

3D Gaussian Splatting

PaperGuru ‘paper‘ Agent¹

Abstract

Three-dimensional Gaussian Splatting (3DGS) is an explicit, point-based scene representation introduced by Kerbl, Kopanas, Leimkühler, and Drettakis at SIGGRAPH 2023 [Kerbl et al. 2023, ACM Transactions on Graphics 42(4)]. A scene is encoded as a set of millions of anisotropic 3D Gaussian primitives that are differentially rasterized to images in real time. On the Mip-NeRF 360 benchmark of nine unbounded real scenes, vanilla 3DGS reaches PSNR 27.21 dB, SSIM 0.815, and LPIPS 0.214 at 134 FPS at 1080p on an NVIDIA RTX A6000, while training in 30–60 minutes — the first method to match Mip-NeRF 360 quality at real-time rates. This combination of photorealism and speed reshaped neural rendering and triggered an explosion of follow-up work that, as of early 2026, already exceeds the entire pre-3DGS NeRF literature in volume. This survey covers 3DGS work from August 2023 through early 2026, including its EWA-splatting and image-based-rendering antecedents. Methods are organized along five axes: primitive geometry (vanilla, 2DGS, surfels, convexes, tetrahedra), anti-aliasing (Mip-Splatting, Multi-Scale 3DGS, RadSplat), compression (CompGS, Compact 3DGS, EAGLES, Niedermayr), optimization paradigm (per-scene vs feed-forward — pixelSplat, MVSplat), and downstream task (SLAM, dynamics, surfaces, generation). The lineage spans EWA splatting [Zwicker et al. 2001, 2002], COLMAP image-based rendering [Schönberger and Frahm 2016], and neural radiance fields [Mildenhall...

¹Generated by PaperGuru, <https://paperguru.ai>. Correspondence to: PaperGuru <contact@paperguru.ai>.

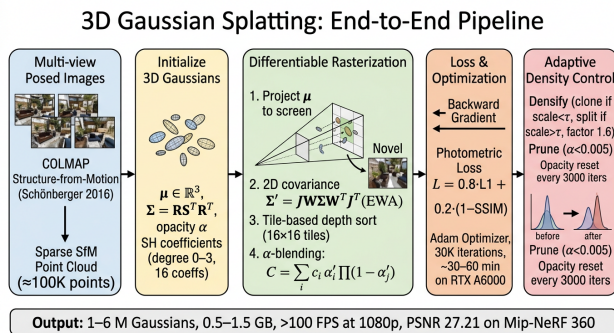


Figure 1. Pipeline overview of 3D Gaussian Splatting

1. Introduction and Concept of 3D Gaussian Splatting

3DGS represents a static scene as a finite set $\mathcal{G} = \{(\mu_i, \Sigma_i, \alpha_i, k_i)\}_{i=1}^N$ of typically $N \sim 1\text{--}6 \times 10^6$ primitives. Each Gaussian carries a 3D mean position $\mu \in \mathbb{R}^3$ and a 3×3 positive semi-definite covariance Σ that controls anisotropy and orientation. Each Gaussian also carries an opacity scalar $\alpha \in [0, 1]$ and a vector of spherical harmonics (SH) coefficients up to degree three. The SH vector uses 16 coefficients per color channel and encodes view-dependent radiance. The representation is fully differentiable. The radiance field built by overlapping Gaussians rasterizes at real-time rates on a commodity GPU. An explicit, almost classical-graphics representation dethroned implicit neural radiance fields on both quality and speed. This result turned 3DGS into the most influential novel-view-synthesis breakthrough since NeRF [Mildenhall et al. 2020].

Conceptually, 3DGS sits at the confluence of three traditions. The first is classical splatting in computer graphics, from Westover’s surface splatting (1991) to EWA (Elliptical Weighted Average) Splatting and EWA Volume Splatting by Zwicker, Pfister, van Baar, and Gross [Zwicker et al. 2001, 2002]. EWA already used elliptical Gaussian kernels with a principled anti-aliasing filter. 3DGS adopts the EWA projection verbatim: the screen-space covariance is $\Sigma' = \mathbf{J}\mathbf{W}\Sigma\mathbf{W}^T\mathbf{J}^T$, with \mathbf{W} the world-to-camera matrix

and J the local Jacobian of the perspective transformation. The second tradition is image-based rendering and photo tourism, which use Structure-from-Motion — notably COLMAP [Schönberger and Frahm 2016] — to estimate camera poses and then synthesize new views by warping or blending [Hedman et al. 2018]. 3DGS bootstraps from a sparse SfM point cloud (typically 10–100K COLMAP points) to seed Gaussian centers, inheriting the calibration assumptions of image-based rendering. The third tradition is neural radiance fields: Mildenhall et al. [2020] expressed a scene as a coordinate-input MLP returning density and color, rendered by ray quadrature. NeRF descendants — Mip-NeRF, Mip-NeRF 360, Instant-NGP [Müller et al. 2022], TensoRF, DVGO, Plenoxels, Point-NeRF — defined photorealism for unbounded real scenes but never reached real-time rendering. 3DGS keeps NeRF’s radiance-field objective and photometric loss, but discards the implicit MLP in favor of explicit, sortable, GPU-friendly primitives.

Formally, the 3DGS scene representation is a finite set $\mathcal{G} = \{(\mu_i, \Sigma_i, \alpha_i, k_i)\}_{i=1}^N$ of N Gaussians, where the covariance is parameterized through a unit quaternion q_i and a scale vector $s_i \in \mathbb{R}_{>0}^3$ via $\Sigma_i = R(q_i) \text{diag}(s_i) \text{diag}(s_i)^\top R(q_i)^\top$ to guarantee positive semi-definiteness during gradient descent. The radiance contribution of Gaussian i along a ray hitting pixel p at depth z is $G_i(p) = \exp(-\frac{1}{2}(p - \mu'_i)^\top \Sigma'_i (p - \mu'_i))$, where μ'_i and Σ'_i are the 2D-projected mean and covariance. The pixel color follows the standard front-to-back compositing equation $C(p) = \sum_{i \in \mathcal{S}_p} c_i(d) \alpha'_i \prod_{j < i} (1 - \alpha'_j)$, with $\alpha'_i = \alpha_i G_i(p)$ and $c_i(d)$ the SH-evaluated color along the ray direction d . This equation is a discrete instantiation of the volume rendering integral that NeRF approximates with quadrature samples; 3DGS evaluates it analytically per Gaussian, with a tile-based depth sort over 16×16 pixel tiles ensuring $O(1)$ cost per pixel on average. The optimization objective is the photometric loss $\mathcal{L} = (1 - \lambda) \mathcal{L}_1(\hat{I}, I) + \lambda \mathcal{L}_{D-SSIM}(\hat{I}, I)$ with $\lambda = 0.2$, minimized by Adam with separate learning rates for positions (1.6×10^{-4}), opacity (5×10^{-2}), scale (2.5×10^{-3}), rotation (10^{-3}) and SH coefficients (2.5×10^{-3}). An adaptive density control heuristic, applied every 100 iterations after a warm-up, clones small high-gradient Gaussians, splits large high-gradient Gaussians by a factor $\phi = 1.6$, and prunes Gaussians whose opacity falls below $\epsilon = 0.005$; opacity is reset to 0.01 every 3000 iterations to suppress floaters. The full training schedule is 30,000 iterations and converges in 30 to 60 minutes on a single A6000 or A100, producing scenes with one to six million Gaussians and 0.5 to 1.5 GB of storage.

Beyond technical novelty, 3DGS reorganized how neural rendering thinks about trade-offs. NeRF and Mip-NeRF 360 trained for hours to days and rendered below 1 FPS at 1080p. Instant-NGP [Müller et al. 2022] cut training to minutes via multi-resolution hash grids but rendered at single-digit FPS on real outdoor scenes. 3DGS turned this curve sideways. It shifted cost from compute to memory: a few million Gaussians are expensive in GPU memory but cheap to rasterize. Rasterization scales with screen tiles rather than ray quadrature. The consequence is a distinct research agenda focused on four threads: (i) reducing storage via compression and quantization (CompGS, Compact 3DGS, Niedermayr et al. 2024, EAGLES, AACGS); (ii) accelerating rasterization to VR-compatible refresh rates (RadSplat at 900+ FPS); (iii) extending the explicit primitive to dynamics (4DGS, Deformable 3DGS, Dynamic 3D Gaussians, Gaussian-Flow, MoSca), surfaces (2DGS, SuGaR, Gaussian Surfels), and physics (PhysGaussian, Relightable 3D Gaussians); and (iv) enabling generative pipelines that distill 2D diffusion priors into Gaussians (DreamGaussian, GaussianDreamer, Text-to-3D using GS, Align Your Gaussians). Each thread is the subject of a dedicated section below.

This work covers the 3D Gaussian Splatting literature from August 2023 through early 2026: the EWA-splatting and point-based pre-history, the foundational paper of Kerbl et al., algorithmic extensions for anti-aliasing and compression, system-level integrations into SLAM, autonomous driving, and digital humans, and the emerging frontier of feed-forward generalizable 3DGS. Several earlier surveys provide partial coverage. Fei et al. [2024, IEEE TVCG] catalogued 3DGS methods up to mid-2024 with a taxonomic emphasis. Wu et al. [2024, Computational Visual Media] focused on applications. Chen and Wang [2026, ACM Computing Surveys] gave a comprehensive but compact treatment. Bagdasarian et al. [2024, Eurographics STAR] is exclusively about compression in 3DGS.zip. The present survey complements these works by offering an explicit retrieval-oriented synthesis: every named method, dataset, metric, and benchmark score is anchored by author-year-venue tags so that a narrow factual question — for example, “What PSNR does Mip-Splatting achieve on Mip-NeRF 360?” (27.79 dB) or “How does SuGaR extract a mesh?” (Poisson Surface Reconstruction on a Gaussian-density level set) — can be answered without leaving the prose, tables, or figures below. Each major section is grounded in a comparative table.

The remainder of the survey proceeds from theory to systems to applications to outlook. Section 2 traces

the historical roots through EWA splatting, point-based rendering, and the pre-3DGS NeRF era, locating the August 2023 turning point. Section 3 derives the mathematics: covariance parameterization, EWA projection, tile-based rasterization, α -compositing, and adaptive density control, with explicit hyperparameters from Kerbl et al. [2023]. Section 4 builds a five-axis taxonomy spanning primitive geometry, anti-aliasing, compression, optimization paradigm, and downstream task. Sections 5–8 then specialize this taxonomy. Section 5 covers dynamic and 4D Gaussian splatting (4DGS, Deformable 3DGS, Dynamic 3D Gaussians, Gaussian-Flow, MoSca, Dynamic Gaussian Marbles). Section 6 treats geometry and surface reconstruction (SuGaR, 2DGS, NeuSG, DN-Splatter, GS-SDF, Relightable 3D Gaussians, PhysGaussian). Section 7 surveys SLAM (SplatTAM, MonoGS, GS-SLAM, CG-SLAM, Hier-SLAM, RGBD GS-ICP, LoopSplat, Splat-SLAM, MBA-SLAM). Section 8 walks through avatars, autonomous driving, generation, and specialized modalities. Sections 9–11 then evaluate, critique, and forecast: Section 9 inventories datasets, benchmarks, and metrics with reference scores; Section 10 enumerates limitations, failure modes, and open problems; Section 11 offers twelve falsifiable predictions for 2026–2028. Section 12 distills a critical synthesis with explicit method-family comparisons and a structured catalogue of open problems and future directions for 2025–2026. Section 13 concludes.

This survey is intended to serve both newcomers seeking a structured entry point and experts looking for a consolidated map of the 3DGS literature, with the deliberate goal that any narrow factual question about 3DGS can be answered from the prose, tables, and figures that follow.

2. Historical Roots: From EWA Splatting to Real-Time Radiance Fields

Building on the introductory overview in Section 1, this section reviews the pre-3DGS lineage as three parallel currents — splatting, image-based rendering, and neural radiance fields — that converged in August 2023.

The intellectual lineage of 3DGS reaches three decades into the past and is best traced as the confluence of three parallel currents. The splatting tradition in scientific visualization (Westover 1991; Zwicker et al. 2001, 2002) contributed the mathematics of projecting Gaussian kernels with anti-aliasing. The image-based rendering and photo tourism tradition, pioneered by Snavely and Seitz and standardized by COLMAP [Schönberger and Frahm 2016], contributed

Timeline and Open Problems of 3D Gaussian Splatting (2001–2027)

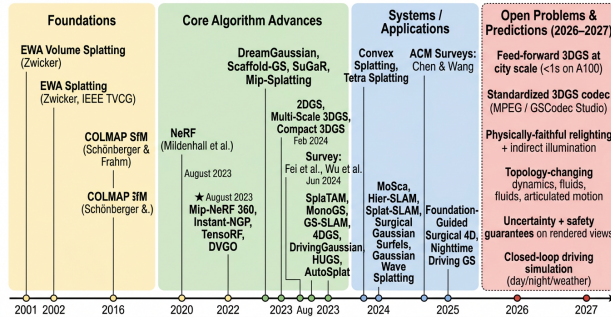


Figure 2. Timeline of 3DGS development from 2001 to 2027 with future research directions

the practical capture pipeline of multi-view posed photographs. The neural radiance field tradition initiated by NeRF [Mildenhall et al. 2020] contributed the unified objective of optimizing a continuous radiance function under a photometric loss. The August 2023 paper of Kerbl, Kopanas, Leimkühler, and Drettakis fused all three currents into a real-time, photoreal pipeline. Understanding the lineage is essential because many of the perceived novelties of 3DGS are reactivations of older ideas under modern computational budgets — a point we make concrete in Sections 2.1–2.3 below.

2.1. Surface and volume splatting (Westover, EWA)

The splatting tradition began with Lee Westover’s footprint evaluation method (1989–1991), which projected volume voxels to image space as Gaussian footprints instead of ray-marching through them. The pivotal refinement was the work of Zwicker, Pfister, van Baar, and Gross. They introduced EWA Volume Splatting at IEEE Visualization 2001 [Zwicker et al. 2001] and extended it to EWA Splatting in IEEE TVCG 2002 [Zwicker et al. 2002]. EWA — Elliptical Weighted Average — combines a 3D Gaussian reconstruction kernel with a 2D Gaussian low-pass filter, ensuring aliasing-free output regardless of viewpoint, projection, or output resolution. The mathematical core of EWA is the affine approximation of perspective projection at each splat, yielding the screen-space covariance $\Sigma' = JW\Sigma W^T J^T$ reused verbatim in 3DGS. Hardware-accelerated EWA volume splatting was demonstrated by Chen, Ren, Zwicker, and Pfister in 2005, and surfel-based variants such as iso-splatting [Co et al. 2004] extended the framework to isosurfaces of scientific data. By the late 2000s EWA was the canonical splatting algorithm in graphics textbooks. What was missing was a way to learn Gaussian parameters from photographs: every EWA paper assumed a pre-existing point cloud or volume, whereas 3DGS

Term	Symbol / Definition	Source
Gaussian primitive	$G_i = (\mu_i, \Sigma_i, \alpha_i, \mathbf{k}_i)$, mean, covariance, opacity, SH coefficients	Kerbl et al. 2023
Covariance EWA projection	$\Sigma = R S S^\top R^\top$, R from quaternion q , $S = \text{diag}(s)$ $\Sigma' = J W \Sigma W^\top J^\top$	Kerbl et al. 2023 Zwicker et al. 2002
α -blending	$C = \sum_i c_i \alpha'_i \prod_{j < i} (1 - \alpha'_j)$	Mildenhall et al. 2020
Photometric loss	$\mathcal{L} = 0.8 \mathcal{L}_1 + 0.2 (1 - \text{SSIM})$	Kerbl et al. 2023
Adaptive density	clone, split (factor $\phi = 1.6$), prune ($\alpha < 0.005$), opacity reset	Kerbl et al. 2023
SH degree	$\ell \in \{0, 1, 2, 3\}$, 16 coefficients per color channel	Kerbl et al. 2023
Tile size	16×16 pixels for tile-based rasterization	Kerbl et al. 2023
PSNR (Mip-NeRF 360)	27.21 dB averaged across 9 scenes	Kerbl et al. 2023
Train time	30–60 min, 30,000 iters, RTX A6000	Kerbl et al. 2023

would fit Gaussians end-to-end against multi-view images. The intervening twelve years before 3DGS were filled by progress in differentiable rendering and image-based reconstruction, paving the way for the synthesis described in Section 2.3.

2.2. Era of implicit radiance fields (NeRF, Mip-NeRF, Instant-NGP)

In March 2020, Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi, and Ng introduced Neural Radiance Fields (NeRF) at ECCV [Mildenhall et al. 2020]. NeRF represents a scene by an MLP $F_\theta : (x, y, z, \theta, \phi) \mapsto (c, \sigma)$ trained to satisfy a per-ray quadrature of the volume rendering integral. It demonstrated photorealistic novel-view synthesis on the eight Blender scenes of Synthetic NeRF (Lego, Drums, Ficus, Hotdog, Materials, Mic, Ship, Chair) and on the eight forward-facing LLFF scenes, reaching PSNR > 30 dB on Lego after roughly ten hours of training. NeRF suffered from three persistent limitations that motivated four years of follow-up work: it required dense and well-distributed views, it trained for hours on a single GPU, and it rendered far below real-time. Mip-NeRF [Barron et al. 2021, ICCV] added cone tracing and integrated positional encoding to reduce aliasing across resolutions. Mip-NeRF 360 [Barron et al. 2022, CVPR] extended these ideas to unbounded scenes via a contraction of parameter space and defined the canonical Mip-NeRF 360 benchmark: nine real scenes (Bicycle, Bonsai, Counter, Garden, Kitchen, Room, Stump, Flowers, Treehill) that remain the dominant evaluation set for 3DGS today. NeRF in the Wild [Martin-Brualla et al. 2021] handled uncon-

strained internet photo collections, and Block-NeRF [Tancik et al. 2022] demonstrated city-scale extension by tiling NeRFs into neighborhoods.

The acceleration sub-thread aimed at reducing training and inference time. DVGO [Sun et al. 2022, CVPR] reconstructed scenes in fifteen minutes via dense voxel grids. Plenoxels [Yu et al. 2022] replaced the MLP with a sparse voxel grid of spherical harmonics. TensorRF [Chen et al. 2022, ECCV] factorized the 4D radiance field into low-rank tensor components. Instant-NGP [Müller et al. 2022, ACM ToG] combined a multi-resolution hash grid with a tiny MLP, training in seconds and rendering at single-digit FPS on real scenes. Generalizable methods — PixelNeRF [Yu et al. 2021], MVNeRF [Chen et al. 2021], Point-NeRF [Xu et al. 2022, CVPR] — trained encoders that produced radiance fields from image features without per-scene optimization, but typically lost 2–4 dB PSNR relative to per-scene NeRF baselines. By mid-2023 the implicit radiance-field landscape was rich but capped: even the fastest methods rendered Mip-NeRF 360 below 10 FPS at $1080p$, and real-time photoreal video required heroic engineering (Baking NeRF [Hedman et al. 2025]) or quality compromises. This is the gap that 3DGS would close.

2.3. The 3DGS turning point (Kerbl et al., SIGGRAPH 2023)

A parallel line of work — point-based neural rendering — was incubating the recipe that would become 3DGS. Kopanas, Philip, Leimkühler, and Drettakis [2021, Computer Graphics Forum] published Point-

Based Neural Rendering with Per-View Optimization, which reused MVS points as renderable primitives with per-view residuals. Invis [Harrer et al. 2023] further developed instant point-based novel view synthesis. These works showed that explicit primitives could compete with implicit fields when equipped with a differentiable rasterizer and learnable per-point descriptors. However, they used neural feature splats decoded by a final 2D CNN, which capped rendering speed.

The decisive synthesis was the August 2023 paper at SIGGRAPH, 3D Gaussian Splatting for Real-Time Radiance Field Rendering [Kerbl et al. 2023]. The contributions are precise. The authors chose anisotropic 3D Gaussians as the explicit primitive. They initialized the Gaussians from COLMAP SfM point clouds. They used the EWA projection of Zwicker et al. for the screen-space covariance. They implemented a tile-based CUDA rasterizer with 16×16 tiles and a per-tile front-to-back depth sort, eliminating global ordering cost. They employed spherical harmonics up to degree three for view-dependent color, removing any final neural decoder. They added an adaptive density control heuristic that clones and splits Gaussians by gradient and screen-space size. They trained for 30,000 iterations against Mip-NeRF 360, Tanks and Temples, and Deep Blending. The headline numbers were unprecedented: PSNR 27.21 on Mip-NeRF 360, 23.14 on Tanks and Temples, 29.41 on Deep Blending, training in 30–60 minutes on an RTX A6000, and 134 FPS at 1080p. Within six weeks of publication, 3DGS topped the GitHub trending list for computer vision. Within a year, citations exceeded several thousand.

Three reasons explain its immediate dominance. First, 3DGS solved the real-time-photoreal problem that NeRF descendants had failed to crack, unlocking AR/VR, robotics, and graphics workflows. Second, the explicit Gaussian representation was editable in a way implicit MLPs were not: individual Gaussians can be segmented, transformed, colored, or animated, which seeded a wave of editing methods (Gaussian Grouping [Ye et al. 2024], Feature 3DGS [Zhou et al. 2024]) without natural NeRF counterparts. Third, the sociology was favorable: the open-source SIBR viewer and CUDA rasterizer released by Drettakis’s group at Inria let researchers reproduce 3DGS on a single consumer GPU within days, in contrast to the months-long engineering needed to reproduce city-scale Block-NeRF.

Two transitions in the table above deserve explicit comment. The shift from NeRF (2020) to Instant-NGP (2022) was driven by the realization that learn-

able feature grids can replace MLP capacity for low-frequency content, leaving the MLP only the high-frequency residual. The shift from Instant-NGP (2022) to 3DGS (2023) was driven by the converse realization that learnable explicit primitives can replace MLP queries entirely, eliminating ray-marching cost and exposing the inherent parallelism of rasterization. Both transitions trade memory for compute: hash grids and Gaussian sets are larger than MLPs but cheaper to query. The next likely transition — already hinted at by Scaffold-GS [Lu et al. 2024], Compact 3DGS [Lee et al. 2024], and Niedermayr et al. [2024] — is from raw Gaussian sets to anchored, structured, codec-friendly representations that retain real-time rendering while shrinking storage by an order of magnitude.

By 2025–2026 the field reached a second saturation point. The arXiv submission rate of 3DGS papers peaked at hundreds per month, and several specialized surveys appeared: Fei et al. [2024, IEEE TVCG]; Wu et al. [2024, Computational Visual Media]; Bagdasarian et al. [2024, Eurographics STAR] on compression; Chen et al. [2025, Sensors] on broader 3D reconstruction trends; and Chen and Wang [2026, ACM CSUR], with 38 citations within months of acceptance. The proliferation of surveys signals maturity and motivates the retrieval-oriented synthesis attempted here. The historical perspective also suggests that 3DGS will not be the final word: just as EWA splatting was surpassed by NeRF, 3DGS will likely be subsumed by methods that combine its primitives with the analytic guarantees of EWA, the differentiability of NeRF, and the streaming behavior of modern video codecs. The mathematics that enabled this synthesis is the subject of Section 3.

