3D Gaussian Splatting: Survey, Technologies, Challenges, and Opportunities

Yanqi Bao, Tianyu Ding, Jing Huo*, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, *Member, IEEE,*
and Jiebo Luo, *Fellow, IEEE*

***Abstract*—3D Gaussian Splatting (3DGS) has emerged as a prominent technique with the potential to become a mainstream method for 3D representations. It can effectively transform multi-view images into explicit 3D Gaussian through efficient training, and achieve real-time rendering of novel views. This survey aims to analyze existing 3DGS-related works from multiple intersecting perspectives, including related tasks, technologies, challenges, and opportunities. The primary objective is to provide newcomers with a rapid understanding of the field and to assist researchers in methodically organizing existing technologies and challenges. Specifically, we delve into the optimization, application, and extension of 3DGS, categorizing them based on their focuses or motivations. Additionally, we summarize and classify nine types of technical modules and corresponding improvements identified in existing works. Based on these analyses, we further examine the common challenges and technologies across various tasks, proposing potential research opportunities.**

***Index Terms*—3D Representations, Rendering, 3DGS.**

I. Introduction

THE advent of Neural Radiance Fields (NeRF) [1] has ignited considerable interest in the pursuit of photorealistic 3D content. Despite substantial recent advancements that have markedly enhanced NeRF's potential for practical applications, its inherent efficiency challenges have remained unresolved. The introduction of 3D Gaussian Splatting (3DGS) has decisively addressed this bottleneck, enabling high-quality real-time ($\geq$30 fps) novel view synthesis at 1080p resolution.

Beyond its computational efficiency gains, 3DGS represents a paradigm shift in neural rendering that bridges traditional geometric reconstruction with neural implicit representations, while its differentiable nature and controllable explicit representation have profound implications for advancing various computer vision tasks. This advancement not only aligns with the broader trend in computer graphics towards continuous, lightweight representations, but also enables practical applications across diverse fields, from enhancing immersive environments in VR/AR and improving spatial awareness in robotics, to supporting urban planning and cultural heritage digitization. This rapid development has quickly attracted researchers and led to a proliferation of related works.

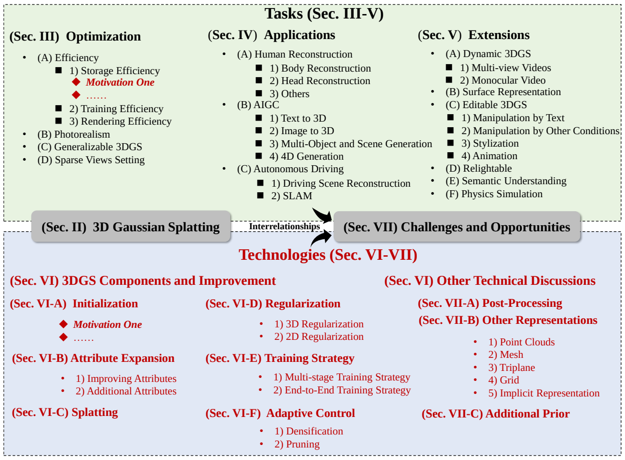

Fig. 1: The introduction of this survey, with the **RED** parts indicating the new content compared to existing reviews.

To assist readers in quickly grasping the progress in 3DGS research, we provide a comprehensive survey of 3DGS and its derivative works. This survey systematically compiles the most important and recent literature on the subject, offering detailed classifications and discussions of their tasks and techniques. In examining the technological commonalities across numerous 3DGS variants, we present a structured analysis of technical improvements in fundamental components of vanilla 3DGS, including Initialization, Attribute Expansion, Splatting, Regularization, Training Strategy, Adaptive Control, Post-Processing, Other Representations and Additional Prior. Based on this summary of techniques, we aim to help readers synthesize the connections among different improved techniques and provide approaches to enhance various components of vanilla 3DGS to meet their customized tasks. Moreover, we conduct a systematic investigation into the relationships between downstream tasks and their enabling technologies in 3DGS, identifying and analyzing four fundamental challenges, including Suboptimal Data, Generalization, Physics Reconstruction and Rendering, and Realness and Efficiency. Through careful examination of these challenges, we propose promising research directions to advance this rapidly evolving field, providing a roadmap for future innovations in 3DGS. Although some comprehensive reviews have documented recent advances

Yanqi Bao, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li and Yang Gao are with the State Key Laboratory for Novel Software Technology, Nanjing University, China, 210023 (e-mail: {yq_bao, yaoliliu, liyuxin16}@smail.nju.edu.cn; {huojing, liwenbin, gaoy}@nju.edu.cn). *Corresponding authors: Jing Huo.

Tianyu Ding is with the Applied Sciences Group, Microsoft Corporation, Redmond, USA (e-mail: tianyuding@microsoft.com).

Jiebo Luo is with the Department of Computer Science, University of Rochester, America (e-mail: jluo@cs.rochester.edu).

This work was supported in part by the National Natural Science Foundation of China under Grant 62276128, Grant 62192783, and Grant 62106100; in part by the Jiangsu Natural Science Foundation under Grant BK20221441; in part by the Young Elite Scientists Sponsorship Program by CAST under Grant 2023QNRC001; in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

in 3DGS [2]–[4], they focus on categorizing and discussing existing works by downstream tasks, overlooking technical connections between different tasks, which leads to redundant discussions. Our work is distinctive in providing discussions at two levels: Tasks and Techniques. Specifically, we categorize and discuss existing downstream tasks according to their different motivations or focuses, rather than reviewing all works sequentially. **More significantly, we present a thorough examination of technical improvements implemented across various modules of the vanilla 3DGS by existing variants, enabling readers to establish clear relationships between different research fields that share similar methodological foundations.** Building upon these, we further investigate the their underlying **commonalities** and delineate core challenges and opportunities, as shown in Fig. 1. Through this approach, we aim to synthesize recent technical breakthroughs and direct researchers' attention to the core unique challenges facing 3DGS. Moreover, we have published an open-source project on GitHub that compiles 3DGS-related articles, and will continue to add new works and technologies into this project. https://github.com/qqqqqqy0227/awesome-3DGS.

II. Preliminaries

3DGS [5] combines the advantages of neural implicit field and point-based rendering methods, achieving the high-fidelity rendering quality of the former while maintaining the real-time rendering capability of the latter. As shown in Fig. 2, the initialized 3DGS is projected onto the image plane through splatting, and through the supervision of regularization terms and adaptive control, the relevant attributes in 3DGS can be continuously optimized until it has the ability to represent the entire 3D space. Specifically, 3DGS defines points in the point cloud as 3D Gaussian primitives with volumetric density:

$$G(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x})^T\boldsymbol{\Sigma}^{\mathbf{-1}}(\boldsymbol{x})\right), \tag{1}$$

where $\boldsymbol{\Sigma}$ is the 3D covariance matrix and $\boldsymbol{x}$ is the position from the point (Gaussian mean) $\boldsymbol{\mu}$. To ensure the semi-positive definiteness of the covariance matrix, 3DGS reparameterizes the covariance matrix as a combination of a rotation matrix $\boldsymbol{R}$ and a scaling matrix $\boldsymbol{S}$: $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{R}^T$, where the 3D scaling matrix $\boldsymbol{S}$ can be represented by a 3D vector $\boldsymbol{s}$, and the rotation matrix $\boldsymbol{R}$ is obtained through a learnable quaternion $\boldsymbol{q}$, resulting in a total of 7 learnable parameters. Compared to the commonly employed Cholesky decomposition, which guarantees the semi-positive-definiteness of matrices, the reparameterization method utilized by 3DGS, albeit introducing an additional learnable parameter, facilitates the imposition of geometric constraints on Gaussian primitives (e.g., constraining the scaling vector to give Gaussian primitives a flattened characteristic). In addition to geometric attributes, each Gaussian primitive also stores an opacity α and a set of learnable Spherical Harmonic (SH) to represent view-dependent appearance. Thus, the collection of all primitives can be regarded as a discretized representation that only stores the non-empty parts of the neural field.

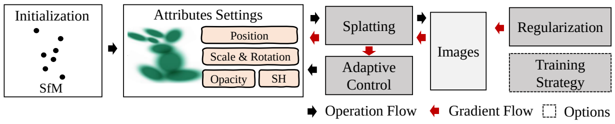

Fig. 2: Pipeline and Technologies of 3D Gaussian Splatting.

At the beginning of training, the initial Gaussian primitives are either initialized from a sparse point cloud provided by Structure-from-Motion or randomly initialized. The initial number of Gaussian primitives may be insufficient for high-quality scene reconstruction; hence, 3DGS offers a method for adaptively controlling Gaussian primitives. This method evaluates whether a primitive is "under-reconstructed" or "over-reconstructed" by observing the gradient of each Gaussian primitive's position attributes in view space. Based on this evaluation, the method increases the number of Gaussian primitives by cloning or splitting the primitives to enhance scene representation capability. Additionally, the opacity of all Gaussian primitives is periodically reset to zero to mitigate the presence of artifacts during the optimization process. This adaptive process allows 3DGS to start optimization with a smaller initial set of Gaussians, thus alleviating the dependency on dense point clouds that previous point-based differentiable rendering methods required.

During rendering, 3DGS projects 3D Gaussian primitives onto the 2D imaging plane using the EWA splatting method [6] and employs α blending to compute the final pixel color.

III. Optimization of 3D Gaussian Splatting

A. *Efficiency*

Efficiency is one of the core metrics for evaluating 3D reconstruction [7]. In this section, we describe it from three perspectives: storage, training, and rendering efficiency.

1) Storage Efficiency: 3DGS requires millions of different Gaussian primitives to fit the geometry and appearance in a scene, leading to high storage overhead: a typical reconstruction of an outdoor scene often requires several hundred megabytes to multiple gigabytes of explicit storage space. Given that the geometric and appearance attributes of different Gaussian primitives may be highly similar, storing attributes for each primitive individually can lead to potential redundancy. Some quantitative reconstruction results are reported in Table I.

Existing works [8]–[10] primarily focus on applying ***Vector Quantization [11] (VQ) techniques*** to compress 3DGS. Among them, Compact3D [9] applies VQ to compress different attributes into four corresponding codebooks and stores the index of each Gaussian in these codebooks to reduce the storage overhead. After establishing the codebooks, the training gradients are copied and backpropagated to the original non-quantized Gaussian parameters via the codebooks, updating both the quantized and non-quantized parameters, and discarding the non-quantized parameters when the training is done. Despite achieving efficient 3DGS compression, these methods inevitably encounter quantization errors following discretization and remain sensitive to hyperparameter configurations.

Furthermore, some works [12], [13] aim at ***developing efficient pruning strategies.*** LightGaussian [13] introduces a Gaussian pruning strategy based on the global significance score and a distillation strategy for high-degree spherical harmonic parameters. Similarly, the work by Lee et al. [12] introduces

TABLE I: Comparison of compression on MipNeRF360 [18].

Method	PSNR↑	SSIM↑	LPIPS↓	Size (MB)↓
3DGS	27.49	**0.813**	**0.222**	744.7
Scaffold-GS [14]	27.50	0.806	0.252	253.9
HAC [17]	**27.53**	0.807	0.238	**15.26**
Compact-3DGS [19]	27.08	0.798	0.247	48.80
EAGLES [8]	27.15	0.808	0.238	68.89
LightGaussian [13]	27.00	0.799	0.249	44.54
Gaussian-SLAM [16]	26.01	0.772	0.259	23.90
Compact3d [9]	27.16	0.808	0.228	50.30

a learnable mask to reduce the number of original Gaussians. Such methods heavily rely on determining which primitives are non-essential.Furthermore, there are works [14]–[17] focused on ***improving efficient Gaussian representations or attributes.***

Scaffold-GS [14] designs anchors and additional attributes for efficient representation, which have the capability to convert to 3DGS. Based on this representation, Scaffold-GS proposes a set of strategies for the growth and pruning of anchors on multi-resolution voxel grids. Despite their widespread implementation, these approaches face inherent compression limitations due to their unstructured characteristics, which HAC [17] later mitigated through the incorporation of structured hash grid.

2) Training Efficiency: DISTWAR [20] introduces an advanced technique aimed at accelerating atomic operations in raster-based differentiable rendering applications, which typically encounter significant bottlenecks during gradient computation due to the high volume of atomic updates. By leveraging intra-warp locality in atomic updates and addressing the variability in atomic traffic among warps, DISTWAR implements warp-level reduction of threads at the SM sub-cores using registers. These strategies enable DISTWAR to achieve an average 2.44× acceleration in performance.

3) Rendering Efficiency: Real-time rendering is one of the core advantages of Gaussian-based methods. Some works that improve storage efficiency can simultaneously enhance rendering performance, for example, by reducing the number of Gaussian primitives. Several studies [21], [22] have specifically addressed this issue. After training the 3DGS, the work by [21] involves pre-identifying and excluding unnecessary Gaussian primitives through offline clustering based on their spatial proximity and potential impact on the final rendered 2D image. Furthermore, this work introduces a specialized hardware architecture designed to support this technique, achieving a speedup of 10.7× compared to a GPU.

B. Photorealism

Photorealism is also a topic worth attention [23]. 3DGS is expected to achieve realistic rendering in various scenarios. Some [24]–[26] focus on ***optimizing under vanilla settings.*** Among them, GaussianPro [24] introduces an innovative paradigm for joint 2D-3D training. Building upon the 3D plane definition and patch matching technology, it proposes a progressive Gaussian propagation strategy, which harnesses the consistency of 3D views and projection relationships to refine the rendered 2D depth and normal maps. Although these methods demonstrate superior capabilities in handling artifacts and 3D inconsistencies compared to 3DGS, further exploration is still needed for complex geometric reconstruction. To further mitigate this issue, the work [27] introduces a scalable and efficient N-dimensional Gaussian Mixture Model for fast, accurate high-dimensional modeling without domain-specific heuristics or sacrificing computational efficiency but is limited by its reliance on dense data and conservative culling.

The sharp decline in ***multi-scale rendering performance*** is also a topic worth attention [28]–[31]. Mip-splatting [29], addressing the issue from the perspective of the sampling rate, introduces a Gaussian low-pass filter and 2D Mip filter based on Nyquist's theorem to constrain the frequency according to the maximal sampling rate across all observed samples. Then, to address the over-smoothing issue [29], Analytic-Splatting [31] proposes a method that analytically approximates the integral of 2D Gaussian signals within pixel window through a conditioned logistic function. While it enhances detail fidelity, it comes at the expense of efficiency.

Other works attempt to reconstruct challenging scenes, such as ***reflective surfaces [32]–[37] and Deblurring [38]–[41].*** GaussianShader [32] reconstructs reflective surfaces by employing a hybrid color representation and integrating the specular GGX [42] and normal estimation module, which encompasses diffuse color, direct specular reflection, and a residual color component that accounts for phenomena such as scattering and indirect light reflections. Although these methods effectively manage complex reflective surfaces, they unavoidably sacrifice computational efficiency relative to vanilla 3DGS. Recently, the work [35] proposes a deferred shading method for 3DGS, which overcomes the challenge of normal estimation for environment map reflection and demonstrates enhanced efficiency by propagating accurate normals across neighboring Gaussians and per-pixel shading in screen space. And, existing deblurring approaches predominantly focus on motion blur or lens defocusing [38]–[40], introducing blur process modeling to enable sharp reconstruction supervised by blurred images. To address a wider range of scenarios, BAGS [41] introduces a Blur Agnostic robust modeling by a Blur Proposal Network and coarse-to-fine optimization scheme.

C. Generalizable 3DGS

The 3DGS's explicit representation has led to a substantial body of works focused on ***using reference images to directly infer corresponding Gaussian primitives on a per-pixel basis***, which are subsequently employed to render images from target views [43], [44]. To achieve this, early works such as Splatter Image [43] propose a novel paradigm for converting images into Gaussian attribute images. MVSplat [45] proposes representing the cost volume using plane sweeps in 3D space and predicting the depths in sparse reference inputs, precisely locating the centers of Gaussian primitives. While demonstrating generalization ability, this technique's application is limited by its restricted synthesis range and the emergence of distractor-data. Subsequent works, FreeSplat [46] and DGGS [47], address these limitations through Pixel-wise Triplet Fusion strategy and distractor-free training and inference paradigms, respectively. Most recently, G3R [48] extends generalizable 3DGS to dynamic scenes, achieving generalized dynamic scene reconstruction through extra LiDAR data.

Furthermore, some studies [49], [50] focus on ***introducing triplane*** to achieve generalization capabilities , which infers Gaussian attributes by querying triplane features. Recent works [51], [52] seek to extend similar paradigms to large-scale

TABLE II: Comparison to works for Body Reconstruction.

