Hybrid Mesh-Gaussian Representation for Efficient Indoor Scene Reconstruction

Binxiao Huang[1] , **Zhihao Li**[2] , **Shiyong Liu**[2] , **Xiao Tang**[2] , **Jiajun Tang**[3] ,
Jiaqi Lin[4] , **Yuxin Cheng**[1] , **Zhenyu Chen**[2] , **Xiaofei Wu**[2] , **Ngai Wong**[1†]
[1]The University of Hong Kong
[2]Huawei Technologies Ltd
[3]Peking University
[4]Tsinghua University
{bxhuang, yxcheng, nwong}@eee.hku.hk, {zhihao.li, liushiyong3, tangxiao12,
wuxiaofei2}@huawei.com, jiajun.tang@pku.edu.cn,
linjq22@mails.tsinghua.edu.cn, zhenyu.chen@tum.de

Abstract

3D Gaussian splatting (3DGS) has demonstrated exceptional performance in image-based 3D reconstruction and real-time rendering. However, regions with complex textures require numerous Gaussians to capture significant color variations accurately, leading to inefficiencies in rendering speed. To address this challenge, we introduce a hybrid representation for indoor scenes that combines 3DGS with textured meshes. Our approach uses textured meshes to handle texture-rich flat areas, while retaining Gaussians to model intricate geometries. The proposed method begins by pruning and refining the extracted mesh to eliminate geometrically complex regions. We then employ a joint optimization for 3DGS and mesh, incorporating a warm-up strategy and transmittance-aware supervision to balance their contributions seamlessly.Extensive experiments demonstrate that the hybrid representation maintains comparable rendering quality and achieves superior frames per second FPS with fewer Gaussian primitives.

1 Introduction

Reconstructing a high-quality 3D representation with multiple unordered images is a critical task in computer vision and graphics. In recent years, an implicit representation approach called the neural radiance field (NeRF) [Mildenhall *et al.*, 2020] shows extraordinary performance. NeRF combines the deep learning and volumetric rendering approach to produce high-fidelity images from novel views. However, the requirement of dense sampling of spatial positions along rays heavily slows down the training and rendering speed. By representing the scene with plenty of anisotropic 3D Gaussian primitives, 3D Gaussian splatting (3DGS) [Mildenhall *et al.*, 2020] has emerged as a charming and efficient scene representation that achieves real-time rendering for scenes with unprecedented fidelity. 3DGS parameterizes the scene as an optimized set of 3D Gaussians obtained from the structure from motion or randomly initialized. It efficiently renders all relevant Gaussian primitives into a 2D plane via splatting-based rasterization to obtain 2D images. [Waczynska *et al.*, 2024; Gao *et al.*, 2024; Fan *et al.*, 2024] utilize the GS to yield accurate reconstructions to extract meshes to simulate physical interactions. Despite the compelling performance achieved by 3D-GS, the number of Gaussian primitives is excessively redundant [Fan *et al.*, 2024; Fang and Wang, 2024]. For tile-wise sorting and pixel-wise alpha-blending in the splatting-based rasterization, massive Gaussian primitives can seriously slow down rendering. Several methods have explored the contributions [Fan *et al.*, 2024; Fan *et al.*, 2023] or important scores [Papantonakis *et al.*, 2024] of each Gaussian to prune the redundant Gaussian for a compact representation. These methods focus on Gaussian attributes to reduce Gaussians. This paper proposes leveraging a GS-mesh hybrid representation to reduce Gaussians. For rich-texture flat areas, vanilla GS requires a large number of Gaussians to offer a precise representation, while textured mesh is capable of expressing high-frequency information through texture. Specifically, we utilize textured mesh (with an opacity set to 1) as the background with certain depths to reduce the number of Gaussians and accelerate the inference rendering speed.

As the distribution of 3D Gaussians shown in Fig. 1, the Gaussians mainly concentrate on complicated geometry and rich texture regions. To precisely represent complex geometries, a substantial number of fine-grained and elaborate meshes are required, which are not obtainable in real-world scene captures. Consequently, we still rely on the Gaussians to represent the complicated regions accurately. For texture-rich areas, we use meshes to represent flat geometric structures and texture maps to express high-frequency content information to reduce the number of Gaussian primitives significantly. Nevertheless, an ideal mesh is not available in real-world scenes. Various methods have been designed to extract mesh from multiple images using NeRF [Mildenhall *et al.*, 2020] or 3D-GS [Kerbl *et al.*, 2023]. Owing to the weak supervision over texture-less areas and the challenges in representing geometrically complex regions, the extracted mesh exhibits noticeable geometric errors. We design some metrics

[†] Corresponding author.

to clear inaccurate and fine-grained meshes, preventing them from misleading the 3D representation. Then, we integrate the textured mesh into the 3DGS method to jointly represent the 3D indoor scenes. To reduce the Gaussians in front of the mesh, we employ texture supervision to narrow the visual disparity between the images rendered from the mesh and the ground truth. Additionally, we introduce a transmittance-aware mask to prevent the projection of front objects' colors onto the mesh. Compared to 3DGS, the proposed hybrid approach significantly reduces Gaussian primitives with a comparable quality.

In summary, our contributions are as follows:

- We propose a hybrid representation that leverages the advantages of Gaussian splatting and textured mesh to reduce the Gaussian primitives.
- We devise certain metrics to prune the flawed meshes in geometrically complex areas and retain them in texture-rich flat regions to facilitate joint rendering.
- We introduce transmittance-aware supervision to facilitate joint optimization of the hybrid representation.

