Fine Granular Scalable Gaussian Splatting Coding for Real-Time Immersive Education

Jiaqi Zou[1], Lingyu Shi[2], Shuzheng Xu[1], Jintao Wang[1], Geert van der Auwera[3], Zhu Li[4]
[1]Department of Electronic Engineering, Tsinghua University, Beijing, China
[2]Independent Contributor
[3]Qualcomm Inc, San Diego, USA
[4]School of Science and Engineering, University of Missouri-Kansas City, Kansas, USA
Email: jqzou@mail.tsinghua.edu.cn,lingyu_shi@163.com, {xusz, wangjintao}@tsinghua.edu.cn, geertv@qti.qualcomm.com, zhu.li@ieee.org

***Abstract*—Immersive education, particularly in experimental teaching, requires students to observe materials from arbitrary viewpoints for a complete understanding. Traditional 2D videos is not enough to meet this requirements. Transmitting dynamic 3D models in real-time is essential for this interactive experience but demands a low-latency streaming solution. To address this challenge, we propose a fine granular scalable coding scheme for Gaussian Splatting (GS) tailored for real-time immersive communication. Our framework creates a scalable GS adaptation sets, enabling a client player to leverage view-dependent quality scalability for minimal latency during streaming and user interaction. Furthermore, to achive high efficiency compression, we introduce a cross-quality layer prediction method whose residuals exhibit high residual correlation across both color and geometry features of GS. We effectively exploit this by using Vector Quantization (VQ), which capitalizes on the joint statistics of Spherical Harmonic (SH) coefficients rather than their marginal distribution, leading to significant Rate-Distortion (R-D) coding gains. A simple GS volume-transparency product based sorting scheme gives scalablility of the representation within the same quality layer. Finally, we integrate our scalable GS representation with DASH's sub-representation scheme to achieve very low-latency 3D visual communication and interaction. Our experiment demonstrates competitive results which shows great potential for high-quality, real-time remote immersive education.**

***Index Terms*—Gaussian Splatting, 3D, scalable compression, DASH, Arithmetic Coding**

I. Introduction

The rapid evolution of digital technology is profoundly transforming the educational landscape, fostering a paradigm shift towards remote and interactive learning. This is particularly critical in experimental courses where a deep, intuitive understanding highly depends on the ability to observe from arbitrary viewpoints. Traditional educational media, primarily 2D videos and images, cannot meet the need. They present a fixed, passive perspective that lacks critical spatial relationships and three-dimensional (3D) dynamics, failing to provide the comprehensive understanding that hands-on laboratory work offers. On contrast, immersive education allows students to interact with dynamic 3D models of experimental setups from any viewpoint, effectively narrowing the gap between remote learning and in-person observation.

To realize this vision of immersive remote education, the real-time transmission of high-fidelity, interactive 3D content is paramount. However, this requirement introduces a significant technical bottleneck: the fundamental trade-off between latency, bandwidth, and visual quality. High-quality 3D assets are inherently data-intensive, and transmitting them with imperceptibly low latency over potentially unpredictable network conditions is a formidable challenge.

Recent breakthrough in 3D visual content capture and representation in 3D Gaussian Splatting (GS) [1] has demonstrated remarkable rendering quality and speed. It has an explicitly interpretable form of representing 3D content, as trainable features in color, geometry and opacity. This representation offers a highly parallelizable rendering pipeline and can capture view-dependent appearance with high photorealism. As a result, GS has attracted significant attention from both academia and industry, and new working items have been initiated within MPEG to investigate compression tools and potential standardization for GS-like data formats [2], [3]. However the existing solutions like [4]–[8] though offering good compression efficiency are too complex in computation for real time streaming applications. This limits their practical usefulness.

In this paper, we propose a comprehensive *scalable coding and streaming framework* for Gaussian Splatting that explicitly addresses these limitations. At the representation level, we construct multiple quality layers by leveraging the natural evolution of GS training and generate a quality scalable bitstream by introducing an inter-quality layer prediction coding [9], [10] and residual Vector Quantization (VQ) scheme. The residual from combined inter-layer prediction and VQ codebooks are therefore encoded with feature-dependent quantization and the combined prediction indices are coded losslessly. Furthermore we introduce a *Gaussian importance sorting* mechanism that ranks Gaussians according to their contribution to the rendered quanlity, enabling progressive fine granularity of each quality layer's representation. This is mated to the DASH-like adaptation set for low latency streaming [11]. This is well-suited for *real-time immersive applications*, such as interactive education.