Pose-dependent Deformation	Novel Pose Animation	Fast Training	Rendering > 60FPS	Monocular Input	Super-resolution	Generalization	
✓	✓	✗	✗	✓	✗	✗	HuGS [64]
-	-	-	-	-	✓	✓	GPS-Gaussian [65]
✗	✓	✓	✓	✓	✗	✗	HUGS [66]
✓	✓	✓	✓	✓	✗	✗	GaussianAvatar [67]
✗	✗	✓	✓	✓	✗	✗	GauHuman [68]
✓	✓	✓	✗	✓	✗	✗	3DGS-Avatar [69]
✓	✓	-	✗	✓	✗	✗	ASH [70]
✓	✓	✗	✗	✗	✗	✗	Animatable Guassian [71]

3D datasets, utilizing transformer-based architectures to enable direct 3D asset inference from sparse image inputs.

D. Sparse Views Setting

Reconstructing from sparse inputs presents significant challenges, wherein the methodology of 3DGS is fundamentally analogous to that of NeRF [53], [54], which aim to ***develop novel regularization strategies and integrate supplementary information, such as depth data or Diffusion model*** [55]–[62]. Specifically, the incorporation of depth data is crucial for 3DGS as an explicit representation, alleviating the model's demand for spatial comprehension under sparse inputs. And multi-view trained diffusion models also can provide important prior knowledge for expanding sparse views into refined dense information. Nevertheless, these methods typically depend on the diffusion models' ability to preserve 3D consistency. Additional regularization is commonly achieved through incorporating explicit spatial distribution rules or pseudo-views supervision. In addition to these common methods, some studies have focused on the ***initialization and optimization strategy***. GaussianObject [63] introduces an initialization strategy based on Visual Hull and an optimization method using distance statistical data to eliminate floaters.

IV. Applications of 3D Gaussian Splatting

A. Human Reconstruction

1) Body Reconstruction: Body reconstruction mainly focuses on reconstructing deformable human avatars from multi-view or monocular videos [72], as well as providing real-time rendering. We list comparisons of recent works in Tab. II.

Most works [64], [66]–[69] prefer to use ***well-preconstructed human models like SMPL [73] or SMPL-X [74]*** as strong prior knowledge. Nevertheless, SMPL is limited to introducing prior knowledge about the human body itself, thus posing challenges for the reconstruction and deformation of outward features such as garments and hair. ***For the reconstruction of outward appearance,*** HUGS [66] utilizes SMPL and LBS only at the initial stage, allowing Gaussian primitives to deviate from the initial mesh to accurately represent garments and hair. Then, ***some studies project the problem space from 3D to 2D***, thereby reducing complexity and introducing well-established 2D networks for parameter learning [70], [71]. Among them, ASH [70] generates a motion-related template mesh via a deformation network and predicts Gaussian parameters through a 2D network using a motion-related texture map derived from this mesh. These methods are often limited by the difficulty of isolating clothing from the reconstructed 3DGS.

2) Head Reconstruction: In the domain of human head reconstruction, the works [75]–[79], following the prevalent approach of utilizing SMPL as a strong prior, incorporates FLAME [80] meshes to provide geometric guidance or coarse reconstruction for 3DGS, achieving superior rendering quality. However, Gaussian Head Avatar [81] challenges the conventional use of FLAME meshes and LBS for facial deformation, arguing that these simplified linear operations are inadequate for capturing intricate facial expressions. As an alternative, it introduces an MLP-based approach that directly predicts Gaussian displacements during the transition from neutral to target expressions, enabling high-resolution head rendering at up to 2K resolution. And then, the works [82], [83] replace FLAME-based initialization with geometric guidance derived from signed distance field and DMTet. Several works [84], [85] focus on rendering efficiency, achieving frame rates exceeding 300 FPS. For downstream tasks, various studies [86]–[88] seek to integrate audio for controlling dynamic head reconstruction, thereby achieving audio-visual synchronization.

3) Others: 3DGS has introduced solutions in other human-related areas [89], [90]. GaussianHair [89] focuses on the reconstruction of human hair, using linked cylindrical Gaussian modeling for the strands. Additionally, some research [91]–[93] have explored the integration of 3DGS with generative models.

B. Artificial Intelligence-Generated Content (AIGC)

1) Text to 3D Objects: Extensive existing research endeavors to utilize the superior generative capabilities of 2D generative models to achieve coherent 3D content creation. Benefiting from reduced dependency on extensive 3D training data, Score Distillation Sampling-based paradigms garnered significant attention in early research [94]. Some works [95]–[98] focus on improving the framework to ***apply score distillation loss to 3DGS.*** Building upon Score Distillation Sampling (SDS), DreamGaussian [95] ensures the geometric consistency of the generated models by extracting explicit Mesh representations from the 3DGS and refines texture in the UV space to enhance the quality of the renderings. However, the mode-seeking paradigm of score distillation frequently leads to ***oversaturation, excessive smoothing, and lack-detail*** in the generated outcomes [99]–[103]. Among them, LucidDreamer [100] addresses the challenges of over-smoothing and insufficient sampling steps inherent in traditional SDS. Using deterministic diffusion trajectories and interval-based score matching mechanisms, it achieves superior quality and efficiency.

To further mitigate the inherent limitations of SDS, several works [104]–[108] seek to leverage ***video or multi-view generative models*** to obtain more data for reconstruction. Although these approaches introduce direct prior guidance for 3D generation, the inherent lack of guaranteed 3D consistency in both multi-view and video generation still leads to instability in reconstruction. To enhance the efficiency of 3D asset generation, works [109]–[111] aim to generate using only ***feed-forward networks*** without the need for scene-specific training. BrightDreamer [109] predicts positional offsets following fixed initialization and employs a text-guided triplane generator to process extracted textual features for predicting additional 3DGS attributes, achieving text-to-3D model conversion in

77ms, albeit at the cost of some reconstruction quality. For enhanced geometric detail control, SketchDream [112] proposes a framework for sketch-based text-to-3D generation and editing, which integrates hand-drawn sketches and text prompts to achieve fine-grained geometry and appearance control, enabling high-quality 3D content creation and local editing through a two-stage coarse-to-fine approach.

More directly, several works [113]–[116] aim to incorporate 3DGS representations into 3D generative models. Among these, L3DG [115] proposes a latent diffusion framework for compressed 3D Gaussian representation, achieving superior visual quality and real-time rendering. However, these approaches are inherently limited by the availability of 3D data, which impacts their reconstruction capability for complex targets.

Some works [91], [92], [117] also attempt to apply this generative paradigm to areas such as ***digital human generation.*** HumanGaussian [92] combines RGB and depth rendering to improve the SDS, thereby jointly supervising the optimization of the structural perception of human appearance and geometry.

2) Image to 3D Object: Similar to works on NeRF, recent studies [118]–[120] have also focused on generating entire 3DGS from a single image. These approaches share fundamental similarities with text-to-3D object methods. As an example, following a process similar to DreamGaussian [95], Repaint123 [119] employs Zero-123 [121] and SDS for coarse 3DGS, followed by a fine stage where mesh representation is extracted and refined using depth-guided and visibility-aware repainting on novel views for consistent 3DGS fine-tuning.

3) Multi-Object and Scene Generation: In addition to single-object generation, multi-object and scene generation is more crucial in most application scenarios.

Multi-Object Generation: Several studies [122]–[125] have explored the generation of multiple composite objects, which not only ***concentrate on the individual objects*** but also aim to ***investigate the interactions between multiple objects.*** For predicting the interactions between multiple objects, CG3D [122] leverages SDS and probabilistic graph models extracted from text to predict the relative relationships between objects and incorporates priors such as gravity and contact relationships between objects, CG3D achieves models with realistic physical interactions.

Scene Generation: Unlike object-centric generation, scene generation typically requires the ***incorporation of additional information***, such as pre-trained depth estimation models [126], [127] or Large Language Models [124], [128], where the former provides spatial understanding for projecting images into 3D space, while the latter enhances the quality of text prompts. LucidDreamer[2] [126] employs a two-stage approach: first initializing point clouds using text-to-image and depth estimation models with inpainting [129] for consistency, then converting to 3DGS with extended image supervision.

4) 4D Generation: Analogous to static scene generation using text-to-image SDS, it is natural to consider that ***text-to-video SDS could potentially generate dynamic scenes [130]–[135].*** These works primarily focus on designing video-based SDS losses or exploring hybrid supervision with T2I (Text to Image) and T2V (Text to Video) models. As an example, Align Your Gaussians [130] adopts a two-stage approach: first reconstructing static 3DGS using MVDream [136] and text-to-image supervision, then extending to 4DGS with text-to-video guidance and simplified score distillation loss. Although these methods are effective, the inherent limitations of the aforementioned SDS-based paradigm persist. To mitigate this issue, subsequent works focus on ***generating pseudo-labeled images*** from additional views to facilitate dynamic 3DGS reconstruction [137]–[140]. Among them, 4DGen [137] generates multi-view pseudo-labels per frame, while employing Hexplane's [141] multi-scale features to maintain temporal consistency in 4DGS generation. Furthermore, some studies focus on ***animating static canonical 3DGS*** [142], [143] to achieve better control. Among them, BAGS [142] introduces neural bones and skinning weights to describe the spatial deformation based on canonical space. Using diffusion model priors and rigid body constraints, BAGS can be manually manipulated to achieve novel pose rendering.

C. Autonomous Driving

1) Autonomous Driving Scene Reconstruction: Reconstruction driving scenes is a challenging task, involving multiple technical domains such as large-scale scene reconstruction, dynamic object reconstruction, static object reconstruction, and Gaussian mixture reconstruction. Existing works [144]–[146] partitions the entire process into ***static background and dynamic target reconstruction.*** DrivingGaussian [144] reconstructs large-scale driving scenes by combining depth-binned static 3DGS for background and dynamic Gaussian graphs for multiple targets, utilizing multi-sensor data. Then StreetGaussians [145] extends this approach by incorporating semantic attributes and employing Fourier transforms for efficient SH temporal modeling in dynamic 3DGS. Subsequent works [147], aiming to further improve reconstruction efficiency, attempt to directly reconstruct entire scenes by Tightly Coupled LiDAR-Camera Gaussian Splatting. Moreover, 3DGS have been applied to ***multimodal spatiotemporal calibration tasks*** [148]. By leveraging the LiDAR point cloud as a reference for the Gaussians' positions, 3DGS-Calib [148] constructs a continuous scene representation and enforces both geometrical and photometric consistency across all sensors, achieving accurate and robust calibration with improved efficiency compared to NeRF-based works.

2) Simultaneous Localization and Mapping (SLAM): SLAM is a key problem in robotics and computer vision, where a device builds a map of an unknown environment while locating itself within it. Some studies [149]–[154] have retained the traditional inputs and approached this from two perspectives: ***online tracking and incremental mapping.*** In early work, GS-SLAM [149] utilizes 3DGS for SLAM with adaptive primitive expansion and employs a coarse-to-fine optimization strategy: first optimizing camera poses using sparse pixels, then refining them through selective re-rendering of reliable Gaussians. In parallel, Photo-SLAM [150] combines ORB features [155] and Gaussian attributes in a Hyper Primitives Map, utilizing LM optimization [156] and loop closure [155] for photorealistic SLAM reconstruction. While these approaches achieve higher efficiency than NeRF-based methods, further optimization of computational performance is essential for real-

TABLE III: Comparison of 3DGS-SLAM on Replica [163].

Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**Tracking RMSE↓**
GS-SLAM [164]	34.27	0.98	0.08	0.50
Photo-SLAM [150]	34.96	0.94	0.06	0.60
SplaTAM [151]	34.11	0.97	0.10	0.36
GS-ICP SLAM [154]	38.83	0.98	**0.04**	**0.16**
MotionGS [165]	**39.60**	0.98	**0.04**	0.49
LoopSplat [166]	36.63	**0.99**	0.11	0.26

world deployment. Therefore, CG-SLAM [153] leverages an uncertainty-aware 3D Gaussian field and a GPU-accelerated framework, achieving superior efficiency. Then, RGBD GS-ICP SLAM [154] enhances efficiency by integrating G-ICP [157] with shared covariances and scale alignment techniques for faster convergence. However, these methods remain susceptible to sensor noise in practical applications. Some quantitative reconstruction results are reported in Table III.

Incorporating scene understanding capabilities is equally crucial in SLAM tasks [158]–[160], thus prompting several works to ***integrate semantic information***. Among them, SGS-SLAM [158] employs multi-channel geometric, appearance, and semantic features for rendering and optimization and proposes a keyframe selection strategy based on geometric and semantic constraints to enhance performance and efficiency.

Additionally, there are several works focusing on related issues such as ***localization [161] and navigation [162].*** Specifically, 3DGS-ReLoc [161] leverages LiDAR initialization and 2D voxelized submaps with KD-tree for efficient memory usage, while achieving precise localization through feature-based PnP optimization. In the context of indoor navigation, GaussNav [162] focuses on the instance image navigation task. Based on reconstructed 3DGS maps, GaussNav proposes an image target navigation algorithm, achieving impressive performance through classification, matching, and path planning.

V. EXTENSIONS OF 3D GAUSSIAN SPLATTING

A. Dynamic 3D Gaussian Splatting

1) Multi-view Video Inputs: Some works [167], [168] attempted to ***directly construct dynamic 3DGS frame by frame.*** An early work [167] extends 3DGS to dynamic scenes by enabling temporal Gaussian motion while preserving static attributes. It employs online temporal reconstruction with previous-frame initialization and incorporates physical priors, including local rigidity, local rotational-similarity, and long-term local-isometry, for motion regularization. Despite promising results, these methods are limited to reconstructing only scene elements visible in the initial frame. To address this limitation, other works [169], [170] aim to achieve such performance by ***predicting deformations.*** SWAGS [169] proposes a window-based 4DGS with flow-guided adaptive window division and dynamic MLP optimization, employing inter-window consistency loss for seamless temporal reconstruction.

2) Monocular Video or Multi-view Image Inputs: Some works [171]–[177] tend to ***divide into two stages: canonical reconstruction and deformation prediction.*** The studies [171], [172] reconstruct static 3DGS and predicts temporal deformation in terms of positions, rotations and scales through position-time encoding. Similarly, 4D-GS [173] introduces the multi-scale HexPlane [141] as the foundational representation to encode temporal and spatial information. To further decouple motion and shape parameters in 4D-GS, ST-4DGS [176] introduces a spatial-temporally consistent 4DGS framework that incorporates motion-aware shape regularization and spatial-temporal density control to learn compact 4D representations. Although these methods achieve stable performance, they struggle to handle abrupt motions and sudden object appearances.

TABLE IV: Comparison of Dynamic 3DGS on D-NeRF [190].

Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**Train↓**	**FPS↑**
K-Planes [NeRF-based]	31.07	0.97	0.02	54 min	1.20
Deformable3DGS [171]	39.31	0.99	0.01	26 min	85.45
CoGS [191]	37.90	0.983	0.027	-	-
SC-GS [181]	**43.31**	**0.997**	**0.0063**	-	-
GauFRe [172]	34.5	0.98	0.02	13mins	112
Deformable4DGS [173]	32.99	0.97	0.05	13 min	104.00
RealTime4DGS [183]	32.71	0.97	0.03	10 min	289.07
4DRotorGS [184]	34.26	0.97	0.03	**5 min**	**1257.63**

Instead of discrete offsets, ***exploring temporally continuous motion*** can promote smoothness in the time dimension [178]–[181]. Gaussian-Flow [180] aims to develop a representation capable of fitting variable motion by analyzing the advantages and disadvantages of polynomial [179], [182] and Fourier series fitting [178]. It then proposes a dual-domain deformation model with adaptive time-step scaling and temporal-rigid constraints for stable and continuous motion prediction.

Recent works aim to ***extend 3DGS to 4D space*** for dynamic 3D scenes representation. Among them, the work [183] achieves end-to-end 4D training by jointly modeling spatial-temporal variables with 4D Gaussian primitives, incorporating 4D rotation, scaling, and temporal-aware spherical harmonics for color variation. Similarly, the work [184] introduces a rotor-based 4DGS representation with eight-component rotation decomposition, enabling temporal slicing for dynamic objects and enforcing 4D consistency through a dedicated loss. Although these approaches demonstrate robustness in complex scene reconstruction, the compressibility of their representations remains a notable consideration. Quantitative reconstruction results are reported in Table IV.

