SAGS: Structure-Aware 3D Gaussian Splatting

Evangelos Ververas[1,2*], Rolandos Alexandros Potamias[1,2*†], Jifei Song[2], Jiankang Deng[1,2], and Stefanos Zafeiriou[1]

[1] Imperial College London, UK
[2] Huawei Noah's Ark Lab, UK
`https://eververas.github.io/SAGS/`
* Equal contribution, † Corresponding Author

Abstract. Following the advent of NeRFs, 3D Gaussian Splatting (3D-GS) has paved the way to real-time neural rendering overcoming the computational burden of volumetric methods. Several extensions of 3D-GS have been proposed to achieve compressible and high-fidelity performance. However, by employing a geometry-agnostic optimization scheme, these methods neglect the inherent 3D structure of the scene, thereby restricting the expressivity and the quality of the representation, resulting in various floating points and artifacts. In this work, we propose a structure-aware Gaussian Splatting method (SAGS) that implicitly encodes the geometry of the scene, which reflects to state-of-the-art rendering performance and reduced storage requirements on benchmark datasets. SAGS is founded on a local-global graph representation that facilitates the learning of complex scenes and enforces meaningful point displacements that preserve the scene's geometry. Additionally, we introduce a lightweight version of SAGS, using a simple yet effective mid-point interpolation scheme, which showcases a compact representation of the scene with up to 24× size reduction without the reliance on any compression strategies. Extensive experiments across multiple benchmark datasets demonstrate the superiority of SAGS compared to state-of-the-art 3D-GS methods under both rendering quality and model size. Besides, we demonstrate that our structure-aware method can effectively mitigate floating artifacts and irregular distortions of previous methods while obtaining precise depth maps.

Keywords: Novel View Synthesis · 3D Gaussian Splatting · GNNs

1 Introduction

Novel View Synthesis (NVS) is a long-studied problem that aims to generate images of a scene from a specific point of view, using only a sparse set of images from different viewpoints with known camera parameters. Due to its diverse applications spanning from Virtual Reality (VR) [7] to content creation [4, 33], novel view synthesis has garnered significant attention. With the advent of Neural Radiance Field (NeRF) [22], an enormous amount of methods have been proposed to utilize volumetric rendering, achieving remarkable rendering results.
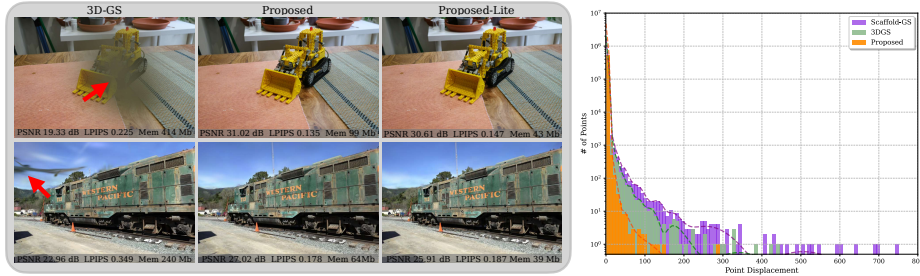
Fig. 1: Structure-Aware GS (SAGS) leverages the intrinsic structure of the scene and enforces point interaction using graph neural networks outperforming the structure agnostic optimization scheme of 3D-GS [15]. The 3D-GS method optimizes each Gaussian independently which results in 3D floaters and large point displacements from their original position (left). This can be also validated in the histogram of displacements (right) between the initial and the final position (mean) of a 3D Gaussian. Optimization-based methods neglect the scene structure and displace points far from their initial position to minimize rendering loss, in contrast to SAGS that predicts displacements that preserve the initial structure. The 3D-GS figures are taken directly from the original 3D-GS website.

However, albeit achieving highly detailed results, volumetric rendering methods fail to produce real-time renderings that hinder their real-world applications.

Recently, Kerbl *et al.* [15] introduced 3D Gaussian Splatting (3D-GS) to tackle this limitation using a set of differentiable 3D Gaussians that can achieve state-of-the-art rendering quality and real-time speed on a single GPU, outperforming previous NeRF-based methods [1–3, 22]. In 3D-GS, the scene is parametrised using a set of 3D Gaussians with learnable shape and appearance attributes, optimized using differentiable rendering. To initialize the 3D Gaussians, Kerbl *et al.* [15] relied on the point clouds derived from COLMAP [31], neglecting any additional scene structure and geometry during optimization.

Undoubtedly, one of the primary drawbacks of the 3D-GS method is the excessive number of points needed to produce high-quality scene renderings. Following the success of 3D-GS, numerous methods have been proposed to reduce the storage requirements using compression and quantization schemes [9, 17, 24] while retaining the rendering performance. However, similar to 3D-GS, each Gaussian is optimized independently to fit the ground truth views, without the use of any additional structural inductive biases to guide the optimization. As can be seen in Fig. 1 (right), such a naive setting results in points being displaced far away from their initialization, thus neglecting their initial point structure and introducing floating points and artifacts [38] (highlighted in with red arrows). Apart from a significant degradation in the rendering quality, neglecting the scene's geometry directly influences the scene's properties, including depth, which thereby limits its VR/AR applications.

In this study, we propose a structure-aware Gaussian splatting method that aims to implicitly encode the scene's geometry and learn inductive biases that

lead to point displacements that maintain the topology of the scene. Intuitively, points within the same local region often share common attributes and features, such as normals and color, that are neglected by current 3D-GS methods. Inspired by the success of Point Cloud analysis [28], we found our method on a graph constructed from the input scene and learn to model 3D Gaussians, as displacements from their original positions. The constructed graph serves as an inductive bias that aims to encode and preserve the scene structure while learning robust Gaussian attributes. Unlike the 3D-GS, the proposed method leverages the inter- and intra-connectivity between 3D point positions and learns to predict the Gaussian attributes using graph neural networks. Using both local and global structural information the network can not only reduce artifact floaters, but it can also increase the expressivity of the network leading to accurate scene representations compared to structure agnostic 3D-GS methods.