The contributions of this paper are summarized as follows:

- Combined prediction from inter-quality layer prediction

979-8-3315-7111-5/26/$31.00 ©2026 IEEE

[10] and GS features codebook based residual prediction, that exploits inter-component correlations in both geometry and color features. In particular, we design a hybrid KNN-based prediction module and employ Vector Quantization (VQ) to efficiently encode SH_{DC} residuals by leveraging their low-dimensional joint statistics.

- We propose a **Gaussian importance sorting** strategy that sorts Gaussians based on physically interpretable criteria, i.e, the product of the GS volume and transparency, enabling progressive, fine granular rendering quality reconstruction of the scene with monotonic quality improvement.
- We integrate the above elements into a **DASH-like sub-representation framework**, where each operating point corresponds to a sorted Gaussian prefix and a refinement mode, allowing practical receiver-driven adaptation with low latency and high visual fidelity, utilizing non-TCP transport like WebRTC [12].

The details are presented in the following sections.

II. Scalable GS coding and Fine Granularity Packetization

This section introduces a complete scalable representation framework for 3D Gaussian Splatting (3DGS). The goal is to enable efficient compression, progressive reconstruction, and network-adaptive quality selection, while preserving the rendering characteristics of the original scene.

A. Gaussian Splatting quality layer generation and inter-layer prediction

In our previous work [10], we adopt a structured multi-quality layering strategy that reshapes the temporal evolution of 3DGS training into a scalable representation suitable for low-latency transmission and progressive reconstruction. During GS optimization, the rendering PSNR increases gradually and forms several stable plateaus, each corresponding to a meaningful quality stage, defined as (1) Base layer (L_0): an early iteration capturing the dominant geometry and primary color components. (2) Enhancement layers (L_1 and L_2): later iterations introducing finer geometric details and radiance refinements. Our goal is to reorganize these training stages into a predictable, compressible, and progressively decodable scalable structure.

We treat enhancement layers as a refinement sequence built upon L_0 and employ cross-layer residual prediction to exploit the strong inter-layer correlation. For each Gaussian in L_1 or L_2, we first establish local correspondences via KNN search in the 3D position domain. The k neighbors from L_0 serve as candidate predictors for both geometry and color features.

We adopt a lightweight dual-predictor structure: the bilateral feature prediction and nearest-feature prediction. After that, we compare the residuals of the two predictors and select the better one. The corresponding prediction mode is recorded for each Gaussian (0 for bilateral, $1, \cdots, k$ for nearest-feature). Because the mode usage is highly sparse, its entropy is typically around 2–2.5 bits per Gaussian, which is far below the theoretical bound of $log_2(K+1)$.

Experimental results in our previous work indicate that refining SH_{DC} alone can recover the majority of the enhancement-layer quality, with a typical PSNR loss within 0.03 dB compared to full refinement. This observation is the key motivation for performing vector quantization only on SH_{DC} residuals in this work.

B. Prediction Residual Coding with Vector Quantization

In the previous section, we applied KNN-based bilateral prediction and nearest-neighbor prediction to the SH_{DC} , opacity, scale, and rotation parameters of each Gaussian, and obtained the corresponding residuals. In this work, however, we only perform vector quantization (VQ) on the residuals of the SH_{DC} component between the enhancement layer and the base layer. This design choice is motivated by a combination of factors, including the perceptual contribution of SH_{DC} , its statistical behavior across optimization iterations, and the overall impact on bitrate efficiency.

Both bilateral prediction and nearest-neighbor prediction rely on the assumption that SH_{DC} exhibits strong local coherence in feature space—Gaussians with similar attributes tend to have similar low-frequency color components. Owing to the low dimensionality and smooth variation of SH_{DC} , this assumption holds particularly well, making its residuals inherently more compressible.

Moreover, SH_{DC} approximates the mean radiance (average color and brightness) of each Gaussian, which dominates the rendered image's MSE and PSNR. Empirically, replacing only the SH_{DC} values with those from a later training iteration already recovers most of the enhancement-layer rendering quality. In contrast, differences in geometry and higher-order SH coefficients mostly affect local details or view-dependent variations, contributing far less to global PSNR improvement. Therefore, focusing VQ resources on SH_{DC} residuals provides the best trade-off between rate and distortion.