B. Surface Representation

Although 3DGS enables highly realistic rendering, extracting surface representations remains challenging. In this line of works, ***Signed Distance Functions*** are an indispensable topic [185]–[188].In early work, SuGaR [185] proposes an idealized SDF to constrain Gaussian surface alignment, enabling efficient mesh extraction through Poisson reconstruction and optional mesh-guided Gaussian refinement for high-quality results. Similarly, 3DGSR [187] integrates neural implicit SDF with 3DGS through a differentiable SDF-to-opacity transformation, maintaining consistency between volumetric and 3DGS-derived depth properties. Another line of research [186], [188] focuses on jointly optimizing NeuS [189] and 3DGS for surfaces. However, these methods exhibit limitations in handling unbounded scenes and computational overhead.

Other studies [192]–[195] aim to address this issue by ***improving 3DGS representation.*** The work [192] proposes Gaussian Surfels with depth-normal consistency loss and volumetric cutting for improved surface reconstruction, followed by screened Poisson mesh generation. Similarly, 2D Gaussian Splatting [194] (2DGS) replaces 3DGS with planar disks to represent surfaces, which are defined within the local tangent plane. Then, Gaussian Opacity Fields (GOF) [193] are developed based on 3DGS, wherein 3DGS is normalized along the ray to form a 1DGS for volume rendering. Although these methods achieve precise geometry reconstruction, they inevitably compromise rendering fidelity and face challenges in handling semi-transparent surfaces.

C. Editable 3D Gaussian Splatting

1) Manipulation by Text: To address this challenge, the existing works can be classified into two distinct categories. The first type introduces the ***score distillation loss***. These methods require editing prompts as additional conditions to guide the editing process [196], [197]. GaussianEditor [196] enables semantic-controlled 3DGS editing through SDS, utilizing hierarchical 3DGS and anchor loss for stability, while incorporating 2D inpainting guidance for object manipulation. Following Dreamgaussian, GSEdit [197] uses the pre-trained InstructPix2Pix [198] model instead of the image generation model for SDS. However, such methods remain constrained by pre-trained diffusion models, particularly when handling complex editing prompts. The second type focuses on ***editing multi-view images*** before reconstructing. GaussianEditor^2 [199] employs multimodal, language, and segmentation models to locate editable regions from text inputs, then optimizes targeted Gaussians based on 2D-edited images. However, this paradigm introduces an intuitive problem: ***how to ensure consistency in multi-view editing*** [200], [201]. GaussCtrl [200] introduces a depth-guided image editing network, ControlNet [202], utilizing its ability to perceive geometry and maintain multi-view consistency in the editing network. It also introduces a latent code alignment strategy in the attention layers, ensuring that the edited multiview images remain consistent with the references.

Unlike editing methods for 3DGS, recent discussions have increasingly focused on ***editing 4DGS***. Recent work, Control4D [203] leverages 4D GaussianPlanes to structurally decompose four-dimensional space, ensuring spatiotemporal consistency through Tensor4D representation, while incorporating a super-resolution GAN-based 4D generator [204] that learns from diffusion-generated edited images. However, it remains challenged on non-rigid movements.

2) Manipulation by Other Conditions: In addition to text-controlled editing, existing works have explored 3DGS editing methods under various conditions, such as mixed conditions [205] and points [206]. TIP-Editor [205] enables fine-grained 3DGS control through text, reference image, and location inputs, utilizing stepwise 2D personalization and coarse-fine editing strategies to support diverse tasks like object insertion and stylization. And Point'n Move [206] enables object-level editing through point annotations, utilizing a dual-stage process of segmentation, inpainting, and recomposition, which demonstrates improved control capability. Recent research [207] has introduced a training-free 3DGS splitting paradigm that achieves editing plane control by formulating it as a constrained minimization problem, preserving visual fidelity through moment conservation while mitigating Gaussian overflow via an analytically derived closed-form solution.

3) Stylization: In the realm of style transfer for 3DGS, early explorations have been made by [208]. Similar to traditional style transfer works [209], this work designs a 2D stylization module on the rendered images and a 3D color module on the 3DGS. By aligning the stylized 2D results of both modules, this approach achieves multi-view consistent 3DGS stylization without altering the geometry.

4) Animation: As described in Sec. V-A, some dynamic 3DGS works, such as SC-GS [181], can achieve animation effects by animating sparse control points. AIGC-related works, such as BAGS [142], aim to utilize video input and generation models to animate existing 3DGS. Similar research has also been mentioned in the context of Human Reconstruction. Additionally, CoGS [191] discusses how to control animation. Based on dynamic representations [167], [171], it uses a small MLP to extract relevant control signals and align the deformation of each Gaussian primitives.

D. Relightable

Relightable 3DGS has also emerged as one of the recent challenges gaining significant attention. ***Decoupling texture and lighting*** represents a common approach in relighting tasks. In early work, Relightable 3D Gaussian [210] and GS-IR [211] represent scenes using points with normal, BRDF, and decomposed lighting attributes for relighting. However, these methods face challenges in handling reflective scenes. Therefore, follow-up work [32] introduce accurate normal estimation and residual color terms to effectively model view-dependent reflections and complex lighting interactions. To address limitations in ***complex materials like semi-transparent volumes and furs***, OLAT-GS [212] decomposes observed color into the attenuated light intensity, received incident illumination and scattering value. And then, GS^3 [213] combines spatial and angular Gaussians with a triple splatting process to model geometry and reflectance properties, incorporating neural networks for self-shadowing and global illumination. Despite their capability in handling challenging geometry and appearance, further exploration is needed for transparent materials and indirect lighting.

E. Semantic Understanding

Endowing 3DGS with semantic understanding capabilities allows for the extension of 2D semantic models into 3D space, thereby enhancing the model's comprehension in 3D environments. This can be applied to various tasks such as 3D detection, segmentation, and editing. Many works attempt to leverage ***pre-trained 2D semantic-aware models*** for extra supervision on semantic attributes [214]–[217]. Feature 3DGS [215] leverages pre-trained 2D models to create joint 3DGS and feature fields, enabling spatial understanding through feature rasterization and regularization for downstream promptable tasks. However, these approaches remain constrained by multi-view consistency and open-world perception challenges. Subsequent research [218], [219] aims to incorporate contrastive learning losses as auxiliary supervision to achieve interactive 3D segmentation.

Others focus on ***incorporating text-visual alignment features*** for open-world understanding [220]–[222]. A significant challenge is the high dimensionality of CLIP features, which makes direct training and storage difficult compared to original Gaussian attributes. The work [220] introduces corresponding continuous semantic vectors into the 3DGS by extracting and discretizing dense features from CLIP [223] and DINO [224], which are used to predict semantic indices m in Discrete

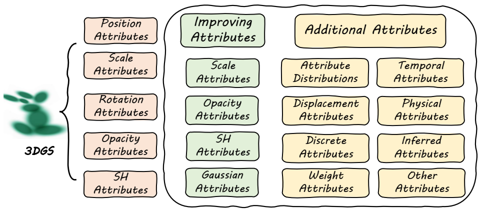

Fig. 3: Overview of Attribute Expansion Strategies.

Feature Space by MLPs as in VQ-VAE [225]. Subsequently, FMGS [222] mitigates the issue of large CLIP feature dimensions by introducing multi-resolution hash encoders [226].

F. Physics Simulation

Recent efforts aim to extend 3DGS to simulation. Based on the "what you see is what you simulate" philosophy, PhysGaussian [227] reconstructs a static 3DGS as the discretization of the scene to be simulated, and then incorporates continuum mechanics theory along with the Material Point Method [228] solver to endow 3DGS with physical properties. Similar approaches have also been explored in PhysDreamer [229]. Nevertheless, these physics-based simulations typically entail significant computational overhead. VR-GS [230] design an efficient interactive simulation system for VR, offering users a rich platform for 3D content manipulation.

VI. 3DGS Components and Improvements

A. Initialization

Proper initialization has been shown to be crucial, as it directly affects the optimization process [231]. The initialization of 3DGS is typically performed using sparse points derived from Structure-from-Motion (SfM) or through random generation. However, these methods are often unreliable, especially under weak supervision signals.

Combining pre-trained models is an optional approach. Pretraining a 3D model on a limited number of 3D samples and using it as an initialization prior is a viable strategy [50]. This approach can enhance the performance of initialization to some extent, although its effectiveness is contingent upon the data used. To address this limitation, pretrained 3D generative models [96], [97], [101] or monocular depth estimation models [61], [126], [127] are also introduced for initialization. Based on these, the work [96] also introduces some perturbation points to achieve a more comprehensive geometric representation.

Improving initialization strategies is also important. Based on the analysis of the role of SfM in capturing low-frequency signals within the spectrum, Sparse-large-variance initialization is designed to effectively focus on the low-frequency distribution identified by SfM [231].

Utilizing other representations can also enhance initialization capabilities. By determining the Local Volumes from a coarse model, a small number of Gaussians are initialized within each volume, thereby avoiding excessive assumptions [232]. Similarly, some initialization based on Visual Hull [63], Flame [77] or Mesh [79], [212], [233] are proposed, enabling the acquisition of structural geometric priors.

Discussion: The incorporation of pre-trained or structured information during the initialization of 3DGS is crucial, as high-fidelity initialization can mitigate training instabilities, particularly in under-determined scenarios. However, excessive reliance on improving initialization performance inevitably sacrifices efficiency, and depending on different downstream tasks, this is related to what is discussed in Sec. VII-B and Sec. VII-C. Therefore, reducing the impact of initialization on the reconstruction process remains a potential challenge.

B. Attribute Expansion

The original attributes of 3DGS include the position, scale, rotation, Spherical Harmonic (SH) coefficients, and opacity value. Some works have extended these attributes to make them more suitable for downstream tasks. It can be categorized into improvements of existing attributes or the introduction of novel attributes, as shown in Fig. 3.

1) Improving Attributes: Certain attributes of vanilla Gaussian can be custom-tailored, thereby making 3DGS suitable for a wider range of tasks.

Scale: By collapsing the z-scale to zero and incorporating additional supervision on depth, normal, or shell maps, the works [24], [91], [185], [192], [194] aim to improve Gaussian primitives to make them flatter and more suitable for surface reconstruction, where z direction can be approximated as the normal direction. Conversely, a scale constraint, which limits the ratio of the major axis length to the minor axis length [148], [227], [234], ensures that the Gaussian primitives remain spherical to mitigate the issue of unexpected plush artifacts caused by overly skinny kernels.

SH: By combining hash grids and MLP, the corresponding color attributes are encoded, effectively addressing the storage issues caused by a large number of SH parameters [12].

Opacity: By constraining the transparency to approach either 0 or 1, thereby minimizing the number of semi-transparent Gaussian primitives, the works [32], [185] achieve clearer Gaussian surfaces, effectively alleviating artifacts.

Gaussian: By introducing shape parameters, an attempt is made to replace the original Gaussians with a Generalized Exponential Family (GEF) mixture [15]. Traditional 3DGS can be viewed as a special case of the GEF mixture ($\beta = 2$), enhancing the representational efficiency of Gaussians,

2) Additional Attributes: By adding new attributes and corresponding supervisions, the original representation capabilities of 3DGS can be augmented.

Semantic Attributes: By introducing them and corresponding supervision, works such as [145], [146], [158]–[160], [215], [218], [219], [235] are endowed with enhanced spatial semantic awareness, which is crucial for tasks such as SLAM and editing. After the semantic attributes' splatting, the 3DGS's semantic attributes are supervised using 2D semantic segmentation maps. Additionally, methods to improve the extraction of semantic information [206] and introducing high-dimensional semantic-text, such as CLIP and DINO features [220]–[222], have been employed to address a wider range of downstream tasks. Similar to semantic attributes, the identity encoding attributes can group 3DGS that belong to the same instance or stuff [214], which is more effective for multi-object scenes.

Attribute Distributions: Learning position distributions with reparameterization techniques instead of a fixed value is an effective approach to prevent local minima [99] and mitigate its reliance on Adaptive Control of 3DGS [44]. In addition to these works focusing on the distribution prediction of position attributes, the distribution of the scale has also been incorporated [99]. By sampling the predicted distributions, Gaussian primitives can be obtained.

Temporal Attributes: Replacing the original static attributes with temporal attributes is key to animating the 3DGS [138], [145], [183], [184]. For 4D attributes, including rotation, scale, and position, existing works render 3DGS on timestep t by either taking time slices [184] or decoupling the t dimension from 4D attributes [138], [183]. Also, the introduction of 4D SH is crucial for time-varying color attributes. For this, the Fourier series is typically used as the adopted basis functions to endow SH with temporal capabilities [145], [183]. Note that due to involving different timesteps, these attributes often require video-based training. This regularization primarily aims to improve attributes in Gaussian primitives [24], [79], [123], [227], [234], as in Sec. VI-B.

Displacement Attributes: They can describe the relationship between the final and initial attributes in Gaussian primitives and be classified based on their condition. Condition-independent displacement attributes are often used to refine coarse attributes, which can be directly optimized in the same manner as other attributes [43]. Condition-dependent displacement attributes can describe the changes of static 3DGS, thereby achieving dynamic representations and controllable representations. This approach often involves introducing a small MLP to predict displacement based on timestep t [171]–[173], expression and other control signals [39], [69], [75], [78], [82]–[85], [191].

Physical Attributes: They encompass a broad range of properties describing the physical laws governing Gaussian primitives, thus endowing 3DGS with more realistic representation. For instance, shading-related attributes like diffuse color, direct specular reflection, residual color, shadow, and anisotropic spherical Gaussian can be used for specular reconstruction and relighting [32]–[34], [174], [212], [213]. Additionally, the velocity attributes can represent the transient information of Gaussian, essential for describing dynamic objects [175]. These attributes are typically optimized by considering the influence of physical laws at specific rendering positions [32], [34], [174] or by incorporating supplementary information, such as flow maps [175].

Discrete Attributes: Utilizing discrete attributes in place of continuous ones is an effective method for compressing high-dimensional representations and representing complex motion. This is often achieved by storing the index values of the VQ codebook [8]–[10], [12], [115] or the motion coefficient for motion basis [179] as the discrete attributes in Gaussian primitives. However, discrete attributes inevitable lead to performance degradation; combining them with compressed continuous attributes may be a potential solution [220].

Inferred Attributes: These attributes do not require optimization; they are inferred from other attributes. The *Parameter-Sensitivity attributes* reflects the impact of parameter changes on reconstruction performance and are represented by the gradient of the parameter, guiding compression clustering [10]. The *Pixel-Coverage attributes* determines the relative size of Gaussian primitives at the current resolution. It is related to the horizontal or vertical size of the Gaussian primitives and guides their scale to meet sampling requirements in multi-scale rendering [28].

Weight Attributes: They rely on structured representations, such as Local Volumes [232], Gaussian-kernel RBF [181], Mesh [233], and SMPL [236], to determine the attributes of query points by calculating the weights of structured points.

Other Attributes: The *Uncertainty Attributes* can help maintain training stability by reducing the loss weight in areas with high uncertainty [57], [220]. The *ORB-Features Attributes*, extracted from image frames [155], play a crucial role in establishing 2D-to-2D and 2D-to-3D correspondences [150].

Discussion: The modification of Gaussian attributes facilitates the execution of a wider range of downstream tasks, offering an efficient approach as it obviates the need for additional structural elements. Moreover, the integration of new attributes with supplementary information constraints also has the potential to significantly enhance the representational efficacy of the original 3DGS. For instance, semantic attributes can, in certain scenarios, yield more precise object boundaries. Note that adding new attributes, while endowing 3DGS with new capabilities, also leads to increased storage requirements, as discussed in Sec. V-E. Therefore, reasonable compression and additional regularization are necessary.

C. Splatting

The role of Splatting is to efficiently transform 3D Gaussian primitives into high-quality 2D images, ensuring smooth, continuous projections and significantly improving rendering efficiency. As a core technology in traditional computer graphics, there are also efforts aimed at improving it from the perspectives of efficiency and performance.

Several studies focus on ***enhancing splatting mechanisms***. By addressing projection errors caused by local affine approximation, the work [237] proposes an Optimal Projection Strategy, projecting each Gaussian radially onto a tangent plane determined by the line from the Gaussian mean to the camera center. To implement this, a Unit Sphere-Based Rasterizer is introduced, avoiding dense point sampling and enabling adaptability to various camera models such as fisheye and panoramic cameras. Subsequently, unlike traditional 3DGS, which samples Gaussian signals only at pixel centers, Analytic-Splatting [31] analytically approximates the Gaussian integral over the entire pixel area using a conditioned logistic function. By diagonalizing the covariance matrix, it efficiently handles 2D Gaussian integrals, enabling accurate pixel intensity response and robust anti-aliasing across varying resolutions. To enhance 3DGS performance in complex scenarios, a GPU-accelerated ray tracing algorithm is introduced [238] for semi-transparent particle-based representations, achieving real-time rendering with support for complex effects such as secondary rays, depth of field, and distorted cameras.