Under a series of experiments on different datasets and scenes, we evaluate the proposed method in terms of rendering quality, structure preservation, storage, and rendering performance, demonstrating the importance of structure in 3D Gaussian splatting. The proposed method can outperform the rendering quality of 3D-GS [15] while reducing the memory requirements without sacrificing rendering speed. To sum up, our contributions can be summarized as follows:

- We introduce the first structure-aware 3D Gaussian Splatting method that leverages both local and global structure of the scene.
- We bridge the two worlds between point cloud analysis and 3D Gaussian splatting, leveraging graph neural networks on the 3D point space.
- We showcase that our method can produce state-of-the-art rendering quality, while reducing the memory requirements by up to $11.7\times$ and $24\times$ with our full and lightweight models, respectively.

2   Related Work

Traditional scene reconstruction. Traditional 3D scene reconstruction techniques [30] utilize the structure-from-motion (SfM) pipeline [31] for sparse point cloud estimation and camera calibration, and further apply multi-view stereo (MVS) [11] to obtain mesh-based 3D scene reconstructions. Specifically, the traditional pipeline starts with a feature extraction step that obtains robust feature descriptors like [19] and Superpoint [8], followed by a feature matching module, *e.g.*, SuperGlue [29], that matches the 2D image descriptors. Pose estimation and bundle adjustment steps are conducted to obtain all the reconstructed parameters, according to Incremental SfM [6], Global SfM [42], or Hybrid-SfM [5]. Finally, MVS methods [14] are employed to reconstruct depth and normals of the target 3D object and subsequently fuse them to produce the final reconstruction.
NeRF based scene reconstruction. Neural radiance fields (NeRF) [22] introduced an implicit neural representation of 3D scenes, that revolutionized neural rendering based novel-view synthesis achieving remarkable photo-realistic renders. Several methods have extended NeRF model by including a set of appearance embeddings [21] and improved training strategies [32,37,41] to tackle com-

plex and large-scale scenes. MipNeRF360 [2] achieved state-of-the-art rendering quality by addressing the aliasing artifacts observed in previous methods, suffering however from exceptionally slow inference times. To improve the training and rendering efficiency of NeRFs, numerous methods have utilized grid-based structures to store compact feature representations. Interestingly, Plenoxels [10] optimized a sparse voxel grid and achieved high-quality rendering performance without resorting to MLPs. Muller *et al.* [23] highlighted the importance of positional encodings for high-fidelity neural rendering and introduced a set of hash-grid encodings that significantly improved the expressivity of the model. Despite improving the efficiency compared to the global MLP representations, grid-based methods still struggle to achieve real-time rendering performance.

3D-GS based scene reconstruction. Recently, 3D Gaussian splatting [15] has been proposed to construct anisotropic 3D Gaussians as primitives, enabling high-quality and real-time free-view rendering. Similar to NeRFs, 3D-GS attempts to overfit the scene by optimizing the Gaussian properties. However, this usually results in an enormous amount of redundant Gaussians that hinder the rendering efficiency and significantly increase the memory requirements of the model. Several methods [17,24] have attempted to reduce memory consumption by compressing the 3D Gaussians with codebooks, yet with the structure neglected, which could be vital in both synthesizing high-quality rendering and reducing the model size. Most relevant to our work, Scaffold-GS [20] introduced a structured dual-layered hierarchical scene representation to constrain the distribution of the 3D Gaussian primitives. However, Scaffold-GS still relies on structure-agnostic optimization which neglects the scene's global and local geometry, resulting in locally incoherent 3D Gaussians. This not only degrades the rendering quality but also significantly impacts the structural properties of the scene, including its depth. To tackle this, we devise a structure-aware Gaussian splatting pipeline that implicitly encodes the scene structure leveraging the inter-connectivity of the Gaussians. The proposed method introduces an inductive bias that not only facilitates high-fidelity renderings using more compact scene representations but also preserves the scene topology.

3   Method

3.1   Preliminaries: 3D Gaussian Splatting

3D Gaussian Splatting [15] is a state-of-the-art novel-view synthesis method that relies on explicit point-based representation. In contrast to the implicit representation of NeRFs that require computationally intensive volumetric rendering, 3D-GS renders images using the efficient point-based splatting rendering [36,39]. Each of the 3D Gaussians is parametrized by its position $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$, opacity $\boldsymbol{\alpha}$ and color $\boldsymbol{c}$ and has a density function defined as:

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{1}$$

where $\boldsymbol{x}$ is an arbitrary 3D point. To prevent the occurrence of non-positive semi-definite covariance matrices the authors proposed to decompose covariance
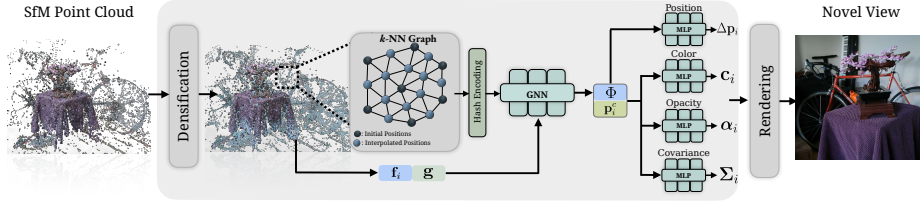
Fig. 2: Overview of the proposed method. Given a point cloud obtained from COLMAP [31], we initially apply a curvature-based densification step to populate under-represented areas. We then apply k-NN search to link points $\mathbf{p}$ within local regions and create a point set graph. Leveraging the inductive biases of graph neural networks, we learn a local-global structural feature $\Phi(\mathbf{p}_i, \mathbf{f}_i)$ for each point. Using a set of small MLPs we decode the structural features to 3D Gaussian attributes, i.e., color $\mathbf{c}$, opacity $\boldsymbol{\alpha}$, covariance $\boldsymbol{\Sigma}$ and point displacements $\Delta \mathbf{p}$ for the initial point position. Finally, we render the 3D Gaussians following the 3D-GS Gaussian rasterizer [15].

matrix using rotation and scaling $\boldsymbol{\Sigma} = \mathbf{RSS^T R^T}$ where the rotation is represented using quaternions. Finally, the 3D Gaussians are splatted on the 2D image, and their corresponding 2D pixel color is calculated by blending the N ordered Gaussians at the queried pixel as:

$$\mathbf{C} = \sum_{i=1}^{N} \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \mathbf{c}_i \tag{2}$$

where $\mathbf{c}_i$ and α_i are the color and the opacity of Gaussian i.