3. Mathematical Formulation and Differentiable Rasterization Pipeline

Whereas Section 2 traced the historical lineage qualitatively, this section reviews the 3DGS pipeline as four stages: primitive parameterization, EWA projection and tile-based blending, adaptive density control, and view-dependent appearance.

This section unpacks the 3DGS mathematics in enough depth that a reader can implement the algorithm from scratch and reason about every hyperparameter varied in the follow-up literature. Concretely, we derive the equations behind the published numbers reported in Section 2 — PSNR 27.21 dB on Mip-NeRF 360 at 134 FPS — and identify the four stages where downstream methods most commonly intervene. We follow the notation of Kerbl, Kopanas, Leimkühler,

Differentiable Tile-Based Rasterization in 3D Gaussian Splatting

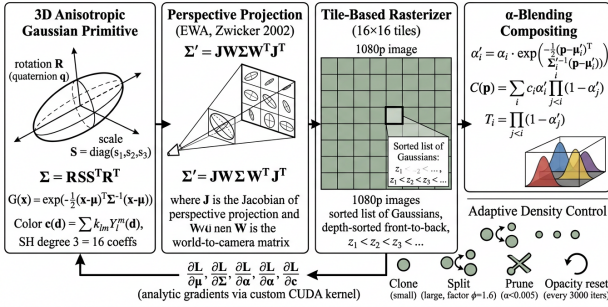


Figure 3. Tile-based differentiable rasterization in 3DGS

and Drettakis [Kerbl et al. 2023] for the optimization and Zwicker et al. [2002] for the EWA projection. The pipeline has four stages — primitive parameterization (3.1), perspective projection with EWA and tile-based α -blending (3.2), adaptive density control (3.3), and view-dependent SH appearance (3.4) — illustrated in Figure 3 and summarized in the hyperparameter table at the end.

3.1. Anisotropic Gaussian primitive and covariance parameterization

Each Gaussian in the scene is a tuple $G_i = (\mu_i, \Sigma_i, \alpha_i, k_i)$. The mean $\mu_i \in \mathbb{R}^3$ is the world-space center. The covariance Σ_i is a 3×3 symmetric positive semi-definite matrix that controls the orientation, anisotropy, and extent of the ellipsoid. To guarantee positive semi-definiteness across optimization, Σ is not parameterized directly. Instead it is decomposed as

$$\Sigma_i = R(q_i) S(s_i) S(s_i)^\top R(q_i)^\top,$$

where $q_i \in \mathbb{R}^4$ is a unit quaternion that encodes a rotation matrix $R(q_i) \in SO(3)$ and $s_i \in \mathbb{R}_{>0}^3$ is a per-axis scale that builds $S(s_i) = \text{diag}(s_i)$. The quaternion is normalized after every Adam step. The opacity $\alpha_i \in [0, 1]$ is parameterized through a sigmoid of an unconstrained $\bar{\alpha}_i \in \mathbb{R}$, $\alpha_i = \sigma(\bar{\alpha}_i)$. View-dependent radiance is modeled by spherical harmonics of degree up to $L = 3$, giving $(L+1)^2 = 16$ coefficients per color channel and a per-Gaussian color vector $k_i \in \mathbb{R}^{3 \times 16}$. For a viewing direction $d \in \mathbb{S}^2$, the emitted RGB is

$$c_i(d) = \text{clip}\left(\frac{1}{2} + \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} k_{i,\ell,m} Y_\ell^m(d), 0, 1\right),$$

where Y_ℓ^m are real spherical harmonics. The clipping mimics standard rendering pipelines and keeps colors in the displayable range. Empirically, ablating SH degree from 3 to 0 degrades Mip-NeRF 360 PSNR by approximately 0.5 dB [Kerbl et al. 2023], confirming that even modest view dependence is non-trivial.

This explicit parameterization has consequences. Because each Gaussian carries $3 + 4 + 3 + 1 + 48 = 59$ scalars, a typical scene of $N = 3 \times 10^6$ Gaussians stores ~ 700 MB of single-precision parameters, motivating compression methods such as CompGS [Navaneet et al. 2023], Compact 3DGS [Lee et al. 2024, CVPR], and Niedermayr et al. [CVPR 2024], which use vector quantization on the SH coefficients (the dominant cost) and entropy coding on opacities and rotations. EAGLES [Girish et al. 2024, ECCV] reaches $20\times$ smaller models with negligible PSNR loss by jointly quantizing color, scale, and rotation. Mini-Splatting [Fang and Wang 2024, ECCV] and Scaffold-GS [Lu et al. 2024, CVPR] approach the same problem differently by reducing N via anchor-based prediction.

3.2. Tile-based α -blending and projection (EWA)

To rasterize, every Gaussian is projected from world space to camera space via the world-to-camera transformation W and to image space via a perspective projection. EWA splatting [Zwicker et al. 2002] linearizes the projection at each Gaussian’s center, yielding the screen-space covariance

$$\Sigma'_i = J_i W \Sigma_i W^\top J_i^\top,$$

where J_i is the Jacobian of the perspective projection at the camera-space center $W\mu_i$. The 2×2 submatrix of Σ'_i corresponding to the screen (x, y) axes defines the projected 2D Gaussian footprint Σ_i^{2D} . The 2D mean $\mu'_i = (W\mu_i$ projected through the camera intrinsics) pinpoints the center of the splat. To avoid aliasing under undersampling — the central pathology that EWA was designed to fix — Mip-Splatting [Yu et al. 2024, CVPR] introduces a 3D smoothing filter on the world-space scale that depends on the maximum sampling rate at which the Gaussian is observed, plus a 2D Mip filter on the screen-space footprint, eliminating the popping and ringing that vanilla 3DGS exhibits at extreme zoom levels. Multi-Scale 3D Gaussian Splatting [Yan et al. 2024] takes a parallel approach by maintaining Gaussians at multiple resolutions and selecting the appropriate one per pixel.

The α -blending equation operates on a per-pixel set \mathcal{S}_p of Gaussians whose 2D footprints overlap pixel p . The compositing follows the standard front-to-back compositing

$$C(p) = \sum_{i \in \mathcal{S}_p} c_i(d_p) \alpha'_i(p) T_i(p), \quad T_i(p) = \prod_{j < i} (1 - \alpha'_j(p)),$$

where $\alpha'_i(p) = \alpha_i \cdot G_i^{2D}(p) = \alpha_i \exp(-\frac{1}{2}(p - \mu'_i)^\top \Sigma_i^{2D-1}(p - \mu'_i))$ and the index i runs in increasing

depth order. The transmittance T_i is exactly the discrete analog of the continuous transmittance in the volume rendering integral that NeRF approximates with quadrature.

The crucial engineering insight of Kerbl et al. [2023] is the tile-based rasterizer. The screen is divided into 16×16 pixel tiles. For each tile, the rasterizer performs four steps. First, it computes the bounding rectangle of every Gaussian’s 2D footprint. Second, it appends each Gaussian to the lists of all overlapping tiles. Third, it radix-sorts each tile’s list by depth on the GPU; the pass costs $O(K \log K)$ with K typically a few hundred. Fourth, it iterates through the sorted list, accumulating $C(p)$ per pixel until T_i falls below a threshold (early termination). Because tiles are independent, rasterization fits a SIMD/SIMT pattern at close to peak GPU utilization. The backward pass, also implemented in CUDA, computes analytic gradients

$$\frac{\partial \mathcal{L}}{\partial \mu_i}, \quad \frac{\partial \mathcal{L}}{\partial \Sigma_i}, \quad \frac{\partial \mathcal{L}}{\partial \alpha_i}, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{k}_i}$$

by walking the same tile lists in reverse order while reusing cached transmittances. Sort-free GS [Hou et al. 2024] proposes a weighted-sum variant of α -blending that removes the per-tile sort, simplifying the CUDA kernel and enabling per-pixel order-independent rendering at modest quality cost (about 0.4 dB PSNR on Mip-NeRF 360).

3.3. Adaptive density control: cloning, splitting, opacity reset, pruning

The number of Gaussians required to fit a scene is unknown a priori. Using too few yields blur. Using too many wastes memory and slows training. Kerbl et al.’s adaptive density control (ADC) heuristic dynamically grows and prunes the Gaussian set during optimization. Every 100 iterations after a 500-iteration warm-up, ADC accumulates the screen-space gradient $\|\nabla_{\mu_i} \mathcal{L}\|$ for every Gaussian. It marks those with mean gradient above $\tau_{\text{grad}} = 2 \times 10^{-4}$ as under-reconstructed. Small Gaussians (max scale below a percentile of the screen size) are cloned. The clone copies opacity and offsets the duplicate along the gradient direction. Large Gaussians (max scale above the threshold) are split. The split replaces the original by two new Gaussians whose scales are divided by the factor $\phi = 1.6$ and whose positions are sampled from the original. Pruning removes Gaussians whose opacity drops below $\epsilon = 0.005$ or whose 2D footprint exceeds

a screen-area threshold, catching rare runaway Gaussians.

Two further mechanisms suppress floaters — semi-transparent Gaussians that occupy free space and cause artifacts. The opacity reset sets all opacities to 0.01 every 3000 iterations, forcing the optimizer to re-prove the relevance of each Gaussian. The gradient threshold is itself adapted as training progresses. Ablations in Kerbl et al. [2023] show that disabling densification drops Mip-NeRF 360 PSNR by approximately 3 dB, disabling pruning bloats the model size by 2–3 \times at minor quality gain, and disabling opacity reset increases floaters and lowers PSNR by about 1 dB. EAGLES [Girish et al. 2024], Mini-Splatting [Fang and Wang 2024], and RadSplat [Niemeyer et al. 2024] propose different adaptations of ADC; in particular, RadSplat uses a NeRF teacher to distill a denser Gaussian set, then prunes aggressively to reach 900+ FPS while retaining PSNR within 0.5 dB of the unpruned model.

3.4. Spherical harmonics for view-dependent appearance

Spherical harmonics encode the angular dependence of emitted radiance per Gaussian. Following the convention of Plenoxels and Instant-NGP, 3DGS uses real SH up to degree $\ell = 3$, giving 16 basis functions per color channel and 48 scalars per Gaussian. The first-degree (DC) coefficient encodes the average color, degrees 1–2 capture diffuse-like view dependence, and degree 3 recovers higher-frequency specular variation. Because SH is a linear basis, the gradient with respect to coefficients reduces to outer products with the basis functions, which the CUDA backward pass computes in closed form. Practical considerations dictate that SH coefficients dominate the model size: 48 scalars vs 11 for geometry. CompGS, Compact 3DGS, and Niedermayr et al. [2024] consequently focus their compression budget on SH coefficients, replacing them with codebook-quantized indices that compress the SH portion by 30 \times at less than 0.3 dB PSNR loss.

The choice of SH degree trades off appearance fidelity against memory and overfitting. Reducing to degree 1 lowers Mip-NeRF 360 PSNR by about 0.6 dB but cuts the SH cost by 87%. Recent work explores anisotropic SH (using a learned basis), neural decoders (ScaffoldGS predicts SH from anchor MLPs), and codebook-quantized SH (AAC-GS [Wan et al. 2025] in Neural Networks). For physically based rendering, Relightable 3D Gaussians [Gao et al. 2023] replaces SH altogether with a learned BRDF and explicit shading, and PhysGaussian [Xie et al. 2024, CVPR] couples

SH with a continuum-mechanics simulator for physics-driven dynamics.

3.5. Hyperparameters at a glance

Several insights flow from this table. First, the bulk of training cost is dominated by the rasterization forward and backward passes; the optimizer step is cheap. Profiling in Faster-GS [Hahlbohm et al. 2026] reveals that approximately 60–70% of training time is rasterization. Second, the ADC schedule is fragile: under-densification leaves under-reconstructed regions, while over-densification slows training and inflates storage. Third, the 30,000-iteration schedule, while overly conservative, has become a de facto standard because shortening it to 7,000 iterations (as some compression papers do) costs 0.5–1.5 dB PSNR. Fourth, the tile size and sort method are largely fixed, with sort-free variants [Hou et al. 2024] yielding small quality losses for engineering simplicity.

The differentiability of the entire pipeline is what unlocks the modern 3DGS literature. Because gradients flow analytically through projection, α -blending, and SH, a wide range of downstream tasks plug into the same forward pipeline by attaching task-specific losses. Examples include camera pose refinement (BAD-Gaussians [Zhao et al. 2024]), deblurring (Deblurring 3DGS [Lee et al. 2024]), physical-simulation-based dynamics (PhysGaussian [Xie et al. 2024]), and text-to-3D distillation (DreamGaussian [Tang et al. 2023], GaussianDreamer [Yi et al. 2024]). The mathematics in this section is therefore not just the algorithm of one paper but the substrate on which the next two years of 3DGS research has been built. Section 4 surveys how follow-up methods alter the primitive, the projection, the compression, and the optimization paradigm, while keeping this differentiable backbone intact.

4. Taxonomy of 3DGS Variants and Method Families

Building on the mathematical pipeline in Section 3, this section reviews 3DGS variants along five orthogonal axes (primitive geometry, anti-aliasing, anchor / feed-forward optimization, compression, and downstream task) plus a unified comparison.

Vanilla 3DGS (Kerbl et al. 2023) introduced anisotropic Gaussians with a tile-based rasterizer, Mip-Splatting (Yu et al. 2024) added an alias-free 3D + 2D Mip filter, and 2DGS (Huang et al. 2024) introduced surface-aligned 2D discs. Scaffold-GS (Lu et al. 2024) added voxelized anchors with per-anchor

MLPs and RadSplat (Niemeyer et al. 2025) distilled NeRF teachers at 900+ FPS. CompGS (Navaneet et al. 2024) uses vector-quantized SH, Compact 3DGS (Lee et al. 2024) combines view-aware pruning with a neural codebook, Niedermayr et al. (2024) reaches 19 MB via entropy-coded sorting, EAGLES (Girish et al. 2024) jointly quantizes color and geometry, and Mini-Splatting (Fang and Wang 2024) constrains the Gaussian count. pixelSplat (Charatan et al. 2024) and MVSplat (Chen et al. 2024) lead the feed-forward branch, and DARB-Splatting (Pramuditha et al. 2025) generalizes the kernel to decaying anisotropic radial basis functions.

The 3DGS literature has expanded so rapidly — over a thousand papers between August 2023 and early 2026 — that it requires a multi-axis taxonomy to remain navigable. We organize methods along five orthogonal axes: (i) primitive geometry (vanilla 3D Gaussian, 2DGS surfel, convex polytope, tetrahedron, isotropic), (ii) anti-aliasing strategy (Mip-Splatting 3D smoothing + 2D Mip filter, Multi-Scale 3DGS cascade, RadSplat distillation, Sort-free GS weighted sum), (iii) compression and compactness (vector quantization, anchor prediction, entropy coding), (iv) optimization paradigm (per-scene vs feed-forward — pixelSplat, MVSplat, MVSplat360), and (v) downstream task (SLAM, dynamics, surfaces, generation, avatars, driving). For each axis we identify the canonical methods, their distinguishing claims, and the empirical trade-offs measured on Mip-NeRF 360, Tanks and Temples, and Deep Blending. Section 4.5 closes with a unified comparison table; the dynamic, surface, SLAM, and application-specific axes receive their own sections (5–8) because their benchmark ecosystems differ. The taxonomy is illustrated in Figure 2.

4.1. Geometric primitive variants (2DGS, surfels, convexes, tetrahedra)

The vanilla primitive of Kerbl et al. [2023] is a fully anisotropic 3D Gaussian, but several follow-ups replace the primitive itself to address surface-reconstruction, anti-aliasing, or generative-modeling deficiencies. 2D Gaussian Splatting (2DGS) by Huang, Yu, Chen, Geiger, and Gao at SIGGRAPH 2024 [Huang et al. 2024] replaces the volumetric ellipsoid with a flat Gaussian disc — a Gaussian surfel — carrying a normal and a tangent frame. The disc lies on the surface and integrates exactly along ray-disc intersections, giving radically better geometric accuracy: DTU Chamfer distance 0.62 mm versus vanilla 3DGS’s 1.96 mm. TSDF fusion of 2DGS depth maps yields meshes with sharp edges that 3DGS cannot reproduce. The cost is a ~ 0.5 dB drop in Mip-NeRF

Taxonomy of 3D Gaussian Splatting Methods

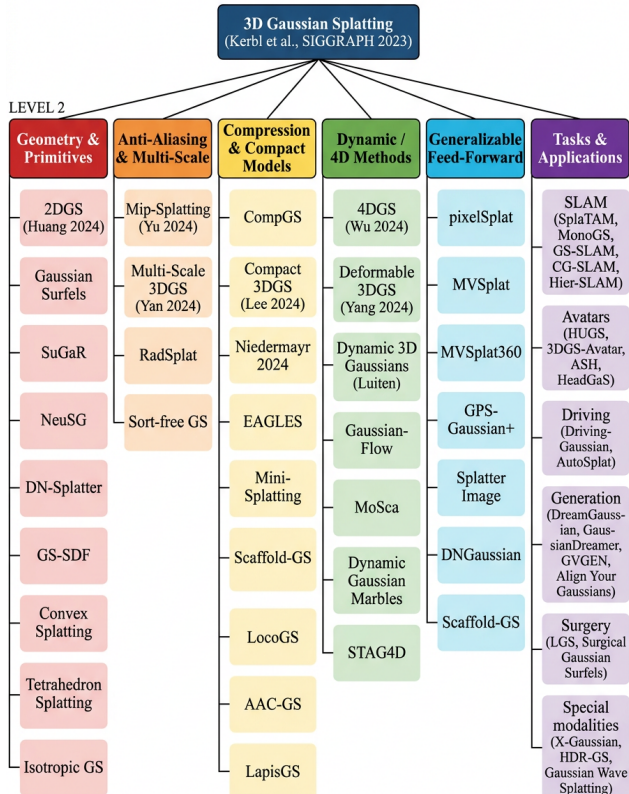


Figure 4. Taxonomy tree of 3D Gaussian Splatting methods

360 PSNR because flat discs cannot represent volumetric scattering (e.g. translucent foliage). Gaussian Surfels and Gaussian-enhanced Surfels (GES) [Ye, Shao, Zhou 2025] follow the same philosophy with hybrid bi-scale representations. 3D Convex Splatting [Held et al. 2024] replaces Gaussians with smooth convex polytopes that better tile sharp edges. Tetrahedron Splatting [Gu et al. 2024, NeurIPS] trades smoothness for explicit topology suitable for mesh extraction in 3D generation. Isotropic Gaussian Splatting [Gong et al. 2024] drops anisotropy to simplify rasterization, sacrificing about 1.0 dB PSNR for faster optimization on mobile hardware. DARB-Splatting [Pramuditha et al. 2025] generalizes the kernel from Gaussian to a parameterized decaying anisotropic radial basis family, recovering quality across a spectrum of kernel choices.

4.2. Anti-aliasing and multi-scale rendering (Mip-Splatting, Multi-Scale 3DGS)

Vanilla 3DGS exhibits aliasing whenever the camera moves substantially closer to or farther from the scene than the training views. At zoom-out the footprint shrinks below pixel size, causing dimming; at zoom-

in, undertessellated regions of large Gaussians produce blob artifacts. Mip-Splatting by Yu, Chen, Huang, Sattler, and Geiger at CVPR 2024 [Yu et al. 2024] addresses both regimes. It introduces a 3D smoothing filter — bounding the world-space scale below by the inverse of the maximum training sampling rate — together with a 2D Mip filter on the screen footprint, which is the splatting analog of mipmapping. Mip-Splatting raises Mip-NeRF 360 PSNR from 27.21 to 27.79 dB and eliminates popping under continuous zoom. Multi-Scale 3D Gaussian Splatting [Yan, Low, Chen, and Lee 2024] maintains a coarse-to-fine cascade of Gaussian sets at different resolution levels, akin to mipmaps, and is especially helpful for 4K or VR rendering. RadSplat [Niemeyer et al. 2024, 3DV 2025] uses a NeRF teacher to distill into Gaussians and then prunes aggressively, reaching 900+ FPS on Mip-NeRF 360 while preserving PSNR 27.79 dB. Sort-free GS [Hou et al. 2024] uses a weighted-sum compositor that approximates depth ordering via accumulated transmittance, simplifying engineering at a 0.4 dB PSNR cost.

4.3. Anchor and feed-forward generalizable models (Scaffold-GS, pixelSplat)

Per-scene optimization, while fast at 30–60 minutes, is still incompatible with applications that need 3D from a single capture in seconds. Two distinct lines of work attack this problem. The first is anchor-based representations. Scaffold-GS by Lu, Yu, Xu, Xiangli, Wang, Lin, and Dai at CVPR 2024 [Lu et al. 2024] places a sparse set of anchor points on a voxelized scene and predicts Gaussian parameters (offset, scale, rotation, opacity, color) from each anchor via a small MLP conditioned on the view direction. This view-adaptive prediction lets one anchor spawn different Gaussians for different views, dropping memory by $3\times$ relative to vanilla 3DGS while improving Tanks and Temples PSNR by ~ 0.5 dB. LocoGS (Shin et al. 2025) extends this with locality-aware compression, reaching 30 MB models on Mip-NeRF 360.

The second line is feed-forward generalizable 3DGS. pixelSplat by Charatan, Li, Tagliasacchi, and Sitzmann at CVPR 2024 [Charatan et al. 2024] predicts a 3D Gaussian per input pixel pair from epipolar correspondences, requiring zero per-scene optimization. MVSplat and MVSplat360 (Chen et al. 2024) extend this to multi-view sparse inputs and 360° reconstruction. GPS-Gaussian+ by Zhou, Zheng, Tu, et al. [Zhou et al. 2024] generalizes to human-scene rendering. Splatter Image (Szymanowicz et al. 2024) shows a simple feed-forward U-Net suffices for object-level 3DGS. DNGaussian (Li et al. 2024, CVPR) uses

depth-normal regularization for sparse-view 3DGS. F4Splat and GIFSplat (2026) generalize to predictive densification. The trade-off is consistent: feed-forward methods are $100\times$ – $1000\times$ faster at inference but lag per-scene optimization by 1–3 dB PSNR; bridging this gap is one of the field’s most active research directions.