Additional improvements focus on ***optimizing splat ordering*** during the blending process. Unlike traditional global sorting in 3DGS, which causes depth inconsistencies during camera rotation, StopThePop [26] proposes a hierarchical per-pixel sorting strategy, which eliminates popping artifacts and ensures

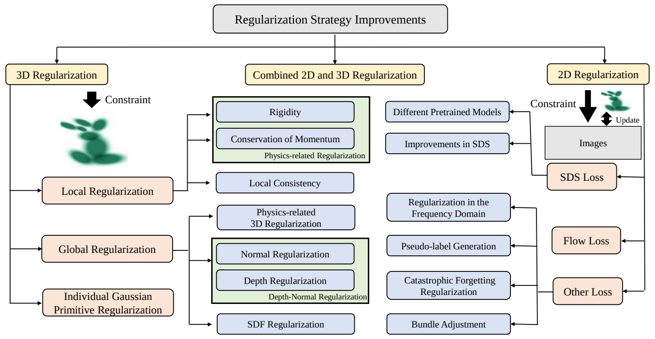

Fig. 4: Overview of Regularization Strategy Improvement.

view-consistent real-time rendering by accurately computing the depth of Gaussians along individual rays. With up to 50% fewer Gaussians, StopThePop achieves 1.6× faster rendering and 2× memory savings compared to 3DGS, while maintaining comparable quality and superior view consistency.

Discussion: Existing research on the advancement of Splatting techniques remains limited, and current approaches struggle to balance rendering performance with efficiency in complex scenes while being constrained by various camera models. Therefore, further discussion remains essential.

D. Regularization

Regularization is crucial for 3D reconstruction. We categorize the regularization terms into 2D and 3D regularization, as shown in Fig. 4. The 3D regularization directly constrains 3DGS, while the 2D regularization imposes constraints on the rendered images, thus influencing attribute optimization.

1) 3D Regularization: The 3D regularization has garnered significant attention due to its intuitive constraint capabilities. These efforts categorized based on their targeted objectives into individual Gaussian primitive, local, and global regularization.

Individual Gaussian Primitive Regularization: This regularization primarily aims to improve attributes in Gaussian primitives [24], [79], [123], [227], [234], as Sec. VI-B.

Local Regularization: Owing to the explicit representation of 3DGS, it is meaningful to impose constraints on Gaussian primitives within local regions. Such constraints can ensure the continuity and feasibility of Gaussian primitives in the local space. *Physics-related Regularization* is often used to ensure the local rigidity of deformable targets, which includes short-term local-rigidity loss, local rotation similarity loss, and long-term local isometry loss. Short-term local rigidity implies that nearby Gaussians should move following a rigid body transformation between time steps; Local rotation similarity enforces that adjacent Gaussian primitives have the same rotation over time steps; Long-term local isometry loss prevents elements of the scene from drifting apart [130], [132], [167], [174], [179], [181], [191], [232]. Subsequently, some works have also adopted similar paradigms to constrain local rigidity [142], [176], [180].

In addition to rigidity loss, there are some *Local Consistency Regularization* that aim to constrain the Gaussian primitives within local regions to maintain similar attributes, such as semantic [214], [216], [220], position [122], [137], time [180], frame [169], normal [239], weight [36] and depth [55], [61].

Global Regularization: Unlike the local regularization within neighboring regions, global regularization aims to constrain the overall 3DGS. *Physics-related Regularization* introduces real-world physical laws to constrain the state of 3DGS, including gravity loss and contact loss, among others.

Benefiting from the explicit representation, depth and normal attributes can be directly calculated and constrained during training, particularly for surface reconstruction tasks. *Depth-Normal Regularization* achieves depth-normal consistency by comparing the normal computed from depth values with the predicted normal [32], [187], [192], [193]. This method enforces constraints on both normal and depth simultaneously. Additionally, directly constraining either the normal or the depth is also feasible. *Normal Regularization* often adopts a self-supervised paradigm due to the lack of direct supervision signals, which can be implemented by designing pseudo-labels from gradients [186], the shortest axis direction of Gaussian primitives [32], or SDF [187], [188]. Similarly, *Depth Regularization* adopts a similar approach; however, it not only aims for accurate depth values but also seeks to ensure clear surfaces in 3DGS. Depth Distortion loss [18] aggregates Gaussian primitives along the ray. In addition to self-supervised methods, incorporating additional pre-trained models to estimate normal [192] and depth [33], [56], [58], [240] has proven to be more effective in *Normal Regularization* and *Depth Regularization.* SDF Regularization is also a constraint strategy for surface reconstruction. It achieves the desired surface by constraining the SDF that corresponds to 3DGS to an ideal distribution [185]–[188], [241].

2) 2D Regularization: Unlike the intuitive constraints in 3D, 2D regularization is often used to address under-constrained situations where original loss functions alone are insufficient.

SDS loss: An important example is the SDS loss, which uses a pre-trained 2D diffusion model to supervise 3DGS training via distillation paradigms [95], [196]. This approach extends to distill pre-trained 3D diffusion models [242], multi-view diffusion models [243], image editing models [198], and video diffusion models. Introducing 3D [96], [244] and multi-view diffusion models [112], [119], [123], [130]–[132], [137] enhances geometry and multi-view consistency. Image editing models [197] enable controllable edits, while video diffusion models [130] support dynamic temporal scene generation. Additionally, distillation on multi-modal images, like RGB-Depth [92], also holds potential, providing more constraints from pre-trained diffusion models. Some improvements specifically target inherent issues in SDS [100], [102]. Interval Score Matching is proposed to address issues of randomness and single-step sampling. Introducing Negative Prompts [245] is a method [92], [100], [244] to mitigate the impact of random noise ϵ and enhance stability by replacing random noise with negative prompts $[\epsilon_\phi(\boldsymbol{x}_t; y_{\text{neg}})]$. And, LODS incorporates LoRA terms [246] to replace traditional random noise ϵ, thereby alleviating the impact of out-of-distribution [102].

Flow loss: It is a commonly used regularization term for dynamic 3DGS and uses the output of a pre-trained 2D optical flow estimation model as ground truth. Predicted flow is rendered by calculating the displacement of Gaussian primitives over a unit time and splatting these 3D displacements onto a 2D plane [132], [146], [176], [178]. However, this approach has a significant gap since optical flow is a 2D attribute and susceptible to noise. Selecting Gaussians with correct depth and introducing uncertainty through KL divergence to constrain optical flow is a potentially feasible method [175].

Other loss: There are also some 2D regularization terms worth discussing. For example, constraining the differences in amplitude and phase between the rendered image and the ground truth in the frequency domain can serve as a loss function to aid training, thereby alleviating overfitting issues [25]. Introducing pseudo-labels for hypothetical viewpoints through noise perturbation can assist training in sparse-view settings [56]. In large-scale scene mapping, constraining the changes in attributes before and after optimization can prevent catastrophic forgetting in 3DGS [247]. Additionally, bundle adjustment is usually an important constraint in pose estimation problems [149], [150], [159].

Noted that, whether 2D or 3D regularization is used, overall updating is sometimes suboptimal due to the large number of primitives. Some primitives often have an uncontrollable impact on the results. Therefore, it is necessary to guide the optimization by selecting important primitives using methods such as visibility [151], [158], [160], [162].

Discussion: The incorporation of regularization terms serves as an effective approach to enhance the reconstruction performance of 3DGS. These regularization terms can impose constraints on various attributes of 3DGS, including geometry and spatial distribution, etc., in accordance with specific task requirements. Furthermore, in under-determined scenarios, the optimization process can be further constrained by introducing additional prior information, as discussed in Sec. VII-C. Given the varying effectiveness of regularization terms across different tasks and conditions, it is methodologically viable to select and simultaneously integrate multiple complementary constraints for a specific task-oriented optimization.

E. Training Strategy

Training strategy is also an important topic. In this section, we divide it into multi-stage training strategy and end-to-end training strategy, which can be applied to different tasks.

1) Multi-stage Training Strategy: Multi-stage training strategy is a common training paradigm, often involving coarse-to-fine reconstruction. It is widely used for under-determined tasks, such as AIGC, SLAM, etc..

Using different 3D representations in different training stages is a typical multi-stage training paradigm. 3DGS → Mesh (training 3DGS first, converting to Mesh, then optimizing Mesh) [95], [110], [119], [131], [197], [244] ensures geometric consistency in the generated 3D model. Additionally, generating multi-view images [104], [118], [138], [199]–[201], [244] in the first stage to aid reconstruction in the second stage can alleviate optimization difficulties.

Two-stage reconstruction for static and dynamic reconstruction is also important in dynamic 3DGS. This type of work typically involves training a time-independent static 3DGS in the first stage, and then training a time-dependent deformation field in the second stage to characterize dynamic Gaussians [171]–[174], [178], [203]. Additionally, incremental reconstruction of dynamic scenes frame by frame is also a focus in some works, often relying on the performance of previous reconstructions [167], [168].

In multi-objective optimization tasks, multi-stage training paradigms can enhance stability and performance. For example, the *coarse-to-fine camera tracking strategy* first obtains a coarse camera pose from a sparse pixel set, then refines it based on optimized rendering results [149], [161].

Additionally, some works aim to refine the 3DGS trained in the first stage [50], [63], [96], [101], [112], [205], [216], [232] or endow them with additional capabilities, such as semantics [162], [217] and stylization [208]. There are many such training strategies, which are also effective in maintaining training stability and avoiding local optima [13]. Furthermore, iterative optimization of the final result to enhance performance is also feasible [60], [201].

2) End-to-End Training Strategy: These strategies are often more efficient and can be applied to a wider range of downstream tasks.

Progressive Optimization Strategy: This commonly used strategy helps 3DGS prioritize learning global representations before locally optimizing details. In the frequency domain, this can be viewed as progressively learning from low-frequency to high-frequency components. It is often implemented by gradually increasing the proportion of high-frequency signals [25], [231] or introducing progressively larger image/feature sizes for supervision [8], [34], [150], which can also improve efficiency [41], [148]. In generative tasks, progressively selecting the camera pose is also an easy-to-difficult training strategy, optimizing from positions close to the initial viewpoint to those further away [119], [127].

Block Optimization Strategy: This strategy is often used in large-scale scene reconstruction to improve efficiency and alleviate catastrophic forgetting [152], [161], [162]. However, such paradigms are often influenced by block partitioning and training data selection. Consequently, several studies have proposed designing Primitives and Data Division strategies to mitigate workload imbalances caused by numerous empty blocks, while enhancing detail reconstruction capabilities [248]. To improve efficiency, introducing Level of Detail and hierarchical reconstruction prove effective, especially in large-scale scene processing [248], [249]. It can also achieve reconstruction by partitioning the scene into static backgrounds and dynamic objects [144]–[146], [172]. Additionally, this approach is applied in AIGC and Semantic Understanding, where refining submap reconstruction quality enhances overall performance [101], [221]. Unlike submaps divided by spatial regions, Gaussians can be categorized into different generations during their densification process, allowing for the application of distinct regularization strategies to each generation, effectively regulating their fluidity [196]. Categorizing Gaussians into those on smooth surfaces and independent points is also feasible for geometric representation. By designing distinct initialization and densification strategies, better representation can be achieved [239]. Additionally, some works design keyframe (or window) selection strategies based on inter-frame covisibility or geometric overlap ratio in temporal data for reconstructions [149], [151], [158], [169], [234], [247].

Robust Optimization Strategy: Introducing noise perturbations is a common method to enhance the robustness of training [47], [63], [110], [171]. Such perturbations can target camera poses, timesteps, and images, and can be regarded as a form of data augmentation to prevent overfitting. Additionally, some strategies mitigate catastrophic forgetting by avoiding continuous training from a single viewpoint [152], [154].

Distillation-based Strategy: To compress model parameters, some distillation strategies use the original 3DGS as the teacher model and a low-dimensional SH 3DGS as the student model, introducing more pseudo views to enhance the performance of the low-dimensional SH [13].

Discussion: Improving training strategies is an efficient way to optimize the training process of 3DGS and can enhance performance in many tasks. While multi-stage training strategies benefit from separately regularized training phases that often lead to substantial performance gains, they typically compromise efficiency. Consequently, promising future research directions can be pursued through two primary avenues: optimizing the efficiency of multi-stage training strategies and enhancing the performance of end-to-end training approaches.

F. Adaptive Control

Adaptive Control of 3DGS is an important process for regulating the number of Gaussian primitives, including cloning, splitting, and pruning. In the following sections, we will summarize existing techniques from the perspectives of densification (cloning and splitting) and pruning.

1) Densification: Densification is crucial, especially for detail reconstruction. we will analyze it from the perspectives of "Where to densify" and "How to densify". Additionally, we will discuss how to avoid excessive densification.

Where to Densification: Densification techniques focus on identifying positions requiring densification, governed by gradients in the original 3DGS and extendable to dynamic scene reconstruction [168]. Regions with low opacity, silhouette, or high depth-rendered error, considered unreliable, guide densification to fill holes or improve 3D inconsistencies [24], [149], [162], [182], [192], [247]. Some approaches improve based on gradients by weighting the number of pixels covered by each Gaussian in different views to dynamically average view gradients, enhancing point cloud growth conditions [250]. Additionally, SDF value, motion masks and neighbor distance are important criteria, with locations closer to the surface, motion regions and lower compactness being more prone to densification [96], [176], [185], [188].

How to Densification: Numerous works have improved densification methods. Graph structures explore Gaussian relationships and define new Gaussians at edge centers based on proximity scores, mitigating sparse viewpoint impacts [56]. To prevent excessive Gaussian growth, the Candidate Pool Strategy stores pruned Gaussians for densification [113]. Additionally, work [251] introduces three conservation rules and employs integral tensor equations for visual consistency.

Excessive densification is also unnecessary, as it directly impacts the efficiency of 3DGS. In cases where two Gaussian functions are in close proximity, limiting their densification is a straightforward idea, where the distance between Gaussians can be measured by Gaussian Divergent Significance [118] (GDS) or Kullback–Leibler divergence [68].

And DeblurGS [38] incorporates a Gaussian Densification Annealing strategy to prevent the densification of inaccurate Gaussians during the early training stages at imprecise camera motion estimation. Furthermore, in some downstream tasks, densification is sometimes abandoned to prevent 3DGS from overfitting to each image, which could lead to incorrect geometric shapes [148], [149], [151], [234].

2) Pruning: Removing unimportant Gaussian primitives can ensure efficient representation. In the initial 3DGS framework, opacity was employed as the criterion for determining the significance of a Gaussian. Subsequent research has explored the incorporation of scale as a guiding factor or distractor masks for pruning [47], [92]. However, these approaches primarily focus on individual Gaussian primitives, lacking a comprehensive consideration of the global representation. Therefore, subsequent derivative techniques are discussed.

Importance scores: The volume and hit count on training views, along with opacity, can be used to jointly determine the global significance score of a Gaussian primitive [13]. Subsequently, Gaussians are ranked according to their global scores, and the ones with the lowest scores are pruned. Similar importance scores were improved in other works [252], [253].

Multi-view consistency: Multi-view consistency is a key criterion for determining whether Gaussians need to be pruned. For example, [234] prunes newly added Gaussians that are not observed by three keyframes within a local keyframe window, while [160] prunes Gaussians that are invisible in all virtual views but visible in real views.

Distance Metric: Surface-aware methods often use distance to the surface [149] and SDF values [188] to prune Gaussian primitives far from the surface. The distance between Gaussians is also a key metric [176]. GauHuman [68] aims to merge Gaussians with small scale and low KL divergence.

Learnable control parameter: Introducing a learnable mask based on scale and opacity to determine whether Gaussian primitives should be removed effectively prevents 3DGS from becoming overly dense [12].

Others: CoR-GS [62] aims to leverage mismatched regions between two 3DGS models, trained in parallel under identical conditions, as guidance for pruning.

Discussion: Adaptive Control strategies play a pivotal role in enhancing rendering fidelity and computational efficiency. However, excessive densification or pruning can adversely affect both the efficiency and performance of 3D Gaussian Splatting. Therefore, it is crucial to examine and establish an optimal balance between these two strategic approaches.