3.2 Structure-Aware 3D Gaussian Splatting

In this work, we propose a structure-aware 3D Gaussian Splatting method, that takes as input a sparse point cloud $\mathbf{P} \in \mathbb{R}^{M \times 3}$ from COLMAP [31] along with a set of sparse views with known camera parameters and learns a set of 3D Gaussians that fit the input views while maintaining the initial structure. The proposed method can be divided into three main components: a) the curvature-aware densification, b) the structure-aware encoder, and c) the refinement layer. Fig. 2 illustrates the pipeline of the proposed method.

Curvature-Aware Densification. Undoubtedly, the performance of 3D-GS methods is significantly impacted by the sparse initialization of the Gaussians, which relies on COLMAP. In scenarios with challenging environments featuring texture-less surfaces, conventional Structure-from-Motion (SfM) techniques frequently fall short of accurately capturing the scene details, and thus are unable to establish a solid 3D-GS initialization. To tackle such cases, we introduce a densification step that aims to populate areas with zero or few points.

In essence, 3D-GS methods attempt to reconstruct a scene from a sparse point cloud by employing a progressive growing scheme. This approach closely aligns

with the well-explored field of point cloud upsampling. Drawing inspiration from the recent Grad-PU method [12], we incorporate an upsampling step to augment the point cloud's density, particularly in regions characterized by low curvature, which are typically under-represented in the initial COLMAP point cloud. We estimate the Gaussian mean curvature of the point cloud $\mathbf{P}$ following the local-PCA approach [25, 26]. To generate a set of midpoints $\mathbf{p}_m$, we define a k-nearest neighbour graph for each of the low curvature points $\mathbf{p} \in \mathbf{P}_L \subset \mathbf{P}$ and calculate its midpoints as:

$$\mathbf{p}_m = \frac{1}{2}(\mathbf{p} + \mathbf{p}_j), \quad j \in k\text{-NN}(\mathbf{p}) \tag{3}$$

where $\mathbf{P}_L$ is the set of points $\mathbf{p}$ with curvature lower than a threshold and $\mathbf{p}_j$ is a neighbour of $\mathbf{p}$. An illustration of our densification approach is shown in Fig. 3. Leveraging the mid-point densification step, we can train a lightweight
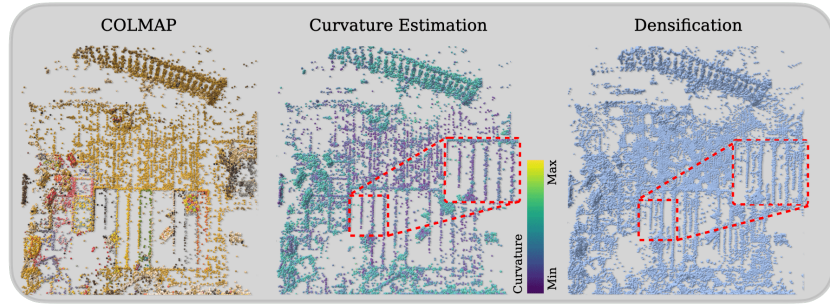

Fig. 3: Overview of the densification. Given an initial SfM [31] point cloud (left) we estimate the curvature following [25]. Curvature values are presented color-coded on the input COLMAP point cloud (middle) where colors with minimum curvature are closer to the purple color. The curvature-aware densification results in more points populating the low-curvature areas (right).

model solely on the initial point set $\mathbf{P}$, while the remaining points, along with their attributes, can be defined on-the-fly. This approach allows us to achieve both good performance and an extremely compact size, without the need for any compression scheme. We will refer to this model as **SAGS-Lite**.

Structure-Aware Encoder. Points that belong to adjacent regions will share meaningful structural features [27] that could improve scene understanding and reduce the floating artifacts. To enable point interactions within local regions and learn structural-aware features, we founded our method on a graph neural network encoder that aggregates local and global information within the scene.

In particular, the first step of the proposed structure-aware network involves creating a k-Nearest Neighbour (NN) graph that links points within a local region. Using such k-NN graph we can enable point interaction and aggregate local features using graph neural networks. To further enhance our encoder with global structural information of the scene, we included a global feature that is

shared across the points. In particular, the structure-aware GNN learns a feature encoding $\mathbf{f}_i$ for each point $\mathbf{p}_i \in \mathbf{P}$, using the following aggregation function:

$$\Phi(\mathbf{p}_i, \mathbf{f}_i) = \phi \left(\sum_{j \in \mathcal{N}(i)} w_{ij} h_{\Theta}(\gamma(\mathbf{p}_j), \mathbf{f}_j - \mathbf{f}_i, \mathbf{g}) \right) \tag{4}$$

where $\gamma(\cdot)$ denotes a positional encoding function that maps point $\mathbf{p}$ to a higher dimensional space D, $\mathbf{g}$ represents a global feature of the scene calculated as the maximum of the feature encoding $\max(\mathbf{f})$, h_{Θ} is a learnable MLP, ϕ represents a non-linear activation function and w_{ij} is an inverse-distance weight defined from the softmax of the inverse distances between the point $\mathbf{p}_i$ and its neighbors $\mathcal{N}(i)$:

$$w_{ij} = \frac{\mathrm{dist}^{-1}(\mathbf{p}_i, \mathbf{p}_j)}{\sum_{j \in \mathcal{N}(i)} \mathrm{dist}^{-1}(\mathbf{p}_i, \mathbf{p}_j)} \tag{5}$$

Following [18,34], we opted to utilize relative features $\mathbf{f}_j - \mathbf{f}_i$ since it is more efficient than aggregating raw neighborhood features and it enriches the expressivity of the network. To encode Gaussian positions we selected the high-performing multi-resolution hash encoding [23] given its lightweight nature and its ability to expressively encode complex scenes.