Let $\mathbf{f}_{DC}^{(0)}$ and $\mathbf{f}_{DC}^{(1)}$ denote as the SH_{DC} coefficients of the base and enhancement layers. Given the optimal predictor output $\hat{f}_{DC}$, selected between bilateral and nearest-neighbor prediction based on residual variance, the SH_{DC} residual of each Gaussian is computed as

$$\mathbf{r}_i = \mathbf{f}_{DC,i}^{(1)} - \hat{f}_{DC}, \tag{1}$$

After obtaining the residual vectors $\mathbf{r}_i$, to better compress the residuals from inter-quality layer prediction, and exploit the joint statistics of the residuals, especially among the color information, i.e, SH_{DC} information, a Vector Quantization (VQ) is introduced. A code book of SH residuals is computed offline and available to both encoder and decoder. The code book size is R-D optimized to reflect the balance between code book index signaling cost and a prediction gain. This can be viewed as the second prediction coding scheme, on top of the inter-layer prediction.

To construct a compact codebook tailored for the current enhancement layer, we train a K-entry VQ dictionary using the

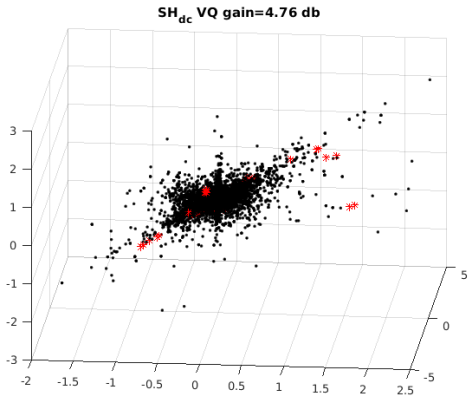

Fig. 1: Visualization of the SH codebook

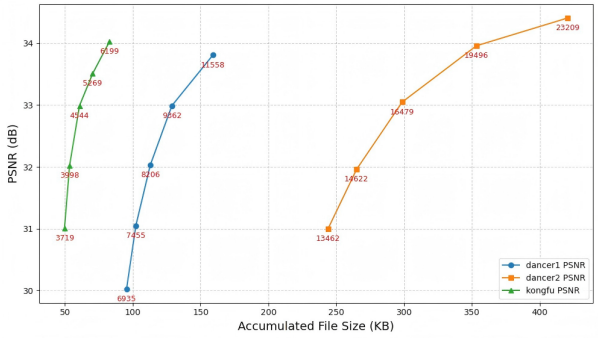

Fig. 2: Progressive Rendering Quality with Importance-Based Gaussian Sorting

standard Lloyd–Max (k-means) procedure.Let $\{\mathbf{r}_i\}_{i=1}^{N}$ denote the residual vectors of all Gaussians. The objective of k-means is to jointly determine the cluster assignments $\{S_k\}_{k=1}^{K}$ and the codewords $\{c_k\}_{k=1}^{K}$ by minimizing the within-cluster sum of squared errors (WCSS)

$$\min_{\{S_k\},\{c_k\}} J = \min_{\{S_k\},\{c_k\}} \sum_{k=1}^{K} \sum_{i \in S_k} \|\mathbf{r}_i - \mathbf{c}_k\|_2^2 . \tag{2}$$

The above optimization is solved via iterative alternating minimization.

Given codewords $\{c_k\}$, the assignment step assigns each residual vector to the nearest codeword

$$\mathrm{idx}_{\mathrm{vq}}(\mathbf{i}) = \arg \min_{k \in \{1,\ldots,K\}} \|\mathbf{r_i} - \mathbf{c}_k\|_2^2 , \tag{3}$$

which yields the cluster sets S_k. Then, given the assignments $\{S_k\}$, the update step recomputes each codeword as the centroid of its assigned samples, denoted as

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{i \in S_k} \mathbf{r}_i \quad k = 1, \ldots, K. \tag{4}$$

These two steps are repeated until convergence or until a maximum iteration limit is reached, yielding a codebook that is best representing the residual distribution. Since each residual vector is only 3-dimensional, the resulting VQ codebook is extremely compact, and its transmission overhead is negligible compared with the total number of Gaussians that will need to be represented.