VII. Other Technical Discussions

A. Post-Processing

Post-processing strategies for pre-trained Gaussians are important, as they can improve the original efficiency and performance. Common post-processing often enhances Gaussian representations through various optimization strategies. This type of work has been discussed in Sec. VI-E.

Representation Conversion: Pre-trained 3DGS can be converted to Mesh using Poisson reconstruction [254] on sampled 3D points [185], [192]. Similarly, GOF [193] uses 3D bounding boxes to convert 3DGS to a Tetrahedral Grid, then extracts meshes using Binary Search of Level Set. Additionally, LGM [110] converts Pre-trained 3DGS to NeRF, then uses NeRF2Mesh [255] for Mesh conversion.

Performance and Efficiency: Some works enhance 3DGS performance in specific tasks through post-processing, such as multi-scale rendering. SA-GS [30] introduces a 2D scale-adaptive filter to dynamically adjust scales based on rendering frequency, enhancing anti-aliasing when zooming out. For efficiency, removing redundant Gaussian primitives from pre-trained 3DGS [21] or introducing a Gaussian caching mechanism [256] can improve rendering efficiency.

B. Integration with Other Representations

The convertible nature of 3D representations facilitates the integration of 3DGS with other representations, leveraging their advantages to improve the original 3DGS.

1) Point Clouds: Point clouds, as a 3D representation related to 3DGS, are often used to initialize positions. Converting point clouds to 3DGS can effectively fill holes [126], [127] or improve reconstruction details [114], typically after high-precision reconstruction. Conversely, 3DGS can be converted into point clouds, voxelized into 3D voxels, and projected onto 2D BEV grids, which guide navigation tasks [162]. Additionally, anchor points in space can assist 3DGS. These methods use voxel centers as anchor points to represent the scene. Each anchor point comprises a local context feature, a scaling factor, and multiple learnable offsets. By decoding other attributes based on these offsets and features, the anchors transform into local neural Gaussians, which helps mitigate redundant expansion [14], [34], [188].

2) Mesh: Meshes have better geometric representation capabilities and can, to some extent, alleviate artifacts or blurry pixels caused by 3DGS [170]. They are still the most widely used 3D representation in downstream tasks [110]. Much work has discussed converting 3DGS to Mesh, as mentioned in Sec. V-B. Once converted, they can be optimized for better geometry and appearance [70], [95], [131], [197]. Jointly optimizing 3DGS and Mesh is also an optional strategy. 3DGS is suitable for constructing complex geometric structures, while Mesh can be used to reconstruct detailed color appearances on smooth surfaces. Combining the two can enhance reconstruction performance [170] and large-scale deformation control [233].

3) Triplane: Triplane, known for its compactness and efficient expressiveness [49], is often used in generalization tasks. It consists of three orthogonal feature planes: X-Y, Y-Z, and X-Z. Features can be obtained by querying positions in the space, and subsequently decoded to predict Gaussian attributes [49], [50], [66], [109]. Recent works [137], [173], [176], [203] extend triplane to 4D space (XYZ-T) using multi-scale HexPlanes [141] or 4D GaussianPlanes [203] to enhance 4DGS continuity in the spatiotemporal dimension.

4) Grid: Grid is also an efficient representation, as it can access grid corners and interpolate to obtain features or attributes at specific positions. Hash grid [226], a representative method, can compress scenes and achieve a more compact and efficient 3DGS [12], [17], [69], [113], [115], [148], [208], [222]. Furthermore, Self-Organizing Gaussian [16] maps unstructured 3D Gaussians onto a 2D grid to preserve local spatial relationships, where adjacent Gaussians have similar attribute values, reducing memory storage and maintaining continuity in 3D space.

Particularly, GaussianVolumes are also used for generalizable representations [113], where a volume is composed of a fixed number of 3DGS. This maintains the efficiency of 3DGS and offers greater manipulability compared to triplane.

5) Implicit Representation: Implicit representations, benefiting from their representational capability, can be used to mitigate the condition difficulty and surface artifacts of 3DGS [90]. Specifically, introducing NeRF to encode color and opacity can significantly enhance the representation's adjustability [257]. Moreover, by designing an SDF-to-opacity transformation function [187] or employing mutual geometry supervision [188] to jointly optimize 3DGS and SDF representations, the surface reconstruction performance of 3DGS can be improved.

Discussion: Given the inherently unstructured characteristics of 3DGS, the incorporation of structured representations emerges as a viable prior, particularly advantageous for tasks such as human body reconstruction, facilitating both cross-representation transformations and geometric reconstruction, thereby enabling enhanced performance in downstream applications. However, excessive reliance on other representations leads to some degradation in rendering performance, potentially due to their additional influence on the distribution of Gaussian primitives. Therefore, investigating extra adaptive control mechanisms emerges as one potential solution to this challenge.

C. Guidance by Additional Prior

When dealing with under-determined problems, such as sparse view settings III-C, introducing additional priors is a straightforward method to improve 3DGS performance.

Pre-trained Models: Introducing pre-trained models is an effective paradigm that can guide the optimization through the model's knowledge. Pre-trained monocular depth models and point cloud prediction models are a common type of priors, where the predicted depth values and positions can be used for the initialization and regularization [55]–[57], [61], [126], [127], [160]. Pre-trained 2D image (or 3D and video) generative models are also important in some AIGC-related tasks. They can be used not only for optimization in combination with SDS Loss [96], [130], [244] but also for directly generating (or editing) images for training [60], [104], [126], [127], [138]. Similarly, some works introduce pre-trained image inpainting networks to alleviate difficulties caused by occlusion as well as overlap [119], [126], [127], [196], [206] or super-resolution models for a high level of detail [127], [203] during the generation process. Additionally, pre-trained ControlNet [202] or Large Language Models can also be used to guide 3D generation. The former can enhance geometric consistency under depth guidance [119], [123], [200], while the latter can predict layout maps to guide spatial relationships in multi-object 3D generation scenarios [123]. Notably, certain pre-trained models can endow 3DGS with additional capabilities, such as semantic understanding models, as discussed in Sec. V-E and spatial understanding models [160].

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3538684

TABLE V: The Relationships among Challenges, Tasks, and Technological Improvement, where the first column represents the core challenges, while the second and third columns denote the related downstream tasks and technological advancements.

Challenges	Major Tasks	Major Technological Improvements
Suboptimal Data Challenges VIII-B	**Limited Number**: Sparse Views III-D, Autonomous Driving Reconstruction IV-C1, Dynamic 3DGS V-A (Monocular Video Part V-A2), AIGC IV-B and Editable 3DGS V-C. **Limited Quality**: Slam IV-C2 and Reconstruction under blurred images (Sec.III-B) or without poses [261], [262]	**Limited Number**: Initialization VI-A, Regularization VI-D, Adaptive Control VI-F Training Strategies VI-E, and Guidance by Additional Prior VII-C. **Limited Quality**: Training Strategies VI-E and Integration with Other Representations (Sec. VII-B)
Generalization Challenges VIII-C	Generalization III-C, and Generalization-related tasks in the Human Reconstruction IV-A and AIGC IV-B	Initialization VI-A, Adaptive Control VI-F, and Integration with Other Representations VII-B
Physics Challenges VIII-D	**Physical Motion**: Dynamic 3DGS V-A, Physics Simulation V-F, Animation V-C4, Dynamic Human IV-A and Autonomous Driving Reconstruction IV-C1. **Physical Rendering**: Photorealism III-B and Physics Simulation V-F.	**Physical Motion and Rendering**: Attribute Expansion VI-B, Regularization Strategy VI-D, and Guidance by Additional Prior VII-C.
Realness and Efficiency Challenges VIII-E	**Realness**: Photorealism III-B, Surface Reconstruction V-B, Semantic Understanding V-E, some AIGC-related IV-B and Autonomous Driving IV-C works. **Efficiency**: Efficiency III-A, some works in Autonomous Driving IV-C and Semantic Understanding V-E	**Realness**: Most of the technologies in V-F. **Efficiency**: Attribute Expansion VI-B, Post-Processing VII-A, Adaptive Control VI-F and Splatting VI-C.

More Sensors: Due to the 3D-agnostic nature of 2D images, reconstructing 3DGS can be challenging, especially in large-scale reconstructions such as SLAM and autonomous driving. Therefore, incorporating additional sensors for 3D information, including depth [152], [154], [159], [160], [247], audio [86]–[88], LiDAR [48], [144], [147], [148], [161], and optical tactile sensors [57], has the potential to alleviate this issue.

Task-specific Priors: Some reconstruction tasks, such as human reconstruction, target subjects with certain common characteristics. These characteristics, such as template models and Linear Blend Skinning, can be extracted as priors to guide the reconstruction of similar targets. In the reconstruction, animation, and generation of non-rigid objects, many works utilize SMPL [73] and SMAL [258] to provide strong priors for representing the motion and deformation of non-rigid objects like humans [64], [66], [68], [69], [92] and animals [142], [236]. Subsequently, based on the SMPL template, Shell Maps [259] and template meshes are also introduced in combination with 3DGS to address issues of low efficiency in 3DGAN [91], [93] and unclear geometry [70], [71]. Similarly, in head and face reconstruction and animation tasks, some works [76], [81] also use the FLAME model [80] as a prior. Linear Blend Skinning [260] is also employed as prior knowledge to assist in the prediction of 3DGS motion [81], [181]. Additionally, in 3D urban scene reconstruction tasks, HUGS [146] introduces the Unicycle Model to model the motion of vehicles, thereby making the motion modeling of moving objects smoother.

Discussion: Accessible auxiliary information has the capability to enhance the performance of 3DGS across numerous tasks, serving as prior knowledge to facilitate spatial comprehension, particularly in inherently ill-posed problems. Although certain priors or sensors may lead to increased computational overhead and costs, they have the capability to significantly enhance the representational capacity of 3DGS.

VIII. Challenges and Opportunities

The preceding discussion indicates that various 3DGS-related tasks share similar technical approaches, which stems from common challenges across different tasks. To provide readers with a deeper understanding of this phenomenon, this section examines the commonalities among different tasks, summarizes four core challenges as well as their corresponding technical solutions, and outlines future opportunities.

A. Interrelationships

We have extensively discussed various 3DGS-related tasks in Sec.III, Sec.IV, and Sec. V, revealing common challenges and techniques across these tasks. As illustrated in Tab.V, we categorize existing tasks according to four core challenges, demonstrating that solutions from different tasks can be mutually instructive. Furthermore, there are some interrelationships between different tasks that have not been mentioned. For instance, Surface Reconstruction techniques (Sec.V-B) are often referenced in the context of Editable 3DGS (Sec.V-C), etc. We anticipate that this analysis will offer valuable insights for future research endeavors in related tasks.

B. Suboptimal Data

Challenges. In real-world scenarios, collecting large volumes of high-quality training data is often impractical. Without access to 3D data and sufficient multi-view images, relying on limited 2D image supervision is insufficient for accurate 3DGS reconstruction. For example, inferring the back appearance from only a frontal image is highly challenging. Additionally, data quality is critical, as accurate poses and clear images directly influence reconstruction performance.

Opportunities. An ideal 3DGS training process requires sufficient high-quality data, but this is often excessively challenging in practical applications. Although introducing priors can mitigate this problem to some extent, optimizing a large number of Gaussians under underconstrained conditions remains difficult. A potential solution is to reduce the number of Gaussian primitives based on their uncertainty while enhancing the representational capacity of individual primitives [15]. This involves finding a trade-off between the number of Gaussians and rendering performance, thereby improving the efficiency of utilizing sparse samples. Then, poor-quality data should also be taken into consideration. Unconstrained in-the-wild images are a typical case, encompassing transient occlusions and dynamic appearance changes, such as varying sky, weather, and lighting, which have been extensively discussed in NeRF [263]–[265]. To enhance efficiency, existing works have addressed this issue in the context of 3DGS [266], [267], attempting to model appearance changes and handle transient objects. However, their performance struggles, especially in scenes with complex lighting changes and frequent occlusions. Thanks to the explicit representation characteristics of 3DGS, decoupling geometric representations and introducing geometric consistency constraints across different scenes is a promising approach to mitigate instability during the training process.

C. Generalization

Challenges. Despite the improved training efficiency compared to NeRF, the scene-specific training paradigm remains a major bottleneck for the application of 3DGS. It is hard to imagine having to train for each target or scene individually, especially in multi-target and scene reconstruction (generation).

Opportunities. Although existing generalization-related works can directly obtain scene representations through forward inference, their performance is often unsatisfactory and limited by the type of scene [43], [45], [49], [109]. We hypothesize that this is due to the difficulty of feedforward networks in performing the adaptive control of 3DGS, as also mentioned in [44]. In future research, designing a reference-feature-based feedforward adaptive control strategy is a potential solution, which can predict the positions requiring adaptive control through reference features and be plug-and-play into existing generalization-related works. Additionally, existing generalization-related works rely on accurate poses, which are often difficult to obtain in practical applications [262], [268], [269]. Therefore, discussing generalizable 3DGS under pose-missing conditions is also promising [256].

D. Physics Reconstruction and Rendering

Challenges. Traditional 3DGS only considers static rendering and neglects the laws of physical motion, which are important in simulations [227]. Additionally, Physically-based rendering is a significant step towards applying 3DGS to simulate the physical world and achieve more realistic effects.

Opportunities. Ensuring that the 3DGS's motion adheres to physical laws is essential for unifying simulation and rendering [227]. Although rigidity-related regularization have been introduced, as described in Sec. VI-D1, most existing works focus on animating 3DGS while neglecting the physical attributes of the Gaussian primitives themselves (Sec. V-A). Some works attempt to introduce velocity attributes [175] and Newtonian dynamics rules [227], but this is not sufficient to fully describe the physical motion of 3DGS. A potential solution is to introduce more physical attributes in Gaussian primitives, such as material [211], acceleration, and force distribution, which can be regularized by priors from certain simulation tools and physics knowledge. Physically-based rendering is also a direction worth attention, as it enables 3DGS to handle relighting and material editing, producing outstanding inverse rendering results [210]. Future works can explore decoupling geometry and appearance in 3DGS, conducting research from the perspectives of normal reconstruction and the modeling of illumination and materials [90], [211], [270].

E. Realness and Efficiency

Challenges. Realness and efficiency challenges are fundamental issues. They are investigated in various works and have been discussed in Sec. III. In this part, we discuss downstream tasks and techniques optimized for performance and efficiency.

Opportunities. The difficulty in reconstructing clear surfaces has always been a significant challenge affecting rendering realism. As discussed in Sec. V-B, some works have addressed it by attempting to represent surfaces with planar Gaussians. However, this can result in a decline in rendering performance, possibly due to the reduced representational capacity of planar Gaussian primitives or the training instability. Therefore, designing Gaussian primitives better suited for surface representation and introducing a multi-stage training paradigm along with regularization are potential solutions. Storage efficiency is one of the critical bottlenecks of 3DGS. Existing works focus on introducing VQ techniques and compressing SH parameters, as discussed in Sec. III-A1. However, such approaches inevitably affect performance. Therefore, exploring how to design more efficient representations based on 3DGS is a potential way to enhance efficiency [14], [15] while maintaining performance.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106.

[2] G. Chen and W. Wang, "A survey on 3d gaussian splatting," *arXiv:2401.03890*.

[3] T. Wu, Y.-J. Yuan, L.-X. Zhang, J. Yang, Y.-P. Cao, L.-Q. Yan, and L. Gao, "Recent advances in 3d gaussian splatting," *Comput. Vis. Media*, pp. 1–30.

[4] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3d gaussian splatting as new era: A survey," *IEEE Trans. Vis. Comput. Graph.*

[5] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–14.

[6] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Ewa volume splatting," in *Proc. IEEE Vis.* IEEE, pp. 29–538.

[7] J. Ding, Y. He, B. Yuan, Z. Yuan, P. Zhou, J. Yu, and X. Lou, "Ray reordering for hardware-accelerated neural volume rendering," *IEEE Trans. Circuits Syst. Video Technol.*

[8] S. Girish, K. Gupta, and A. Shrivastava, "Eagles: Efficient accelerated 3d gaussians with lightweight encodings," *arXiv:2312.04564*.

[9] K. Navaneet, K. P. Meibodi, S. A. Koohpayegani, and H. Pirsiavash, "Compact3d: Compressing gaussian splat radiance field models with vector quantization," *arXiv:2311.18159*.

[10] S. Niedermayr, J. Stumpfegger, and R. Westermann, "Compressed 3d gaussian splatting for accelerated novel view synthesis," *arXiv:2401.02436*.

