead is negligible, while their effect on reconstruction quality becomes particularly important for challenging dynamic content and higher-rate operating points.

Table 4. **Quality ablation at low and high rates.** Average ΔPSNR (dB) relative to the uncompressed RD4DGS baseline, computed over `coffee_martini` and `sear_steak`. Level 1 and Level 6 represent low- and high-rate operating points, respectively.

Configuration	Level 1	Level 6
ECVQ only	−0.11	−1.25
w/o $\Delta\mathbf{q}$ rotation coding	−0.33	−0.70
w/o reference-frame position $\Delta\boldsymbol{\mu}$	−0.41	−1.43
w/o median-calibrated companding f_c	−0.22	−0.40
Full TAC-4DGS	**−0.06**	**−0.32**

5. Limitations and Future Work

TAC-4DGS has several limitations that motivate future work. On very dense scenes such as `Fabien`, where the primitive count exceeds several million Gaussians, the fixed trajectory codebook capacity (M=1024) can become saturated and degrade high-frequency motion details. Adaptive codebook allocation based on scene density could address this limitation. More broadly, TAC-4DGS assumes a uniform keyframe count across Gaussians, which is inefficient for short-lived or transient primitives. Adaptive temporal lifespans and keyframe frequencies could reduce this overhead. The current method also compresses trajectories independently and therefore does not exploit spatial correlations between neighboring Gaussians that share similar motion. Joint spatial-temporal trajectory modeling offers a further source of redundancy. Lastly, the benefit of companding depends on motion magnitude, suggesting motion-adaptive quantization schemes that selectively apply trajectory conditioning could further reduce overhead for quasi-static content.

6. Conclusion

We presented TAC-4DGS, a trajectory-aware compression framework for rate–distortion-optimized 4D Gaussian Splatting. By incorporating uncompressed position wavelets and rotation keyframes into an end-to-end entropy-constrained vector quantization pipeline, our method substantially reduces model storage over state-of-the-art baselines while preserving visual fidelity. Central to our approach, median-calibrated μ-law companding and relative delta parameterizations resolve extreme gradient imbalances during codebook optimization, ensuring stable convergence without requiring structural changes to the underlying rasterization engine. Ultimately, these results demonstrate that significant compression headroom remains in dynamic 3D representations, which can be effectively unlocked by applying tailored classical signal-processing principles to spatio-temporal trajectories.

Acknowledgements

This work has been funded by the Ministry of Culture and Science North Rhine-Westphalia under grant number PB22-063A (InVirtuo 4.0: Experimental Research in Virtual Environments), and by the state of North Rhine-Westphalia as part of the Excellency Start-up Center.NRW (U-BO-GROW) under grant number 03ESCNW18B.

References

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5, 7

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

[3] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, 2024. 2

[4] Philip A Chou, Tom Lookabaugh, and Robert M Gray. Entropy-constrained vector quantization. *IEEE Trans. Acoust., Speech, Signal Process.*, 37(1):31–42, 1989. 1, 3, 5

[5] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. In *Proc. SIGGRAPH*, 2024. 2

[6] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps, 2024. 2

[7] Qiankun Gao, Jiarui Meng, Chengxiang Wen, Jie Chen, and Jian Zhang. Hicom: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2

[8] Sharath Girish, Tianye Li, Amrita Mazumdar, Abhinav Shrivastava, David Luebke, and Shalini De Mello. QUEEN: QUantized efficient ENcoding for streaming free-viewpoint videos. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[9] Cheng-Yuan Ho, He-Bi Yang, Jui-Chiu Chiang, Yu-Lun Liu, and Wen-Hsiao Peng. Ted-4dgs: Temporally activated and embedding-based deformation for 4dgs compression, 2025. 1, 2

[10] Qiang Hu, Zihan Zheng, Houqiang Zhong, Sihua Fu, Li Song, Xiaoyun Zhang, Guangtao Zhai, and Yanfeng Wang. 4dgc: Rate-aware 4d gaussian compression for efficient streamable free-viewpoint video. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 875–885, 2025. 1, 2

