Eliminating Geometric Representation Redundancy for 3D Gaussian Splat Coding

Shanchuan Liu*, Zhiwei Zhu*, Sicheng Li, Yiyi Liao, and Lu Yu†
Zhejiang University, Hangzhou, China
{shanchuanliu, zhuzhiwei21, jasonlisicheng, yiyi.liao, yul}@zju.edu.cn

***Abstract*—3D Gaussian Splatting (3DGS) enables photorealistic, real-time rendering, yet its native representation imposes significant storage and transmission overhead, hindering widespread deployment. Most existing approaches focus on reducing data volume or improving the efficiency of lossy coding. However, they overlook the high data entropy caused by inherent representation ambiguity, where multiple geometric attribute values can define the same geometry. To address this, we introduce a lightweight, plug-and-play preprocessing method that lowers raw data entropy by canonicalizing scale and quaternion attributes. Specifically, our method first resolves geometric representation ambiguity via a deterministic regularization rule that enforces a unique representation; reduces dimensionality by converting 4D quaternions to minimal 3D Rodrigues parameters; and addresses numerical redundancy by clamping perceptually insignificant scale values. Our method is a generic, plug-and-play preprocessing module, fully orthogonal to existing 3DGS compression schemes, and effectively boosts their coding efficiency. When combined with a baseline compression pipeline, it yields an average BD-Rate reduction of 15.81% compared to the same pipeline without our preprocessing.**

***Index Terms*—3D Gaussian Splatting, Compression, Preprocessing.**

I. Introduction

3D Gaussian Splatting (3DGS) has established a new benchmark in novel view synthesis, achieving photorealistic rendering at real-time rates [1]. This performance establishes 3DGS as a foundational technology for immersive applications such as virtual and augmented reality (VR/AR) and digital twins, but the high storage and bandwidth requirements of its native representation pose a critical challenge for practical deployment. Consequently, the development of effective compression and coding methods is essential for its practical deployment. [2]–[5]

Extensive research has been devoted to developing intrinsically compact representations of 3D scenes. A primary strategy focuses on reducing the raw data volume by altering the scene's fundamental composition. This is achieved most directly by reducing the number of primitives, either through post-training pruning or merging [6]–[8], by refining adaptive density control within the training loop [9], [10], or by employing hybrid implicit models that generate a full scene from a small set of features or anchors [11]–[14]. Complementing these efforts, another line of work simplifies the attributes of each primitive to reduce their footprint. This includes reducing the dimensionality of 3D Gaussian attributes, for instance by converting quaternions to Euler angles [15], or simplifying appearance models by reducing the order of Spherical Harmonics [16]. Another approach is learning a compressed representation to replace raw attributes through vector quantization or compact feature mappings [17]–[19]. Other methods focus on optimizing the statistical properties of Gaussian splatting data by integrating entropy-aware training objectives. For example, entropy constraints are imposed on the distribution of quaternion attributes during training [20], or on feature planes when training a tri-plane representation [19].

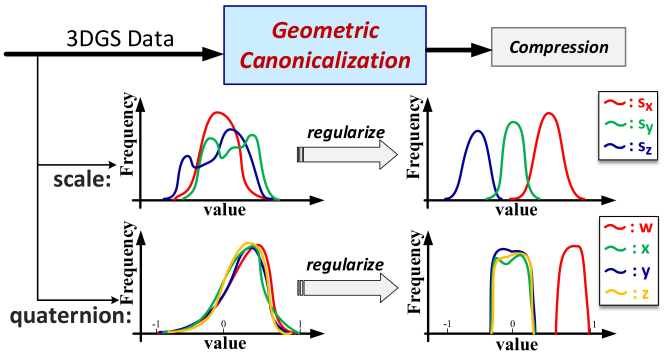

Fig. 1: Our training-free Geometric Canonicalization pipeline. It first regularizes the data distributions of scale and quaternion attributes, reducing their dynamic range to enhance compressibility.

While these strategies are effective, they either require costly retraining or overlook a fundamental source of entropy: inherent redundancies within the geometric parameterization of the standard Gaussian primitive. Chief among these is the representation ambiguity stemming from the rotational symmetry of ellipsoids, which allows a single Gaussian to be described by multiple equivalent combinations of scale and rotation values. This ambiguity, compounded by other redundancies, leads to disordered data distributions that challenge effective compression.