Refinement Network. In the final state of the proposed model, the structure-aware point encodings are decoded to the 3D Gaussian attributes using four distinct networks, one for each of the attributes; namely position $\boldsymbol{\mu} \in \mathbb{R}^3$, color $\mathbf{c} \in \mathbb{R}^3$, opacity $\boldsymbol{\alpha} \in \mathbb{R}^1$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$. Aligned with 3DGS, we parametrize covariance matrix $\boldsymbol{\Sigma}$, with a scale vector $\mathbf{S} \in \mathbb{R}^3$ and a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ represented with quaternions. To enforce high rendering speed, we defined each decoder as a small MLP that takes as input the structure-aware encoding and the view-dependent point positions $\mathbf{p}_i$ and outputs the Gaussian attributes for each point. For example, the color attribute $\mathbf{c}$ can be defined as:

$$\mathbf{c}_i = \mathrm{MLP}_{\mathbf{c}}(\Phi(\mathbf{p}_i, \mathbf{f}_i), \mathbf{p}_i^{\mathbf{c}}) \tag{6}$$

where $\mathrm{MLP}_{\mathbf{c}}(\cdot)$ represents the color attribute MLP layer and $\mathbf{p}_i^{\mathbf{c}}$ are the view-depended point positions, normalized with the camera coordinates $\mathbf{x}_c$ as:

$$\mathbf{p}_i^{\mathbf{c}} = \frac{\mathbf{p}_i - \mathbf{x}_c}{||\mathbf{p}_i - \mathbf{x}_c||_2} \tag{7}$$

Similarly, we predict opacity $\boldsymbol{\alpha}$, scale $\boldsymbol{S}$ and rotation $\boldsymbol{R}$ attributes using view-depended point positions. In contrast to the aforementioned view-dependent Gaussian attributes, we opted to learn the 3D scene, represented from the Gaussian mean positions $\boldsymbol{\mu}$, in a camera-agnostic manner. This way we can enforce the model to learn the underlying 3D geometry solely using the world-space point position and shift the bulk of the view-depended attributes to the rest of the MLPs. Additionally, to enforce stable training, we model the 3D Gaussian positions $\boldsymbol{\mu}$ as displacement vectors from the initial COLMAP positions:

$$\boldsymbol{\mu}_i = \mathbf{p}_i + \Delta\mathbf{p}_i, \quad \Delta\mathbf{p}_i = \mathrm{MLP}(\Phi(\mathbf{p}_i, \mathbf{f}_i)) \tag{8}$$

where $\mathbf{p}_i$ denotes the initial point derived from structure-from-motion.

SAGS-Lite. In addition to our best-performing model, we present a simple but effective strategy to reduce the storage requirements of the model while retaining the high quality of the renderings and its structural properties. Considering that the predominant storage burden in 3D Gaussian Splatting methods stems from the abundance of stored Gaussians, our objective was to devise a pipeline that would yield a much more compact set of Gaussians, without relying on vector quantization or compression techniques. In particular, leveraging the initial densification step, we can define the midpoints using the initial key points of the COLMAP and predict their Gaussian attributes using the interpolated key-point features.

Under this setting, the mid-points can be generated on the fly and their corresponding features will be interpolated from the structure-aware feature encodings $\mathbf{f}'$ as:

$$\mathbf{f}'_m = \frac{1}{2}(\mathbf{f}'_i + \mathbf{f}'_j), \quad (i,j) \in \mathbf{P} \tag{9}$$

where $\mathbf{f}'_i, \mathbf{f}'_j$ are the feature encodings of two keypoints $i, j \in \mathbf{P}$ and $\mathbf{f}'_m$ defines the interpolated feature of their midpoint. Aligned with our full model, the interpolated features along with their corresponding view-depended interpolated positions are fed to the refinement networks to predict their Gaussian attributes.

Training. To train our model we utilized a $\mathcal{L}_1$ loss and a structural-similarity loss $\mathcal{L}_{SSIM}$ on the rendered images, following [15]:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{SSIM} \tag{10}$$

where λ is set to 0.2.

Implementation. We build our model on top of the original 3D-GS [15] PyTorch implementation. Similar to 3D-GS, we train our method for 30,000 iterations across all scenes and apply a growing and pruning step until 15,000 iterations, every 100 iterations, starting from the 1500 iterations. Throughout our implementation, we utilize small MLPs with a hidden size of 32. For additional implementation details, we refer the reader to the supplementary material.

4   Experiments

Datasets. To evaluate the proposed method, on par with the 3D-GS [15], we utilized 13 scenes including nine scenes from Mip-NeRF360 [2], two scenes from Tanks&Temples [16] and two scenes from Deep Blending [13] datasets.

Baselines. We compared the proposed method with NeRF- and 3D-GS-based state-of-the-art works in novel-view synthesis, including the Mip-NeRF360 [2], Plenoxels [10], iNGP [23], 3D-GS [15] along with the recent Scaffold-GS [20].

Metrics. We evaluate the proposed SAGS model in terms of rendering quality, structure preservation, and rendering performance. To measure the rendering quality, we utilized the commonly used PSNR, SSIM [35], and LPIPS [40] metrics. In addition, we report model storage requirements in megabytes (MB) and rendering speed in frames per second (FPS).