An example of SH_{DC} plots from AVS dataset "dancer", along with the offline trained VQ codebook of size $k = 8$, is illustrated above. Once the codebook is obtained, each residual vector is encoded using its VQ index idxvq(i), requiring $log_2 K$ bits per Gaussian before entropy coding. During decoding, the residual is reconstructed by table lookup of the corresponding codeword followed by dequantization, and then added back to the predictor output to recover the enhancement-layer SH_{DC} coefficients.

C. Scalable GS stream with Gaussian Sorting

Beyond compressing the feature space of Gaussians, our scalable framework also aims to support progressive reconstruction, i.e., reconstructing a reasonable approximation of the scene using only a subset of the most important Gaussians, and then refining the rendering quality as more Gaussians are added. To this end, we design a simple yet effective Gaussian importance metric based on opacity and scale, and use it to derive a global sorting of all primitives.

Not all Gaussian primitives contribute equally to the rendered image. Empirically, a Gaussian's opacity determines how much it attenuates or emits light along a ray, while its scale governs the screen-space footprint; together they fix the number of affected pixels and the overall energy contribution. Exploiting this observation, we derive a lightweight GS importance metric as

$$\mathbf{w}_i = \alpha_i \cdot u_i, \tag{5}$$

where

$$\begin{cases} \alpha_i = \sigma(o_i) \\ V_i = \exp(s_{i,x} + s_{i,y} + s_{i,z}) \end{cases} \tag{6}$$

Here $\alpha_i \in (0, 1)$ is the effective opacity obtained from the network output o_i via a logistic sigmoid, and V_i is the volumetric footprint derived from the log-scale vector $s_i \in \mathbb{R}^3$.

A single descending sort on w_i re-orders the entire layer; consecutive prefixes of this ordered list are then written to separate DASH segments. The client downloads the first prefix first, obtaining an immediate low-bitrate preview, and simply appends later prefixes to refine quality without any scene re-initialization.

III. Simulation Results

To validate the effectiveness of the proposed scalable Gaussian representation and adaptive streaming framework, we conduct a series of quantitative evaluations across multiple sequences and coding configurations. Our experiments aim to demonstrate both the compression efficiency and the low decoding complexity achieved by the proposed solution.

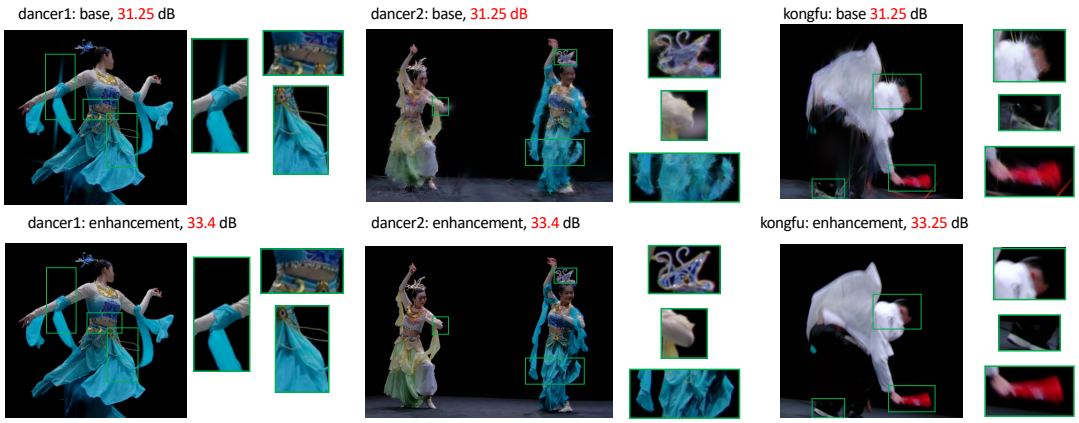

Fig. 3: Subjective quality illustration

A. GS quality layers creation

We first perform a controlled study using the test sequences provided by the AVS VRU [13] working group. Three representative scenes are selected:Dance_Dunhuang_Single_1080 (dancer1), Dance_Dunhuang_Pair_1080 (dancer2), and Kungfu_Fan_Single_1080 (kongfu).These sequences exhibit diverse motion patterns, varying spatial density, and rich color variations, making them well-suited for benchmarking GS compression. Representative frames of these sequences are shown in Fig. 3.