This paper introduces Geometric Canonicalization, a lightweight, training-free method that systematically eliminates these inherent geometric representation redundancies. By applying a deterministic transformation to any pre-trained model, as shown in Fig. 1, our method produces a canonical

*These authors contributed equally to this work. †Corresponding author. The authors are with the College of Information Science and Electronic Engineering, and also with the Zhejiang Provincial Key Laboratory of Information Processing, Communication and Networking (IPCAN), Hangzhou 310027, China.

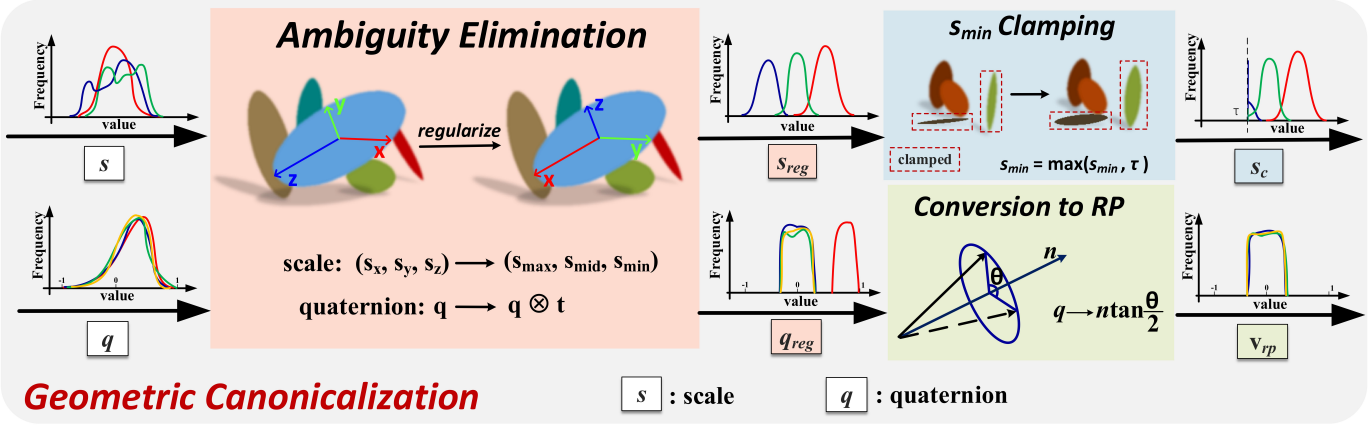

Fig. 2: The illustration of our three-stage canonicalization pipeline. Our method converts raw geometric attributes into a compact canonical form through three steps: resolving many-to-one ambiguities for unique Gaussian representation, reducing 4D quaternions to minimal 3D vectors, and clamping small scale values below a perceptual threshold.

representation that is significantly more compressible without altering the rendered scene. Specifically, our work **1) resolves representation ambiguity through a deterministic regularization rule, creating structured data distributions; 2) reduces dimensionality by converting 4D quaternions to minimal 3D Rodrigues parameters; and 3) addresses numerical redundancy by clamping imperceptible scale values.** As a plug-and-play module, our work enhances the performance of existing and future codecs without costly retraining.

II. Proposed Method

A. Preliminary

3D Gaussian Splatting. A 3DGS scene is represented by a set of anisotropic 3D Gaussians, typically visualized as oriented ellipsoids. Each primitive is parameterized by its color, defined by opacity (α) and Spherical Harmonics (SH) coefficients, and its geometry. The geometry is specified by a position $\boldsymbol{\mu}$, a scaling vector $\mathbf{s} \in \mathbb{R}^3$, and a unit quaternion $\mathbf{q} \in \mathbb{S}^3$. The latter two attributes jointly define the ellipsoid's shape and orientation via a 3D covariance matrix $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \mathbf{RSS}^T\mathbf{R}^T \tag{1}$$

where $\mathbf{R}$ (derived from $\mathbf{q}$) is a rotation matrix defining the orientation of the ellipsoid's principal axes, and $\mathbf{S} = \text{diag}(\mathbf{s})$ is a diagonal matrix defining the scaling along these axes.