4.1   Novel-View Synthesis

Rendering Quality. In Tab. 1, we report the average evaluation performance of the proposed and the baseline methods over the three datasets. As can be easily seen, SAGS outperforms 3D-GS and the recently introduced Scaffold-GS method under all datasets and metrics. Leveraging the inter-connectivity between the 3D Gaussians, the SAGS model can facilitate high-quality reconstruction in challenging cases that the independent and unstructured optimization scheme of 3D-GS and Scaffold-GS methods struggle. As can be qualitatively validated in Fig. 4, the proposed SAGS model can better capture high-frequency details, such as the letters on the train wagon, the door handle, and the desk chair mechanism.

Table 1: Quantitative comparison between the proposed and the baseline methods on Mip-NeRF360 [2], Tanks&Temples [16] and Deep Blending [13] datasets.

Dataset Method|Metrics	Mip-NeRF360 PSNR ↑	SSIM ↑	LPIPS ↓	Tanks&Temples PSNR ↑	SSIM ↑	LPIPS ↓	Deep Blending PSNR ↑	SSIM ↑	LPIPS ↓
3D-GS [15]	28.69	0.870	0.182	23.14	0.841	0.183	29.41	0.903	0.243
Mip-NeRF360 [2]	29.23	0.844	0.207	22.22	0.759	0.257	29.40	0.901	0.245
iNPG [23]	26.43	0.725	0.339	21.72	0.723	0.330	23.62	0.797	0.423
Plenoxels [10]	23.62	0.670	0.443	21.08	0.719	0.379	23.06	0.795	0.510
Scaffold-GS [20]	28.84	0.848	0.220	23.96	0.853	0.177	30.21	0.906	0.254
SAGS-Lite	28.54	0.841	0.225	24.16	0.846	0.181	29.07	0.889	0.292
SAGS	29.65	0.874	0.179	24.88	0.866	0.166	30.47	0.913	0.241

Structure Preservation. Apart from the visual quality of the renderings, preserving the 3D geometry of the scene is extremely crucial for downstream VR/AR applications of NVS. Using the proposed structure-aware encoder, we manage to tackle the structure preservation limitations of previous 3D-GS methods and constrain the point displacements close to their initial positions. As pointed out with the red arrows in Fig. 4, this substantially diminishes floater artifacts, which were prevalent in the 3D-GS method. To quantitatively validate the structural preservation of our method, in Fig. 5, we illustrate the displacements of points, in a color-coded format, on top of their original positions. In particular, we depict the color-coded displacements for the *train* scene from the Tanks&Temples dataset, where points with color closer to purple indicate small displacements and colors closer to yellow indicate large displacements. Aligned with the findings of Fig. 1 (right), SAGS better constrains the Gaussians to lie in the original geometry of the scene compared to 3D-GS and Scaffold-GS methods that rely on structure-agnostic optimization to fit the scene.