For each sequence, we train a 3DGS model using a multi-view image set and record the rendering PSNR at different optimization iterations. To achieve a balance between quality separation and bitrate per Gaussian, each sequence is encoded into one base layer and two enhancement layers. The base layer provides a coarse but complete reconstruction of the scene, while the enhancement layers progressively refine geometric structure and photometric fidelity.

Taking dancer2 as an example, we select the following training checkpoints: Base Layer: iteration = 2000 (PSNR $\approx$ 32.86 dB) Enhancement Layer 1: iteration = 4000 (PSNR $\approx$ 33.40 dB) Enhancement Layer 2: iteration = 16000 (PSNR $\approx$ 34.40 dB)

During streaming, instead of employing traditional multi-description coding (MDC), we utilize our inter-quality-layer prediction mechanism to generate highly compact enhancement-layer bitstreams, resulting in significantly improved rate–distortion efficiency.

B. R-D performance and complexity analysis

To assess the effectiveness of the proposed scalable representation, we evaluate the RD performance of the proposed framework against two baselines:1)Scalable GS without VQ: Our previous framework using only inter-layer prediction and scalar quantization.2)GPCC MDC: Geometry-based point cloud compression using multi-description coding, serving as a conventional baseline.

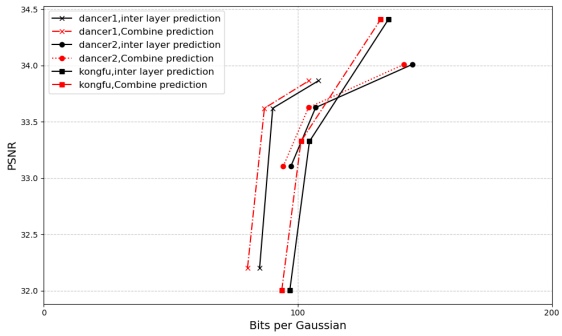

Fig. 4: RD Performance of Proposed Combine Prediction vs. Inter-Layer Prediction

The resulting rate–distortion characteristics are shown in Fig. 4 and Fig. 5. Across all sequences, the benefits of cross-layer prediction are evident: at comparable PSNR levels, the scalable-coded bitstreams reduce bitrate by more than an order of magnitude relative to the GPCC MDC baseline. On the 'dancer1' sequence, for example, achieving approximately 33.5 dB requires more than 850 bits per Gaussian with MDC, whereas our scalable-coded variant requires fewer than 100 bits per Gaussian, representing over 85% bitrate savings. Incorporating the proposed VQ module for SH_{DC} residuals ("combine" curves) further reduces the enhancement-layer bitrate, leveraging the compact, highly correlated structure of the low-frequency color residuals. The VQ-enhanced bitstream consistently achieves the best performance among all tested variants, confirming that SH_{DC} residual quantization is a dominant contributor to overall quality refinement.

We also analyze decoding complexity using two entropy coders: PAQ8L and LZMA. As shown in Table I, PAQ8L achieves higher compression but incurs 100 ms per thousand Gaussians, making it unsuitable for real-time rendering. In contrast, LZMA decoding takes only 1.5 ms per thousand Gaussians, with a modest 9.8% bitrate overhead. This makes LZMA a practical choice for client-side decoding in streaming scenarios.

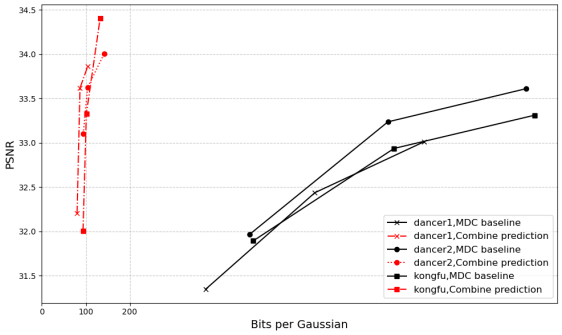

Fig. 5: GS quality layer coding R-D performance compared with baseline

TABLE I
Decoding Time Complexity

Seq Name		Decoding Time(ms/kgs)	
		PAQ8L	LZMA
dancer1	Base Layer	113.1	1.52
	ENH Layer 1	112.7	1.51
	ENH Layer 2	104.9	1.49
dancer2	Base Layer	114.0	1.96
	ENH Layer 1	111.9	1.94
	ENH Layer 2	98.1	1.40
kongfu	Base Layer	111.5	1.64
	ENH Layer 1	108.5	1.61
	ENH Layer 2	102.2	1.53