B. Geometric Representation Ambiguity

Geometric representation ambiguity arises because the same rendered Gaussian can be described by multiple equivalent combinations of a scale vector $\mathbf{s}$ and a rotation quaternion $\mathbf{q}$. This issue stems from the physical symmetry of an ellipsoid; its appearance is invariant to certain transformations of its local coordinate system, specifically axis permutations and flips. Formally, for any given Gaussian, the set $\mathcal{E}$ of all equivalent parameter pairs $(\mathbf{s}', \mathbf{q}')$ is defined by transformations that leave the covariance matrix unchanged:

$$\mathcal{E} = \{(\mathbf{s}', \mathbf{q}') \mid \mathbf{R}'\mathbf{S}'(\mathbf{S}')^T(\mathbf{R}')^T = \mathbf{RSS}^T\mathbf{R}^T\} \tag{2}$$

Any such transformation on the local axes can be represented by an orthogonal matrix $\mathbf{T}$, yielding a new rotation $\mathbf{R}' = \mathbf{RT}$. In the quaternion domain, this corresponds to the product $\mathbf{q}' = \mathbf{q} \otimes \mathbf{t}$, where $\mathbf{t}$ is the quaternion for the transformation $\mathbf{T}$. If the transformation swaps axes, the scale vector $\mathbf{s}$ must be permuted accordingly to produce $\mathbf{s}'$.

Given an initial representation $(\mathbf{s}, \mathbf{q})$, where $\mathbf{s} = (s_x, s_y, s_z)$ and $\mathbf{q} = (w, x, y, z)$, the full set of equivalent pairs can be generated:

a) Axis Flips: When only flipping two local axes, the scale vector remains unchanged ($\mathbf{s}' = \mathbf{s}$). These three transformations yield the following equivalent quaternions:

- *Flip Y, Z axes:* $\mathbf{q}' = (-x, w, z, -y)$
- *Flip X, Z axes:* $\mathbf{q}' = (-y, -z, w, x)$
- *Flip X, Y axes:* $\mathbf{q}' = (-z, y, -x, w)$

b) Pairwise Axis Swaps: These transformations correspond to swapping two local axes and flipping the remaining one; the scale values are permuted accordingly.

- For $\mathbf{s}' = (s_y, s_x, s_z)$: $\mathbf{q}' = \frac{1}{\sqrt{2}}(-x-y, w-z, w+z, x-y)$
- For $\mathbf{s}' = (s_x, s_z, s_y)$: $\mathbf{q}' = \frac{1}{\sqrt{2}}(-y-z, y-z, w-x, w+x)$
- For $\mathbf{s}' = (s_z, s_y, s_x)$: $\mathbf{q}' = \frac{1}{\sqrt{2}}(-x-z, w+y, z-x, w-y)$

c) Cyclic Axis Permutations: These transformations correspond to a cyclic permutation of all three axes.

- For $\mathbf{s}' = (s_y, s_z, s_x)$: $\mathbf{q}' = \frac{1}{2}(w-x-y-z, w+x+y-z, w-x+y+z, w+x-y+z)$
- For $\mathbf{s}' = (s_z, s_x, s_y)$: $\mathbf{q}' = \frac{1}{2}(w+x+y+z, -w+x-y+z, -w+x+y-z, -w-x+y+z)$

In total, there are 6 unique scale permutations. Each combines with 4 orientation-preserving symmetries (including identity), yielding $6 \times 4 = 24$ unique pairs of $(\mathbf{s}', \mathbf{R}')$. Accounting for the twofold sign ambiguity of quaternions, where both $\mathbf{q}'$ and $-\mathbf{q}'$ represent the same rotation [21], the full set $\mathcal{E}$ has a cardinality of 48. This high degree of redundancy creates a disordered data distribution detrimental to compression.

C. The Geometric Canonicalization Method

To resolve the inherent geometric representation redundancies, we introduce a three-stage method that transforms the raw

attributes $(\mathbf{s}, \mathbf{q})$ into a canonical representation that is compact and perceptually optimized, as illustrated in Fig. 2.