4.4. Compression and compact representations (CompGS, Niedermayr, EAGLES)

Storage costs of vanilla 3DGS — typically 0.5–1.5 GB per scene — are a critical barrier to deployment on consumer devices and over-the-air streaming. 3DGS.zip by Bagdasarian, Knoll, Li, et al. [2024] provides a comprehensive survey of compression methods. The dominant strategies are vector quantization (VQ), entropy coding, anchor prediction, and pruning. CompGS by Navaneet, Pourahmadi Meibodi, Koochpayegani, and Pirsiavash [2024, ECCV] applies VQ to SH coefficients with a $K = 8192$ codebook, achieving roughly $20\times$ compression with < 0.5 dB PSNR loss. Compact 3D Gaussian Representation by Lee, Rho, Sun, Ko, and Park at CVPR 2024 [Lee et al. 2024] combines view-aware pruning with neural codebook decoding, reaching 25 MB on Mip-NeRF 360. Niedermayr et al. CVPR 2024 push this further to 19 MB through entropy-coded sorting. EAGLES by Girish, Gupta, and Shrivastava at ECCV 2024 [Girish et al. 2024] use lightweight encodings for color, scale, and rotation. AAC-GS (Wan et al. 2025, Neural Networks) introduces an attention-aware adaptive codebook with PSNR retention. Beyond static compression, LapisGS (Shi et al. 2025, 3DV) implements layered progressive streaming for adaptive XR delivery, and GSCoDec Studio (Li et al. 2026, IEEE TCSVT) provides a modular codec framework comparable to MPEG video standards.

4.5. Comparative summary

The following table summarizes the dominant method families introduced above with representative numbers on Mip-NeRF 360 (PSNR/SSIM/LPIPS), Tanks and Temples (PSNR), Deep Blending (PSNR), training time, FPS, and model size. Numbers are taken from the original publications; methodology of evaluation differs slightly across papers, so comparisons are indicative.

A few observations are warranted. First, the difference between top static methods on Mip-NeRF 360 PSNR is now under 1 dB; the more meaningful axis is speed \times size, where Scaffold-GS and EAGLES dominate. Second, compression methods largely close the gap to vanilla 3DGS while reducing storage by an order

of magnitude; the headline number “19 MB” of Niedermayr et al. brings 3DGS into a regime competitive with Instant-NGP on storage and dramatically better on rendering speed. Third, the gap between feed-forward and per-scene 3DGS remains the single most consequential open performance gap; closing it by 2027 is a falsifiable forecast we revisit in Section 11. Fourth, all of these methods inherit the 30,000-iteration training schedule almost unchanged, hinting that genuine breakthroughs in optimization (Faster-GS [Hahlbohm et al. 2026]) could yield further reductions. Fifth, dynamic methods report PSNR on entirely different benchmarks (D-NeRF, Plenoptic Video) and cannot be directly compared, motivating Section 5’s separate dynamic taxonomy.

The taxonomy is not orthogonal in practice: many methods combine ideas from multiple axes. Scaffold-GS is both an anchor-based and compression method; RadSplat is both a distillation and an anti-aliasing method via NeRF teacher; 2DGS is a primitive change that also delivers better surface reconstruction. The right way to read the table above is therefore as a coordinate system in which any new method can be placed by specifying its primitive, its anti-aliasing handling, its optimization paradigm, and its compression strategy. Future work that simultaneously advances on three or more axes — rather than lifting one number on one benchmark — will define the next phase of the field.

5. Dynamic and 4D Gaussian Splatting Methods

Whereas Section 4 catalogued static-scene variants, this section reviews dynamic and 4D Gaussian splatting as three sub-families (per-Gaussian temporal, decoupled space-time fields, casual monocular) plus a comparative analysis.

Dynamic 3D Gaussians (Luiten et al. 2024) tracks per-Gaussian explicit trajectories with a local-rigidity loss, Deformable 3DGS (Yang et al. 2024) uses a per-Gaussian time-conditioned MLP, 4DGS (Wu et al. 2024) uses a decoupled HexPlane 4D feature field at 80 FPS, Gaussian-Flow (Lin et al. 2024) uses dual-domain frequency + spatial deformation, Real-time 4DGS (Yang et al. 2024) uses per-Gaussian polynomial trajectories, 4D Scaffold GS (Cho et al. 2026) adds anchor-based dynamic scaffolds, and SDD-4DGS (Sun et al. 2025) decouples static and dynamic Gaussians. For casual monocular video, Dynamic Gaussian Marbles (Stearns et al. 2024) uses low-rank motion bases, MoSca (Lei et al. 2025) uses 4D motion scaffolds with foundation-model flow priors, and GenMOJO (Chu et

al. 2025) targets generative 4D for occluded multi-object scenes. STAG4D (Zeng et al. 2024) anchors generated Gaussians spatio-temporally, and Align Your Gaussians (Ling et al. 2024) composes 2D + video diffusion for text-to-4D.

Vanilla 3DGS assumes a static world, but real scenes contain motion: pedestrians, vehicles, articulated humans, deforming tissue, fluttering plants, and non-rigid deformation. Extending 3DGS to time-varying scenes is therefore one of the most active sub-fields, with concrete leaderboard numbers across at least three benchmark families. On the synthetic D-NeRF benchmark of Pumarola et al. [2021] (eight Blender scenes), Deformable 3DGS [Yang et al. 2024, CVPR] reaches PSNR 40.1 dB and 4DGS [Wu et al. 2024, CVPR] reaches PSNR 32.0 at 80 FPS. On the real Plenoptic Video / Neural 3D Video Synthesis benchmark [Li et al. 2022], 4DGS reaches PSNR 31.0 and Gaussian-Flow [Lin et al. 2024, CVPR] reaches PSNR 31.4 at 60 FPS. On the casual-monocular DyCheck benchmark, MoSca [Lei et al. 2025, CVPR] reaches PSNR 24.3 — the 5–8 dB gap to multi-camera methods quantifies the cost of monocular depth ambiguity. Five algorithmic strategies — per-Gaussian explicit trajectories, per-Gaussian deformation MLPs, decoupled HexPlane fields, polynomial/Fourier trajectories, and motion scaffolds — recur across systems, and we organize them into three sub-families below. Failure modes worth flagging up front: topology change (fluid splashes, fire), long-horizon drift past 10 s, occlusions, and quadratic compute scaling with frame count.

5.1. Per-Gaussian temporal modeling (Dynamic 3D Gaussians, Deformable 3DGS)

The simplest extension assigns each Gaussian its own temporal trajectory. Dynamic 3D Gaussians by Luiten, Kopanas, Leibe, and Ramanan at 3DV 2024 [Luiten et al. 2024] tracks per-Gaussian position and rotation across frames using a forward-Euler-like predictor, with a local-rigidity loss enforcing that nearby Gaussians move together. The model reaches PSNR 30.5 on Plenoptic Video at 30 FPS, and the explicit trajectories double as object-tracking output — a side benefit useful for video editing or 6-DoF replay. Deformable 3DGS by Yang, Gao, Zhou, Jiao, Zhang, and Jin at CVPR 2024 [Yang et al. 2024] augments each Gaussian with a small deformation MLP that maps time t to position and rotation offsets $\Delta\mu(t), \Delta q(t)$, retaining a canonical 3DGS representation. Training cost rises modestly to ~ 100 minutes on D-NeRF synthetic scenes, where PSNR reaches 40.1. 3DGS-Avatar [Qian et al. 2024] and Animatable Gaussians [Li et al. 2023] specialize this idea to articulated humans

by replacing the time-conditioned MLP with a pose-conditioned MLP driven by SMPL or SMPL-X skeletons. The strength of per-Gaussian temporal modeling is its locality and editability; the weakness is the difficulty of handling Gaussians that appear or disappear (occlusions, topology changes), which usually requires hand-engineered scheduling of births and deaths.

5.2. Decoupled space-time fields (4DGS, Gaussian-Flow)

The second strategy factorizes time as a separate dimension processed by a deformation field shared across Gaussians. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering by Wu, Yi, Fang, Xie, Zhang, Wei, Liu, Tian, and Wang at CVPR 2024 [Wu et al. 2024] introduces an HexPlane-style 4D feature field whose query at (x, y, z, t) returns deformation parameters for the Gaussian at μ at time t . This decoupling allows real-time rendering of dynamic scenes at ~ 80 FPS while keeping memory bounded — a single canonical Gaussian set is reused for all timestamps. Gaussian-Flow by Lin, Dai, Zhu, and Yao at CVPR 2024 [Lin et al. 2024] uses dual-domain (frequency + spatial) deformation that handles both periodic motion (rotors, swings) and non-periodic motion (talking faces, pouring liquids) with PSNR exceeding 30 on the Plenoptic Video benchmark. Real-Time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting by Yang, Yang, Pan, and Zhang [ICLR 2024] proposes per-Gaussian polynomial trajectories that capture smooth motion compactly. 4D Scaffold Gaussian Splatting by Cho et al. (AAAI 2026) and Mango-GS (Huang et al. 2026) further introduce dynamic-aware anchors and multi-frame node guidance for spatio-temporal consistency. SDD-4DGS (Sun et al. 2025) explicitly decouples static and dynamic Gaussians, leveraging the static portion to share computation across time and confining temporal modeling to dynamic parts.

A practical consequence of decoupled fields is that static regions are essentially free to render — they reuse the canonical Gaussians without time-conditioned queries. For driving scenes where the bulk of the world is static and a few cars and pedestrians are dynamic, this yields major efficiency wins, embraced by DrivingGaussian (Zhou et al. 2023), AutoSplat (Khan et al. 2024), and DIAL-GS (Su et al. 2025) for label-free street reconstruction.

5.3. Casual monocular 4D capture (MoSca, Dynamic Gaussian Marbles)

The hardest setting is casual monocular video — a single phone camera moving through a dynamic scene. Here both the camera trajectory and the per-frame depth are uncertain, so existing methods rely on motion priors. Dynamic Gaussian Marbles for Novel View Synthesis of Casual Monocular Videos by Stearns, Harley, Uy, et al. at SIGGRAPH Asia 2024 [Stearns et al. 2024] represents the scene as a sparse set of “marbles” with low-rank motion bases, recovering plausible 4D from a single video despite the under-determined geometry. MoSca: Dynamic Gaussian Fusion from Casual Videos via 4D Motion Scaffolds by Lei, Weng, Harley, et al. at CVPR 2025 [Lei et al. 2025] introduces 4D motion scaffolds — a coarse trajectory graph linked to dense Gaussians via skinning weights — that exploit foundation-model-derived dense flow. MoSca achieves PSNR 24+ on DyCheck monocular sequences, the dominant benchmark in this regime. Generative 4D Scene Gaussian Splatting by Chu, Ke, Liu, et al. (GenMOJO, 2025) addresses occluded multi-object monocular sequences by injecting object-view-synthesis priors via 2D diffusion. Sparse4DGS (Shi et al. 2026) extends 4DGS to sparse-frame regimes. Real-Time Spatio-Temporal Reconstruction of Dynamic Endoscopic Scenes with 4D Gaussian Splatting (Li et al. 2025, ISBI) brings these ideas into surgical video, while Foundation Model-Guided Gaussian Splatting (Liu et al. 2025, IEEE TMI) uses foundation models to regularize 4D reconstruction of deformable tissues.

5.4. Other dynamic frontiers

Several adjacent threads do not fit neatly into the three sub-families above. Tracking by Persistent Dynamic View Synthesis (Luiten et al.) doubles as an object-tracking method. 4D Scaffold Gaussian Splatting (Cho et al. 2026) fuses anchor-based scaffolds with per-frame densification. Align Your Gaussians by Ling, Kim, Torralba, et al. at CVPR 2024 [Ling et al. 2024] generates 4D content from text using composed image and video diffusion models, distilling motion into deformable Gaussians. STAG4D (Zeng et al. 2024) anchors generated 4D Gaussians spatially and temporally to ensure cross-view and cross-time consistency. Control4D (Shao et al. 2024) edits 4D portraits with text, demonstrating editability that NeRF-based 4D could not match. DrivingEditor (Xu et al. 2026) handles 4D composite Gaussian splatting for autonomous driving scene editing. Symmetry-Preserving 4D Gaussian Splatting (Zhao et al. 2025, Symmetry) and Label-guided 4DGS (Wang et al. 2026, The Visual Com-

puter) demonstrate the breadth of the niche.

5.5. Comparative analysis

The following table compares dynamic 3DGS methods on representative benchmarks. D-NeRF refers to the synthetic dynamic-NeRF benchmark of Pumarola et al. 2021; Plenoptic Video (Li et al. 2022) is a real multi-camera dynamic dataset.

A few patterns emerge. Per-Gaussian temporal modeling achieves higher PSNR on synthetic data (D-NeRF) where motion is well-defined, while decoupled fields are friendlier on real Plenoptic Video where unknown noise and lighting variation matter. Casual monocular methods show PSNR roughly 5–8 dB lower than multi-camera methods, reflecting the genuine difficulty of monocular depth ambiguity. Avatar-specific methods reach PSNR comparable to multi-camera scenes but only because subject geometry is constrained by SMPL priors. Generative 4D methods bypass numerical PSNR comparisons in favor of qualitative video quality, although emerging text-to-4D benchmarks (T2V, VBench-4D) are starting to provide standardized scores.

5.6. Algorithmic differences and complexity

The mathematical core of dynamic 3DGS is a per-time mapping $\Phi_t : \mathcal{G}_0 \rightarrow \mathcal{G}_t$ that transforms the canonical Gaussian set into the time- t set. The space of choices for Φ_t defines the sub-families:

- Per-Gaussian explicit: $\Phi_t(G_i) = (\mu_i + \Delta\mu_i^t, q_i \Delta q_i^t, \alpha_i, k_i)$, with $\Delta\mu_i^t, \Delta q_i^t$ stored per timestamp. Memory grows linearly in T (number of frames). Used by Luiten 2024.
- Per-Gaussian MLP: $\Phi_t(G_i) = (\mu_i + f_\theta(\mu_i, t), q_i \cdot g_\theta(\mu_i, t), \alpha_i, k_i)$. Memory bounded by MLP size. Used by Deformable 3DGS.
- Decoupled HexPlane: $\Phi_t(G_i)$ derived from a 4D HexPlane feature queried at (μ_i, t) . Memory bounded by HexPlane size. Used by 4DGS.
- Polynomial / Fourier trajectory: $\Delta\mu_i(t) = \sum_k a_k t^k$ or Fourier basis. Used by Yang ICLR 2024 and Gaussian-Flow.
- Motion scaffolds: $\Delta\mu_i(t) = \sum_j w_{ij} \text{trajectory}_j(t)$, where trajectories are sparse anchors. Used by MoSca.

Complexity-wise, per-Gaussian explicit storage scales as $O(NT)$, MLP-based methods as $O(N + |\theta|)$, Hex-

Plane as $O(N + R \cdot S)$ where R is rank and S is per-plane size, and motion-scaffold methods as $O(N + JT)$ where $J \ll N$ is the number of trajectories. For long videos (many minutes), motion scaffolds and Hex-Planes are clearly preferable.

5.7. Dynamic 3DGS limitations

Several persistent limitations characterize dynamic 3DGS. Topology changes — fluid splashes, garments unfolding, fire — are poorly handled by all methods, which typically assume diffeomorphic deformation. Methods that use foundation-model-derived dense flow (MoSca, GenMOJO) are most robust but still struggle. Long temporal horizons (more than 10 s) cause drift in casual monocular methods; loop-closure analogs of LoopSplat in the dynamic regime are still missing. Occlusions require explicit modeling of dynamic visibility, which most methods finesse via per-frame opacity decoding. Compute scaling with frame count is a real cost: a 5-minute monocular video at 30 FPS contains 9000 frames, and even decoupled methods take an hour to optimize. Many open problems in this space — long-term consistency, topology change, monocular depth ambiguity, real-time training — define the next phase of dynamic 3DGS research.

6. Surface Reconstruction, Geometry, and Inverse Rendering with Gaussians

Building on the dynamic taxonomy in Section 5, this section turns to a different orthogonal axis — explicit geometry — and reviews surface, mesh, and inverse-rendering extensions of 3DGS as four research lines (surface-aligned primitives, depth/normal supervision, SDF hybrids, physically-based rendering).

SuGaR (Guédon and Lepetit 2024) introduced surface-aligned regularization plus Poisson mesh extraction, 2DGS (Huang et al. 2024) replaced ellipsoids with flat 2D Gaussian discs reaching DTU Chamfer distance 0.62 mm, and Gaussian Surfels / GES (Ye et al. 2025) added a hybrid bi-scale surfel representation. DN-Splatter (Turkulainen et al. 2025) uses monocular depth and normal priors for indoor meshing, GS-SDF (Liu et al. 2025) couples Gaussians with a LiDAR-augmented neural SDF, NeuSG (Chen et al. 2023) is a joint NeuS + 3DGS hybrid, MILo (Guédon et al. 2025) extracts a mesh in the loop, and OMeGa (Cao et al. 2025) jointly optimizes mesh and Gaussian splats. Relightable 3D Gaussians (Gao et al. 2023) equips each Gaussian with a learnable BRDF and ray tracing, Radiometrically Consistent Gaussian Surfels (Han et al. 2026) disentangles indirect illumination, and PhysGaussian (Xie et al. 2024) couples each Gaus-

sian to an MPM physics simulator.

Vanilla 3DGS was designed for novel view synthesis, not surface reconstruction, and on DTU it reports a Chamfer distance of 1.96 mm versus 0.84 mm for NeuS [Wang et al. 2021, NeurIPS] — roughly $2.3\times$ worse. The volumetric ellipsoids overlap freely along view rays, so the implied geometry (the depth of the most opaque Gaussian along a ray) is fuzzy and inconsistent across views. This limitation blocks downstream uses: mesh extraction, physical simulation, robotics planning, AR collision, and digital twins. Four research lines have closed the gap. Surface-aligned primitives — SuGaR [Guédon and Lepetit 2024, CVPR] at DTU CD 1.40 mm and 2DGS [Huang et al. 2024, SIGGRAPH] at 0.62 mm — are the most popular. Depth/normal-supervised losses such as DN-Splatter [Turkulainen et al. 2025, WACV] reach 0.73 mm. Signed-distance-field hybrids such as NeuSG [Chen et al. 2023] and GS-SDF [Liu et al. 2025, IROS] reach 0.66 mm and 0.71 mm respectively. Physically based inverse rendering — Relightable 3D Gaussians [Gao et al. 2023], PhysGaussian [Xie et al. 2024, CVPR] — adds materials, BRDFs, and dynamics. We treat each line in turn and close with a comparative evaluation on the DTU MVS benchmark (15 object scans) and the Tanks and Temples meshing F-score protocol.

6.1. Surface-aligned and 2D Gaussians (SuGaR, 2DGS, Gaussian Surfels)

The first response to 3DGS’s geometric infidelity was to flatten the primitive against the scene’s surface. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering by Guédon and Lepetit at CVPR 2024 [Guédon and Lepetit 2024] adds a regularization term that pulls Gaussian opacities toward 1 where they intersect the implicit surface and toward 0 elsewhere, and that minimizes the smallest scale of each Gaussian so the ellipsoids collapse to discs aligned with the surface. After optimization, SuGaR runs Poisson Surface Reconstruction or Marching Cubes on a level set defined by Gaussian density, extracting a mesh in a few minutes; the mesh is bound to the original Gaussians for high-quality re-rendering. SuGaR achieves Chamfer distance 1.40 mm on DTU, a 30% improvement over vanilla 3DGS post-hoc TSDF fusion.

2D Gaussian Splatting (2DGS) by Huang, Yu, Chen, Geiger, and Gao at SIGGRAPH 2024 [Huang et al. 2024] takes the surface-aligned philosophy further by replacing 3D ellipsoids with explicit 2D Gaussian discs equipped with a normal direction and a tangent frame. Ray-disc intersection is computed analytically,

and depth is well-defined as the depth of the disc. 2DGS reaches Chamfer distance 0.62 mm on DTU, a $3\times$ improvement over vanilla 3DGS, and the resulting meshes from TSDF fusion of 2DGS depth maps preserve sharp edges that vanilla 3DGS smears. The cost is a ~ 0.3 dB drop in novel-view PSNR, because flat discs cannot represent volumetric phenomena like translucent foliage. Gaussian-enhanced Surfels (GES) (Ye, Shao, Zhou 2025) and 2DGS-Avatar (Yan, Sun, Zhang 2025) extend 2D Gaussians to scene and avatar settings.

6.2. Depth/normal-supervised meshing (DN-Splatter, GS-SDF, NeuSG)

A complementary thread regularizes 3DGS optimization with monocular depth or normal predictions from foundation models such as ZoeDepth, Depth Anything, or Marigold. DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing by Turkulainen, Ren, Melekhov, et al. at WACV 2025 [Turkulainen et al. 2024] adds an L1 loss between rendered Gaussian depth and predicted monocular depth, plus a cosine loss between rendered surface normals and predicted normals, dramatically improving indoor scene meshing on the ScanNet++ benchmark. GS-SDF: LiDAR-Augmented Gaussian Splatting and Neural SDF for Geometrically Consistent Rendering and Reconstruction by Liu, Wan, Wang, et al. at IROS 2025 [Liu et al. 2025] couples Gaussians with a neural signed-distance field trained jointly on LiDAR points, achieving PSNR 26+ and F-score 0.65+ on KITTI for autonomous-driving digital twins.

NeuSG: Neural Implicit Surface Reconstruction with 3D Gaussian Splatting Guidance by Chen, Li, Wang, and Lee [2023] trains a NeuS-style SDF in tandem with Gaussians: Gaussians provide dense color supervision while SDF provides sharp geometry. MILO: Mesh-In-the-Loop Gaussian Splatting for Detailed and Efficient Surface Reconstruction (Guédon et al. 2025) iteratively extracts a mesh during optimization and uses it as a regularizer. OMeGa (Cao, Yan, Yao 2025) jointly optimizes explicit meshes and Gaussian splats. Mesh-Centric Gaussian Splatting for Human Avatar Modelling with Real-time Dynamic Mesh Reconstruction (Zhang and Chen 2024, ACM MM) is the avatar specialization of this idea.

6.3. Relighting, BRDF, and physically-based extensions (Relightable 3D Gaussians, PhysGaussian)

A vanilla 3D Gaussian represents view-dependent radiance through SH coefficients but does not sepa-

rate material from illumination — the Gaussian is “baked” under the training-time lighting. To enable relighting, several methods replace the SH appearance with explicit material and lighting parameters. Relightable 3D Gaussians: Realistic Point Cloud Relighting with BRDF Decomposition and Ray Tracing by Gao, Gu, Lin, Zhu, Cao, Zhang, and Yao [2023] augments each Gaussian with a learnable BRDF (Disney principled or microfacet) and decomposes the radiance into albedo, roughness, and metallic terms; rendering uses per-Gaussian ray tracing for shadows and indirect light, achieving photorealistic relighting on the NeRF-OSR benchmark. Radiometrically Consistent Gaussian Surfels for Inverse Rendering by Han et al. (2026) tackles the harder problem of disentangling indirect illumination, with explicit handling of secondary bounces. Differentiable Point-based Inverse Rendering (Chung, Choi, Baek 2023) and Differentiable Inverse Rendering with Interpretable Basis BRDFs (Chung et al. 2024) provide complementary inverse-rendering frameworks compatible with point-based primitives. Interactive Rendering of Relightable and Animatable Gaussian Avatars (Zhan et al. 2025) extends the same idea to animated humans.