[11] He Huang, Qi Yang, Mufan Liu, Yiling Xu, and Zhu Li. Adc-gs: Anchor-driven deformable and compressed gaussian splatting for dynamic scene reconstruction. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 1179–1187. International Joint Conferences on Artificial Intelligence Organization, 2025. Main Track. 2

[12] ITU-T. Recommendation G.711: Pulse code modulation (PCM) of voice frequencies, 1988. 3

[13] Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. A compact dynamic 3d gaussian representation for real-time dynamic view synthesis. In *ECCV*, 2024. 1

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), 2023. 1, 2

[15] Hyeongmin Lee and Kyungjune Baek. Temporal smoothness-aware rate-distortion optimized 4d gaussian splatting. In *Advances in Neural Information Processing Systems*, 2025. 1, 2, 5, 6, 7

[16] Junoh Lee, ChangYeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. In *Proceedings of the Neural Information Processing Systems*, 2024. 1, 2, 5, 7

[17] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *CVPR*, 2024. 2

[18] Minseo Lee, Byeonghyeon Lee, Lucas Yunkyu Lee, Eunsoo Lee, Sangmin Kim, Seunghyeon Song, Joo Chan Lee, Jong Hwan Ko, Jaesik Park, and Eunbyung Park. Optimized minimal 4d gaussian splatting, 2025. 2

[19] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video, 2022. 2, 5, 6, 7

[20] Lei Liu, Zhenghao Chen, Wei Jiang, Wei Wang, and Dong Xu. Hemgs: A hybrid entropy model for 3d gaussian splatting data compression, 2025. 2

[21] Mufan Liu, Qi Yang, He Huang, Wenjie Huang, Zhenlong Yuan, Zhu Li, and Yiling Xu. Light4gs: Lightweight compact 4d gaussian splatting generation via context model. *arXiv preprint arXiv:2503.13948*, 2025. 2

[22] Xiangrui Liu, Xinju Wu, Pingping Zhang, Shiqi Wang, Zhu Li, and Sam Kwong. Compgs: Efficient 3d scene representation via compressed gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024. 2

[23] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[25] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *CVPR*, 2024. 2

[26] Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, and Valerie Allié. Dataset and pipeline for multi-view light-field video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1743–1753, 2017. 5, 6, 8

[27] Sara Fridovich-Keil and Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 5, 7

[28] Bernard Smith. Instantaneous companding of quantized signals. *Bell System Technical Journal*, 36(3):653–709, 1957. 3

[29] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 5, 7

[30] Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 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 (CVPR)*, pages 20675–20685, 2024. 5, 7

[31] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 3

[32] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19649–19659, 2023. 5, 7

[33] Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex C. Kot, and Bihan Wen. ContextGS : Compact 3d gaussian splatting with anchor level context model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[34] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits and Systems for Video Technology*, 13(7):560–576, 2003. 3

[35] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 1, 2, 5, 7

[36] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 2

[37] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 5, 7

[38] Yu-Ting Zhan, Cheng-Yuan Ho, Hebi Yang, Yi-Hsin Chen, Jui Chiu Chiang, Yu-Lun Liu, and Wen-Hsiao Peng. CAT-3DGS: A context-adaptive triplane approach to rate-distortion-optimized 3DGS compression. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, 2025. 2

[39] Wenkang Zhang, Yan Zhao, Qiang Wang, Zhixin Xu, Li Song, and Zhengxue Cheng. D-fcgs: Feedforward compression of dynamic gaussian splatting for free-viewpoint videos, 2025. 2

[40] Xinjie Zhang, Zhening Liu, Yifan Zhang, Xingtong Ge, Dailan He, Tongda Xu, Yan Wang, Zehong Lin, Shuicheng Yan, and Jun Zhang. Mega: Memory-efficient 4d gaussian splatting for dynamic scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27828–27838, 2025. 1, 2