1) Ambiguity Elimination via Regularization: This first stage regularizes the parameter space by establishing a unique representation for each Gaussian. We select a **regularized pair** $(\mathbf{s}_{reg}, \mathbf{q}_{reg})$ from the set of 48 equivalent representations $\mathcal{E}$ using a two-level selection rule. First, we enforce a descending order on the scale components:

$$\mathcal{E}^* = \{(\mathbf{s}', \mathbf{q}') \in \mathcal{E} \mid s'_x \geq s'_y \geq s'_z\} \tag{3}$$

Second, from the resulting subset, we select the representation that maximizes the quaternion's scalar component:

$$(\mathbf{s}_{reg}, \mathbf{q}_{reg}) = \underset{(\mathbf{s}', \mathbf{q}') \in \mathcal{E}^*}{\arg\max} \{w'\} \tag{4}$$

This regularization process yields low-entropy distributions ideal for compression by systematically reducing the value range of both scale and rotation attributes, enhancing their suitability for fixed-point quantization. Sorting the scales $(s'_x \geq s'_y \geq s'_z)$ decomposes the original data into three distinct, more concentrated component distributions. Concurrently, maximizing the quaternion's scalar component w' collapses the rotation distribution into a sharp peak near the identity. Both transformations constrict the dynamic range of the attributes, which not only improves quantization precision but also enables attribute-aware coding strategies, such as allocating fewer bits to the smallest scale component.

2) Conversion to Rodrigues Parameters: **Quaternion Representation.** A unit quaternion $\mathbf{q} = (w, x, y, z)$, where $w^2 + x^2 + y^2 + z^2 = 1$, provides a compact, singularity-free representation for 3D rotations. For a rotation of angle θ around a unit axis $\mathbf{n} = (n_x, n_y, n_z)$, the components are given by:

$$w = \cos(\frac{\theta}{2}), \quad (x, y, z) = (n_x, n_y, n_z) \sin(\frac{\theta}{2}) \tag{5}$$

To eliminate dimensional redundancy, we convert this 4D vector into a minimal 3D vector using **Rodrigues Parameters** (RP) [22]:

$$\mathbf{v}_{rp} = \mathbf{n} \tan(\frac{\theta}{2}) = \left(\frac{x_{reg}}{w_{reg}}, \frac{y_{reg}}{w_{reg}}, \frac{z_{reg}}{w_{reg}} \right) \tag{6}$$

The non-linear nature of this conversion (proportional to $\tan(\theta/2)$) beneficially transforms the peaked distribution from the previous stage into a more uniform one. This structure better utilizes the dynamic range of a quantizer, thereby reducing average quantization error. Crucially, the prior maximization of w_{reg} robustly avoids the singularity at $w_{reg} = 0$, ensuring the conversion is always numerically stable.

3) Rendering-Quality-Guided Scale Clamping: The final stage targets numerical redundancy in the scale values. In the 3DGS rendering pipeline, a Gaussian's influence depends on its projected 2D footprint. At typical viewing distances encountered in interactive applications, Gaussians with extremely small scales have a negligible footprint and are effectively invisible. We eliminate this redundancy by clamping the regularized log-scale values to a minimum threshold, τ:

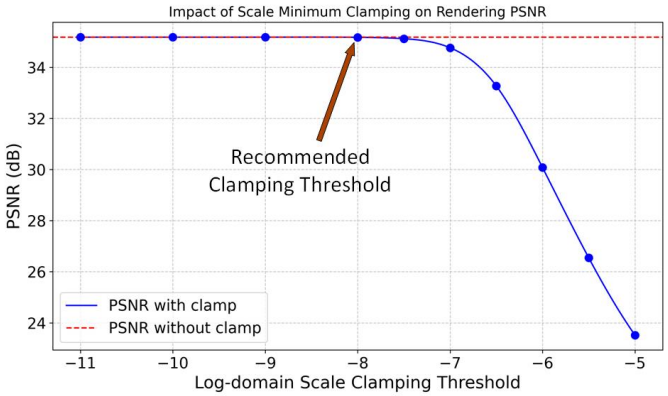

Fig. 3: Relationship between rendered view PSNR and scale minimum clamping threshold.

$$\log(\mathbf{s}_c) = \max(\log(\mathbf{s}_{reg}), \tau) \tag{7}$$

Based on our empirical evaluation (Fig. 3), we select $\tau = -8.0$. This operation truncates the distribution's tail, simultaneously reducing entropy and constricting its dynamic range, which allows a quantizer to allocate its full precision to the perceptually relevant range of values.