In addition, preserving the geometry of the scene ensures the preservation of spatial relationships and distances between objects, which is significantly important in 3D modeling. The depth information provides crucial cues that verify the spatial distances within a scene and can easily validate the suppression of floater artifacts that are not visible in the rendered image. Therefore, in Fig. 6, we qualitatively evaluate the depth maps that correspond to various scene renderings, generated by the SAGS and the Scaffold-GS methods. The proposed
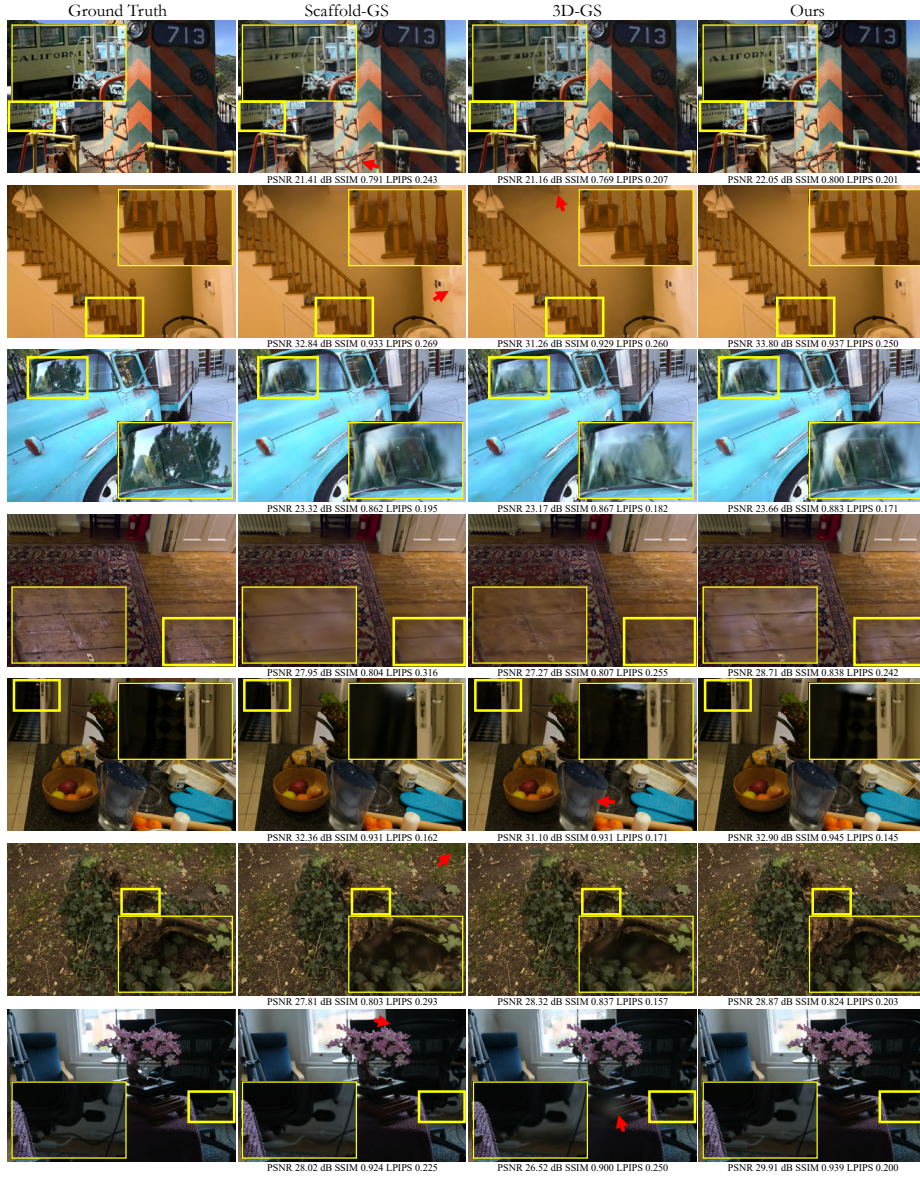
Fig. 4: **Qualitative comparison.** We qualitatively evaluate the proposed and the baseline methods (3D-GS [15] and Scaffold-GS [20]) across six scenes from different datasets. We highlight some detailed differences between the three methods using a magnified crop in yellow. We also emphasize additional visual artifacts using red arrows. The proposed method consistently captures more structural and high-frequency details while minimizing floaters and artifacts compared to the baseline methods.
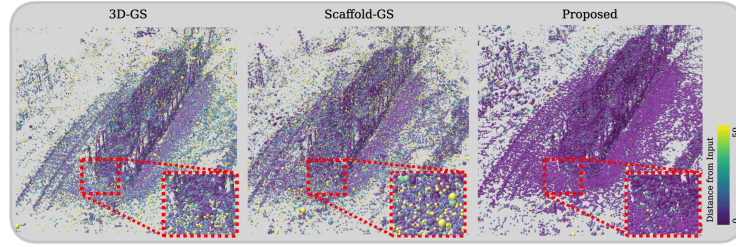
Fig. 5: Color Coded Gaussian Displacements. We measured the Gaussians' displacements from their original positions, on the "train" scene from Tanks&Temples [16] dataset, and encoded them in a colormap scale. Colors closer to purple color indicate small displacements. Both the 3D-GS and Scaffold-GS methodologies depend on a rudimentary point optimization approach, that neglects the local topology and fails to guide the Gaussians in a structured manner.
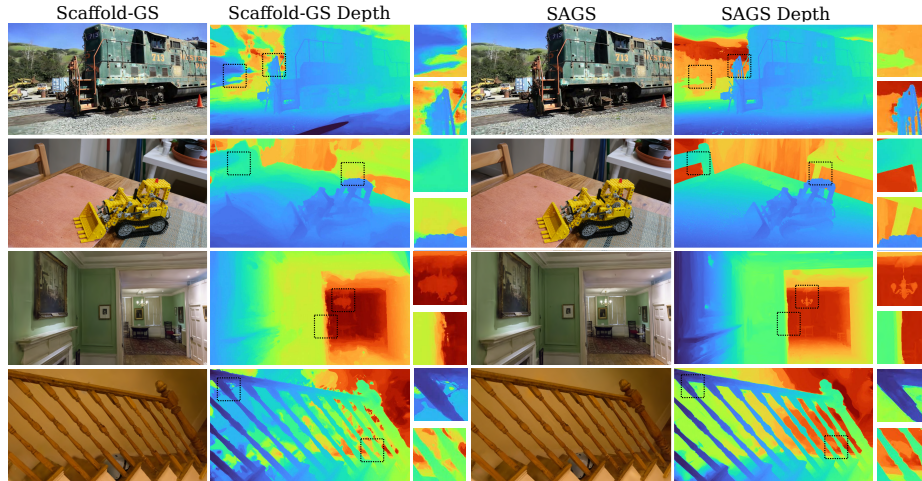

Fig. 6: Depth Structural Preservation. Comparison between the proposed and the Scaffold-GS method on the scene's structure preservation. The proposed method can accurately capture sharp edges and suppress "floater" artifacts that are visible on the Scaffold-GS depth maps.

method can not only achieve sharp edges and corners (*e.g.*, on the table and the door) but also accurate high-frequency details (*e.g.*, the tire track of the bulldozer and the staircase banister). On the contrary, Scaffold-GS method provides noisy depth maps that fail to capture the scene's geometry, *e.g.*, the chair back and the chandelier are modeled with an enormous set of points that do not follow any 3D representation. This is caused by the unstructured nature of the Gaussian optimization that attempts to minimize only the rendering constraints without any structural guidance. Furthermore, Scaffold-GS method falls short in accurately representing flat surfaces, as can be seen in the walls and the table,

Table 2: Performance comparison between the proposed and the 3D-GS based methods. We also report the storage reduction of each model compared to original 3D-GS method [15].

Dataset	Mip-NeRF360		Tanks&Temples		Deep Blending	
Methods	FPS	Mem (MB)	FPS	Mem (MB)	FPS	Mem (MB)
3D-GS [15]	97	693	123	411	109	676
Scaffold-GS [20]	102	252 (2.8× ↓)	110	87 (4.7× ↓)	139	66 (10.2× ↓)
Ours	**110**	135 (5.1× ↓)	108	75 (5.5× ↓)	138	58 (11.7× ↓)
Ours-Lite	101	**76 (9.1× ↓)**	**112**	**35 (12× ↓)**	**142**	**28 (24× ↓)**

Table 3: Ablation study on the components of SAGS. The ablation was performed on the Deep Blending and the Tanks&Temples datasets.