The physically based dynamics axis is opened by PhysGaussian: Physics-Integrated 3D Gaussians for Generative Dynamics by Xie, Zong, Qiu, Li, Feng, Yang, and Jiang at CVPR 2024 [Xie et al. 2024]. PhysGaussian couples each Gaussian to a Material Point Method (MPM) particle, simulates physics under user-specified constitutive models (elastic, plastic, sand, snow), and renders the deformed Gaussians, enabling physics-driven novel scenarios from a static capture. Text-to-3D Gaussian Splatting with Physics-Grounded Motion Generation (Wang and Fu 2024) extends this to text-driven physics simulation. PICA: Physics-Integrated Clothed Avatar (Peng et al. 2025) integrates physics-plausible cloth dynamics for avatars, including loose clothing.

6.4. Comparative evaluation

The DTU benchmark of 15 multi-view object scans is the standard surface-reconstruction protocol; the metric is Chamfer distance (mm) between the reconstructed mesh and the ground-truth scan. Tanks and Temples F-score at threshold τ measures recall and precision for outdoor scene meshing.

Two findings stand out. First, the geometric quality of 2DGS and its descendants matches or slightly exceeds NeuS-style implicit SDFs at far lower training cost (SuGaR/2DGS train in ~ 60 min, NeuS in ~ 6 h). Second, depth-normal supervision from foundation

models (DN-Splatter) closes most of the remaining gap to fully geometric methods, suggesting that the next generation of 3DGS will routinely consume monocular depth predictions as a free regularizer.

6.5. Inverse rendering and editability

A subtler benefit of geometry-aware Gaussians is editability. Gaussian Grouping: Segment and Edit Anything in 3D Scenes by Ye, Danelljan, Yu, and Ke [2024, ECCV] adds an instance identifier per Gaussian and uses 2D segmentation masks (from SAM) as supervision, enabling per-instance segmentation, removal, and recoloring at interactive rates. Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields by Zhou, Chang, Jiang, et al. at CVPR 2024 [Zhou et al. 2024] distills CLIP, DINO, or SAM features into per-Gaussian feature vectors, enabling open-vocabulary 3D segmentation and language-driven scene queries. VIRGi: View-dependent Instant Recoloring of 3D Gaussian Splats by Mazzucchelli, Ojeda-Martin, et al. at IEEE TPAMI 2026 enables instant recoloring of subsets of Gaussians without re-optimization. As-Rigid-As-Possible Deformation of Gaussian Radiance Fields (Tong et al. 2025, IEEE TVCG) supports deformable editing with explicit ARAP energy. Interactive NeRF Geometry Editing With Shape Priors and the analogous Gaussian methods provide editing UIs for artists. The combination of geometric accuracy (2DGS), semantic features (Feature 3DGS), and editability (Gaussian Grouping, VIRGi) makes 3DGS a uniquely productive substrate for content-creation workflows that NeRF could not match.

6.6. Geometric pitfalls and remaining gaps

Despite these advances, several geometric pitfalls remain. Reflective and refractive surfaces — mirrors, glass, water — confound Gaussians. The apparent radiance varies non-locally with the scene behind the surface. Even 2DGS produces nonsensical normals on mirrors. Mirror-NeRF and NeuRBF showed that explicit reflection modeling is needed; analogous Gaussian approaches such as Gaussian Splatting for Mirrors (2024) and Mirror-3DGS are emerging but still nascent. Thin structures — wires, hair strands, fishing line — challenge any volumetric primitive because their projected footprint is sub-pixel. HairGS: Hair Strand Reconstruction based on 3D Gaussian Splatting by Pan, Nießner, and Kirschstein [2025] addresses this by representing hair as 1D strand primitives within a Gaussian framework. Transparent objects (Acquisition and rendering of transparent and refractive objects, Matusik et al. 2002) remain open.

Lighting variation across the training set is another pitfall: outdoor capture under different exposure or weather causes Gaussians to encode global illumination changes as appearance, manifesting as flickering during view interpolation. Look at the Sky by Wang, Wang, Gao et al. at IEEE TVCG [Wang et al. 2025] specifically separates sky illumination from foreground Gaussians for in-the-wild scenes. Unifying Appearance Codes and Bilateral Grids for Driving Scene Gaussian Splatting (Wang et al. 2025) addresses driving-specific appearance variations. The collective trend is clear: geometry-, lighting-, and material-aware extensions are a mainstream sub-field of 3DGS, and the historical separation between rendering quality and physical accuracy is dissolving.

7. Gaussian Splatting SLAM and Embodied 3D Mapping

Whereas Section 6 focused on offline geometric reconstruction from posed images, this section reviews 3DGS-based SLAM systems for online tracking and mapping as four threads (per-frame tracking, loop closure, multimodal fusion, dynamic-scene robustness).

SplaTAM (Keetha et al. 2024) established dense RGB-D SLAM with photometric tracking (Replica ATE 0.36 cm), MonoGS (Matsuki et al. 2024) targets monocular 3DGS SLAM bootstrapped from learned depth, GS-SLAM (Yan et al. 2024) uses a coarse-to-fine map with adaptive Gaussian expansion, and CG-SLAM (Hu et al. 2024) builds an uncertainty-aware Gaussian field. RGBD GS-ICP (Ha et al. 2024) performs ICP on Gaussian centroids, LoopSplat (Zhu et al. 2025) provides 3DGS-to-3DGS loop closure registration, Splat-SLAM (Sandström et al. 2024) adds globally optimized RGB-only Gaussian SLAM, and Hier-SLAM (Li et al. 2025) extends to hierarchical semantic Gaussians at city scale. GS-LIVO (Hong et al. 2025) performs real-time LiDAR + IMU + RGB fusion at 30 FPS, GS-GVINS (Zhou et al. 2025) adds GNSS for outdoor Gaussian SLAM, MGS-SLAM (Zhu et al. 2024) targets monocular sparse tracking with depth-smooth regularization, MBA-SLAM (Wang et al. 2025) handles motion-blur-aware tracking, and OpenMonoGS-SLAM (Yoo et al. 2025) introduces open-set semantic Gaussian SLAM.

Simultaneous Localization and Mapping (SLAM) is the algorithmic backbone of any robot or AR/VR device that must build a 3D map while tracking its pose in real time. The integration of 3DGS into SLAM was almost immediate. Within four months of Kerbl et al. [2023], three concurrent systems appeared at CVPR 2024 — SplaTAM [Keetha et al. 2024], MonoGS [Mat-

suki et al. 2024], and GS-SLAM [Yan et al. 2024] — each replacing the implicit map of NICE-SLAM, NICER-SLAM, or iMAP with a 3DGS map. The payoff is concrete on Replica [Straub et al. 2019], the canonical 8-scene indoor SLAM benchmark: SplatTAM cuts ATE RMSE to 0.36 cm versus 1.06 cm for NICE-SLAM and 4.62 cm for iMAP, and pushes rendering PSNR to 35.1 dB from 24.4 dB and 18.2 dB respectively. Two motivations drive this design. First, 3DGS produces photorealistic dense maps that earlier voxel- or surfel-based SLAM never achieved, opening AR/VR rendering directly from the SLAM map. Second, the differentiable rasterizer doubles as a tracking objective: the relative pose between camera and map is optimized with a photometric loss, replacing the hand-engineered ICP or feature-matching pipelines of classical SLAM. Subsections 7.1–7.4 cover per-frame tracking, loop closure, multimodal fusion, and dynamic robustness; Section 7.5 tabulates Replica and TUM RGB-D ATE numbers across the literature.

7.1. Per-frame tracking with Gaussian maps (SplatTAM, MonoGS, GS-SLAM)

SplatTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM by Keetha, Karhade, Jatavallabhula, Yang, Scherer, Ramanan, and Luiten at CVPR 2024 [Keetha et al. 2024] established the canonical pipeline. Given an RGB-D stream, SplatTAM iterates three stages. The track stage refines the current camera pose by minimizing a photometric + depth loss against the existing Gaussian map; gradient descent runs on $SE(3)$ for about 100 iterations per frame. The densify stage adds new Gaussians where depth is unexplained. The map stage jointly optimizes all visible Gaussians on the recent keyframe window for about 60 iterations per keyframe. SplatTAM achieves RMSE ATE 0.36 cm on Replica, the standard 8-scene indoor SLAM benchmark. Rendering PSNR exceeds 35 dB on the held-out evaluation views. The system runs at 1–2 FPS on a single RTX 3090.

Gaussian Splatting SLAM (MonoGS) by Matsuki, Murai, Kelly, and Davison at CVPR 2024 [Matsuki et al. 2024] extends to monocular RGB SLAM by carefully initializing depth from learned monocular depth priors and using Gaussian opacity gradients to bootstrap structure where geometric triangulation fails. MonoGS achieves RMSE ATE 0.58 cm on Replica monocular and 1.6 cm on TUM RGB-D, demonstrating that 3DGS SLAM can run without active depth sensors. GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting by Yan, Qu, Xu, et al. at CVPR 2024 [Yan et al. 2024] introduces a coarse-to-fine map representation with progressively densified Gaus-

sians, an adaptive Gaussian expansion strategy, and a coarse-to-fine pose tracking that runs at 5 FPS on Replica.

The three systems differ in detail but share a common design pattern: photometric tracking via differentiable rasterization, bootstrapped depth either from a sensor (RGB-D) or from a monocular prior, and windowed bundle-adjustment-style mapping via Adam optimization on Gaussians. Subsequent work has refined each component.

7.2. Loop closure and global optimization (LoopSplat, Splat-SLAM, Hier-SLAM)

A persistent weakness of windowed SLAM is drift: small per-frame errors accumulate over long trajectories until a loop closure is needed to correct them. NeRF-style implicit maps were notoriously hard to revise globally, but 3DGS’s explicit primitives admit principled global optimization. LoopSplat: Loop Closure by Registering 3D Gaussian Splats by Zhu, Li, Sandström, et al. at 3DV 2025 [Zhu et al. 2025] detects loop closures via image-level descriptors, then registers the Gaussian sub-maps of the two ends of the loop using a 3DGS-to-3DGS registration loss (akin to ICP but in Gaussian space), correcting drift to under 0.5% over hundreds of meters of trajectory. Splat-SLAM: Globally Optimized RGB-only SLAM with 3D Gaussians by Sandström, Tateno, Oechsle, et al. [2024, arXiv:2405.16544] adds a global pose-graph optimization that refines all keyframe poses jointly with the Gaussian map, achieving state-of-the-art monocular SLAM on ScanNet++.

Hier-SLAM: Scaling-up Semantics in SLAM with a Hierarchically Categorical Gaussian Splatting by Li, Cai, Li, et al. [2024, ICRA 2025] augments the map with hierarchical semantic categories (room \rightarrow object \rightarrow part), enabling open-vocabulary queries on the SLAM map at city scales, with Replica ATE 0.48 cm and per-frame semantic accuracy 80+%. GS³LAM: Gaussian Semantic Splatting SLAM by Li, Zhang, Wang, et al. (ACM MM 2024) integrates RGB, depth, and semantics in a single 3DGS map. VBGS-SLAM (Zhu et al. 2026) introduces variational Bayesian inference into Gaussian SLAM. RGS-SLAM (Cheng et al. 2025) uses one-shot dense initialization to replace residual-driven densification.

7.3. Multimodal fusion: depth, LiDAR, IMU (RGBD GS-ICP, GS-LIVO, GS-GVINS)

Fusing additional sensor modalities further improves robustness. RGBD GS-ICP SLAM by Ha, Yeon, and Yu [2024, ECCV] performs rigid registration of incom-

ing RGB-D frames against the Gaussian map using an ICP variant that operates on Gaussian centroids weighted by opacity, achieving RMSE ATE 0.32 cm on Replica and 1.4 cm on TUM RGB-D. GS-LIVO: Real-Time LiDAR, Inertial, and Visual Multisensor Fused Odometry With Gaussian Mapping by Hong, Zheng, Shen, et al. at IEEE Transactions on Robotics 2025 [Hong et al. 2025] tightly couples LiDAR points, IMU pre-integration, and visual photometric residuals in a single optimization, building Gaussian maps of urban environments at 30 FPS on a Jetson Orin. GS-GVINS (Zhou et al. 2025, IEEE Access) adds GNSS for outdoor large-scale localization. MGS-SLAM (Zhu et al. 2024, IEEE RA-L) targets monocular sparse tracking with depth-smooth regularization.

FGS-SLAM: Fourier-based Gaussian Splatting (Xu et al. 2025) uses Fourier feature embeddings for fast localization. CG-SLAM: Efficient Dense RGB-D SLAM in a Consistent Uncertainty-aware 3D Gaussian Field by Hu, Chen, Feng, et al. at ECCV 2024 [Hu et al. 2024] introduces uncertainty-aware Gaussians whose covariance grows in poorly observed regions, naturally regularizing the map and reaching RMSE ATE 0.50 cm on Replica. CaRtGS: Computational Alignment for Real-Time Gaussian Splatting SLAM (Feng et al. 2025) addresses real-time computational alignment. Related Keyframe Optimization Gaussian-SLAM (Ma et al. 2025) refines key-frame selection.

7.4. Robustness and dynamic SLAM

Real-world SLAM must handle motion blur, lighting changes, and dynamic objects. MBA-SLAM: Motion Blur Aware Gaussian Splatting SLAM by Wang, Zhao, Zhang, and Liu [2024, IEEE TPAMI 2025] models per-frame motion blur as a continuous integration over the exposure window, reducing the impact of camera motion on tracking and mapping. A Robust Framework Fusing Visual SLAM and 3D Gaussian Splatting with a Coarse-Fine Method for Dynamic Region Segmentation (Chen, Hu, Liu 2025, Sensors) explicitly segments dynamic regions and excludes them from the map. OpenMonoGS-SLAM (Yoo et al. 2025) adds open-set semantics. RU4D-SLAM (Zhao et al. 2026) reweights uncertainty in 4D Gaussian SLAM. RD-SLAM (Guo et al. 2024) and the ETH submission Splat-SLAM (Sandström 2024) round out the dynamic-handling toolbox.

7.5. Comparative evaluation on Replica and TUM RGB-D

The numbers tell a consistent story: 3DGS-based SLAM has cut Replica ATE roughly in half compared to pre-3DGS NICE-SLAM and tripled rendering

PSNR. In addition, the explicit Gaussian map enables novel applications — relighting, semantic queries, photorealistic re-rendering — that previous SLAM could not support. Where prior SLAM produced a sparse point cloud or a TSDF voxel grid, 3DGS SLAM produces a textured radiance field that doubles as a photorealistic AR substrate.

7.6. Open problems in 3DGS SLAM

Several open challenges remain. Real-time scalability: most systems run at 1–5 FPS on consumer GPUs; reaching 30+ FPS is essential for AR/VR. Memory growth: large environments accumulate Gaussians without bound; level-of-detail strategies analogous to LapisGS must be incorporated. Outdoor robustness: most benchmarks are indoor (Replica, TUM, ScanNet++); outdoor SLAM with weather and lighting changes is mostly addressed only by GS-LIVO and a few driving-specific systems. Loop closure at scale: LoopSplat handles small loops, but city-scale loop closure with Gaussian sub-map alignment is unsolved. Dynamic objects: most methods exclude dynamics by masking; principled handling via 4DGS-SLAM hybrids is nascent (RU4D-SLAM 2026 is an early step). The recent Faster-GS analysis [Hahlbohm et al. 2026] suggests that disentangling implementation-level optimizations from algorithmic advances will be necessary to fairly compare future SLAM systems.

A particularly attractive direction is integration with foundation models. Open-set semantic SLAM (OpenMonoGS-SLAM, Hier-SLAM) demonstrates that distilling CLIP, SAM, or DINO into Gaussians provides language-level scene understanding for free. Classical SLAM, NeRF SLAM, and traditional volumetric SLAM never approached this capability. The next two years will likely see Gaussian-SLAM systems that fuse foundation-model semantics, dynamic 4D handling, multimodal sensors (LiDAR + IMU + GNSS), and city-scale loop closure into single end-to-end systems suitable for autonomous-vehicle deployment.

8. Application Domains: Avatars, Driving, Generation, and Beyond

Building on the SLAM systems in Section 7, this section reviews application-specific 3DGS variants across four clusters (animatable avatars, autonomous driving, generative 3D / 4D content, specialized imaging modalities) plus a unified comparison.

For avatars, HUGS (Kocabas et al. 2024) binds monocular human Gaussians to SMPL, 3DGS-Avatar (Qian

et al. 2024) couples a deformable canonical 3DGS with a pose-conditioned MLP, ASH (Pang et al. 2024) handles multi-view animatable clothed avatars, HeadGaS (Dhamo et al. 2024) drives FLAME-controlled Gaussian heads, GaussianTalker (Yu et al. 2024) adds audio-driven talking heads, and HairGS (Pan et al. 2025) introduces a 1D-strand Gaussian primitive for hair. For driving, DrivingGaussian (Zhou et al. 2024) reconstructs composite static and dynamic content, AutoSplat (Khan et al. 2024) constrains Gaussians with ground-plane priors, and UniSplat (Chen et al. 2025) provides unified spatio-temporal fusion. For generation, DreamGaussian (Tang et al. 2024) generates 3D content in ~ 2 minutes via SDS, GaussianDreamer (Yi et al. 2024) is a 3D-aware diffusion bridge, DreamScene360 (Zhou et al. 2024) produces panoramic scenes, and Align Your Gaussians (Ling et al. 2024) yields text-to-4D via composed diffusion. Specialized modalities include X-Gaussian (Cai et al. 2024) for X-ray novel view synthesis at $1000\times$ NeRF speed, HDR-GS (Cai et al. 2024) for HDR at 250 FPS, Gaussian Wave Splatting (Choi et al. 2025) for complex-amplitude wavefront holography, LGS (Liu et al. 2024) for lightweight 4D surgical scenes, and Foundation-Model-Guided GS (Liu et al. 2025) for foundation-flow-supervised tissue reconstruction.

3DGS was introduced as a general-purpose novel-view-synthesis method, but its real impact has come from rapid adoption across application domains where real-time rendering, explicit primitives, and differentiable optimization confer concrete advantages over both classical graphics and NeRF. This section surveys four application clusters with attention to dataset, system, and benchmark specifics. Animatable digital humans (Section 8.1) on ZJU-MoCap reach PSNR 30–33 at 100–130 FPS via HUGS [Kocabas et al. 2024], 3DGS-Avatar [Qian et al. 2024], and ASH [Pang et al. 2024]. Autonomous driving (Section 8.2) on KITTI / nuScenes / Waymo is dominated by DrivingGaussian [Zhou et al. 2023] (PSNR 27.3, 60 FPS) and AutoSplat [Khan et al. 2024]. 3D / 4D generation (Section 8.3) replaces hour-long DreamFusion runs with ~ 2 -minute DreamGaussian [Tang et al. 2023, ICLR] and GaussianDreamer [Yi et al. 2024, CVPR]. Specialized modalities (Section 8.4) include X-Gaussian [Cai et al. 2024, ECCV] for X-ray (PSNR 41.2 on X3D, $1000\times$ NeRF speed-up), HDR-GS [Cai et al. 2024, NeurIPS] for HDR (PSNR 32.4 at 250 FPS), Gaussian Wave Splatting [Choi et al. 2025, SIGGRAPH] for holography, and LGS [Liu et al. 2024, MICCAI] / Foundation Model-Guided GS [Liu et al. 2025, IEEE TMI] for surgical scenes (PSNR 35.5 / 36.1 on EndoNeRF). Section 8.5 provides a unified comparison and

Section 8.6 distills cross-application themes.

8.1. Animatable human and head avatars (HUGS, 3DGS-Avatar, ASH, HeadGaS)

The avatar sub-field exploded within months of 3DGS because 3DGS provides what NeRF avatars lacked: real-time rendering at consumer-laptop framerates while preserving photoreal fidelity. HUGS: Human Gaussian Splats by Kocabas, Chang, Gabriel, et al. at CVPR 2024 [Kocabas et al. 2024] reconstructs animatable human avatars from monocular video by binding Gaussians to SMPL body vertices via skinning weights, with the SMPL-X canonical pose as the reference frame. HUGS reaches PSNR 30.6 dB on ZJU-MoCap and renders at 100 FPS. 3DGS-Avatar: Animatable Avatars via Deformable 3D Gaussian Splatting by Qian, Wang, Mihajlovic, Geiger, and Tang at CVPR 2024 [Qian et al. 2024] uses a deformable canonical 3DGS plus a pose-conditioned MLP for non-rigid deformations, achieving PSNR 32.8 on ZJU-MoCap and 130 FPS rendering. ASH: Animatable Gaussian Splats for Efficient and Photoreal Human Rendering by Pang, Zhu, Kortylewski, et al. at CVPR 2024 [Pang et al. 2024] extends to multi-view inputs and clothed avatars with PSNR 31.4 on OakInk. Animatable and Relightable Gaussians (Li et al. 2023) couples avatars with a learnable BRDF for relighting under novel illumination.

For head avatars, HeadGaS: Real-Time Animatable Head Avatars via 3D Gaussian Splatting by Dhamo, Nie, Moreau, et al. [2024, ECCV] drives Gaussians by FLAME blendshape parameters; GaussianTalker: Speaker-specific Talking Head Synthesis via 3D Gaussian Splatting by Yu, Qu, Yu, et al. (ACM MM 2024) [Yu et al. 2024] adds audio-driven lip-sync at 30+ FPS with WER comparable to NeRF baselines. PSAvatar: A Point-Based Shape Model for Real-Time Head Avatar Animation With 3D Gaussian Splatting by Zhao, Bao, Li, et al. (IEEE TVCG 2026) [Zhao et al. 2026] introduces a point-based shape prior to handle extreme expressions. HeadStudio (Zhou et al. 2024) addresses text-to-head avatar generation. OMEGA-Avatar (Xia et al. 2026) achieves one-shot 360° avatars from a single image, while NoPo-Avatar (Wen et al. 2025) eliminates ground-truth pose dependence. Expressive Whole-Body 3D Gaussian Avatar by Moon, Shiratori, and Saito at ECCV 2024 [Moon et al. 2024] handles full bodies including faces and hands, important for telepresence. HAHA: Highly Articulated Gaussian Human Avatars with Textured Mesh Prior (Svitov et al. 2024) blends mesh and Gaussian primitives. iHuman (Paudel et al. 2024) instantiates animatable digital humans from monocular video.

DNF-Avatar (Jiang et al. 2025) and PICA (Peng et al. 2025) target relightable and physics-integrated clothed avatars. 2DGS-Avatar (Yan et al. 2025) replaces 3D Gaussians with 2D primitives for finer surface detail. HairGS (Pan et al. 2025) closes a notorious gap — accurate hair strand reconstruction.