[11] W. H. Equitz, "A new vector quantization clustering algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 10, pp. 1568–1575.

[12] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park, "Compact 3d gaussian representation for radiance field," *arXiv:2311.13681*.

[13] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, and Z. Wang, "Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps," *arXiv:2311.17245*.

[14] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering," *arXiv:2312.00109*.

[15] A. Hamdi, L. Melas-Kyriazi, J. Mai, G. Qian, R. Liu, C. Vondrick, B. Ghanem, and A. Vedaldi, "Ges: Generalized exponential splatting for efficient radiance field rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 19 812–19 822.

[16] W. Morgenstern, F. Barthel, A. Hilsmann, and P. Eisert, "Compact 3d scene representation via self-organizing gaussian grids," *arXiv:2312.13299*.

[17] Y. Chen, Q. Wu, W. Lin, M. Harandi, and J. Cai, "Hac: Hash-grid assisted context for 3d gaussian splatting compression," in *Eur. Conf. Comput. Vis.* Springer, pp. 422–438.

[18] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 5470–5479.

[19] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park, "Compact 3d gaussian representation for radiance field," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 21 719–21 728.

[20] S. Durvasula, A. Zhao, F. Chen, R. Liang, P. K. Sanjaya, and N. Vijaykumar, "Distwar: Fast differentiable rendering on raster-based rendering pipelines," *arXiv:2401.05345*.

[21] J. Jo, H. Kim, and J. Park, "Identifying unnecessary 3d gaussians using clustering for fast rendering of 3d gaussian splatting," *arXiv:2402.13827*.

[22] J. Lee, S. Lee, J. Lee, J. Park, and J. Sim, "Gscore: Efficient radiance field rendering via architectural support for 3d gaussian splatting," in *Proc. 29th ACM Int. Conf. Archit. Support Program. Lang. Oper. Syst., Volume 3*, pp. 497–511.

[23] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* IEEE, pp. 5481–5490.

[24] K. Cheng, X. Long, K. Yang, Y. Yao, W. Yin, Y. Ma, W. Wang, and X. Chen, "Gaussianpro: 3d gaussian splatting with progressive propagation," *arXiv:2402.14650*.

[25] J. Zhang, F. Zhan, M. Xu, S. Lu, and E. Xing, "Fregs: 3d gaussian splatting with progressive frequency regularization," *arXiv:2403.06908*.

[26] L. Radl, M. Steiner, M. Parger, A. Weinrauch, B. Kerbl, and M. Steinberger, "Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering," *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–17.

[27] S. Diolatzis, T. Zirr, A. Kuznetsov, G. Kopanas, and A. Kaplanyan, "N-dimensional gaussians for fitting of high dimensional functions," in *ACM SIGGRAPH Conf. Papers*, pp. 1–11.

[28] Z. Yan, W. F. Low, Y. Chen, and G. H. Lee, "Multi-scale 3d gaussian splatting for anti-aliased rendering," *arXiv:2311.17089*.

[29] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, "Mip-splatting: Alias-free 3d gaussian splatting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 19 447–19 456.

[30] X. Song, J. Zheng, S. Yuan, H.-a. Gao, J. Zhao, X. He, W. Gu, and H. Zhao, "Sa-gs: Scale-adaptive gaussian splatting for training-free anti-aliasing," *arXiv:2403.19615*.

[31] Z. Liang, Q. Zhang, W. Hu, L. Zhu, Y. Feng, and K. Jia, "Analytic-splatting: Anti-aliased 3d gaussian splatting via analytic integration," in *Eur. Conf. Comput. Vis.* Springer, pp. 281–297.

[32] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, W. Wang, and Y. Ma, "Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 5322–5332.

[33] J. Meng, H. Li, Y. Wu, Q. Gao, S. Yang, J. Zhang, and S. Ma, "Mirror-3dgs: Incorporating mirror reflections into 3d gaussian splatting," *arXiv:2404.01168*.

[34] Z. Yang, X. Gao, Y. Sun, Y. Huang, X. Lyu, W. Zhou, S. Jiao, X. Qi, and X. Jin, "Spec-gaussian: Anisotropic view-dependent appearance for 3d gaussian splatting," *arXiv:2402.15870*.

[35] K. Ye, Q. Hou, and K. Zhou, "3d gaussian splatting with deferred reflection," in *ACM SIGGRAPH Conf. Papers*, pp. 1–10.

[36] X. Wu, J. Xu, C. Wang, Y. Peng, Q. Huang, J. Tompkin, and W. Xu, "Local gaussian density mixtures for unstructured lumigraph rendering," in *ACM SIGGRAPH Conf. Papers*, ser. SA '24. New York, NY, USA: Association for Computing Machinery. [Online]. Available: https://doi.org/10.1145/3680528.3687659

[37] J. Liu, X. Tang, F. Cheng, R. Yang, Z. Li, J. Liu, Y. Huang, J. Lin, S. Liu, X. Wu *et al.*, "Mirrorgaussian: Reflecting 3d gaussians for reconstructing mirror reflections," in *Eur. Conf. Comput. Vis.* Springer, pp. 377–393.

[38] J. Oh, J. Chung, D. Lee, and K. M. Lee, "Deblurgs: Gaussian splatting for camera motion blur," *arXiv:2404.11358*.

[39] B. Lee, H. Lee, X. Sun, U. Ali, and E. Park, "Deblurring 3d gaussian splatting," in *Eur. Conf. Comput. Vis.* Springer, pp. 127–143.

[40] O. Seiskari, J. Ylilammi, V. Kaatrasalo, P. Rantalankila, M. Turkulainen, J. Kannala, E. Rahtu, and A. Solin, "Gaussian splatting on the move: Blur and rolling shutter compensation for natural camera motion," in *Eur. Conf. Comput. Vis.* Springer, pp. 160–177.

[41] C. Peng, Y. Tang, Y. Zhou, N. Wang, X. Liu, D. Li, and R. Chellappa, "Bags: Blur agnostic gaussian splatting through multi-scale kernel modeling," in *Eur. Conf. Comput. Vis.* Springer, pp. 293–310.

[42] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet models for refraction through rough surfaces," in *Proc. 18th Eurographics Conf. Rendering Tech.*, pp. 195–206.

[43] S. Szymanowicz, C. Rupprecht, and A. Vedaldi, "Splatter image: Ultra-fast single-view 3d reconstruction," *arXiv:2312.13150*.

[44] D. Charatan, S. Li, A. Tagliasacchi, and V. Sitzmann, "Pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction," *arXiv:2312.12337*.

[45] Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai, "Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images," *arXiv:2403.14627*.

[46] Y. Wang, T. Huang, H. Chen, and G. H. Lee, "Freesplat: Generalizable 3d gaussian splatting towards free-view synthesis of indoor scenes," *arXiv:2405.17958*.

[47] Y. Bao, J. Liao, J. Huo, and Y. Gao, "Distractor-free generalizable 3d gaussian splatting," *arXiv:2411.17605*.

[48] Y. Chen, J. Wang, Z. Yang, S. Manivasagam, and R. Urtasun, "G3r: Gradient guided generalizable reconstruction," in *Eur. Conf. Comput. Vis.* Springer, pp. 305–323.

[49] Z.-X. Zou, Z. Yu, Y.-C. Guo, Y. Li, D. Liang, Y.-P. Cao, and S.-H. Zhang, "Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers," *arXiv:2312.09147*.

[50] D. Xu, Y. Yuan, M. Mardani, S. Liu, J. Song, Z. Wang, and A. Vahdat, "Agg: Amortized generative 3d gaussians for single image to 3d," *arXiv:2401.04099*.

[51] K. Zhang, S. Bi, H. Tan, Y. Xiangli, N. Zhao, K. Sunkavalli, and Z. Xu, "Gs-lrm: Large reconstruction model for 3d gaussian splatting," *Eur. Conf. Comput. Vis.*

[52] Y. Xu, Z. Shi, W. Yifan, H. Chen, C. Yang, S. Peng, Y. Shen, and G. Wetzstein, "Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation," *arXiv:2403.14621*.

[53] S. Guo, Q. Wang, Y. Gao, R. Xie, L. Li, F. Zhu, and L. Song, "Depth-guided robust point cloud fusion nerf for sparse input views," *IEEE Trans. Circuits Syst. Video Technol.*

[54] Y. Bao, Y. Li, J. Huo, T. Ding, X. Liang, W. Li, and Y. Gao, "Where and how: Mitigating confusion in neural radiance fields from sparse inputs," in *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 2180–2188.

[55] J. Chung, J. Oh, and K. M. Lee, "Depth-regularized optimization for 3d gaussian splatting in few-shot images," *arXiv:2311.13398*.

[56] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, "Fsgs: Real-time few-shot view synthesis using gaussian splatting," *arXiv:2312.00451*.

[57] A. Swann, M. Strong, W. K. Do, G. S. Camps, M. Schwager, and M. Kennedy III, "Touch-gs: Visual-tactile supervised 3d gaussian splatting," *arXiv:2403.09875*.

[58] J. Li, J. Zhang, X. Bai, J. Zheng, X. Ning, J. Zhou, and L. Gu, "Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization," *arXiv:2403.06912*.

[59] Z. Liu, J. Su, G. Cai, Y. Chen, B. Zeng, and Z. Wang, "Georgs: Geometric regularization for real-time novel view synthesis from sparse inputs," *IEEE Trans. Circuits Syst. Video Technol.*

[60] X. Liu, J. Chen, S.-H. Kao, Y.-W. Tai, and C.-K. Tang, "Deceptive-nerf/3dgs: Diffusion-generated pseudo-observations for high-quality sparse-view reconstruction," in *Eur. Conf. Comput. Vis.* Springer, pp. 337–355.

[61] A. Paliwal, W. Ye, J. Xiong, D. Kotovenko, R. Ranjan, V. Chandra, and N. K. Kalantari, "Coherentgs: Sparse novel view synthesis with coherent 3d gaussians," in *Eur. Conf. Comput. Vis.* Springer, pp. 19–37.

[62] J. Zhang, J. Li, X. Yu, L. Huang, L. Gu, J. Zheng, and X. Bai, "Cor-gs: sparse-view 3d gaussian splatting via co-regularization," in *Eur. Conf. Comput. Vis.* Springer, pp. 335–352.

[63] C. Yang, S. Li, J. Fang, R. Liang, L. Xie, X. Zhang, W. Shen, and Q. Tian, "Gaussianobject: Just taking four images to get a high-quality 3d object with gaussian splatting," *arXiv:2402.10259*.

[64] A. Moreau, J. Song, H. Dhamo, R. Shaw, Y. Zhou, and E. Pérez-Pellitero, "Human gaussian splatting: Real-time rendering of animatable avatars," *arXiv:2311.17113*.

[65] S. Zheng, B. Zhou, R. Shao, B. Liu, S. Zhang, L. Nie, and Y. Liu, "GPS-Gaussian: Generalizable Pixel-wise 3D Gaussian Splatting for Real-time Human Novel View Synthesis."

[66] M. Kocabas, J.-H. R. Chang, J. Gabriel, O. Tuzel, and A. Ranjan, "Hugs: Human gaussian splats," *arXiv:2311.17910*.

[67] L. Hu, H. Zhang, Y. Zhang, B. Zhou, B. Liu, S. Zhang, and L. Nie, "Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians," *arXiv:2312.02134*.

[68] S. Hu and Z. Liu, "Gauhuman: Articulated gaussian splatting from monocular human videos," *arXiv:2312.02973*.

[69] Z. Qian, S. Wang, M. Mihajlovic, A. Geiger, and S. Tang, "3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting," *arXiv:2312.09228*.

[70] H. Pang, H. Zhu, A. Kortylewski, C. Theobalt, and M. Habermann, "Ash: Animatable gaussian splats for efficient and photoreal human rendering," *arXiv:2312.05941*.

[71] Z. Li, Z. Zheng, L. Wang, and Y. Liu, "Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling," *arXiv:2311.16096*.

[72] Z. Sheng, F. Liu, M. Liu, F. Zheng, and L. Nie, "Open-set synthesis for free-viewpoint human body reenactment of novel poses," *IEEE Trans. Circuits Syst. Video Technol.*

[73] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, oct.

[74] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 10 975–10 985.

[75] Y. Chen, L. Wang, Q. Li, H. Xiao, S. Zhang, H. Yao, and Y. Liu, "Monogaussianavatar: Monocular gaussian point-based head avatar," in *ACM SIGGRAPH Conf. Papers*, pp. 1–9.

[76] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner, "Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians," *arXiv:2312.02069*.

[77] Z. Zhao, Z. Bao, Q. Li, G. Qiu, and K. Liu, "Psavatar: A point-based morphable shape model for real-time head avatar creation with 3d gaussian splatting," *arXiv:2401.12900*.

[78] H. Dhamo, Y. Nie, A. Moreau, J. Song, R. Shaw, Y. Zhou, and E. Pérez-Pellitero, "Headgas: Real-time animatable head avatars via 3d gaussian splatting," in *Eur. Conf. Comput. Vis.* Springer, pp. 459–476.

[79] K. Teotia, H. Kim, P. Garrido, M. Habermann, M. Elgharib, and C. Theobalt, "Gaussianheads: End-to-end learning of drivable gaussian head avatars from coarse-to-fine representations," *ACM Trans. Graph.*, vol. 43, no. 6, pp. 1–12.

[80] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 194:1–194:17. [Online]. Available: https://doi.org/10.1145/3130800.3130813

[81] Y. Xu, B. Chen, Z. Li, H. Zhang, L. Wang, Z. Zheng, and Y. Liu, "Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians," *arXiv:2312.03029*.

[82] ——, "Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 1931–1941.

[83] Y. Xu, L. Wang, Z. Zheng, Z. Su, and Y. Liu, "3d gaussian parametric head model," in *Eur. Conf. Comput. Vis.* Springer, pp. 129–147.

[84] S. Ma, Y. Weng, T. Shao, and K. Zhou, "3d gaussian blendshapes for head avatar animation," in *ACM SIGGRAPH Conf. Papers*, pp. 1–10.

[85] J. Xiang, X. Gao, Y. Guo, and J. Zhang, "Flashavatar: High-fidelity head avatar with efficient gaussian embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 1802–1812.

[86] J. Li, J. Zhang, X. Bai, J. Zheng, X. Ning, J. Zhou, and L. Gu, "Talkinggaussian: Structure-persistent 3d talking head synthesis via gaussian splatting," in *Eur. Conf. Comput. Vis.* Springer, pp. 127–145.

[87] K. Cho, J. Lee, H. Yoon, Y. Hong, J. Ko, S. Ahn, and S. Kim, "Gaussiantalker: Real-time talking head synthesis with 3d gaussian splatting," in *Proc. ACM Int. Conf. Multimedia*, pp. 10 985–10 994.

[88] H. Yu, Z. Qu, Q. Yu, J. Chen, Z. Jiang, Z. Chen, S. Zhang, J. Xu, F. Wu, C. Lv *et al.*, "Gaussiantalker: Speaker-specific talking head synthesis via 3d gaussian splatting," in *Proc. ACM Int. Conf. Multimedia*, pp. 3548–3557.

[89] H. Luo, M. Ouyang, Z. Zhao, S. Jiang, L. Zhang, Q. Zhang, W. Yang, L. Xu, and J. Yu, "Gaussianhair: Hair modeling and rendering with light-aware gaussians," *arXiv:2402.10483*.

[90] L. Bolanos, S.-Y. Su, and H. Rhodin, "Gaussian Shadow Casting for Neural Characters."

[91] R. Abdal, W. Yifan, Z. Shi, Y. Xu, R. Po, Z. Kuang, Q. Chen, D.-Y. Yeung, and G. Wetzstein, "Gaussian shell maps for efficient 3d human generation," *arXiv:2311.17857*.

[92] X. Liu, X. Zhan, J. Tang, Y. Shan, G. Zeng, D. Lin, X. Liu, and Z. Liu, "Humangaussian: Text-driven 3d human generation with gaussian splatting," *arXiv:2311.17061*.

[93] T. Kirschstein, S. Giebenhain, J. Tang, M. Georgopoulos, and M. Nießner, "Gghead: Fast and generalizable 3d gaussian heads," in *ACM SIGGRAPH Asia Conf. Papers*, pp. 1–11.

[94] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv:2209.14988*.

[95] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," *arXiv:2309.16653*.

[96] Z. Chen, F. Wang, and H. Liu, "Text-to-3d using gaussian splatting," *arXiv:2309.16585*.

[97] T. Yi, J. Fang, J. Wang, G. Wu, L. Xie, X. Zhang, W. Liu, Q. Tian, and X. Wang, "Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models," *arXiv preprint arXiv*, vol. 2310.