Scene	Deep Blending			Tanks&Temples		
Ablation|Metrics	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
w/o Curvature-Aware Densification	29.87	0.901	0.259	23.97	0.851	0.175
w/o GNN	29.94	0.905	0.254	24.19	0.844	0.181
w/o Positional-Encoding $\gamma(\mathbf{p})$	30.21	0.909	0.252	24.31	0.852	0.169
w/o Global Feature $\mathbf{g}$	30.17	0.911	0.250	24.42	0.861	0.174
w/o View Dependent Positions $\mathbf{p}_i^{\mathbf{c}}$	30.07	0.903	0.256	24.37	0.849	0.173
SAGS	**30.47**	**0.913**	**0.241**	**24.88**	**0.866**	**0.166**

compared to SAGS which accurately captured flat surfaces.

Performance. Apart from the rendering quality we evaluated the performance of the proposed and the baseline methods in terms of rendering speed (FPS) and storage size (MB) under all datasets. In Tab. 2, we report the FPS and memory sizes for each method averaged per dataset. Both SAGS and SAGS-Lite models achieve a real-time rendering speed, with over 100 FPS under all scenes, on par with the Scaffold-GS and 3D-GS methods. Importantly, SAGS reduces the storage requirements of 3D-GS by 5× on the challenging MipNeRF360 dataset [2] achieving state-of-the-art performance with an average model of 135MB. The storage requirements are reduced even more with our lightweight model that can achieve up to 24×-storage reduction compared to 3D-GS without relying on any compression scheme.

Comparison with SAGS-Lite. In Fig. 7, we qualitatively evaluate the lightweight version of the proposed SAGS model. As shown in Tab. 2, the SAGS-Lite model can drastically reduce the storage requirements of 3D-GS by up to 28 times, while achieving similar rendering performance with 3D-GS [15]. Despite lacking some sharp details when compared to our full SAGS model, SAGS-Lite can accurately represent the scene while at the same time avoiding the floater artifacts caused by 3D-GS optimization.
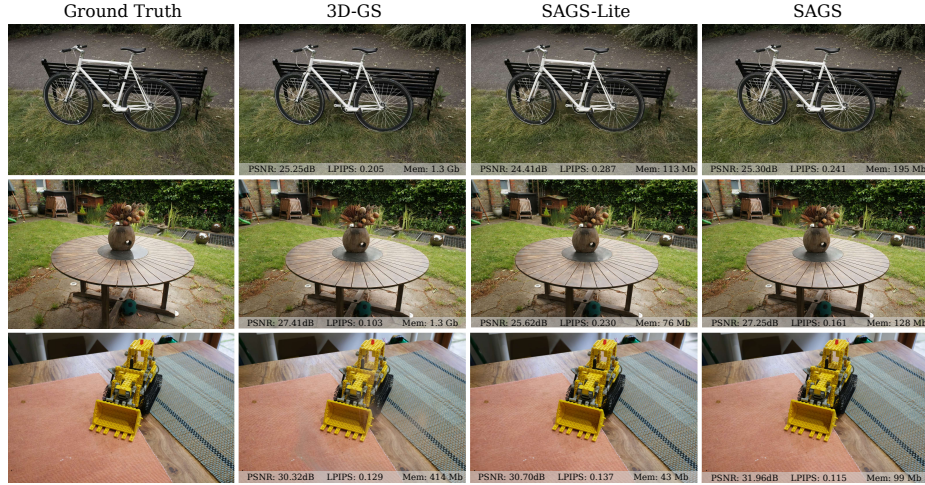
Ground Truth   3D-GS   SAGS-Lite   SAGS

PSNR: 25.25dB   LPIPS: 0.205   Mem: 1.3 Gb   PSNR: 24.41dB   LPIPS: 0.287   Mem: 113 Mb   PSNR: 25.30dB   LPIPS: 0.241   Mem: 195 Mb

PSNR: 27.41dB   LPIPS: 0.103   Mem: 1.3 Gb   PSNR: 25.62dB   LPIPS: 0.230   Mem: 76 Mb   PSNR: 27.25dB   LPIPS: 0.161   Mem: 128 Mb

PSNR: 30.32dB   LPIPS: 0.129   Mem: 414 Mb   PSNR: 30.70dB   LPIPS: 0.137   Mem: 43 Mb   PSNR: 31.96dB   LPIPS: 0.115   Mem: 99 Mb

Fig. 7: Comparison with SAGS-Lite. We qualitatively compared the proposed SAGS-Lite model against SAGS and 3DGS. SAGS-Lite can achieve to maintain high quality renderings with up to 24× storage reduction compared to 3DGS.

4.2 Ablation Study

To evaluate the impact of individual components within the proposed method, we conducted a series of ablation studies in the Deep Blending and Tanks&Temples datasets. Quantitative and qualitative results are presented in Tab. 3 and Fig. 8.

Effect of the Densification. First, removing the curvature-aware densification step leads to parts of the scene being under-represented in the resulting point cloud, as the gradient-based growing struggles to sufficiently fill them. This is particularly evident in areas of COLMAP which are initially undersampled, such as the floor in the *drjohnson* scene and the gravel road in the *train* scene. In contrast, our curvature-aware densification adequately fills those areas supporting further growing during training.

Effect of the Structure-Aware Encoder. Replacing the aggregation layer in Eq. (4) with an MLP diminishes the inductive biases and the expressivity of the model to encode the scene's structure, leading to Gaussians with locally inconsistent attributes. As can be seen in Fig. 8, the absence of structure results in renderings with more floaters and artifacts.

Feature Analysis. Using the points' positions directly in Eq. (4) instead of their positional encodings $\gamma(\mathbf{p})$, results in lower resolution representations of the scene which implies less high frequency details in renderings. A similar effect is caused by removing the global structure information offered by $\mathbf{g}$, which leads to less expressive feature encodings $\Phi(\mathbf{p}_i, \mathbf{f}_i)$ limiting the capacity of the refinement network and the quality of its predictions. For both previous configurations, the floor in the *drjohnson* scene and areas of the *train* scene demonstrate parts with flat textures. Last, by ablating the view dependent positions $\mathbf{p}_i^c$ from the ap-

Fig. 8: **Ablation study on the components of SAGS.** We perform a series of ablation experiments on the Deep Blending and the Tanks&Temples datasets and demonstrate qualitative results from the *drjohnson* and the *train* scenes. We emphasise differences across model configurations over the same crop of the resulting images and highlight additional visual artifacts using red arrows.

pearance related attributes resulted in missing reflections, for example on the floor in *drjohnson*, and floating points as on the black cable in the *train* scene.