The entire three-stage process defines a deterministic mapping, termed the Geometric Canonicalization operator $\mathcal{C}$:

$$\mathcal{C} : (\mathbf{s}, \mathbf{q}) \mapsto (\mathbf{s}_c, \mathbf{v}_{rp}) \tag{8}$$

This resulting pair $(\mathbf{s}_c, \mathbf{v}_{rp})$ constitutes our unique **canonical representation**.

III. Experimental Results

A. Experimental Setup

Anchor. To benchmark our method, we adopt the GSCodec Studio [20] as the anchor, which is a video-based approach: it first quantizes Gaussian attributes with a uniform fixed-point quantizer, then maps them onto 2D grids to create pseudo-images, and compressed with a standard video codec (e.g., HEVC). This architecture enables efficient decoding on consumer devices via widely supported hardware codecs. Our method is integrated as a training-free, plug-and-play preprocessing module that operates on Gaussian geometric attributes before quantization.

Dataset and Evaluation. We conduct experiments on the first frame of three dynamic sequences from the MPEG GSC dataset: Bartender, Breakfast, and Cinema [23]. We measure compression gains using the Bjøntegaard Delta Rate (BD-Rate) [24], calculating the average bitrate savings for the same PSNR.

B. Objective Results

As detailed in Table I for the *Bartender* sequence, our method consistently achieves higher rendering quality (PSNR) while simultaneously reducing file size compared to the anchor. This efficiency translates into significant BD-Rate

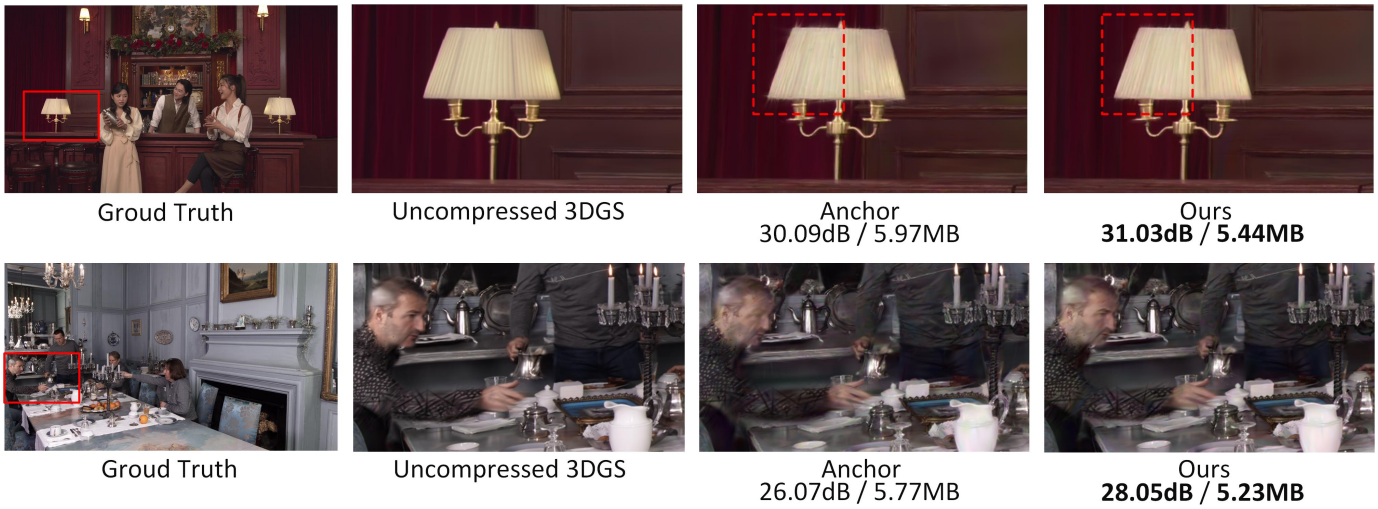

Fig. 4: Subjective comparison under high compression. Our method enhances the compression robustness of geometric attributes, effectively mitigating the severe geometric distortion seen in the baseline under high compression.

reductions across all test scenes, with gains of **-15.57%** on *Bartender*, **-17.14%** on *Breakfast*, and **-14.71%** on *Cinema*, culminating in an average saving of **-15.81%**. These consistent gains validate the successful removal of geometric attribute redundancy.