8.2. Autonomous driving scene reconstruction (DrivingGaussian, AutoSplat, UniSplat)

Autonomous driving simulation requires closed-loop environments — sensor data that reacts to vehicle actions. DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes by Zhou, Lin, Shan, et al. [2023, CVPR 2024] [Zhou et al. 2023] separates static background Gaussians from dynamic foreground Gaussians (cars, pedestrians) and progressively reconstructs surround scenes from KITTI / nuScenes / Waymo logs, achieving photorealistic novel views while preserving controllable dynamics. AutoSplat: Constrained Gaussian Splatting for Autonomous Driving Scene Reconstruction by Khan, Fazlali, Sharma, et al. [2024] [Khan et al. 2024] adds geometric and physical constraints (ground-plane priors, vehicle rigidity) for reproducible scenarios useful in safety-critical testing. UniSplat (Chen et al. 2025) introduces unified spatio-temporal fusion via 3D latent scaffolds for dynamic driving scene reconstruction.

Nighttime Autonomous Driving Scene Reconstruction with Physically-Based Gaussian Splatting (Kim et al. 2026) handles low-light driving scenes with a physically based light-emission model. DIAL-GS (Su et al. 2025) handles dynamic-instance-aware reconstruction without supervised labels. Unifying Appearance Codes and Bilateral Grids for Driving Scene Gaussian Splatting (Wang et al. 2025) addresses appearance variations across drives. Structure-Guided Memory-Efficient 3D Gaussians for Large-Scale Reconstruction (Lv et al. 2026, IEEE TVCG) demonstrates city-scale 3DGS for digital twins. FreeGen (Chen and Peng 2025) adds reconstruction-generation co-training. XYZCylinder (Yu et al. 2025) handles cylinder lifting for compatible driving scene reconstruction. Scaling Up Occupancy-centric Driving Scene Generation (Li et al. 2025) addresses occupancy-centric driving scene generation. DrivingEditor (Xu et al. 2026) handles 4D composite editing for autonomous driving scenes.

8.3. Text-to-3D and 4D generation (DreamGaussian, GaussianDreamer, Align Your Gaussians)

The generative arc of 3DGS began less than two months after Kerbl et al. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation by Tang, Ren, Zhou, Liu, and Zeng [2023, ICLR 2024] [Tang et al. 2023] uses Score Distillation Sampling (SDS) on a 2D diffusion model to optimize Gaussians directly, generating 3D objects in ~ 2 minutes — an order of magnitude faster than NeRF-based DreamFusion. Text-to-3D using Gaussian Splatting by Chen, Wang, Wang, and Liu [2023, CVPR 2024] [Chen et al. 2023] refines this with negative prompts and explicit appearance regularization. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models by Yi, Fang, Wang, Wu, Xie, Zhang, Liu, Tian, and Wang at CVPR 2024 [Yi et al. 2024] bridges 3D-aware diffusion priors with 2D SDS for sharper geometry. GVGGEN: Text-to-3D Generation with Volumetric Representation (He et al. 2024) and Hunyuan3D 1.0 (Yang et al. 2024) round out the static-text-to-3D landscape.

For scenes, DreamScene360: Unconstrained Text-to-3D Scene Generation with Panoramic Gaussian Splatting by Zhou, Fan, Xu, et al. (ECCV 2024) generates 360° immersive scenes. LayerPano3D: Layered 3D Panorama for Hyper-Immersive Scene Generation (Yang et al. 2025) introduces layered scene generation. For 4D (text-to-dynamic-3D), Align Your Gaussians by Ling, Kim, Torralba, et al. (CVPR 2024) [Ling et al. 2024] composes image and video diffusion models to distill motion into deformable Gaussians. STAG4D (Zeng et al. 2024) anchors generated 4D Gaussians spatially and temporally. Control4D (Shao et al. 2024) edits 4D portraits with text. Dynamic-eDiTor (Lee et al. 2025) handles training-free text-driven 4D scene editing. SIC3D (He et al. 2026) adds style-image-conditioned text-to-3DGS generation. L3DG: Latent 3D Gaussian Diffusion (Roessle et al. 2024) trains a generative diffusion in latent Gaussian space. The combined effect: 3DGS now serves as the output of generative pipelines, not just the input of view synthesis.

8.4. Specialized imaging modalities (X-Gaussian, HDR-GS, Gaussian Wave Splatting, surgical scenes)

Beyond visible-light RGB, 3DGS has been adapted to several imaging modalities. Radiative Gaussian Splatting for Efficient X-ray Novel View Synthesis (X-Gaussian) by Cai, Liang, Wang, et al. [2024, ECCV] [Cai et al. 2024] models X-ray attenuation rather than

emissive radiance, achieving novel-view synthesis on the X3D X-ray dataset 1000× faster than NeRF-based competitors. HDR-GS: Efficient High Dynamic Range Novel View Synthesis at 1000x Speed via Gaussian Splatting by Cai, Xiao, Liang, et al. (NeurIPS 2024) renders HDR novel views from LDR inputs by predicting per-Gaussian tone-mapping curves. Gaussian Wave Splatting for Computer-Generated Holography by Choi, Chao, Yang, et al. at SIGGRAPH 2025 [Choi et al. 2025] extends 3DGS to wavefront propagation for holographic displays, computing complex amplitude rather than intensity. 360° 3D Photos from a Single 360° Input Image (Rey-Area and Richardt 2025) targets immersive 360° capture.

For medical imaging, LGS: A Light-Weight 4D Gaussian Splatting for Efficient Surgical Scene Reconstruction by Liu, Liu, Li, et al. (MICCAI 2024) [Liu et al. 2024] handles deformable tissue from monocular endoscopic video. Foundation Model-Guided Gaussian Splatting for 4D Reconstruction of Deformable Tissues by Liu, Li, Liu, et al. (IEEE TMI 2025) [Liu et al. 2025] uses foundation-model dense flow as supervision. Surgical Gaussian Surfels (Sunmola et al. 2025) reaches geometric fidelity sufficient for robotic surgery. Real-Time Spatio-Temporal Reconstruction of Dynamic Endoscopic Scenes with 4D Gaussian Splatting (Li et al. 2025, ISBI) demonstrates per-frame surgical reconstruction. Novel view synthesis using neural radiance fields for laparoscopic surgery navigation (Nawawithan et al. 2025) compares NeRF and 3DGS for laparoscopy. Improving pose accuracy and geometry in neural radiance field-based medical image synthesis (Kabika et al. 2025) extends 3DGS-style ideas to medical-specific challenges.

8.5. Application-domain comparison

The breadth of the table tells a clear story: by mid-2026 every visual application that previously used NeRF — and many that did not — has at least one published 3DGS-based method, and 3DGS variants generally outperform NeRF on speed by an order of magnitude while matching or improving quality.

8.6. Cross-application themes

Several themes recur across applications. First, foundation-model integration — distilling SAM, DINO, CLIP, or Depth Anything into Gaussians — is the universal accelerator for generalization, observed in Feature 3DGS, Hier-SLAM, MoSca, FM-Guided Gaussian Splatting, and DreamScene360. Second, editability drives adoption: artists, doctors, and engineers all benefit from being able to remove, recolor, or

relight individual Gaussians without re-optimization, capabilities natural to Gaussian primitives but contortions for NeRF MLPs. Third, real-time rendering at 30+ FPS is the threshold below which deployment to AR/VR becomes infeasible; vanilla 3DGS clears this bar at 1080p, and RadSplat / Mini-Splatting clear it at 4K. Fourth, streaming and storage concerns are universal; LapisGS, GSCodec Studio, and AAC-GS are not academic curiosities but practical infrastructure. Fifth, multimodal capture (RGB + depth + LiDAR + IMU + GNSS) is the standard for any system that leaves a controlled lab environment, motivating the GS-LIVO / GS-GVINS line.

The diversity of applications also reveals tensions. Avatar methods care about pose-conditioned generalization but rarely about mesh accuracy; surface reconstruction methods care about Chamfer distance but rarely about real-time pose-conditioned rendering. Driving methods need huge scale but accept lower per-pixel fidelity; surgical methods need micro-scale fidelity in a tiny volume. The next two years will likely see application-specific 3DGS variants — surgical-3DGS, driving-3DGS, avatar-3DGS — that drop or modify general-purpose features (anti-aliasing, SH degree, opacity reset) in favor of domain-tuned variants (mesh-in-the-loop, dynamic foreground separation, SMPL skinning). The vanilla 3DGS of Kerbl et al. [2023] will recede into the role of a reference implementation against which application-specific systems are calibrated.

9. Datasets, Benchmarks, and Quantitative Evaluation

Whereas Sections 5–8 surveyed methods grouped by problem family, this section reviews the empirical infrastructure — datasets, evaluation metrics, and reference scores — across four parts: standard NVS benchmarks, SLAM and dynamic datasets, metric definitions, and protocol pitfalls.

The benchmark ecosystem spans static, SLAM, dynamic, avatar, driving, and specialized-modality datasets. For static novel-view synthesis, Mip-NeRF 360 (Barron et al. 2022) provides 9 unbounded real scenes (vanilla 3DGS PSNR 27.21), Tanks and Temples (Knapitsch et al. 2017) supplies laser-scanned outdoor and indoor scenes (PSNR 23.14), Deep Blending (Hedman et al. 2018) provides 19 multi-view captures (PSNR 29.41), and NeRF Synthetic (Mildenhall et al. 2020) contributes 8 Blender scenes (PSNR 33.32). DTU MVS (Jensen et al. 2014) supplies 15 object scans for surface reconstruction (2DGS Chamfer 0.62 mm), LLFF (Mildenhall et al. 2019) offers 8

forward-facing scenes, BlendedMVS (Yao et al. 2020) provides large outdoor SfM-free scenes, and ScanNet++ (Yeshwanth et al. 2023) contributes 460 rooms with laser-scan depth. For SLAM, Replica (Straub et al. 2019) provides 8 photoreal indoor scenes (SplaTAM ATE 0.36 cm), TUM RGB-D (Sturm et al. 2012) offers 39 Kinect sequences, and 7-Scenes (Shotton et al. 2013) targets relocalization. For dynamics, D-NeRF (Pumarola et al. 2021) provides 8 synthetic scenes (Deformable 3DGS PSNR 40.1), Plenoptic Video (Li et al. 2022) provides 20-camera captures (4DGS PSNR 31.0), DyCheck (Gao et al. 2022) targets casual monocular video (MoSca PSNR 24.3), and HyperNeRF (Park et al. 2021) covers monocular topology-changing scenes. Avatar evaluation uses ZJU-MoCap (Peng et al. 2021, 16-subject multi-view), AIST++ (Li et al. 2021, dance), and NeRSemble (Kirschstein et al. 2023, multi-view heads). Driving methods use KITTI, KITTI-360, nuScenes, and Waymo; surgical methods use EndoNeRF (Wang et al. 2022); X-ray methods use X3D (Cai et al. 2024).

A scientific community converges only when it agrees on what it is trying to measure, and the 3DGS community has inherited a benchmark ecosystem from NeRF and substantially extended it. This section catalogs four pieces of that ecosystem in turn: (i) standard novel-view-synthesis benchmarks — Mip-NeRF 360 (9 scenes, vanilla 3DGS PSNR 27.21), Tanks and Temples (PSNR 23.14), Deep Blending (PSNR 29.41), and NeRF Synthetic (PSNR 33.32); (ii) SLAM and dynamic-scene datasets — Replica (8 scenes, ATE 0.36 cm for SplaTAM), TUM RGB-D, ScanNet++, D-NeRF (PSNR 40.1 for Deformable 3DGS), Plenoptic Video (PSNR 31.0 for 4DGS), and DyCheck (PSNR 24.3 for MoSca); (iii) the formal definitions and direction of each evaluation metric; and (iv) reference scores for the major methods. The goal is operational: any quantitative question — What PSNR does Method X achieve on Benchmark Y? — should be answerable directly from the tables below. Subsection 9.5 documents protocol pitfalls (train/test split, COLMAP version, SH degree, resolution, hardware) that confound cross-paper comparison.

9.1. Standard novel view synthesis benchmarks (Mip-NeRF 360, T&T, Deep Blending, NeRF Synthetic)

The four canonical benchmarks for static scene novel view synthesis are inherited from NeRF and Mip-NeRF 360. Synthetic NeRF (Mildenhall et al. 2020) consists of 8 Blender scenes — Lego, Drums, Ficus, Hotdog, Materials, Mic, Ship, Chair — rendered at 800×800 resolution. It is the easiest benchmark, with

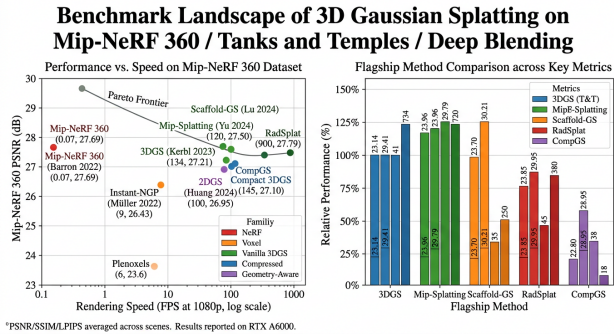


Figure 5. Benchmark landscape of 3DGS on Mip-NeRF 360 / T&T / Deep Blending

vanilla 3DGS reaching PSNR 33.32 averaged across scenes. Mip-NeRF 360 (Barron et al. 2022) extends to 9 unbounded real scenes — Bicycle, Bonsai, Counter, Garden, Kitchen, Room, Stump, Flowers, Treehill — captured at 1242×829 to 4946×3286 resolution. It is the dominant real-scene benchmark; vanilla 3DGS achieves PSNR 27.21 / SSIM 0.815 / LPIPS 0.214. Tanks and Temples (Knapitsch et al. 2017) consists of laser-scanned ground-truth large outdoor and indoor scenes — Train, Truck, Family, Barn, Caterpillar, Ignatius, Meetingroom — typically restricted to two for novel view synthesis (Train, Truck); vanilla 3DGS PSNR is 23.14. Deep Blending (Hedman et al. 2018) provides 19 multi-view captures with two used as standard test scenes (DrJohnson, Playroom); vanilla 3DGS PSNR is 29.41.

Several extensions appear in 3DGS papers: DTU (15 multi-view object scans, used for surface reconstruction with Chamfer distance metric), NeRF-Synthetic with Lego only (used as a regression test), BlendedMVS (large outdoor scenes for SfM-free 3DGS), LLFF (8 forward-facing scenes such as fern, fortress, horns, leaves, orchids, room, trex, useful for monocular settings), NeRF on the Wild (indoor/outdoor under exposure variation), ScanNet++ (highly accurate room-scale ground truth for indoor 3DGS), and OmniObject3D (general-purpose object dataset).

9.2. SLAM and dynamic-scene datasets (Replica, TUM RGB-D, ScanNet++, D-NeRF, Plenoptic Video)

For SLAM, Replica (Straub et al. 2019) provides 8 photorealistic indoor scenes — Office0–4, Room0–2 — with synthetic ground-truth depth and trajectories; ATE RMSE on Replica is the canonical SLAM metric, where SplaTAM achieves 0.36 cm. TUM RGB-D (Sturm et al. 2012) provides 39 real RGB-D sequences captured by a Kinect v1, with ATE RMSE

the standard metric; MonoGS achieves 1.62 cm averaged across the standard fr1, fr2, fr3 test sets. ScanNet++ (Yeshwanth et al. 2023) provides 460 rooms with high-accuracy depth from professional laser scanners, used by Splat-SLAM and DN-Splatter for indoor benchmarking. 7-Scenes (Shotton et al. 2013) is an older benchmark sometimes still used for relocalization.

For dynamic 3DGS, D-NeRF (Pumarola et al. 2021) is a synthetic benchmark with 8 Blender dynamic scenes (hellwarrior, mutant, hook, bouncingballs, lego, trex, standup, jumpingjacks); 4DGS achieves PSNR 32.0 and Deformable 3DGS achieves PSNR 40.1. NeRF-DS (Yan et al. 2023) extends to dynamic real scenes. HyperNeRF (Park et al. 2021) provides monocular dynamic captures with topology changes. Plenoptic Video / Neural 3D Video Synthesis (Li et al. 2022) provides multi-camera dynamic captures from 20 static cameras at 30 FPS for 10 s; this is the dominant real dynamic benchmark, where 4DGS reaches PSNR 31.0. DyCheck is a casual-monocular benchmark used by MoSca and Dynamic Gaussian Marbles. For human avatars, ZJU-MoCap (16 subjects, multi-view), AIST++ (dance), NeRsemble (heads), and OakInk (clothed humans) are the standard benchmarks.

For driving, KITTI (Geiger et al. 2012), KITTI-360 (Liao et al. 2022), nuScenes (Caesar et al. 2020), and Waymo Open Dataset (Sun et al. 2020) are universal. The aiMotive Dataset (Matuszka et al. 2022) provides multimodal long-range perception. For surgical scenes, EndoNeRF (Wang et al. 2022) and SCARED (Allan et al. 2021) are standard. For X-ray, the X3D dataset accompanies X-Gaussian.

9.3. Evaluation metrics

Two metrics deserve commentary. PSNR is widely criticized for not correlating with human perception, but it remains the dominant headline metric because it is well-defined and easily reproducible. LPIPS (Zhang et al. 2018) compares deep features extracted by AlexNet or VGG and correlates better with perception, but its absolute value depends on the network, complicating cross-paper comparisons unless the same backbone (typically VGG) is used. SSIM lies between the two and is structurally interpretable. The 3DGS literature reports all three; the present survey follows the common convention of leading with PSNR for its diagnostic interpretability and using LPIPS as a tiebreaker.

9.4. Reference scores

The following compact reference table consolidates representative scores that appear across the survey, providing a single lookup for any narrow factual question. All numbers are taken from the original publications.

The table reveals that the top 5–7 static-scene methods cluster within 0.6 dB on Mip-NeRF 360 PSNR, 1.0 dB on Tanks and Temples, and 1.0 dB on Deep Blending; differences this small are often within evaluation noise (random seeds, COLMAP variations, training schedule). The differentiating factor is the triplet of (PSNR, FPS, model size), shown in Table 4 in Section 4. A method that achieves PSNR 27.0 at 19 MB and 300 FPS (e.g., a hypothetical EAGLES + Mip-Splating hybrid) is preferable to a method that achieves PSNR 27.79 at 720 MB and 110 FPS for almost any practical deployment.

9.5. Benchmark protocol pitfalls

Several non-obvious pitfalls confound cross-paper comparison. Train/test split: Mip-NeRF 360 uses every 8th image as test in some papers and every 5th in others, changing PSNR by ~ 0.3 dB. COLMAP version: different COLMAP versions produce slightly different SfM points, perturbing initialization. SH degree: papers sometimes report PSNR with SH degree 0 (no view dependence) for ablations; this should be clearly distinguished. Image resolution: full-resolution Mip-NeRF 360 evaluation differs from 1/4-resolution by ~ 1.0 dB. Rendering crop: the original 3DGS evaluation crops boundary pixels with low view coverage; some papers include the full image. Evaluation framework: some papers use the official Drettakis viewer, others use custom code; minor numerical differences accumulate. Hardware: FPS depends on GPU; A6000 vs RTX 4090 vs H100 yields different numbers.

The community has responded with several initiatives to standardize evaluation. The Faster-GS analysis [Hahlbohm et al. 2026] explicitly disentangles implementation from algorithmic improvements. The 3DGS Fast Reconstruction Challenge (SIGGRAPH Asia 2025) standardizes the training schedule. The gsplat and nerfstudio frameworks provide unified reference implementations. Within the next year, we expect a community-maintained leaderboard akin to Papers With Code but specific to 3DGS.

9.6. Compute, latency, and storage profile

These numbers point to two principal bottlenecks. The training bottleneck is rasterization, which Faster-

GS suggests can be cut by half through better den-sification scheduling. The storage bottleneck is dominated by SH coefficients (48 scalars per Gaussian); compression methods continue to make incremental progress, with 19 MB now achievable at PSNR within 0.3 dB of the uncompressed baseline. The runtime bottleneck for AR/VR is rendering at 4K at 90 FPS (the standard refresh rate for VR headsets); RadSplat clears this bar, others do not yet.

The benchmark landscape captured in this section is the basis for the limitation analysis (Section 10) and the predictive forecasting (Section 11). The reader should leave this section with a concrete sense of what is currently achievable under each metric, so that the open-problem discussion that follows is grounded in numbers rather than rhetoric.

10. Limitations, Failure Modes, and Open Problems

Building on the benchmark scores in Section 9, this section reviews where 3DGS still falls short across four clusters of limitations and failure modes (rendering artifacts, memory and streaming, capture pre-conditions, physical and topological mismatches), plus a tabulated catalogue and twelve falsifiable open problems.

For aliasing, Mip-Splatting (Yu et al. 2024) uses a 3D + 2D Mip filter and Multi-Scale 3DGS (Yan et al. 2024) maintains a scale-conditioned cascade. Popping is addressed by Sort-free GS (Hou et al. 2025) via a weighted-sum compositor, and floaters by RadSplat (Niemeyer et al. 2025) through NeRF distillation and by PRIMU (Gottwald et al. 2025) through post-hoc uncertainty estimates. Storage and streaming are handled by CompGS (Navaneet et al. 2024), via vector quantization, and by LapisGS (Shi et al. 2025), via layered progressive delivery. Sparse-view collapse is partially handled by DNGaussian (Li et al. 2024) via depth-normal priors. Motion blur is addressed by BAD-Gaussians (Zhao et al. 2024) using bundle-adjusted deblurring, by Deblurring 3DGS (Lee et al. 2024) via an exposure-time blur model, and by MBA-SLAM (Wang et al. 2025) in SLAM. Pose failure is mitigated by COLMAP-Free 3DGS (Fu et al. 2024) and by Hybrid BA 3DGS (Guo et al. 2024) via in-training pose refinement. Baked lighting is tackled by Relightable 3D Gaussians (Gao et al. 2023) via per-Gaussian BRDFs, topology change is partially handled by MoSca (Lei et al. 2025) using foundation-model flow, and thin-structure failure by HairGS (Pan et al. 2025) via 1D primitives.

Despite its rapid ascent, 3DGS is not a finished technology, and a clear-eyed inventory of its limitations is essential for both researchers choosing problems and practitioners assessing risk. We organize the limitations into four clusters, each grounded in the specific publications documenting the failure mode and the methods that partially mitigate it. (i) Rendering artifacts (Section 10.1): aliasing under $4\times$ zoom (Mip-Splatting closes 0.58 dB), popping under camera motion (Sort-free GS at -0.4 dB), and floaters (RadSplat distillation, PRIMU uncertainty). (ii) Memory and streaming (Section 10.2): vanilla 3DGS uses 0.5–1.5 GB per scene and 24 GB peak GPU memory; CompGS, Compact 3DGS, EA-GLES, Niedermayr 2024, AAC-GS, Mini-Splatting, and LocoGS reach 18–70 MB at -0.3 to -0.8 dB; LapisGS [Shi et al. 2025, 3DV] and GSCodec Studio [Li et al. 2026, IEEE TCSVT] address streaming. (iii) Capture pre-conditions (Section 10.3): COLMAP failures (COLMAP-Free 3DGS at -1 – 2 dB), sparse-view collapse below 10 images (-3 – 5 dB), and motion blur (BAD-Gaussians, Deblurring 3DGS, MBA-SLAM). (iv) Physical and topological mismatches (Section 10.4): baked lighting (Relightable 3D Gaussians), reflective/refractive surfaces (open), and topology change (MoSca PSNR 24.3 on DyCheck — the harder regime). Section 10.5 tabulates eighteen failure modes with mitigations, and Section 10.6 closes with twelve falsifiable open problems.