5   Conclusion

In this paper, we present Structure-Aware Gaussian Splatting (SAGS), a novel Gaussian Splatting approach that leverages the intrinsic scene structure for high-fidelity neural rendering. Motivated by the shortcomings of current 3D Gaussian Splatting methods to naively optimize Gaussian attributes independently neglecting the underlying scene structure, we propose a graph neural network based approach that predicts Gaussian's attributes in a structured manner. Using the proposed graph representation, neighboring Gaussians can share and aggregate information facilitating scene rendering and the preservation of its geometry. We showcase that the proposed method can outperform current state-of-the-art methods in novel view synthesis while retaining the real-time rendering of 3D-GS. We further introduce a simple yet effective mid-point interpolation scheme that attains up to 24×-storage reduction compared to 3D-GS method while retaining high-quality rendering, without the use of any compression and quantization algorithm. Overall, our findings demonstrate the benefits of structure in 3D-GS.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 5835–5844 (2021), `https://api.semanticscholar.org/CorpusID:232352655`

2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022)

3. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: The Conference on Computer Vision and Pattern Recognition (CVPR) (2023)

4. Chen, Z., Wang, F., Liu, H.: Text-to-3d using gaussian splatting. arXiv preprint arXiv:2309.16585 (2023)

5. Cui, H., Gao, X., Shen, S., Hu, Z.: Hsfm: Hybrid structure-from-motion. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1212–1221 (2017)

6. Cui, H., Shen, S., Gao, X., Hu, Z.: Batched incremental structure-from-motion. In: 2017 International Conference on 3D Vision (3DV). pp. 205–214. IEEE (2017)

7. Deng, N., He, Z., Ye, J., Duinkharjav, B., Chakravarthula, P., Yang, X., Sun, Q.: Fov-nerf: Foveated neural radiance fields for virtual reality. IEEE Transactions on Visualization and Computer Graphics **28**(11), 3854–3864 (2022)

8. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)

9. Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z.: Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps (2023)

10. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022)

11. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: 2007 IEEE 11th International Conference on Computer Vision. pp. 1–8. IEEE (2007)

12. He, Y., Tang, D., Zhang, Y., Xue, X., Fu, Y.: Grad-pu: Arbitrary-scale point cloud upsampling via gradient descent with learned distance functions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)

13. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering **37**(6), 257:1–257:15 (2018)

14. Kaya, B., Kumar, S., Oliveira, C., Ferrari, V., Van Gool, L.: Uncertainty-aware deep multi-view photometric stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12601–12611 (2022)

15. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), `https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/`

16. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017)
17. Lee, J.C., Rho, D., Sun, X., Ko, J.H., Park, E.: Compact 3d gaussian representation for radiance field. arXiv preprint arXiv:2311.13681 (2023)
18. Li, G., Muller, M., Thabet, A., Ghanem, B.: Deepgcns: Can gcns go as deep as cnns? In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9267–9276 (2019)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**, 91–110 (2004)
20. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2024)
21. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
22. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
23. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
24. Niedermayr, S., Stumpfegger, J., Westermann, R.: Compressed 3d gaussian splatting for accelerated novel view synthesis (2023)
25. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces. In: IEEE Visualization, 2002. VIS 2002. pp. 163–170. IEEE (2002)
26. Potamias, R.A., Bouritsas, G., Zafeiriou, S.: Revisiting point cloud simplification: A learnable feature preserving approach. In: European Conference on Computer Vision. pp. 586–603. Springer (2022)
27. Potamias, R.A., Neofytou, A., Bintsi, K.M., Zafeiriou, S.: Graphwalks: efficient shape agnostic geodesic shortest path estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2968–2977 (2022)
28. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
29. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
30. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
31. Snavely, N., Seitz, S.M., Szeliski, R.: Photo Tourism: Exploring Photo Collections in 3D. Association for Computing Machinery, New York, NY, USA, 1 edn. (2023), https://doi.org/10.1145/3596711.3596766
32. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8248–8258 (2022)
33. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023)

34. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (tog) **38**(5), 1–12 (2019)

35. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004). `https://doi.org/10.1109/TIP.2003.819861`

36. Wiles, O., Gkioxari, G., Szeliski, R., Johnson, J.: Synsin: End-to-end view synthesis from a single image. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 7465–7475 (2019), `https://api.semanticscholar.org/CorpusID:209405397`

37. Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., Dai, B., Lin, D.: Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXII. Lecture Notes in Computer Science, vol. 13692, pp. 106–122. Springer (2022). `https://doi.org/10.1007/978-3-031-19824-3_7`, `https://doi.org/10.1007/978-3-031-19824-3_7`

38. Xiong, H., Muttukuru, S., Upadhyay, R., Chari, P., Kadambi, A.: Sparsegs: Real-time 360 {\deg} sparse view synthesis using gaussian splatting. arXiv preprint arXiv:2312.00206 (2023)

39. Yifan, W., Serena, F., Wu, S., Öztireli, C., Sorkine-Hornung, O.: Differentiable surface splatting for point-based geometry processing. ACM Transactions on Graphics (TOG) **38**(6), 1–14 (2019)

40. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)

41. Zheng, J., Jang, Y., Papaioannou, A., Kampouris, C., Potamias, R.A., Papantoniou, F.P., Galanakis, E., Leonardis, A., Zafeiriou, S.: Ilsh: The imperial light-stage head dataset for human head view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1112–1120 (2023)

42. Zhuang, B., Cheong, L.F., Lee, G.H.: Baseline desensitizing in translation averaging. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4539–4547 (2018)