TABLE I: Quantitative comparison of compression efficiency on the Bartender dataset.

Method	Metric	RP0	RP1	RP2	RP3
Anchor	Size (MB)	23.78	12.30	8.69	5.97
	PSNR (dB)	34.84	33.67	32.21	30.09
Ours	Size (MB)	22.83	11.45	8.00	5.44
	PSNR (dB)	34.90	33.82	32.68	31.03

Ablation Study. To systematically evaluate the contribution of each component, we conducted an ablation study in which we progressively activated every module of our method. The rate-distortion (RD) curves in Fig. 5 validate that each stage provides a distinct and cumulative benefit, aligning with our theoretical analysis in Sec. II-C. The initial regularization (+reg) yields a notable quality improvement by concentrating the attribute distributions. Adding the conversion to Rodrigues Parameters (+reg+RP) provides a clear bitrate reduction due to dimensionality reduction. Finally, scale clamping (+reg+RP+clamp) further improves RD performance, especially in low-bitrate regimes, by focusing quantization precision on the most perceptually relevant scales.

C. Subjective Results

Fig. 4 provides a qualitative comparison at similar bitrates. The anchor exhibits noticeable deformation artifacts, particularly along object edges, which become more pronounced at low bitrates. In contrast, our method delivers superior visual quality at a comparable bitrate. Its canonicalized representation is more robust to quantization, significantly mitigating such artifacts and preserving geometric fidelity.

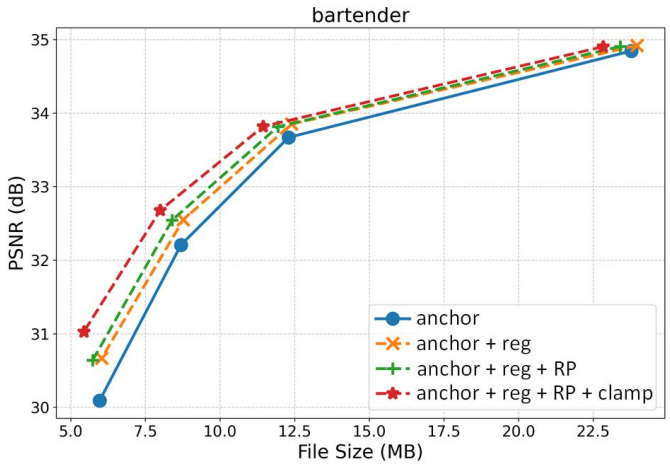

Fig. 5: Ablation study. The curves show the cumulative rate-distortion improvement as each stage of our method is progressively added to the anchor.

IV. Conclusion

This paper introduces Geometric Canonicalization, a training-free method that eliminates inherent geometric redundancies in 3DGS to increase the data's compressibility. Our method first resolves representation ambiguity, and subsequently addresses dimensional and numerical redundancies. This process transforms raw Gaussian attributes into a canonical representation that is significantly more robust for compression. Experimental results validate our approach, showing our lightweight preprocessing step achieves an average BD-Rate saving of -15.81% over the baseline scheme. As a practical, plug-and-play module, our method offers a foundational optimization for efficient 3D Gaussian splat coding.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants U21B2004 and 62571485.

REFERENCES

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics (TOG)*, 2023.

[2] M. T. Bagdasarian, P. Knoll, Y. Li, F. Barthel, A. Hilsmann, P. Eisert, and W. Morgenstern, "3DGS.Zip: A Survey on 3D Gaussian Splatting Compression Methods," in *Computer Graphics Forum*, 2024.