C. MPEG Subrepresentation for Low Latency Streaming

To evaluate the effectiveness of the proposed scalable Gaussian representation under a practical streaming scenario, we simulate a receiver-driven adaptive transmission setting following the design principles of DASH-like multi-representation switching.Each subrepresentation corresponds to a combination of a Gaussian prefix and a feature refinement mode, and the client dynamically selects the appropriate subrepresentation based on available network bandwidth. The importance-based Gaussian ordering ensures that even small prefixes produce coherent scene reconstructions, allowing rapid session startup. As bandwidth increases, the client progressively incorporates more Gaussians, yielding smooth quality improvements without visible popping or the need to reinitialize the scene. Fig. 2 illustrates the progression of reconstruction quality as more Gaussians are transmitted, with red annotations indicating the cumulative count. The progressively additive structure of both geometry and feature refinements enables stable transitions under fluctuating bandwidth, while the compactness of the enhancement layers ensures efficient use of network resources.

IV. Conclusion

In this paper, we propose a novel scalable coding framework for Gaussian Splatting, leveraging inter-quality layer prediction and residual coding to enable efficient compression and adaptive streaming. The solution is designed to be lightweight, with the ability to seamlessly integrate into MPEG DASH-like streaming frameworks, facilitating low-latency, high-quality 3D visual communication. To our knowledge, this is the first approach that effectively makes GS streaming both feasible and highly efficient. Future work will focus on refining the algorithm by deploying it on a real-world DASH streaming testbed, enabling further validation of its compression performance and streaming efficiency in dynamic, interactive scenarios.

Acknowledgment

The work is supported in part by a Tsinghua University Laboratory Innovation Fundings, and a Qualcomm Gift to UMKC.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[2] Y.-C. Sun, Y. Shi, C.-T. Lee, M. Zhu, W. T. Ooi, Y. Liu, C.-Y. Huang, and C.-H. Hsu, "Lts: A dash streaming system for dynamic multi-layer 3d gaussian splatting scenes," in *Proceedings of the 16th ACM Multimedia Systems Conference*, ser. MMSys '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 136–147. [Online]. Available: https://doi.org/10.1145/3712676.3714445

[3] H. Xu, X. Wu, and X. Zhang, "3dgs compression with sparsity-guided hierarchical transform coding," 2025. [Online]. Available: https://arxiv.org/abs/2505.22908

[4] X. Liu, X. Wu, P. Zhang, S. Wang, Z. Li, and S. Kwong, "Compgs: Efficient 3d scene representation via compressed gaussian splatting," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 2936–2944.

[5] Y. Chen, Q. Wu, W. Lin, M. Harandi, and J. Cai, "Hac: Hash-grid assisted context for 3d gaussian splatting compression," in *European Conference on Computer Vision*. Springer, 2024, pp. 422–438.

[6] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 20 310–20 320.

[7] Y. Shi, "3d gaussian-based immersive media streaming in networked extended reality," in *Proceedings of the 16th ACM Multimedia Systems Conference*, ser. MMSys '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 356–360. [Online]. Available: https://doi.org/10.1145/3712676.3719673

[8] J. Zhang, T. Chen, H. Zhu, D. Wang, D. Ding, and Z. Ma, "Compressing 3d gaussian splatting via a generalizable neural coder," in *2024 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2024, pp. 1–5.

[9] M. A. Talha, Akhtar, Z. Li, and G. Awuera, "Intergs: Inter-predictive coding of gaussian splatting sequences," in *IEEE Visual Communication & Image Processing Conf (VCIP), Klugenfurt, Austria*, 2025.

[10] L. Shi, J. Zou, S. Sun, G. Auwera, and Z. Li, "Low latency immersive visual communication with scalable gaussian splatting coding," in *IEEE Multimedia Signal Processing Workshop*. IEEE Signal Processing Society, 2026.

[11] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.

[12] S. Zhao, Z. Li, and D. Medhi, "Low delay MPEG DASH streaming over the webrtc data channel," in *2016 IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops 2016, Seattle, WA, USA, July 11-15, 2016*. IEEE Computer Society, 2016, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICMEW.2016.7574765

[13] X. Zheng, L. Liao, X. Li, J. Jiao, R. Wang, F. Gao, S. Wang, and R. Wang, "Pku-dymvhumans: A multi-view video benchmark for high-fidelity dynamic human modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 530–22 540.