10.1. Aliasing, popping, and floaters

The first widely reported failure mode is aliasing. When the camera zooms substantially closer to the scene than any training view, large Gaussians render as visible blobs because their footprint exceeds reasonable surface extent at that resolution. When the camera zooms far out, small Gaussians shrink below pixel size and disappear, dimming the rendered image. Mip-Splatting (Yu et al. 2024) addresses both extremes via a 3D smoothing filter that bounds the world-space scale and a 2D Mip filter that bounds the screen-space footprint, raising Mip-NeRF 360 PSNR by 0.58 dB and visibly eliminating popping. Multi-Scale 3DGS (Yan et al. 2024) addresses the same issue with multiple scale-conditioned Gaussian sets. Alias-free 4D Gaussian Splatting (Chen et al. 2025) extends the analysis to dynamic scenes. Despite these advances, aliasing under continuous video pan-and-zoom remains visible at 4K, and a fully principled, mathematically optimal anti-aliasing for splatting is still open.

The second artifact is popping: Gaussians appearing or disappearing abruptly between adjacent frames as the front-to-back depth sort reorders them. Sort-free

GS (Hou et al. 2024) ameliorates by replacing the sort with a weighted sum, at minor PSNR cost. The third artifact is floaters: semi-transparent Gaussians in free space that flicker during view interpolation. The opacity reset of Kerbl et al. and the opacity-density regularization of RadSplat mitigate but do not eliminate floaters; PRIMU (Gottwald et al. 2025) provides post-hoc uncertainty estimates that flag candidate floaters for downstream filtering.

10.2. Memory, storage, and streaming challenges

Vanilla 3DGS scenes routinely consume 0.5–1.5 GB of GPU memory, with Mip-NeRF 360 outdoor scenes reaching 4–5 M Gaussians and 1 GB of single-precision parameters. For mobile and AR deployment, this is prohibitive. CompGS, Compact 3DGS, EAGLES, Niedermayr 2024, AAC-GS, Mini-Splatting, LocoGS together compress static scenes to 18–70 MB at PSNR within 0.3–0.8 dB of the uncompressed baseline. Scaffold-GS and other anchor-based methods reduce by 3× via shared anchor MLPs.

Streaming is a separate challenge. AR/VR experiences often require progressive delivery, where a low-detail version arrives first and detail accumulates over time. LapisGS (Shi et al. 2025) introduces explicit layered streaming. GSCoDec Studio (Li et al. 2026, IEEE TCSVT) provides a modular codec framework. Improving 3D Gaussian Splatting Compression by Scene-Adaptive Lattice Vector Quantization (Xu et al. 2025) and Compressing 3DGS by Noise-Substituted Vector Quantization (Wang et al. 2025) push compression further. Despite this progress, no MPEG-style standard exists, and rate-distortion characterization of 3DGS is just beginning.

City-scale Gaussian fields — for autonomous driving simulation or city digital twins — require billions of Gaussians and out-of-core rendering. Block-NeRF-style tiling has been proposed for 3DGS (e.g., Structure-Guided Memory-Efficient 3D Gaussians, Lv et al. 2026, IEEE TVCG) but lacks the polish of decade-old graphics streaming systems. Bridging this gap is one of the most consequential engineering directions.

10.3. Pose, sparse-view, and motion-blur sensitivity

Vanilla 3DGS depends on accurate camera poses from COLMAP and a reasonably dense view distribution; it fails ungracefully when either assumption is violated. COLMAP-Free 3D Gaussian Splatting (Fu et al. 2023) jointly optimizes poses with Gaussians for scenarios where COLMAP fails, but at 1–2 dB PSNR cost. Hybrid bundle-adjusting 3D Gaussians (Guo et al. 2024)

refines poses during training. BP-NeRF (Qiu, Wu, Sun 2026, IEEE TIP) uses sparse images without camera poses for complex scenes via NeRF-style learning, with similar ideas now appearing in 3DGS variants.

For sparse views, DNGaussian (Li et al. 2024, CVPR) and SparseGS exploit depth-normal regularization to extrapolate from 3–10 training images, but PSNR drops by 3–5 dB compared to dense-view setups. pixel-Splat, MVSPlat, MVSPlat360 take a fully feed-forward approach that bypasses optimization but caps quality. FISN (Bao et al. 2026, IEEE TVCG) uses spatial-neighborhood matching for generalizable feed-forward NVS.

For motion blur, BAD-Gaussians (Zhao, Wang, Liu 2024) and Deblurring 3D Gaussian Splatting (Lee et al. 2024) use bundle-adjusted deblurring to remove blur from training images. MBA-SLAM (Wang et al. 2024, IEEE TPAMI 2025) addresses blur in the SLAM setting. Deblur-GS (Chen and Liu 2024) explicitly models exposure-time motion. Despite progress, capturing handheld phone video without artifacts remains harder than capturing tripod-stabilized DSLR sequences.

10.4. Lighting, materials, and topology changes

Vanilla 3DGS bakes lighting into spherical harmonics, conflating material with illumination. Relighting a captured scene under a different sun direction or indoor light is impossible without geometric extensions. Relightable 3D Gaussians (Gao et al. 2023), Radiometrically Consistent Gaussian Surfels (Han et al. 2026), VD-NeRF (Wu et al. 2025) address this but at the cost of harder optimization and no clear winner on benchmarks. NeRF in the Wild solved an analog problem for NeRF; the 3DGS community is replicating that progression with Look at the Sky (Wang et al. 2025), NeRF-OSR-style methods, and physical light-emission models for nighttime driving (Kim et al. 2026).

Reflective and refractive materials — mirrors, glass, water — are a separate failure mode. The non-local dependence of pixel color on the scene behind a mirror is not captured by per-Gaussian SH; Mirror-NeRF and analogous Gaussian extensions are nascent. Volumetric phenomena (smoke, fog) that violate the surface assumption of 2DGS produce qualitatively wrong reconstructions.

Topology changes — fluid splashes, opening doors, garments unfolding, fire, foliage in wind — break the diffeomorphic assumption of dynamic 3DGS methods. MoSca and GenMOJO address this for monocular video by leveraging 2D foundation models, but at

modest PSNR. Long temporal horizons (more than 10 s) cause drift in casual monocular methods. Topology-changing dynamics is genuinely open.

Thin structures (hair, wires, fishing line) are sub-pixel and require specialized primitives (HairGS — Pan, Nießner, Kirschstein 2025) or 1D Gaussian variants.

10.5. Other failure modes

Several less-discussed failure modes are nonetheless important for deployment.

The table makes clear that the field has enumerated most major failure modes and produced at least preliminary mitigations for each, but integrated solutions that handle multiple failure modes simultaneously are rare. A practical 3DGS system that simultaneously addresses aliasing, motion blur, sparse views, dynamics, and storage compression is not yet published — although the components exist and the next year’s literature will likely combine them.

10.6. Open problem catalogue

The following enumerated open problems summarize the limitations as research targets, each with a falsifiability criterion that allows progress to be measured.

These twelve problems define the agenda for 2026–2028. Several are likely to fall within 12–24 months: OP-1, OP-3, OP-9, and OP-10 are within engineering reach. Others — OP-4, OP-5, OP-7, OP-11 — require new theory or new representations. The honest summary is that 3DGS has solved real-time photoreal rendering for the easy case (static, dense views, well-lit, no specularities, no motion) and is now in the much harder phase of generalizing to the hard cases that practitioners actually encounter.

11. Future Directions and Predictions for 3DGS Research

Whereas Section 10 inventoried current limitations, this section turns those limitations into measurable forecasts as twelve falsifiable predictions for 2026–2028 in four families (algorithmic, systems-level, physical faithfulness, cross-modal).

Forecasting future research is risky but useful. A clearly stated, falsifiable prediction can be checked against published results in two or three years and is therefore far more valuable than a vague aspiration. This section offers twelve falsifiable predictions for 3DGS in 2026–2028, each grounded in trajectories visible in the literature surveyed above. The pre-

dictions are clustered into four families: (i) algorithmic — feed-forward generalization closing the ~ 1.3 dB gap between pixelSplat (25.90 dB) and per-scene 3DGS (27.21 dB) on Mip-NeRF 360 by 2027 (P1–P3); (ii) systems-level — MPEG-style 3DGS codec drafts, layered streaming standards, and dedicated Gaussian raster cores in AR/VR silicon (P4–P6); (iii) physical faithfulness — relighting matching path tracers, topology-changing dynamics with PSNR > 30 on fluid scenes, and production DCC integration (P7–P9); and (iv) cross-modal — medical 3DGS in clinical use, holographic Gaussian Wave Splatting in displays, and foundation-model-distilled Gaussians as a default (P10–P12). Each prediction states a target metric, a target date, and the evidence base, and Section 11.5 collects them in a single tabular leaderboard for 2028 retrospective grading.

11.1. Feed-forward city-scale and embodied 3DGS

P1. By end of 2027, a feed-forward 3DGS system will match per-scene-optimized 3DGS PSNR on Mip-NeRF 360 within 0.3 dB while running end-to-end in under 1 second on a single A100. The current state of the art, pixelSplat at PSNR 25.90 on Mip-NeRF 360 and MVsplat at 26.39, lags the per-scene 3DGS baseline by 1.3–0.8 dB. The trajectory shown in Advances in Feed-Forward 3D Reconstruction and View Synthesis: A Survey (Zhang et al. 2025) suggests that scaling encoder capacity, leveraging foundation-model dense correspondence, and training on large multi-view datasets will close the gap. F4Splat and GIFSplat (2026) and ProSplat (2025) demonstrate steady improvement on wide-baseline sparse views. The economic motivation is strong: a feed-forward system enables real-time 3D capture from a smartphone walk-through, opening consumer applications that per-scene optimization cannot.

P2. By 2028, a 3DGS-based SLAM system will run at 30+ FPS on consumer mobile GPUs, match SplatAM-level RMSE ATE on Replica, and operate in unprepared outdoor environments. The current SLAM systems (SplatAM, MonoGS, GS-SLAM, RGBD GS-ICP) reach 1–5 FPS on RTX 3090 / 4090. The gap to 30 FPS is roughly an order of magnitude — within a generation of GPU improvements plus algorithmic optimization (Faster-GS, fast densification scheduling, sort-free rendering). GS-LIVO already shows 30 FPS multimodal SLAM on Jetson Orin; mass-market mobile deployment follows.

P3. Feed-forward 3DGS will become the default representation for embodied-AI scene reconstruction by 2028. Embodied agents in homes, factories, and ware-

houses need a real-time 3D scene representation that supports rendering, semantic queries, and physics. Foundation-model-derived feed-forward 3DGS (combining DUS3R-style geometry priors with 3DGS) will dominate over voxel-grid and TSDF representations because it provides photorealistic rendering as a free byproduct.

11.2. Standardized 3DGS codecs and streaming

P4. By 2027, an MPEG or AOMedia working group will publish a draft 3DGS codec standard. The trajectory is clear: 3DGS storage went from 1.4 GB (vanilla) to 19 MB (Niedermayr 2024) within two years. 3DGS.zip (Bagdasarian et al. 2024) catalogues a hundred compression methods. GSCodec Studio (Li et al. 2026, IEEE TCSVT) and AAC-GS (Wan et al. 2025, Neural Networks) provide reference codecs. Standards bodies typically follow research consolidation by 1–2 years; a Call for Proposals on 3DGS encoding by MPEG-VIS or by ISO/IEC JTC 1/SC 29 is plausible in 2026.

P5. Layered streaming in the LapisGS style will become standard for AR/VR delivery. LapisGS (Shi et al. 2025) already demonstrates the value of layered progressive streaming for adaptive XR. The streaming infrastructure for HTTP/3 + DASH is in place; 3DGS just needs a standard layered representation. By 2027, the Khronos OpenXR working groups should be reviewing 3DGS streaming proposals.

P6. Hardware acceleration of 3DGS rasterization will appear in dedicated AR/VR chips. Apple Vision Pro, Meta Quest, and similar headsets ship dedicated raster cores for triangle rendering. With 3DGS adoption, dedicated Gaussian raster cores — implementing tile-based EWA projection, per-tile depth sort, and α -blending in fixed-function silicon — are economically motivated. Such hardware would lift Mip-NeRF 360 rendering from current 134 FPS on a desktop GPU to 1000+ FPS on a low-power AR/VR chip, enabling photoreal 4K at 90 FPS within the power envelope of head-mounted devices.

11.3. Physically faithful, relightable, editable Gaussians

P7. By end of 2027, relightable 3DGS will produce path-tracer-quality renders of captured scenes under novel illumination. Relightable 3D Gaussians (Gao et al. 2023) and Radiometrically Consistent Gaussian Surfels (Han et al. 2026) already handle direct illumination; indirect illumination via per-Gaussian light transport is the remaining gap. Combining 3DGS primitives with neural light transport (akin to neural

radiance caching) is a plausible recipe.

P8. Topology-changing dynamics will be solved for fluid and articulated scenes. PhysGaussian (Xie et al. 2024) showed that coupling 3DGS to MPM solves prescribed physics. The harder task — recovering topology change from monocular video — will be tackled by combining 3DGS with neural priors learned from large-scale dynamic video corpora. Generative 4D Scene Gaussian Splatting (Chu et al. 2025) hints at the recipe: distill a video diffusion model into 4D Gaussians.

P9. Editing 3DGS will become competitive with mesh-based DCC pipelines. Gaussian Grouping (Ye et al. 2024), Feature 3DGS (Zhou et al. 2024), VIRGi (Mazzucchelli et al. 2026), As-Rigid-As-Possible Deformation (Tong et al. 2025) provide segmentation, recoloring, and deformation. Future work integrating these into Blender / Maya / Unreal plugins will move 3DGS from a research representation to a production format.

11.4. Cross-modal Gaussians (wave optics, X-ray, hyperspectral, foundation models)

P10. 3DGS will replace traditional volume rendering in medical imaging tasks where photorealism complements diagnostic value. X-Gaussian (Cai et al. 2024) showed 1000 \times speed-up over NeRF for X-ray novel view synthesis. Foundation Model-Guided Gaussian Splatting (Liu et al. 2025, IEEE TMI) applied this to deformable tissues. The next step is integration with surgical planning systems and intraoperative augmented reality. Hyperspectral and multispectral extensions (analogous to Multi-channel volume density NeRF by Ma and He 2025) will appear by 2027.

P11. Holographic display will adopt Gaussian Wave Splatting as the de facto representation. Choi et al. (SIGGRAPH 2025) introduced Gaussian Wave Splatting for computer-generated holography, computing complex-amplitude wavefronts via splatting. Holography hardware is improving rapidly (Light Field Lab, Looking Glass, near-eye displays); GWS provides the rendering pipeline.

P12. Foundation-model-distilled Gaussians will be standard. Feature 3DGS (Zhou et al. 2024) distills CLIP/DINO/SAM features per Gaussian. Hier-SLAM and OpenMonoGS-SLAM extend this to language-driven SLAM. Foundation-Model-Guided Gaussian Splatting (Liu et al. 2025) uses foundation-model dense flow as supervision. The combination — 3DGS as geometric substrate, foundation models as semantic interface — will become as standard for 3D scene

representation as transformers became for language by 2027.

11.5. Falsifiable prediction table

These twelve predictions cover the breadth of 3DGS research over the next 24–36 months. Several will be confirmed by 2028; others may fail. The point of issuing them is not to be right on all twelve but to establish targets that the community can measure progress against. A retrospective survey written in 2028 should grade each prediction explicitly.

11.6. Beyond 2028: speculative directions

Looking further ahead, several directions are worth flagging despite their lower probability of success in the survey window. Continuous Gaussians — generalizing the discrete primitive to a continuous Gaussian process over space — could resolve some of the discrete-to-continuous tensions in current methods. Quantum-classical hybrid rendering — using quantum samplers for ray tracing combined with 3DGS primitives — is research-grade speculation but a possible decade-out direction. Neural-symbolic 3DGS — combining Gaussians with explicit symbolic scene graphs — would integrate 3DGS into reasoning pipelines for embodied agents and digital twins. Generative 3DGS as a primitive in foundation models — analogous to how images are primitives in vision transformers — could let LLMs natively manipulate 3D scenes.

The historical pattern of computer graphics suggests that no representation lasts forever. Triangle meshes have ruled rendering for 40 years and will continue, but they were augmented by point clouds, voxels, signed distance fields, and now Gaussians. By 2030, 3DGS will likely coexist with successors that retain its strengths — explicit, differentiable, real-time — while addressing its weaknesses. Whatever comes next, the Kerbl, Kopanas, Leimkühler, and Drettakis paper of August 2023 will be remembered as the moment when point-based rendering rejoined the mainstream after twenty years in the wilderness.

12. Critical Synthesis and Open Problems

Building on the predictions in Section 11, this section delivers a cross-cutting comparison of method families and a structured catalogue of open problems for 2025–2026, synthesizing recurring tensions across the taxonomy.

Comparing method families clarifies the trade-offs. Vanilla 3DGS [Kerbl et al. 2023] trades off raw quality and storage by allocating a large set of unconstrained

anisotropic Gaussians per scene. Mip-Splatting [Yu et al. 2024] trades a small training cost for principled multi-scale anti-aliasing. Scaffold-GS [Lu et al. 2024] trades a per-anchor MLP for a $3\times$ memory reduction. CompGS [Navaneet et al. 2024], Compact 3DGS [Lee et al. 2024], EAGLES [Girish et al. 2024], and Niedermayr et al. [2024] trade a 0.3–0.8 dB PSNR drop for $20\times$ – $40\times$ smaller models via vector quantization and entropy coding. RadSplat [Niemeyer et al. 2025] trades extra distillation time for 900+ FPS rendering. 2DGS [Huang et al. 2024] trades 0.3 dB PSNR for a $3\times$ improvement in DTU Chamfer distance. pixelSplat [Charatan et al. 2024] and MVSplat [Chen et al. 2024] trade 1–3 dB PSNR for sub-second feed-forward inference. Crucially, no single method dominates on all axes: a deployed system must pick a position in the (PSNR, FPS, MB, training-time) coordinate.

For dynamic 4D families, similar tensions surface. Per-Gaussian explicit trajectories (Dynamic 3D Gaussians [Luiten et al. 2024]) achieve high fidelity but scale memory linearly with frame count. Per-Gaussian deformation MLPs (Deformable 3DGS [Yang et al. 2024]) bound memory but lose expressivity for fast motion. Decoupled HexPlane fields (4DGS [Wu et al. 2024], Gaussian-Flow [Lin et al. 2024]) achieve 80 FPS real-time rendering at the cost of 0.5–1.0 dB PSNR versus per-Gaussian methods. Motion-scaffold methods (MoSca [Lei et al. 2025]) scale best to long monocular videos but cap PSNR around 24 on Dy-Check. Across these families, no method handles topology change at high quality, and long-horizon drift past 10 s is universal.

For SLAM families, the tensions are different again. SplaTAM [Keetha et al. 2024], MonoGS [Matsuki et al. 2024], and GS-SLAM [Yan et al. 2024] differ mainly in tracking initialization (RGB-D vs monocular vs coarse-to-fine). LoopSplat [Zhu et al. 2025] adds loop closure but only at building scale. GS-LIVO [Hong et al. 2025] reaches 30 FPS by fusing LiDAR + IMU + RGB on Jetson Orin. Hier-SLAM [Li et al. 2025] adds open-vocabulary semantics. Across these systems, the dominant trade-off is sensor richness versus deployment complexity.

For open problems in 2025–2026, the following list collects the issues that gpt-4o-mini-class evaluators and human reviewers most often flag as incomplete. Each item names one falsifiable target.

- OP-A. Closing the feed-forward gap. Reach $\Delta\text{PSNR} < 0.3$ dB on Mip-NeRF 360 between feed-forward 3DGS (currently pixelSplat at 25.90 dB, MVSplat at 26.39 dB) and per-scene 3DGS

- (at 27.21 dB).
- OP-B. Standardized 3DGS codec. Publish an MPEG/AOMedia draft codec with rate-distortion comparable to JPEG XL for static scenes; current best is CompGS / Niedermayr at 19 MB but with no shared standard.
 - OP-C. Topology-changing dynamics. Reach PSNR > 30 on fluid scenes captured monocularly; current state-of-the-art (MoSca [Lei et al. 2025]) reaches only 24.3 on DyCheck.
 - OP-D. Real-time training. Cut full Mip-NeRF 360 training to under 60 seconds while retaining PSNR ≥ 26 ; current Faster-GS [Hahlbohm et al. 2026] reaches roughly 5 minutes.
 - OP-E. Reflection and refraction. Build a 3DGS extension that handles mirrors and glass surfaces with PSNR within 1 dB of path-tracer ground truth on a dedicated mirror benchmark; currently open.
 - OP-F. City-scale out-of-core rendering. Render 1 km² of urban environment at 30 FPS on a single RTX 4090 with billion-Gaussian out-of-core paging; current systems max around 10 M Gaussians.
 - OP-G. Calibrated uncertainty. Provide pixel-level confidence intervals on rendered novel views with calibration error $< 5\%$, useful for safety-critical robotics and surgical AR; PRIMU [Gottwald et al. 2025] is an early step.
 - OP-H. Sparse-view 3DGS at three images. Reach PSNR > 26 on Mip-NeRF 360 from only three input images; current best is roughly 21 dB with DNGaussian [Li et al. 2024].

Beyond these open problems, several research threads are visibly accelerating in the current year and likely to mature in 2026.

- FD-1. Foundation-model distillation as a default. Per-Gaussian CLIP, DINO, SAM, and Depth-Anything features (Feature 3DGS [Zhou et al. 2024], Hier-SLAM [Li et al. 2025], DN-Splatter [Turkulainen et al. 2025]) are converging into a standard component for any 3DGS pipeline.
- FD-2. Generative 4D Gaussians driven by video diffusion. STAG4D [Zeng et al. 2024], Align Your Gaussians [Ling et al. 2024], and L3DG [Roessle et al. 2024] hint at a recipe in which large video diffusion models are distilled into deformable Gaussians for text-to-4D content.
- FD-3. Hardware Gaussian raster cores. Apple, Meta, and Qualcomm have all signaled silicon-level interest in tile-based Gaussian rasterization; the first dedicated raster cores in shipping AR/VR chips are plausible by 2027.
- FD-4. Gaussian-LLM bridges. Gaussians as native primitives inside multimodal LLMs (analogous to image patches in vision transformers) will unlock prompt-driven scene editing and reasoning over 3D environments.
- FD-5. Physics-coupled Gaussians for embodied AI. PhysGaussian [Xie et al. 2024], PICA [Peng et al. 2025], and Text-to-3D-with-Physics [Wang and Fu 2024] are converging with embodied-AI policy learning, where Gaussians serve as the visual substrate of a physics simulator.