[3] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3D Gaussian Splatting as New Era: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[4] T. Wu, Y.-J. Yuan, L.-X. Zhang, J. Yang, Y.-P. Cao, L.-Q. Yan, and L. Gao, "Recent Advances in 3D Gaussian Splatting," *Computational Visual Media*, 2024.

[5] M. S. Ali, C. Zhang, M. Cagnazzo, G. Valenzise, E. Tartaglione, and S.-H. Bae, "Compression in 3D Gaussian Splatting: A Survey of Methods, Trends, and Future Directions," *arXiv preprint arXiv:2502.19457*, 2025.

[6] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, Z. Wang *et al.*, "LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction and 200+ FPS," *Advances in Neural Information Processing Systems*, 2024.

[7] A. Hanson, A. Tu, V. Singla, M. Jayawardhana, M. Zwicker, and T. Goldstein, "PUP 3D-GS: Principled Uncertainty Pruning for 3D Gaussian Splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[8] C. Wang, S. N. Sridhara, E. Pavez, A. Ortega, and C. Chang, "Adaptive Voxelization for Transform Coding of 3D Gaussian Splatting Data," *arXiv preprint arXiv:2506.00271*, 2025.

[9] S. Kheradmand, D. Rebain, G. Sharma, W. Sun, J. Tseng, H. Isack, A. Kar, A. Tagliasacchi, and K. M. Yi, "3D Gaussian Splatting as Markov Chain Monte Carlo," 2025.

[10] S. Pateux, M. Gendrin, L. Morin, T. Ladune, and X. Jiang, "BOGausS: Better Optimized Gaussian Splatting," *arXiv preprint arXiv:2504.01844*, 2025.

[11] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park, "Compact 3D Gaussian Representation for Radiance Field," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[12] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[13] 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*, 2024.

[14] Y. Wang, Z. Li, L. Guo, W. Yang, A. Kot, and B. Wen, "ContextGS: Compact 3D Gaussian Splatting with Anchor Level Context Model," *Advances in Neural Information Processing Systems*, 2024.

[15] S. Xie, W. Zhang, C. Tang, Y. Bai, R. Lu, S. Ge, and Z. Wang, "MeSoNGS: Post-training Compression of 3D Gaussians via Efficient Attribute Transformation," in *European Conference on Computer Vision*, 2024.

[16] P. Papantonakis, G. Kopanas, B. Kerbl, A. Lanvin, and G. Drettakis, "Reducing the Memory Footprint of 3D Gaussian Splatting," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2024.

[17] H. Wang, M. H. Vali, and A. Solin, "Compressing 3D Gaussian Splatting by Noise-Substituted Vector Quantization," in *Scandinavian Conference on Image Analysis*, 2025.

[18] A. Rai, D. Wang, M. Jain, N. Sarafianos, K. Chen, S. Sridhar, and A. Prakash, "UVGS: Reimagining Unstructured 3D Gaussian Splatting Using UV Mapping," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[19] S. Lee, F. Shu, Y. Sanchez, T. Schierl, and C. Hellge, "Compression of 3D Gaussian Splatting with Optimized Feature Planes and Standard Video Codecs," *arXiv preprint arXiv:2501.03399*, 2025.

[20] S. Li, C. Wu, H. Li, X. Gao, Y. Liao, and L. Yu, "GSCodec Studio: A Modular Framework for Gaussian Splat Compression," *arXiv preprint arXiv:2506.01822*, 2025.

[21] Wikipedia contributors, "Quaternion," https://simple.wikipedia.org/wiki/Quaternion, 2025, online; Accessed: 2025-07-19.

[22] H. Schaub, J. L. Junkins *et al.*, "Stereographic Orientation Parameters for Attitude Dynamics: A Generalization of the Rodrigues Parameters," *Journal of the Astronautical Sciences*, 1996.

[23] J. Y. Jeong, R. Koo, K.-J. Oh, H.-C. Shin, and G. Lee, "[gsc][jee6.1-related] training method for generating temporally consistent per-frame i-3dgs dataset," ISO/IEC JTC1/SC29/WG4, Genova, Online, Working Document m71763, Jan. 2025, presented at ISO/IEC JTC1/SC29/WG4 Meeting.

[24] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-Curves," *ITU-T SG16, Doc. VCEG-M33*, 2001.