[98] Q. Shen, X. Yang, M. B. Mi, and X. Wang, "Vista3d: Unravel the 3d darkside of a single image," in *Eur. Conf. Comput. Vis.* Springer, pp. 405–421.

[99] X. Li, H. Wang, and K.-K. Tseng, "Gaussiandiffusion: 3d gaussian splatting for denoising diffusion probabilistic models with structured noise," *arXiv:2311.11221*.

[100] Y. Liang, X. Yang, J. Lin, H. Li, X. Xu, and Y. Chen, "Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching," *arXiv:2311.11284*.

[101] D. Di, J. Yang, C. Luo, Z. Xue, W. Chen, X. Yang, and Y. Gao, "Hyper-3dg: Text-to-3d gaussian generation via hypergraph," *arXiv:2403.09236*.

[102] X. Yang, Y. Chen, C. Chen, C. Zhang, Y. Xu, X. Yang, F. Liu, and G. Lin, "Learn to optimize denoising scores for 3d generation: A unified and improved diffusion prior on nerf and 3d gaussian splatting," *arXiv:2312.04820*.

[103] W. Zhuo, F. Ma, H. Fan, and Y. Yang, "Vividdreamer: Invariant score distillation for hyper-realistic text-to-3d generation," in *Eur. Conf. Comput. Vis.* Springer, pp. 122–139.

[104] L. Melas-Kyriazi, I. Laina, C. Rupprecht, N. Neverova, A. Vedaldi, O. Gafni, and F. Kokkinos, "Im-3d: Iterative multiview diffusion and reconstruction for high-quality 3d generation," *arXiv:2402.08682*.

[105] V. Voleti, C.-H. Yao, M. Boss, A. Letts, D. Pankratz, D. Tochilkin, C. Laforte, R. Rombach, and V. Jampani, "Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion," in *Eur. Conf. Comput. Vis.* Springer, pp. 439–457.

[106] R. Gao, A. Holynski, P. Henzler, A. Brussee, R. Martin-Brualla, P. Srinivasan, J. T. Barron, and B. Poole, "Cat3d: Create anything in 3d with multi-view diffusion models," *arXiv:2405.10314*.

[107] J. Han, F. Kokkinos, and P. Torr, "Vfusion3d: Learning scalable 3d generative models from video diffusion models," in *Eur. Conf. Comput. Vis.* Springer, pp. 333–350.

[108] H. Yang, Y. Chen, Y. Pan, T. Yao, Z. Chen, C.-W. Ngo, and T. Mei, "Hi3d: Pursuing high-resolution image-to-3d generation with video diffusion models," in *Proc. ACM Int. Conf. Multimedia*, pp. 6870–6879.

[109] L. Jiang and L. Wang, "Brightdreamer: Generic 3d gaussian generative framework for fast text-to-3d synthesis," *arXiv:2403.11273*.

[110] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," *arXiv:2402.05054*.

[111] B. Zhang, Y. Cheng, J. Yang, C. Wang, F. Zhao, Y. Tang, D. Chen, and B. Guo, "Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling," *arXiv:2403.19655*.

[112] F.-L. Liu, H. Fu, Y.-K. Lai, and L. Gao, "Sketchdream: Sketch-based text-to-3d generation and editing," *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–13.

[113] X. He, J. Chen, S. Peng, D. Huang, Y. Li, X. Huang, C. Yuan, W. Ouyang, and T. He, "Gvgen: Text-to-3d generation with volumetric representation," *arXiv:2403.12957*.

[114] L. Lu, H. Gao, T. Dai, Y. Zha, Z. Hou, J. Wu, and S.-T. Xia, "Large point-to-gaussian model for image-to-3d generation," in *Proc. ACM Int. Conf. Multimedia*, pp. 10 843–10 852.

[115] B. Roessle, N. Müller, L. Porzi, S. Rota Bulò, P. Kontschieder, A. Dai, and M. Nießner, "L3dg: Latent 3d gaussian diffusion," in *ACM SIGGRAPH Asia Conf. Papers*, pp. 1–11.

[116] H. Yu, W. Gong, J. Chen, and H. Ma, "Get3dgs: Generate 3d gaussians based on points deformation fields," *IEEE Trans. Circuits Syst. Video Technol.*

[117] Y. Yuan, X. Li, Y. Huang, S. De Mello, K. Nagano, J. Kautz, and U. Iqbal, "Gavatar: Animatable 3d gaussian avatars with implicit mesh learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 896–905.

[118] Q. Feng, Z. Xing, Z. Wu, and Y.-G. Jiang, "Fdgaussian: Fast gaussian splatting from single image via geometric-aware diffusion model," *arXiv:2403.10242*.

[119] J. Zhang, Z. Tang, Y. Pang, X. Cheng, P. Jin, Y. Wei, W. Yu, M. Ning, and L. Yuan, "Repaint123: Fast and high-quality one image to 3d generation with progressive controllable 2d repainting," *arXiv:2312.13271*.

[120] Y. Chen, J. Fang, Y. Huang, T. Yi, X. Zhang, L. Xie, X. Wang, W. Dai, H. Xiong, and Q. Tian, "Cascade-zero123: One image to highly consistent 3d with self-prompted nearby views," in *Eur. Conf. Comput. Vis.* Springer, pp. 311–330.

[121] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9298–9309.

[122] A. Vilesov, P. Chari, and A. Kadambi, "Cg3d: Compositional generation for text-to-3d via gaussian splatting," *arXiv:2311.17907*.

[123] X. Zhou, X. Ran, Y. Xiong, J. He, Z. Lin, Y. Wang, D. Sun, and M.-H. Yang, "Gala3d: Towards text-to-3d complex scene generation via layout-guided generative gaussian splatting," *arXiv:2402.07207*.

[124] H. Li, H. Shi, W. Zhang, W. Wu, Y. Liao, L. Wang, L.-h. Lee, and P. Y. Zhou, "Dreamscene: 3d gaussian-based text-to-3d scene generation via formation pattern sampling," in *Eur. Conf. Comput. Vis.* Springer, pp. 214–230.

[125] Y. Chen, T. Wang, T. Wu, X. Pan, K. Jia, and Z. Liu, "Comboverse: Compositional 3d assets creation using spatially-aware diffusion guidance," in *Eur. Conf. Comput. Vis.* Springer, pp. 128–146.

[126] J. Chung, S. Lee, H. Nam, J. Lee, and K. M. Lee, "Luciddreamer: Domain-free generation of 3d gaussian splatting scenes," *arXiv:2311.13384*.

[127] H. Ouyang, K. Heal, S. Lombardi, and T. Sun, "Text2immersion: Generative immersive scene with 3d gaussians," *arXiv:2312.09242*.

[128] S. Zhou, Z. Fan, D. Xu, H. Chang, P. Chari, T. Bharadwaj, S. You, Z. Wang, and A. Kadambi, "Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting," in *Eur. Conf. Comput. Vis.* Springer, pp. 324–342.

[129] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic

models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 11 461–11 471.

[130] H. Ling, S. W. Kim, A. Torralba, S. Fidler, and K. Kreis, "Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models," *arXiv:2312.13763*.

[131] J. Ren, L. Pan, J. Tang, C. Zhang, A. Cao, G. Zeng, and Z. Liu, "Dreamgaussian4d: Generative 4d gaussian splatting," *arXiv:2312.17142*.

[132] Q. Gao, Q. Xu, Z. Cao, B. Mildenhall, W. Ma, L. Chen, D. Tang, and U. Neumann, "Gaussianflow: Splatting gaussian dynamics for 4d content creation," *arXiv:2403.12365*.

[133] S. Bahmani, I. Skorokhodov, V. Rong, G. Wetzstein, L. Guibas, P. Wonka, S. Tulyakov, J. J. Park, A. Tagliasacchi, and D. B. Lindell, "4d-fy: Text-to-4d generation using hybrid score distillation sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 7996–8006.

[134] S. Bahmani, X. Liu, W. Yifan, I. Skorokhodov, V. Rong, Z. Liu, X. Liu, J. J. Park, S. Tulyakov, G. Wetzstein *et al.*, "Tc4d: Trajectory-conditioned text-to-4d generation," in *Eur. Conf. Comput. Vis.* Springer, pp. 53–72.

[135] Y. Zheng, X. Li, K. Nagano, S. Liu, O. Hilliges, and S. De Mello, "A unified approach for text-and image-guided 4d scene generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 7300–7309.

[136] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang, "Mvdream: Multi-view diffusion for 3d generation," *arXiv:2308.16512*.

[137] Y. Yin, D. Xu, Z. Wang, Y. Zhao, and Y. Wei, "4dgen: Grounded 4d content generation with spatial-temporal consistency," *arXiv:2312.17225*.

[138] Z. Pan, Z. Yang, X. Zhu, and L. Zhang, "Fast dynamic 3d object generation from a single-view video," *arXiv:2401.08742*.

[139] Q. Sun, Z. Guo, Z. Wan, J. N. Yan, S. Yin, W. Zhou, J. Liao, and H. Li, "Eg4d: Explicit generation of 4d object without score distillation," *arXiv:2405.18132*.

[140] Y. Zeng, Y. Jiang, S. Zhu, Y. Lu, Y. Lin, H. Zhu, W. Hu, X. Cao, and Y. Yao, "Stag4d: Spatial-temporal anchored generative 4d gaussians," in *Eur. Conf. Comput. Vis.* Springer, pp. 163–179.

[141] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 130–141.

[142] T. Zhang, Q. Gao, W. Li, L. Liu, and B. Chen, "Bags: Building animatable gaussian splatting from a monocular video with diffusion priors," *arXiv:2403.11427*.

[143] Z. Wu, C. Yu, Y. Jiang, C. Cao, F. Wang, and X. Bai, "Sc4d: Sparse-controlled video-to-4d generation and motion transfer," in *Eur. Conf. Comput. Vis.* Springer, pp. 361–379.

[144] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," *arXiv:2312.07920*.

[145] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians for modeling dynamic urban scenes," *arXiv:2401.01339*.

[146] H. Zhou, J. Shao, L. Xu, D. Bai, W. Qiu, B. Liu, Y. Wang, A. Geiger, and Y. Liao, "Hugs: Holistic urban 3d scene understanding via gaussian splatting," *arXiv:2403.12722*.

[147] C. Zhao, S. Sun, R. Wang, Y. Guo, J.-J. Wan, Z. Huang, X. Huang, Y. V. Chen, and L. Ren, "Tclc-gs: Tightly coupled lidar-camera gaussian splatting for surrounding autonomous driving scenes," *arXiv:2404.02410*.

[148] Q. Herau, M. Bennehar, A. Moreau, N. Piasco, L. Roldao, D. Tsishkou, C. Migniot, P. Vasseur, and C. Demonceaux, "3dgs-calib: 3d gaussian splatting for multimodal spatiotemporal calibration," *arXiv:2403.11577*.

[149] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," *arXiv:2311.11700*.

[150] H. Huang, L. Li, H. Cheng, and S.-K. Yeung, "Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras," *arXiv:2311.16728*.

[151] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track & map 3d gaussians for dense rgb-d slam," *arXiv:2312.02126*.

[152] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-slam: Photo-realistic dense slam with gaussian splatting," *arXiv:2312.10070*.

[153] J. Hu, X. Chen, B. Feng, G. Li, L. Yang, H. Bao, G. Zhang, and Z. Cui, "Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field," in *Eur. Conf. Comput. Vis.* Springer, pp. 93–112.

[154] S. Ha, J. Yeon, and H. Yu, "Rgbd gs-icp slam," *arXiv:2403.12550*.

[155] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. Int. Conf. Comput. Vis. (ICCV)*. IEEE, pp. 2564–2571.

[156] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Proc. Dundee Biennial Conf. Numer. Anal.* Springer, pp. 105–116.

[157] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: Sci. Syst. (RSS)*, vol. 2, no. 4. Seattle, WA, p. 435.

[158] M. Li, S. Liu, and H. Zhou, "Sgs-slam: Semantic gaussian splatting for neural dense slam," *arXiv:2402.03246*.

[159] S. Zhu, R. Qin, G. Wang, J. Liu, and H. Wang, "Semgauss-slam: Dense semantic gaussian splatting slam," *arXiv:2403.07494*.

[160] Y. Ji, Y. Liu, G. Xie, B. Ma, and Z. Xie, "Neds-slam: A novel neural explicit dense semantic slam framework using 3d gaussian splatting," *arXiv:2403.11679*.

[161] P. Jiang, G. Pandey, and S. Saripalli, "3dgs-reloc: 3d gaussian splatting for map representation and visual relocalization," *arXiv:2403.11367*.

[162] X. Lei, M. Wang, W. Zhou, and H. Li, "Gaussnav: Gaussian splatting for visual navigation," *arXiv:2403.11625*.

[163] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv:1906.05797*.

[164] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 19 595–19 604.

[165] X. Guo, W. Zhang, R. Liu, P. Han, and H. Chen, "Motiongs: Compact gaussian splatting slam by motion filter," *arXiv:2405.11129*.

[166] L. Zhu, Y. Li, E. Sandström, S. Huang, K. Schindler, and I. Armeni, "Loopsplat: Loop closure by registering 3d gaussian splats," *arXiv:2408.10154*.

[167] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," *arXiv:2308.09713*.

[168] J. Sun, H. Jiao, G. Li, Z. Zhang, L. Zhao, and W. Xing, "3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos," *arXiv:2403.01444*.

[169] R. Shaw, J. Song, A. Moreau, M. Nazarczuk, S. Catley-Chandar, H. Dhamo, and E. Perez-Pellitero, "Swags: Sampling windows adaptively for dynamic 3d gaussian splatting," *arXiv:2312.13308*.

[170] Y. Xiao, X. Wang, J. Li, H. Cai, Y. Fan, N. Xue, M. Yang, Y. Shen, and S. Gao, "Bridging 3d gaussian and mesh for freeview video rendering," *arXiv:2403.11453*.

[171] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," *arXiv:2309.13101*.

[172] Y. Liang, N. Khan, Z. Li, T. Nguyen-Phuoc, D. Lanman, J. Tompkin, and L. Xiao, "Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis," *arXiv:2312.11458*.

[173] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," *arXiv:2310.08528*.

[174] B. P. Duisterhof, Z. Mandi, Y. Yao, J.-W. Liu, M. Z. Shou, S. Song, and J. Ichnowski, "Md-splatting: Learning metric deformation from 4d gaussians in highly deformable scenes," *arXiv:2312.00583*.

[175] Z. Guo, W. Zhou, L. Li, M. Wang, and H. Li, "Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction," *IEEE Trans. Circuits Syst. Video Technol.*

[176] D. Li, S.-S. Huang, Z. Lu, X. Duan, and H. Huang, "St-4dgs: Spatial-temporally consistent 4d gaussian splatting for efficient dynamic scene rendering," in *ACM SIGGRAPH Conf. Papers*, pp. 1–11.

[177] Z. Lu, X. Guo, L. Hui, T. Chen, M. Yang, X. Tang, F. Zhu, and Y. Dai, "3d geometry-aware deformable gaussian splatting for dynamic view synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 8900–8910.

[178] K. Katsumata, D. M. Vo, and H. Nakayama, "An efficient 3d gaussian representation for monocular/multi-view dynamic scenes," *arXiv:2311.12897*.

[179] A. Kratimenos, J. Lei, and K. Daniilidis, "Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting," *arXiv:2312.00112*.

[180] Y. Lin, Z. Dai, S. Zhu, and Y. Yao, "Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particles," *arXiv:2312.03431*.

[181] Y.-H. Huang, Y.-T. Sun, Z. Yang, X. Lyu, Y.-P. Cao, and X. Qi, "Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes," *arXiv:2312.14937*.

[182] Z. Li, Z. Chen, Z. Li, and Y. Xu, "Spacetime gaussian feature splatting for real-time dynamic view synthesis," *arXiv:2312.16812*.

[183] Z. Yang, H. Yang, Z. Pan, X. Zhu, and L. Zhang, "Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting," *arXiv:2310.10642*.

[184] Y. Duan, F. Wei, Q. Dai, Y. He, W. Chen, and B. Chen, "4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes," in *ACM SIGGRAPH Conf. Papers*, pp. 1–11.

[185] A. Guédon and V. Lepetit, "Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering," *arXiv:2311.12775*.

[186] H. Chen, C. Li, and G. H. Lee, "Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance," *arXiv:2312.00846*.