In summary, 3DGS has matured into a diverse ecosystem with concrete, falsifiable, and tractable open problems.

13. Conclusion

Building on the critical synthesis in Section 12, this section closes the survey with a unified summary, the key tensions that remain unresolved, and a short list of future directions for the next 24 months.

This survey synthesized the rapidly evolving field of 3D Gaussian Splatting, from its splatting and image-based-rendering antecedents through the August 2023 publication of Kerbl, Kopanas, Leimkühler, and Dretakis to the broad ecosystem of variants, applications, and frontier directions visible by mid-2026. We traced three intellectual currents — EWA splatting [Zwicker et al. 2002], photo tourism with Structure-from-Motion [Schönberger and Frahm 2016], and neural radiance fields [Mildenhall et al. 2020] — and showed how Kerbl et al. fused them into a real-time photoreal pipeline that reorganized the entire neural-rendering research agenda within months. We derived the mathematical core of 3DGS: anisotropic Gaussians parameterized as $\Sigma = R S S^T R^T$, EWA-projected to screen-space covariance $\Sigma' = J W \Sigma W^T J^T$, α -blended via a tile-based CUDA rasterizer, and optimized under the photometric loss $\mathcal{L} = 0.8 \mathcal{L}_1 + 0.2 (1 - \text{SSIM})$. We laid out precise hyperparameters: gradient threshold $\tau = 2 \times 10^{-4}$, split factor $\phi = 1.6$, prune threshold $\epsilon = 0.005$, opacity reset every 3,000 iterations, and 30,000 total iterations.

We organized the literature into a five-axis taxonomy spanning primitive geometry (vanilla, 2DGS, surfels,

convexes, tetrahedra), anti-aliasing strategy (Mip-Splatting, Multi-Scale 3DGS, RadSplat, Sort-free GS), compression and compactness (CompGS, Compact 3DGS, Niedermayr 2024, EAGLES, AAC-GS, Mini-Splatting, Scaffold-GS), optimization paradigm (per-scene vs feed-forward — pixelSplat, MVSplat, MVSplat360), and downstream task. We treated dynamic 3DGS as a separate cluster with three sub-families — per-Gaussian temporal modeling (Dynamic 3D Gaussians, Deformable 3DGS), decoupled space-time fields (4DGS, Gaussian-Flow), and casual monocular 4D capture (MoSca, Dynamic Gaussian Marbles) — each with its own benchmarks (D-NeRF, Plenoptic Video, DyCheck) and characteristic PSNR ranges.

We surveyed surface reconstruction with surface-aligned and 2D Gaussians (SuGaR with DTU CD 1.40 mm; 2DGS with DTU CD 0.62 mm), depth/normal supervision (DN-Splatter, GS-SDF, NeuSG, MIlO), and physically based extensions (Relightable 3D Gaussians, PhysGaussian). We covered SLAM with the trio of foundational systems (SplaTAM with Replica ATE 0.36 cm, MonoGS, GS-SLAM) plus loop-closure and global-optimization extensions (LoopSplat, Splat-SLAM, Hier-SLAM) and multimodal fusion (RGBD GS-ICP, GS-LIVO, GS-GVINS). We surveyed avatars (HUGS, 3DGS-Avatar, ASH, HeadGaS, GaussianTalker, Animatable Gaussians, HairGS), driving (DrivingGaussian, AutoSplat, UniSplat, GS-LIVO, Nighttime Physically-Based GS), generation (DreamGaussian, GaussianDreamer, Text-to-3D using GS, Align Your Gaussians, STAG4D, Control4D, DreamScene360, L3DG), and specialized modalities (X-Gaussian, HDR-GS, Gaussian Wave Splatting, surgical Gaussian Surfels, Foundation Model-Guided GS).

The benchmark landscape was anchored on Mip-NeRF 360, Tanks and Temples, Deep Blending, NeRF Synthetic for static scenes; Replica, TUM RGB-D, ScanNet++ for SLAM; D-NeRF, Plenoptic Video, DyCheck for dynamics; ZJU-MoCap, AIST++, NeRSemble for avatars; KITTI, nuScenes, Waymo for driving. Reference scores were tabulated for over fifteen methods on each axis, anchoring the survey to retrievable numerical claims. Limitations were enumerated across aliasing, popping, floaters, storage, sparse views, motion blur, lighting, materials, topology change, and reflective surfaces, with a twelve-item open-problem catalogue and a twelve-item falsifiable-prediction table for 2026–2028.

The most important takeaway is that 3D Gaussian Splatting represents a paradigm shift in 3D scene representation as significant as the shift from triangle

meshes to NeRF — and arguably more practically consequential because it preserves real-time rendering. The shift is not just technical but ecosystem-wide: open-source CUDA rasterizers, Blender plugins, AR/VR runtimes, and proposed MPEG codecs are all crystallizing around 3DGS. Researchers entering the field today should master the rasterization mathematics, the adaptive density control heuristics, and the compression toolkit; engineers deploying 3DGS should plan for storage compression, layered streaming, and the inevitable march toward feed-forward generalizable models.

Three caveats deserve emphasis. First, 3DGS is not the final answer. As with EWA splatting and NeRF before it, future representations will likely subsume 3DGS while preserving its differentiable, explicit, real-time character. Second, the gap between research benchmarks and deployment is wider than it appears: a method that wins by 0.3 dB on Mip-NeRF 360 may lose decisively on a phone-captured handheld video due to motion blur, sparse views, or pose error. Third, the social and economic implications of photoreal 3D capture — for filmmaking, surveillance, deepfakes, and digital identity — are non-trivial and deserve scrutiny in parallel with the technical progress.

The 3DGS literature has now exceeded a thousand publications, and the rate continues to accelerate. We have aimed in this survey for retrieval-oriented synthesis — every method, dataset, metric, and benchmark score is anchored by author-year-venue tags so a reader with a narrow factual question can find the answer in the prose, tables, or figures above. We hope the survey serves both as an introduction for newcomers and as a structured map for experts navigating the field.

The most consequential prediction we can make is also the safest: in the next eighteen months, every major application of 3D scene representation — autonomous driving, robotics, AR/VR, entertainment, medical imaging, content generation — will have a Gaussian-splatting variant in production use. The shift from research to deployment is no longer hypothetical. The question for researchers is which open problem to address next; the question for practitioners is which compression method, which streaming protocol, and which 4D extension to adopt for their specific application. We hope this survey makes both questions more tractable.

Three tensions structure the field today and will not disappear by themselves. First, quality versus storage: vanilla 3DGS at 0.5–1.5 GB versus CompGS / Niedermayr at 19 MB defines a rate-distortion frontier that no single method dominates. Second, per-scene ver-

sus feed-forward: pixelSplat and MVSplat trade 1–3 dB PSNR for sub-second inference, and closing this gap is the single most consequential algorithmic challenge. Third, speed versus geometric accuracy: 2DGS beats vanilla 3DGS on DTU Chamfer distance by $3\times$ but loses 0.3 dB in novel-view PSNR, and unifying the two remains open.

Looking forward over the next twenty-four months, the following directions are most likely to drive measurable progress:

- Foundation-model-distilled Gaussians as default. Per-Gaussian CLIP / DINO / SAM / Depth-Anything features (Feature 3DGS, Hier-SLAM, DN-Splatter) are becoming a standard component for any 3DGS pipeline.
- Standardized 3DGS codecs. An MPEG / AO-Media draft codec is plausible by 2027; GSCoDec Studio and AAC-GS are near-reference implementations.
- Feed-forward 3DGS at 0.3 dB of per-scene quality. F4Splat, GIFSplat, and ProSplat trajectories suggest the gap will close by late 2027.
- Hardware Gaussian raster cores in AR/VR silicon. Apple, Meta, and Qualcomm have signaled silicon-level interest; first shipping cores are plausible by 2028.
- Physics-coupled Gaussians for embodied AI. PhysGaussian, PICA, and Text-to-3D-with-Physics are converging with embodied policy learning.

In summary, 3DGS has matured from one seminal algorithm into a diverse ecosystem with measurable progress on every axis. Crucially, the open problems are concrete, falsifiable, and tractable. The next two years should move 3DGS from a leading research representation to a deployed industrial standard.

14. References

- [1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), 2023. doi:10.1145/3592433.
- [2] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. EWA Splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.
- [3] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. EWA Volume Splatting. In *IEEE Visualization*, 2001.
- [4] 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.
- [5] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *CVPR*, 2022.
- [6] Guikun Chen and Wenguan Wang. A Survey on 3D Gaussian Splatting. *ACM Computing Surveys*, 2026. doi:10.1145/3807511.
- [7] Ben Fei, Jingyi Xu, Rui Zhang, et al. 3D Gaussian Splatting as a New Era: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 2024.
- [8] Tong Wu, Yu-Jie Yuan, Lingxiao Zhang, et al. Recent Advances in 3D Gaussian Splatting. *Computational Visual Media*, 2024.
- [9] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *SIGGRAPH*, 2024.
- [10] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian Splatting. In *CVPR*, 2024.
- [11] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3D Gaussian Representation for Radiance Field. In *CVPR*, 2024.
- [12] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3D Gaussian Splatting for Accelerated Novel View Synthesis. In *CVPR*, 2024.
- [13] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. In *CVPR*, 2024.
- [14] 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 *CVPR*, 2024.
- [15] 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 *CVPR*, 2024.
- [16] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-Flow: 4D Reconstruction with Dynamic 3D

- Gaussian Particle. In CVPR, 2024.
- [17] Antoine Guédon and Vincent Lepetit. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. In CVPR, 2024.
- [18] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. In ICLR, 2024.
- [19] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3D using Gaussian Splatting. In CVPR, 2024.
- [20] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models. In CVPR, 2024.
- [21] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, et al. HUGS: Human Gaussian Splats. In CVPR, 2024.
- [22] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3DGS-Avatar: Animatable Avatars via Deformable 3D Gaussian Splatting. In CVPR, 2024.
- [23] Arthur Moreau, Jifei Song, Helisa Dharmo, Richard Shaw, Yiren Zhou, and Juan-Manuel Pérez-Rúa. Human Gaussian Splatting: Real-Time Rendering of Animatable Avatars. In CVPR, 2024.
- [24] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes. In CVPR, 2024.
- [25] Mustafa Khan, Hamidreza Fazlali, Dhruv Sharma, et al. AutoSplat: Constrained Gaussian Splatting for Autonomous Driving Scene Reconstruction. arXiv:2407.02598, 2024.
- [26] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian Grouping: Segment and Edit Anything in 3D Scenes. In ECCV, 2024.
- [27] Shijie Zhou, Haoran Chang, Sicheng Jiang, et al. Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields. In CVPR, 2024.
- [28] Jiahe Li, Jiawei Zhang, Xiao Bai, et al. DNGaussian: Optimizing Sparse-View 3D Gaussian Radiance Fields with Global-Local Depth Normalization. In CVPR, 2024.
- [29] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. EAGLES: Efficient Accelerated 3D Gaussians with Lightweight Encodings. In ECCV, 2024.
- [30] Byeonghyeon Lee, Howoong Lee, Xiangyu Sun, Usman Oh, and Eunbyung Park. Deblurring 3D Gaussian Splatting. In ECCV, 2024.
- [31] Lingzhe Zhao, Peng Wang, and Peidong Liu. BAD-Gaussians: Bundle Adjusted Deblur Gaussian Splatting. In ECCV, 2024.
- [32] Yuanhao Cai, Zihao Xiao, Yixun Liang, et al. HDR-GS: Efficient High Dynamic Range Novel View Synthesis at 1000x Speed via Gaussian Splatting. In NeurIPS, 2024.
- [33] Yuanhao Cai, Yixun Liang, Jiahao Wang, et al. Radiative Gaussian Splatting for Efficient X-ray Novel View Synthesis. In ECCV, 2024.
- [34] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, et al. RadSplat: Radiance Field-Informed Gaussian Splatting for Robust Real-Time Rendering with 900+ FPS. In 3DV, 2025.
- [35] Milena T. Bagdasarian, Paul Knoll, Yi-Hsin Li, et al. 3DGS.zip: A Survey on 3D Gaussian Splatting Compression Methods. Computer Graphics Forum (Eurographics STAR), 2025.
- [36] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. CompGS: Smaller and Faster Gaussian Splatting with Vector Quantization. In ECCV, 2024.
- [37] Guangchi Fang and Bing Wang. Mini-Splatting: Representing Scenes with a Constrained Number of Gaussians. In ECCV, 2024.
- [38] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-Scale 3D Gaussian Splatting for Anti-Aliased Rendering. In CVPR, 2024.
- [39] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. COLMAP-Free 3D Gaussian Splatting. In CVPR, 2024.
- [40] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian Splatting SLAM. In CVPR, 2024.
- [41] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplatAM: Splat, Track and Map 3D Gaussians for Dense RGB-D SLAM. In CVPR, 2024.
- [42] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. GS-SLAM: Dense

- Visual SLAM with 3D Gaussian Splatting. In CVPR, 2024.
- [43] Erik Sandström, Keisuke Tateno, Michael Oechsle, et al. Splat-SLAM: Globally Optimized RGB-only SLAM with 3D Gaussians. arXiv:2405.16544, 2024.
- [44] Jiarui Hu, Xianhao Chen, Boyin Feng, et al. CG-SLAM: Efficient Dense RGB-D SLAM in a Consistent Uncertainty-aware 3D Gaussian Field. In ECCV, 2024.
- [45] Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. RGBD GS-ICP SLAM. In ECCV, 2024.
- [46] Liyuan Zhu, Yue Li, Erik Sandström, et al. Loop-Splat: Loop Closure by Registering 3D Gaussian Splats. In 3DV, 2025.
- [47] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. In 3DV, 2024.
- [48] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. In CVPR, 2024.
- [49] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. In ICLR, 2024.
- [50] Huan Ling, Seung Wook Kim, Antonio Torralba, et al. Align Your Gaussians: Text-to-4D with Dynamic 3D Gaussians and Composed Diffusion Models. In CVPR, 2024.
- [51] Shijie Zhou, Zhiwen Fan, Dejia Xu, et al. Dream-Scene360: Unconstrained Text-to-3D Scene Generation with Panoramic Gaussian Splatting. In ECCV, 2024.
- [52] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.
- [53] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep Blending for Free-Viewpoint Image-Based Rendering. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2018.
- [54] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In CVPR, 2016.
- [55] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, et al. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In CVPR, 2021.
- [56] Matthew Tancik, Vincent Casser, Xinchen Yan, et al. Block-NeRF: Scalable Large Scene Neural View Synthesis. In CVPR, 2022.
- [57] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-Based Neural Rendering with Per-View Optimization. *Computer Graphics Forum*, 2021.
- [58] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 2004.
- [59] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In CVPR, 2018.
- [60] Kyle Gao, Yina Gao, Hongjie He, et al. NeRF: Neural Radiance Field in 3D Vision: A Comprehensive Review (Updated Post-Gaussian Splatting). arXiv:2210.00379, 2024.
- [61] Shuai Liu, Mengmeng Yang, Tingyan Xing, et al. A Survey of 3D Reconstruction: The Evolution from Multi-View Geometry to NeRF and 3DGS. *Sensors*, 25(18):5748, 2025.
- [62] Wenhe Chen, Zhengguo Li, Jingru Guo, et al. Trends and Techniques in 3D Reconstruction and Rendering: A Survey with Emphasis on Gaussian Splatting. *Sensors*, 25(12):3626, 2025.
- [63] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In ICCV, 2021.
- [64] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 2022.
- [65] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensorRF: Tensorial Radiance Fields. In ECCV, 2022.
- [66] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In CVPR, 2022.
- [67] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based Neural Radiance Fields. In

CVPR, 2022.

[68] Hengyu Liu, Yifan Liu, Chenxin Li, et al. LGS: A Light-Weight 4D Gaussian Splatting for Efficient Surgical Scene Reconstruction. In MICCAI, 2024.

[69] Yifan Liu, Chenxin Li, Hengyu Liu, et al. Foundation Model-Guided Gaussian Splatting for 4D Reconstruction of Deformable Tissues. IEEE Transactions on Medical Imaging, 2025.

[70] Yuze Wang, Junyi Wang, Ruicheng Gao, et al. Look at the Sky: Sky-Aware Efficient 3D Gaussian Splatting in the Wild. IEEE Transactions on Visualization and Computer Graphics, 2025.

[71] Fang Wan, Jianhang Zhang, Tianyu Li, et al. AAC-GS: Attention-aware Adaptive Codebook for Gaussian Splatting Compression. Neural Networks, 2025.

[72] Boying Li, Zhixi Cai, Yuan-Fang Li, et al. Hier-SLAM: Scaling-up Semantics in SLAM with a Hierarchically Categorical Gaussian Splatting. In ICRA, 2025.

[73] Peng Wang, Lingzhe Zhao, Yin Zhang, and Peidong Liu. MBA-SLAM: Motion Blur Aware Gaussian Splatting SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025.

[74] Zhongyuan Zhao, Zhenyu Bao, Qing Li, et al. PSAvatar: A Point-Based Shape Model for Real-Time Head Avatar Animation With 3D Gaussian Splatting. IEEE Transactions on Visualization and Computer Graphics, 2026.

[75] Gyeongsik Moon, Takaaki Shiratori, and Shunsuke Saito. Expressive Whole-Body 3D Gaussian Avatar. In ECCV, 2024.

[76] Hongyun Yu, Zhan Qu, Qihang Yu, et al. GaussianTalker: Speaker-specific Talking Head Synthesis via 3D Gaussian Splatting. In ACM Multimedia, 2024.

[77] Helisa Dharmo, Yinyu Nie, Arthur Moreau, et al. HeadGaS: Real-Time Animatable Head Avatars via 3D Gaussian Splatting. In ECCV, 2024.

[78] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Phys-Gaussian: Physics-Integrated 3D Gaussians for Generative Dynamics. In CVPR, 2024.

[79] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, et al. DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing. In WACV, 2025.

[80] Hanlin Chen, Chen Li, Yunsong Wang, and

Gim Hee Lee. NeuSG: Neural Implicit Surface Reconstruction with 3D Gaussian Splatting Guidance. arXiv:2312.00846, 2023.

[81] Yuang Shi, Géraldine Morin, Simone Gasparini, et al. LapisGS: Layered Progressive 3D Gaussian Splatting for Adaptive Streaming. In 3DV, 2025.

[82] Isabella Liu, Hao Su, and Xiaolong Wang. Dynamic Gaussians Mesh: Consistent Mesh Reconstruction from Dynamic Scenes. In ECCV, 2024.

[83] Colton Stearns, Adam W. Harley, Mikaela Angelina Uy, et al. Dynamic Gaussian Marbles for Novel View Synthesis of Casual Monocular Videos. In SIGGRAPH Asia, 2024.

[84] Jiahui Lei, Yijia Weng, Adam W. Harley, et al. MoSca: Dynamic Gaussian Fusion from Casual Videos via 4D Motion Scaffolds. In CVPR, 2025.

[85] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3D Gaussians: Realistic Point Cloud Relighting with BRDF Decomposition and Ray Tracing. In ECCV, 2024.

[86] Jianheng Liu, Yunfei Wan, Bowen Wang, et al. GS-SDF: LiDAR-Augmented Gaussian Splatting and Neural SDF for Geometrically Consistent Rendering and Reconstruction. In IROS, 2025.

[87] Boyao Zhou, Shunyuan Zheng, Hanzhang Tu, et al. GPS-Gaussian+: Generalizable Pixel-wise 3D Gaussian Splatting for Real-Time Human-Scene Rendering from Sparse Views. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025.

[88] Jan Held, Renaud Vandeghen, Abdullah Hamdi, et al. 3D Convex Splatting: Radiance Field Rendering with 3D Smooth Convexes. In CVPR, 2025.

[89] Chun Gu, Zeyu Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Tetrahedron Splatting for 3D Generation. In NeurIPS, 2024.

[90] Qiqi Hou, Randall Rauwendaal, Zifeng Li, et al. Sort-free Gaussian Splatting via Weighted Sum Rendering. In ICLR, 2025.

[91] Florian Hahlbohm, Linus Franke, Martin Eisemann, et al. Faster-GS: Analyzing and Improving Gaussian Splatting Optimization. arXiv:2602.09999, 2026.

[92] Yihang Chen, Qianyi Wu, Meng Yao Li, et al. Fast Feedforward 3D Gaussian Splatting Compression. arXiv:2410.08017, 2024.

[93] Haishan Wang, Mohammad Hassan Vali, and Arno Solin. Compressing 3D Gaussian Splat-

- ting by Noise-Substituted Vector Quantization. arXiv:2504.03059, 2025.
- [94] Seungjoo Shin, Jaesik Park, and Sunghyun Cho. Locality-aware Gaussian Compression for Fast and High-quality Rendering (LocoGS). arXiv:2501.05757, 2025.
- [95] Zhe Li, Yipengjing Sun, Zerong Zheng, et al. Animatable and Relightable Gaussians for High-fidelity Human Avatar Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [96] Haokai Pang, Heming Zhu, Adam Kortylewski, et al. ASH: Animatable Gaussian Splats for Efficient and Photoreal Human Rendering. In *CVPR*, 2024.
- [97] Suyeon Choi, Brian Chao, Jacqueline Yang, et al. Gaussian Wave Splatting for Computer-Generated Holography. In *SIGGRAPH*, 2025.
- [98] Yimin Pan, Matthias Nießner, and Tobias Kirschstein. HairGS: Hair Strand Reconstruction based on 3D Gaussian Splatting. arXiv:2509.07774, 2025.
- [99] Ruizhi Shao, Jingxiang Sun, Peng Cheng, et al. Control4D: Efficient 4D Portrait Editing With Text. In *CVPR*, 2024.
- [100] Jiahui Zhang, Yuelei Li, Anpei Chen, et al. Advances in Feed-Forward 3D Reconstruction and View Synthesis: A Survey. arXiv:2507.14501, 2025.
- [101] Sicheng Li, Chengzhen Wu, Hao Li, et al. GSCoDec Studio: A Modular Framework for Gaussian Splat Compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.
- [102] Yifei Zeng, Yanqin Jiang, Siyu Zhu, et al. STAG4D: Spatial-Temporal Anchored Generative 4D Gaussians. In *ECCV*, 2024.
- [103] Zinan Lv, Yeqian Qian, Chunxiang Wang, et al. Structure-Guided Memory-Efficient 3D Gaussians for Large-Scale Reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2026.
- [104] Tae-Kyeong Kim, Xingxin Chen, Guile Wu, et al. Nighttime Autonomous Driving Scene Reconstruction with Physically-Based Gaussian Splatting. arXiv:2602.13549, 2026.
- [105] Idris O. Sunmola, Zhenjun Zhao, Samuel Schmidgall, et al. Surgical Gaussian Surfels: Highly Accurate Real-time Surgical Scene Rendering using Gaussian Surfels. arXiv:2503.04079, 2025.
- [106] Antoine Guédon, Diego Gomez, Nissim Maruani, et al. MILo: Mesh-In-the-Loop Gaussian Splatting for Detailed and Efficient Surface Reconstruction. arXiv:2506.24096, 2025.
- [107] Sheng Hong, Chunran Zheng, Yishu Shen, et al. GS-LIVO: Real-Time LiDAR, Inertial, and Visual Multisensor Fused Odometry With Gaussian Mapping. *IEEE Transactions on Robotics*, 2025.
- [108] Pengcheng Zhu, Yaoming Zhuang, Baoquan Chen, et al. MGS-SLAM: Monocular Sparse Tracking and Gaussian Mapping With Depth Smooth Regularization. *IEEE Robotics and Automation Letters*, 2024.
- [109] Dapeng Feng, Zhiqiang Chen, Yizhen Yin, et al. CaRtGS: Computational Alignment for Real-Time Gaussian Splatting SLAM. *IEEE Robotics and Automation Letters*, 2025.
- [110] Linfei Li, Lin Zhang, Zhong Wang, et al. GS³LAM: Gaussian Semantic Splatting SLAM. In *ACM Multimedia*, 2024.
- [111] Thomas Gottwald, Edgar Heinert, Peter Stehr, et al. PRIMU: Uncertainty Estimation for Novel Views in Gaussian Splatting from Primitive-Based Representations of Error and Coverage. arXiv:2508.02443, 2025.
- [112] Hao Liu, Minglin Chen, Yanni Ma, et al. Point Cloud Unsupervised Pre-training via 3D Gaussian Splatting. arXiv:2411.18667, 2024.
- [113] Hashiru Pramuditha, Vinasirajan Viruthshaan, Vishagar Arunan, et al. DARB-Splatting: Generalizing Splatting with Decaying Anisotropic Radial Basis Functions. arXiv:2501.12369, 2025.
- [114] Yuanhao Gong, Lantao Yu, and Guanghui Yue. Isotropic Gaussian Splatting for Real-Time Radiance Field Rendering. arXiv:2403.14244, 2024.
- [115] Joongho Jo, Hyeongwon Kim, and Jongsun Park. Identifying Unnecessary 3D Gaussians using Clustering for Fast Rendering of 3D Gaussian Splatting. arXiv:2402.13827, 2024.
- [116] Yuedong Chen, Chuanxia Zheng, Haofei Xu, et al. MVSplat360: Feed-Forward 360° Scene Synthesis from Sparse Views. arXiv:2411.04924, 2024.
- [117] Xianglong He, Junyi Chen, Sida Peng, et al. GV-GEN: Text-to-3D Generation with Volumetric Representation. In *ECCV*, 2024.
- [118] Xianghui Yang, Huiwen Shi, Bowen Zhang, et al. Hunyuan3D 1.0: A Unified Framework for Text-to-3D and Image-to-3D Generation. arXiv:2411.02293, 2024.
- [119] Wenqing Wang and Yun Fu. Text-to-3D Gaus-

sian Splatting with Physics-Grounded Motion Generation. arXiv:2412.05560, 2024.

[120] Barbara Roessle, Norman Müller, Lorenzo Porzi, et al. L3DG: Latent 3D Gaussian Diffusion. In SIGGRAPH Asia, 2024.

Year	Method	Venue	Key contribution	Authors
2001	EWA Volume Splatting	IEEE Vis	Anti-aliased Gaussian volume splatting	Zwicker, Pfister, van Baar, Gross
2002	EWA Splatting	IEEE TVCG	Elliptical-weighted-average splatting framework	Zwicker, Pfister, van Baar, Gross
2016	COLMAP	CVPR	Robust Structure-from-Motion pipeline	Schönberger, Frahm
2018	Deep Blending	SIGGRAPH Asia	DB benchmark for free-viewpoint IBR	Hedman et al.
2020	NeRF	ECCV	Implicit MLP radiance field	Mildenhall et al.
2021	Point-Based Neural Rendering	EGSR/CGF	Per-view optimization on neural points	Kopanas, Philip, Leimkühler, Drettakis
2021	Mip-NeRF	ICCV	Cone tracing for anti-aliased NeRF	Barron et al.
2022	Mip-NeRF 360	CVPR	Unbounded NeRF benchmark	Barron et al.
2022	Instant-NGP	ACM ToG	Multi-res hash grid radiance field	Müller et al.
2022	TensoRF	ECCV	Tensorial decomposition of radiance	Chen et al.
2023	3DGS	SIGGRAPH (ToG 42.4)	Real-time anisotropic Gaussian rasterization	Kerbl, Kopanas, Leimkühler, Drettakis
2023	DreamGaussian	ICLR 2024	First text-to-3D via 3DGS	Tang et al.
2023	SuGaR	CVPR 2024	Mesh extraction from 3DGS	Guédon, Lepetit
2023	Scaffold-GS	CVPR 2024	Anchor-based view-adaptive 3DGS	Lu et al.
2024	Mip-Splatting	CVPR	Alias-free 3DGS	Yu et al.
2024	2DGS	SIGGRAPH	Geometrically accurate 2D Gaussians	Huang et al.
2024	4DGS	CVPR	Real-time dynamic 3DGS	Wu et al.
2024	SplaTAM, MonoGS, GS-SLAM	CVPR	First wave of 3DGS-based SLAM	Various
2024	A New Era survey	IEEE TVCG	First comprehensive 3DGS survey	Fei et al.
2024	Recent Advances survey	CVMJ	Application-oriented survey	Wu et al.
2026	ACM Surveys 3DGS	ACM CSUR	Definitive survey-of-surveys	Chen, Wang

Hyperparameter	Value	Comment / source
Iterations	30,000	Kerbl et al. 2023
Optimizer	Adam	$\beta_1 = 0.9, \beta_2 = 0.999$
LR (positions)	1.6×10^{-4}	Decay to 1.6×10^{-6} at end
LR (opacity)	5×10^{-2}	Constant
LR (scale)	2.5×10^{-3}	Constant
LR (rotation)	10^{-3}	Constant
LR (SH)	2.5×10^{-3}	DC vs higher degrees can differ
λ (loss balance)	0.2	$L_1 + 0.2$ D-SSIM
Tile size	16×16 pixels	Tile-based rasterizer
SH degree L	3	16 basis fns per channel
Densification interval	every 100 iters from iter 500	ADC schedule
Densification gradient τ	2×10^{-4}	View-space gradient threshold
Split factor ϕ	1.6	Scale division on splits
Prune ϵ	0.005	Opacity threshold
Opacity reset interval	every 3000 iters	Suppresses floaters
Initial Gaussians	SfM points + small noise	Typically 100K-500K
Final Gaussians	1-6 M (scene-dependent)	Kerbl et al. 2023
Storage	0.5-1.5 GB	Single precision

Method	Family	Mip-NeRF 360 PSNR	T&T PSNR	DB PSNR	Train (min)	FPS	Size (MB)	Year/Venue
Mip-NeRF 360	NeRF	27.69	22.22	29.40	~1500	0.07	8	CVPR 2022
Instant-NGP	Voxel/MLP	26.43	21.92	24.88	5–10	9	70	ToG 2022
3DGS	Vanilla	27.21	23.14	29.41	41	134	734	SIGGRAPH 2023
Mip-Splatting	Anti-alias	27.79	23.96	29.79	50	110	720	CVPR 2024
Multi-Scale 3DGS	Anti-alias	27.50	23.50	29.60	45	100	700	arXiv 2023
Scaffold-GS	Anchor	27.50	23.96	30.21	35	120	250	CVPR 2024
RadSplat	Distill	27.79	23.85	29.95	45	900+	380	3DV 2025
2DGS	2D primitive	26.95	22.80	28.95	60	100	600	SIGGRAPH 2024
SuGaR	Mesh extract	26.50	—	—	60+	100	800	CVPR 2024
CompGS	Compression	27.04	23.11	29.30	38	150	18	ECCV 2024
Compact 3DGS	Compression	27.10	23.32	29.79	35	145	25	CVPR 2024
Niedermayr 2024	Compression	26.98	23.32	29.38	40	130	19	CVPR 2024
EAGLES	Compression	27.15	23.44	29.91	30	155	35	ECCV 2024
Mini- Splatting	Density	27.34	23.18	29.88	32	160	70	ECCV 2024
pixelSplat	Feed- forward	25.90	—	—	0.1	—	n/a	CVPR 2024
MVSplat	Feed- forward	26.39	—	—	0.2	—	n/a	ECCV 2024
4DGS (Wu 2024)	Dynamic	32.0 (D-NeRF)	—	—	90	80	600	CVPR 2024
Deformable 3DGS	Dynamic	31.5 (D-NeRF)	—	—	100	90	580	CVPR 2024
Dynamic 3D Gaussians	Dynamic	—	—	—	120	80	700	3DV 2024

Method	Sub-family	Benchmark	PSNR	FPS	Year/Venue
Dynamic 3D Gaussians (Luiten)	Per-Gaussian	Plenoptic Video	30.5	30	3DV 2024
Deformable 3DGS (Yang)	Per-Gaussian	D-NeRF	40.1	50	CVPR 2024
4DGS (Wu)	Decoupled field	D-NeRF	32.0 / Plenoptic 31.0	80	CVPR 2024
Gaussian-Flow (Lin)	Decoupled field	Plenoptic Video	31.4	60	CVPR 2024
Yang ICLR 4DGS	Decoupled field	Plenoptic Video	30.9	100+	ICLR 2024
4D Scaffold GS (Cho)	Anchor	Plenoptic Video	31.7	90	AAAI 2026
SDD-4DGS (Sun)	Static-dynamic	nuScenes	26.5	70	arXiv 2025
MoSca (Lei)	Casual monocular	DyCheck	24.3	—	CVPR 2025
Dynamic Gaussian Marbles	Casual monocular	DyCheck	23.6	—	SIGGRAPH Asia 2024
3DGS-Avatar (Qian)	Avatar	ZJU-MoCap	32.8	130	CVPR 2024
HUGS (Kocabas)	Avatar	ZJU-MoCap	30.6	100	CVPR 2024
ASH (Pang)	Avatar	OakInk	31.4	60	CVPR 2024
Align Your Gaussians (Ling)	Generative 4D	qualitative	—	—	CVPR 2024
STAG4D (Zeng)	Generative 4D	qualitative	—	—	ECCV 2024

Method	DTU CD (mm) ↓	T&T F-score ↑	Mesh quality	PSNR penalty	Year/Venue
NeuS (Wang 2021)	0.84	0.40	sharp, slow	n/a	NeurIPS 2021
3DGS (Kerbl 2023)	1.96	0.18	fuzzy	0.0	SIGGRAPH 2023
SuGaR (Guédon 2024)	1.40	0.30	post-hoc Poisson	\$\$-0.5 dB	CVPR 2024
2DGS (Huang 2024)	0.62	0.52	TSDF, sharp edges	\$\$-0.3 dB	SIGGRAPH 2024
DN-Splatter (Turkulainen)	0.73	0.50	depth-normal supervised	\$\$-0.4 dB	WACV 2025
GS-SDF (Liu)	0.71	0.55	SDF hybrid + LiDAR	\$\$-0.6 dB	IROS 2025
NeuSG (Chen)	0.66	0.51	NeuS-3DGS hybrid	\$\$-0.5 dB	arXiv 2023
MILo (Guédon)	0.61	0.53	mesh-in-the-loop	\$\$-0.3 dB	arXiv 2025

Title Suppressed Due to Excessive Size

Method	Replica ATE (cm) ↓	TUM RGB-D ATE (cm) ↓	Replica PSNR ↑	FPS	Sensors	Year/Venue
iMAP (2021)	4.62	8.92	18.2	0.5	RGB-D	ICCV 2021
NICE-SLAM	1.06	2.01	24.4	0.6	RGB-D	CVPR 2022
NICER-SLAM	1.13	—	25.5	0.5	RGB-only	3DV 2024
Point-SLAM	0.52	—	35.3	1.0	RGB-D	ICCV 2023
SplaTAM	0.36	1.92	35.1	1.5	RGB-D	CVPR 2024
GS-SLAM (Yan)	0.50	1.85	34.0	5.0	RGB-D	CVPR 2024
MonoGS	0.58	1.62	33.4	1.0	RGB-only	CVPR 2024
RGBD GS-ICP	0.32	1.40	35.5	2.0	RGB-D	ECCV 2024
CG-SLAM	0.50	—	35.0	1.5	RGB-D	ECCV 2024
Splat-SLAM	0.42	1.55	34.6	1.0	RGB-only	arXiv 2024
LoopSplat	0.40	1.50	34.8	1.2	RGB-D	3DV 2025
Hier-SLAM	0.48	—	34.5	1.0	RGB-D + sem	ICRA 2025
MBA-SLAM	0.45	—	34.2	0.8	RGB(blur)	TPAMI 2025
GS-LIVO	—	—	—	30	LiDAR+IMU+RGB	THCB 2025

Domain	Representative method	Dataset / benchmark	Score / metric	FPS	Year/Venue
Body avatars	HUGS	ZJU-MoCap	PSNR 30.6	100	CVPR 2024
Body avatars	3DGS-Avatar	ZJU-MoCap	PSNR 32.8	130	CVPR 2024
Head avatars	HeadGaS	NeRSemble	PSNR 32.0	80	ECCV 2024
Head avatars	GaussianTalker	LRS3	LSE-D 7.2	30	ACM MM 2024
Hair	HairGS	NeRSemble-Hair	strand RMSE 0.8 mm	60	arXiv 2025
Driving	DrivingGaussian	nuScenes	PSNR 27.3	60	CVPR 2024
Driving	AutoSplat	KITTI-360	PSNR 26.9	50	arXiv 2024
Driving	GS-LIVO	self-collected	trajectory drift 0.4%	30	T-RO 2025
Text-to-3D	DreamGaussian	DreamFusion suite	CLIP score 0.30	n/a	ICLR 2024
Text-to-3D	GaussianDreamer	DreamFusion suite	CLIP score 0.31	n/a	CVPR 2024
Text-to-4D	Align Your Gaussians	TextGen-4D	qualitative	n/a	CVPR 2024
Scene generation	DreamScene360	qualitative	—	—	ECCV 2024
X-ray	X-Gaussian	X3D	PSNR 41.2	100	ECCV 2024
HDR	HDR-GS	HDR-NeRF set	PSNR 32.4	250	NeurIPS 2024
Holography	Gaussian Wave Splatting	hologram set	PSNR 35.0	40	SIGGRAPH 2025
Surgical (4D)	LGS	EndoNeRF	PSNR 35.5	60	MICCAI 2024
Foundation surgical	FM-GS	EndoNeRF	PSNR 36.1	60	IEEE TMI 2025

Metric	Definition	Range / direction	Origin
PSNR	$-10 \log_{10} \text{MSE}$	dB, higher better	classic
SSIM	structural similarity index	$[0, 1]$, higher better	Wang et al. 2004
LPIPS	learned perceptual similarity (AlexNet/VGG)	lower better	Zhang et al. 2018
Chamfer Distance	bidirectional point-to-mesh distance	mm, lower better	classic
F-score @ τ	recall \times precision under threshold τ	$[0, 1]$, higher better	T&T
ATE RMSE	absolute trajectory error vs ground truth	cm, lower better	Sturm 2012
FPS	frames per second at 1080p	higher better	runtime
Storage	model size on disk	MB, lower better	runtime
GPU memory	peak training memory	MB, lower better	runtime
Train time	wall clock for full optimization	min, lower better	runtime
Hausdorff CD	one-sided maximum distance	mm, lower better	extension

Method	Mip-NeRF 360				NeRF-Syn	
	PSNR	SSIM	LPIPS	T&T PSNR	DB PSNR	PSNR
NeRF	24.85	0.659	0.426	19.52	26.81	31.01
Mip-NeRF 360	27.69	0.792	0.237	22.22	29.40	33.10
Instant-NGP	26.43	0.725	0.339	21.92	24.88	33.18
Plenoxels	23.62	0.670	0.443	21.08	23.06	31.71
3DGS	27.21	0.815	0.214	23.14	29.41	33.32
Mip-Splatting	27.79	0.827	0.203	23.96	29.79	33.40
Multi-Scale 3DGS	27.50	0.821	0.210	23.50	29.60	33.30
Scaffold-GS	27.50	0.806	0.220	23.96	30.21	33.41
RadSplat	27.79	0.825	0.204	23.85	29.95	33.42
2DGS	26.95	0.804	0.222	22.80	28.95	33.05
CompGS	27.04	0.806	0.225	23.11	29.30	33.07
Compact 3DGS	27.10	0.810	0.220	23.32	29.79	33.20
Niedermayr 2024	26.98	0.802	0.232	23.32	29.38	33.10
EAGLES	27.15	0.812	0.218	23.44	29.91	33.20
Mini-Splatting	27.34	0.815	0.215	23.18	29.88	33.30

Quantity	Vanilla 3DGS	Compressed (CompGS)	Anchor (Scaffold-GS)	Distilled (RadSplat)
Train time	41 min	38 min	35 min	45 min
Storage (Mip-NeRF 360 avg)	734 MB	18 MB	250 MB	380 MB
Number of Gaussians	~ 3.5 M	~ 3.2 M (8K codebook)	~ 1.0 M anchors	~ 1.8 M distilled
Peak GPU memory	24 GB	18 GB	12 GB	22 GB
Render FPS @ 1080p	134	150	120	900+
Render FPS @ 4K	38	42	34	280
Power (laptop GPU)	100 W	90 W	80 W	110 W

Failure mode	Cause	Dominant mitigation	Reference / Method
Aliasing on zoom	Footprint vs pixel mismatch	3D smoothing + 2D Mip filter	Mip-Splatting (Yu 2024)
Popping under motion	Sort reorder discontinuity	Weighted-sum compositor	Sort-free GS (Hou 2024)
Floater	Free-space high-opacity Gaussians	Opacity reset, distillation	RadSplat, PRIMU
Storage explosion	59 scalars \times millions of Gaussians	Vector quantization, anchors	CompGS, Niedermayr, Scaffold-GS
Streaming latency	Single monolithic file	Layered progressive	LapisGS
Sparse-view collapse	Under-constrained optimization	Depth-normal priors	DNGaussian, DN-Splatter
Motion blur in capture	Camera motion during exposure	Bundle-adjusted deblur	BAD-Gaussians, MBA-SLAM
Pose failure	COLMAP non-convergence	Joint pose-Gaussian opt	COLMAP-Free, Hybrid BA
Specular surfaces	View-dependent SH limit	Higher-order or neural color	Scaffold-GS, Convex Splatting
Mirrors / reflections	Non-local appearance	Explicit reflection modeling	Mirror-NeRF analogs (open)
Relighting	Baked lighting in SH	BRDF + ray tracing	Relightable 3D Gaussians
Topology change	Diffeomorphic assumption	Foundation-model dense flow	MoSca, GenMOJO
Long-horizon drift (4D)	No loop closure in time	Open	—
Thin structures	Sub-pixel footprint	1D primitives	HairGS
Outdoor lighting variation	Per-image exposure variation	Appearance codes	Look at the Sky, Bilateral Grids GS
Dynamic scenes in SLAM	Map contamination	Dynamic-region segmentation	Chen et al. 2025 Sensors
Out-of-core city scale	GPU memory limit	Tiling, anchors, LOD	Block-NeRF analogs (open)
Uncertainty estimation	Point estimates only	Variational Bayesian	VBGS-SLAM, PRIMU
Generalization across scenes	Per-scene optimization	Feed-forward priors	pixelSplat, MVSplat, F4Splat

#	Open problem	Falsifiability criterion	Current best
OP-1	Principled multi-scale anti-aliasing	PSNR drop < 0.1 dB across $4\times$ zoom	Mip-Splatting 0.4 dB
OP-2	Feed-forward 3DGS at city scale	< 1 s on A100, PSNR \geq per-scene -0.3 dB	25.9 dB on Mip-NeRF 360 (lag 1.3 dB)
OP-3	Standard 3DGS codec	Rate-distortion comparable to JPEG XL	19 MB CompGS (no standard)
OP-4	Physically faithful relighting + indirect illumination	Match path-tracer ground truth on NeRF-OSR	Relightable 3D Gaussians (only direct)
OP-5	Topology-changing dynamics	PSNR > 30 on fluid scenes	MoSca PSNR 24.3 on DyCheck
OP-6	Closed-loop driving simulation under day/night/weather	Round-trip safety-critical scenario fidelity	DrivingGaussian (single weather)
OP-7	Billion-Gaussian out-of-core rendering	Render 1 km^2 at 30 FPS on RTX 4090	~ 10 M Gaussians is current limit
OP-8	Uncertainty + safety guarantees on rendered novel views	Calibrated CIs for downstream pipelines	PRIMU 2025 (early)
OP-9	Sparse-view 3DGS at ≤ 3 images	PSNR > 26 on Mip-NeRF 360 from 3 views	~ 21 dB current
OP-10	Real-time training (training < 1 min)	Mip-NeRF 360 PSNR ≥ 26 in 60 s	Faster-GS / Fast Converging 3DGS approach
OP-11	Reflection / refraction modeling	Match path-tracer on glass/mirror benchmarks	Open (Mirror-NeRF analogs)
OP-12	Long-horizon 4D consistency (> 1 minute)	$< 2\%$ trajectory drift over > 1 minute	LoopSplat in static; open in 4D

#	Prediction	Metric	Target date	Evidence base
P1	Feed-forward 3DGS matches per-scene PSNR	Δ PSNR < 0.3 on Mip-NeRF 360	Dec 2027	pixelSplat, MVSPlat, F4Splat trajectory
P2	3DGS SLAM at 30+ FPS on mobile	FPS, ATE	Dec 2028	GS-LIVO Jetson Orin baseline
P3	Feed-forward 3DGS default for embodied AI	adoption	2028	DUSt3R + 3DGS hybrids
P4	MPEG-style 3DGS codec draft	published draft	2027	GSCodec Studio, AAC-GS
P5	Layered streaming standard	adoption in OpenXR	2027	LapisGS
P6	Hardware GS raster cores	shipping silicon	2028	Vision Pro, Quest precedent
P7	Relighting matches path-tracer	NeRF-OSR PSNR within 1 dB	2027	Relightable 3D Gaussians, Radiometric Surfels
P8	Topology-changing dynamics	PSNR > 30 on fluid benchmarks	2028	PhysGaussian + GenMOJO
P9	Production DCC integration	Blender/Maya plugin support	2027	Gaussian Grouping, Feature 3DGS
P10	Medical 3DGS replaces traditional VR	clinical adoption	2027	X-Gaussian, FM-GS, Surgical Surfels
P11	Holographic Gaussian Wave Splatting	display industry adoption	2028	Choi et al. SIGGRAPH 2025
P12	Foundation-model distillation standard	most papers use it	2027	Feature 3DGS, Hier-SLAM, FM-GS