[187] X. Lyu, Y.-T. Sun, Y.-H. Huang, X. Wu, Z. Yang, Y. Chen, J. Pang, and X. Qi, "3dgsr: Implicit surface reconstruction with 3d gaussian splatting," *arXiv:2404.00409*.

[188] M. Yu, T. Lu, L. Xu, L. Jiang, Y. Xiangli, and B. Dai, "Gsdf: 3dgs meets sdf for improved rendering and reconstruction," *arXiv:2403.16964*.

[189] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv:2106.10689*.

[190] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 10 318–10 327.

[191] H. Yu, J. Julin, Z. Á. Milacski, K. Niinuma, and L. A. Jeni, "Cogs: Controllable gaussian splatting," *arXiv:2312.05664*.

[192] P. Dai, J. Xu, W. Xie, X. Liu, H. Wang, and W. Xu, "High-quality surface reconstruction using gaussian surfels," *arXiv:2404.17774*.

[193] Z. Yu, T. Sattler, and A. Geiger, "Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes," *arXiv:2404.10772*.

[194] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2d gaussian splatting for geometrically accurate radiance fields," *arXiv:2403.17888*.

[195] C. Reiser, S. Garbin, P. Srinivasan, D. Verbin, R. Szeliski, B. Mildenhall, J. Barron, P. Hedman, and A. Geiger, "Binary opacity grids: Capturing fine geometric detail for mesh-based view synthesis," *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–14.

[196] Y. Chen, Z. Chen, C. Zhang, F. Wang, X. Yang, Y. Wang, Z. Cai, L. Yang, H. Liu, and G. Lin, "Gaussianeditor: Swift and controllable 3d editing with gaussian splatting," *arXiv:2311.14521*.

[197] F. Palandra, A. Sanchietti, D. Baieri, and E. Rodolà, "Gsedit: Efficient text-guided editing of 3d objects via gaussian splatting," *arXiv:2403.05154*.

[198] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 18 392–18 402.

[199] J. Fang, J. Wang, X. Zhang, L. Xie, and Q. Tian, "Gaussianeditor: Editing 3d gaussians delicately with text instructions," *arXiv:2311.16037*.

[200] J. Wu, J.-W. Bian, X. Li, G. Wang, I. Reid, P. Torr, and V. A. Prisacariu, "Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing," *arXiv:2403.08733*.

[201] Y. Wang, X. Yi, Z. Wu, N. Zhao, L. Chen, and H. Zhang, "View-consistent 3d editing with gaussian splatting," *arXiv:2403.11868*.

[202] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 3836–3847.

[203] R. Shao, J. Sun, C. Peng, Z. Zheng, B. Zhou, H. Zhang, and Y. Liu, "Control4d: Dynamic portrait editing by learning 4d gan from 2d diffusion-based editor," *arXiv:2305.20082*, vol. 2, no. 6, p. 16.

[204] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144.

[205] J. Zhuang, D. Kang, Y.-P. Cao, G. Li, L. Lin, and Y. Shan, "Tip-editor: An accurate 3d editor following both text-prompts and image-prompts," *arXiv:2401.14828*.

[206] J. Huang and H. Yu, "Point'n move: Interactive scene object manipulation on gaussian splatting radiance fields," *arXiv:2311.16737*.

[207] Q.-Y. Feng, G.-C. Cao, H.-X. Chen, Q.-C. Xu, T.-J. Mu, R. Martin, and S.-M. Hu, "Evsplitting: An efficient and visually consistent splitting algorithm for 3d gaussian splatting," in *SIGGRAPH Asia 2024 Conf. Papers*, pp. 1–11.

[208] A. Saroha, M. Gladkova, C. Curreli, T. Yenamandra, and D. Cremers, "Gaussian splatting in style," *arXiv:2403.08498*.

[209] Y.-H. Huang, Y. He, Y.-J. Yuan, Y.-K. Lai, and L. Gao, "Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 18 342–18 352.

[210] J. Gao, C. Gu, Y. Lin, H. Zhu, X. Cao, L. Zhang, and Y. Yao, "Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing," *arXiv:2311.16043*.

[211] Z. Liang, Q. Zhang, Y. Feng, Y. Shan, and K. Jia, "Gs-ir: 3d gaussian splatting for inverse rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 21 644–21 653.

[212] Z. Kuang, Y. Yang, S. Dong, J. Ma, H. Fu, and Y. Zheng, "Olat gaussians for generic relightable appearance acquisition," in *ACM SIGGRAPH Asia Conf. Papers*, pp. 1–11.

[213] Z. Bi, Y. Zeng, C. Zeng, F. Pei, X. Feng, K. Zhou, and H. Wu, "Gs3: Efficient relighting with triple gaussian splatting," in *ACM SIGGRAPH Asia Conf. Papers*, pp. 1–12.

[214] M. Ye, M. Danelljan, F. Yu, and L. Ke, "Gaussian grouping: Segment and edit anything in 3d scenes," *arXiv:2312.00732*.

[215] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi, "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields," *arXiv:2312.03203*.

[216] K. Lan, H. Li, H. Shi, W. Wu, Y. Liao, L. Wang, and P. Zhou, "2d-guided 3d gaussian segmentation," *arXiv:2312.16047*.

[217] B. Dou, T. Zhang, Y. Ma, Z. Wang, and Z. Yuan, "Cosseggaussians: Compact and swift scene segmenting 3d gaussians," *arXiv:2401.05925*.

[218] Q. Gu, Z. Lv, D. Frost, S. Green, J. Straub, and C. Sweeney, "Egolifter: Open-world 3d segmentation for egocentric perception," in *Eur. Conf. Comput. Vis.* Springer, pp. 382–400.

[219] S. Choi, H. Song, J. Kim, T. Kim, and H. Do, "Click-gaussian: Interactive segmentation to any 3d gaussians," in *Eur. Conf. Comput. Vis.* Springer, pp. 289–305.

[220] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, "Language embedded 3d gaussians for open-vocabulary scene understanding," *arXiv:2311.18482*.

[221] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," *arXiv:2312.16084*.

[222] X. Zuo, P. Samangouei, Y. Zhou, Y. Di, and M. Li, "Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding," *arXiv:2401.01970*.

[223] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.* PMLR, pp. 8748–8763.

[224] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9650–9660.

[225] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Adv. Neural Inf. Process. Syst.*, vol. 30.

[226] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–15.

[227] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang, "Physgaussian: Physics-integrated 3d gaussians for generative dynamics," *arXiv:2311.12198*.

[228] A. De Vaucorbeil, V. P. Nguyen, S. Sinaie, and J. Y. Wu, "Material point method after 25 years: Theory, implementation, and applications," *Adv. Appl. Mech.*, vol. 53, pp. 185–398.

[229] T. Zhang, H.-X. Yu, R. Wu, B. Y. Feng, C. Zheng, N. Snavely, J. Wu, and W. T. Freeman, "Physdreamer: Physics-based interaction with 3d objects via video generation," in *Eur. Conf. Comput. Vis.* Springer, pp. 388–406.

[230] Y. Jiang, C. Yu, T. Xie, X. Li, Y. Feng, H. Wang, M. Li, H. Lau, F. Gao, Y. Yang *et al.*, "Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality," in *ACM SIGGRAPH 2024 Conf. Papers*, pp. 1–1.

[231] J. Jung, J. Han, H. An, J. Kang, S. Park, and S. Kim, "Relaxing accurate initialization constraint for 3d gaussian splatting," *arXiv:2403.09413*.

[232] D. Das, C. Wewer, R. Yunus, E. Ilg, and J. E. Lenssen, "Neural parametric gaussians for monocular non-rigid object reconstruction," *arXiv:2312.01196*.

[233] L. Gao, J. Yang, B.-T. Zhang, J.-M. Sun, Y.-J. Yuan, H. Fu, and Y.-K. Lai, "Mesh-based gaussian splatting for real-time large-scale deformation," *arXiv:2402.04796*.

[234] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting slam," *arXiv:2312.06741*.

[235] Y. Yue, A. Das, F. Engelmann, S. Tang, and J. E. Lenssen, "Improving 2d feature representations by 3d-aware fine-tuning," in *Eur. Conf. Comput. Vis.* Springer, pp. 57–74.

[236] J. Lei, Y. Wang, G. Pavlakos, L. Liu, and K. Daniilidis, "Gart: Gaussian articulated template models," *arXiv:2311.16099*.

[237] L. Huang, J. Bai, J. Guo, and Y. Guo, "Gs++: Error analyzing and optimal gaussian splatting," *arXiv:2402.00752*.

[238] N. Moenne-Loccoz, A. Mirzaei, O. Perel, R. de Lutio, J. Martinez Esturo, G. State, S. Fidler, N. Sharp, and Z. Gojcic, "3d gaussian ray tracing: Fast tracing of particle scenes," *ACM Trans. Graph.*, vol. 43, no. 6, pp. 1–19.

[239] Y. Li, C. Lyu, Y. Di, G. Zhai, G. H. Lee, and F. Tombari, "Geogaussian: Geometry-aware gaussian splatting for scene rendering," *arXiv:2403.11324*.

[240] H. Xiong, S. Muttukuru, R. Upadhyay, P. Chari, and A. Kadambi, "Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting," *arXiv:2312.00206*.

[241] L. Bolanos, S.-Y. Su, and H. Rhodin, "Gaussian shadow casting for neural characters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 20 997–21 006.

[242] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv:2212.08751*.

[243] Y. Liu, C. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang, "Syncdreamer: Generating multiview-consistent images from a single-view image," *arXiv:2309.03453*.

[244] Z. Li, Y. Chen, L. Zhao, and P. Liu, "Controllable text-to-3d generation via surface-aligned gaussian splatting," *arXiv:2403.09981*.

[245] M. Armandpour, A. Sadeghian, H. Zheng, A. Sadeghian, and M. Zhou, "Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond," *arXiv:2304.04968*.

[246] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv:2106.09685*.

[247] S. Sun, M. Mielle, A. J. Lilienthal, and M. Magnusson, "High-fidelity slam using gaussian splatting with rendering-guided densification and regularized optimization," *arXiv:2403.12535*.

[248] Y. Liu, C. Luo, L. Fan, N. Wang, J. Peng, and Z. Zhang, "Citygaussian: Real-time high-quality large-scale scene rendering with gaussians," in *Eur. Conf. Comput. Vis.* Springer, pp. 265–282.

[249] B. Kerbl, A. Meuleman, G. Kopanas, M. Wimmer, A. Lanvin, and G. Drettakis, "A hierarchical 3d gaussian representation for real-time rendering of very large datasets," *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–15.

[250] Z. Zhang, W. Hu, Y. Lao, T. He, and H. Zhao, "Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting," *arXiv:2403.15530*.

[251] Q. Feng, G. Cao, H. Chen, T.-J. Mu, R. R. Martin, and S.-M. Hu, "A new split algorithm for 3d gaussian splatting," *arXiv:2403.09143*.

[252] M. Niemeyer, F. Manhardt, M.-J. Rakotosaona, M. Oechsle, D. Duckworth, R. Gosula, K. Tateno, J. Bates, D. Kaeser, and F. Tombari, "Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps," *arXiv:2403.13806*.

[253] G. Fang and B. Wang, "Mini-splatting: Representing scenes with a constrained number of gaussians," *arXiv:2403.14166*.

[254] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. Eurographics Symp. Geom. Process. (SGP)*, vol. 7, no. 4.

[255] J. Tang, H. Zhou, X. Chen, T. Hu, E. Ding, J. Wang, and G. Zeng, "Delicate textured mesh recovery from nerf via adaptive surface refinement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 17 739–17 749.

[256] H. Li, Y. Gao, D. Zhang, C. Wu, Y. Dai, C. Zhao, H. Feng, E. Ding, J. Wang, and J. Han, "Ggrt: Towards generalizable 3d gaussians without pose priors in real-time," *arXiv:2403.10147*.

[257] D. Malarz, W. Smolak, J. Tabor, S. Tadeja, and P. Spurek, "Gaussian splatting with nerf-based color and opacity."

[258] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black, "3d menagerie: Modeling the 3d shape and pose of animals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 6365–6373.

[259] S. D. Porumbescu, B. Budge, L. Feng, and K. I. Joy, "Shell maps," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 626–633.

[260] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," in *ACM SIGGRAPH 2007 Papers*, pp. 80–es.

[261] Z. Fan, W. Cong, K. Wen, K. Wang, J. Zhang, X. Ding, D. Xu, B. Ivanovic, M. Pavone, G. Pavlakos *et al.*, "Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds," *IEEE Trans. Circuits Syst. Video Technol.*

[262] Y. Sun, X. Wang, Y. Zhang, J. Zhang, C. Jiang, Y. Guo, and F. Wang, "icomma: Inverting 3d gaussians splatting for camera pose estimation via comparing and matching," *arXiv:2312.09031*.

[263] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 7210–7219.

[264] X. Chen, Q. Zhang, X. Li, Y. Chen, Y. Feng, X. Wang, and J. Wang, "Hallucinated neural radiance fields in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 12 943–12 952.

[265] Y. Yang, S. Zhang, Z. Huang, Y. Zhang, and M. Tan, "Cross-ray neural radiance fields for novel-view synthesis from unconstrained image collections," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 15 901–15 911.

[266] H. Dahmani, M. Bennehar, N. Piasco, L. Roldao, and D. Tsishkou, "Swag: Splatting in the wild images with appearance-conditioned gaussians," *arXiv:2403.10427*.

[267] D. Zhang, C. Wang, W. Wang, P. Li, M. Qin, and H. Wang, "Gaussian in the wild: 3d gaussian splatting for unconstrained image collections," *arXiv:2403.15704*.

[268] Y. Fu, S. Liu, A. Kulkarni, J. Kautz, A. A. Efros, and X. Wang, "Colmap-free 3d gaussian splatting," *arXiv:2312.07504*.

[269] D. Cai, J. Heikkilä, and E. Rahtu, "Gs-pose: Cascaded framework for generalizable segmentation-based 6d object pose estimation," *arXiv:2403.10683*.

[270] S. Saito, G. Schwartz, T. Simon, J. Li, and G. Nam, "Relightable gaussian codec avatars," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 130–141.

IX. Biography Section

Yanqi Bao is currently pursuing the Ph.D. degree at the Department of Computer Science and Technology, Nanjing University, China. His research interests include 3d vision, NeRF and 3DGS.

Tianyu Ding is currently a Senior Researcher at Microsoft, Redmond, USA. He received his PhD degree in Applied Mathematics and Statistics from Johns Hopkins University (JHU). His research interests focus on improving efficiency in machine learning and artificial intelligence, especially in areas like computer vision and generative models.

Jing Huo received the PhD degree from the Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 2017. She is currently an Associate Professor with the Department of Computer Science and Technology, Nanjing University. Her current research interests include machine learning and computer vision, with a focus on image and video generation techniques.

Yaoli Liu is currently an undergraduate student at Software Institute of Nanjing University. His research interests include generalizable 3DGS.

Yuxin Li is currently a graduate student at the Department of Computer Science and Technology, Nanjing University, China. His research interests include 3d vision and neural rendering.

Wenbin Li received his Ph.D. degree from the Department of Computer Science and Technology at Nanjing University in 2019. He is currently an Associate Researcher in the Department of Computer Science and Technology at Nanjing University, China. His research interests include machine learning and computer vision, particularly in metric learning, few-shot learning and their applications to image classification.

Yang Gao received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2000. He is currently a Professor and also the Deputy Director of the Department of Computer Science and Technology, Nanjing University, where he is also directing the Reasoning and Learning Research Group. He has published more than 100 papers in top-tier conferences and journals. His current research interests include artificial intelligence and machine learning. He also serves as the program chair and area chair for many international conferences.

Jiebo Luo (IEEE Fellow) is Professor of Computer Science at the University of Rochester which he joined in 2011 after a prolific career of fifteen years at Kodak Research Laboratories. He has authored over 500 technical papers and holds over 90 U.S. patents. His research interests include computer vision, NLP, machine learning, data mining, computational social science, and digital health. He has been involved in numerous technical conferences, including serving as program co-chair of ACM Multimedia 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, as well as general co-chair of ACM Multimedia 2018 and IEEE ICME 2024. He has served on the editorial boards of the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Multimedia (TMM), IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), IEEE Transactions on Big Data (TBD), ACM Transactions on Intelligent Systems and Technology (TIST), Pattern Recognition, Knowledge and Information Systems (KAIS), Machine Vision and Applications, and Intelligent Medicine. He is the current Editor-in-Chief of the IEEE Transactions on Multimedia. Professor Luo is also a Fellow of ACM, AAAI, SPIE, and IAPR.