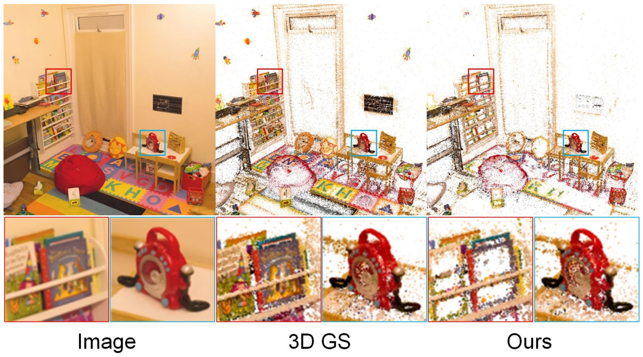


Figure 1: Visualization of Gaussian distributions and corresponding rendered images. The hybrid representation significantly reduces the number of Gaussians in texture-rich regions while maintaining Gaussian density in geometrically complex areas.

2 Related Works

2.1 3D Gaussian Splatting

Recently, 3DGS [Kerbl *et al.*, 2023] has emerged as an advancement in novel view synthesis, achieving real-time, high-fidelity rendering. Different from implicit neural fields with volume rendering based on ray marching, 3DGS represents the 3D scene with explicit Gaussians that can be rendered through tile-based sorting and rasterization processes. A plethora of subsequent studies have surfaced, innovating technology across diverse fields such as human avatar rendering [Luo *et al.*, 2024; Liu *et al.*, 2024], content creation [Li *et al.*, 2024; Zhou *et al.*, 2024], and scene rendering [Wu *et al.*, 2024; Lin *et al.*, 2025; Lin *et al.*, 2024].

However, the performance of 3DGS models is constrained by the large number of Gaussians due to the rasterization process. [Bulò *et al.*, 2024] leverages a per-pixel error function as the density criterion and introduces a mechanism to control the total number of Gaussians. Taming 3DGS [Mallick *et al.*, 2024] designs a steerable densification method that yields any desired number of Gaussians after training. [Papantonakis *et al.*, 2024] assigns a redundancy score for each Gaussian to measure how necessary a Gaussian is to represent the scene. Then, Gaussians whose redundancy scores are higher than the adaptive threshold are pruned. Mini-splatting [Fang and Wang, 2024] proposes an intersection-preserving technique to retain Gaussians that act as intersection points and adopts an importance-weighted sampling approach to maintain a sparse set of Gaussians. LightGaussian [Fan *et al.*, 2023] computes a global significance score of each Gaussian considering the Gaussian's volume, opacity, transmittance, and hit count of each training view. It then ranks all Gaussians by their global significance scores to quantitatively guide the pruning of the lower-ranked Gaussians. All these methods rely solely on GS to represent the 3D scene and take into account the attributions of Gaussians to prune the unimportant ones. As illustrated in Fig. 1, the Gaussian primitives predominantly concentrate on the texture-rich and geometrically complex regions. Our hybrid representation aims to reduce the Gaussians representing the texture-rich area via textured mesh while behaving the same as 3DGS in regions with intricate geometry. Theoretically, our method is perpendicular to previous approaches.

2.2 Hybrid representation

In order to combine the benefits of multiple 3D expressions (*i.e.* NeRF, mesh, GS), various hybrid approaches have been proposed [Wen *et al.*, 2024; Dhamo *et al.*, 2024; Wu and Tuytelaars, 2024]. VMesh [Guo *et al.*, 2023] depicts an object with a textured mesh and an auxiliary sparse volume for efficient view synthesis. HybridNeRF [Turki *et al.*, 2024] combines the best of surface and volume-based rendering into a single model to achieve a real-time rendering with high quality. SuGaR [Guédon and Lepetit, 2024] regulates the 3D Gaussians to align well with the surface to extract the mesh and jointly optimizes the Gaussians and the mesh. GSDF [Yu *et al.*, 2024a] and NeuSG [Chen *et al.*, 2023] improve the quality of surface reconstruction by combining the benefits of 3DGS with neural implicit fields. MeshGS [Choi *et al.*, 2024] introduces several regularization techniques to precisely bind Gaussian splats with the extracted mesh surface for high-quality rendering. Some methods use a GS-mesh hybrid representation to make the GS editable and controllable. GaMeS [Waczynska *et al.*, 2024] parameterizes each Gaussian primitive by the vertices of the mesh face to allow modifying the Gaussians in a mesh manner. SplattingAvatar [Shao *et al.*, 2024] disentangles the motion and appearance of an avatar with explicit mesh faces and implicit appearance modeling with GS. It can directly control the mesh to empower its compatibility with animation techniques. GaussianAvatars [Qian *et al.*, 2024] places Gaussian splats onto a parametric morphable face mode to enable control in terms of expression, pose, and viewpoint of head avatars. HAHA [Svitov *et al.*, 2024] models human avatars utilizing Gaussian splatting and a textured mesh for efficient rendering with much fewer Gaussians. This method assumes the availability of a water-tight SMPL-X parametric mesh and doesn't need to consider the occlusion of objects in the scene. Kim *et.al* [Kim and Lim, 2024] integrates mesh to represent the room layout and employs GS for other objects with a prior segmentation. Nevertheless, the textured mesh is only used for layouts that demand minimal Gaussians, and the manual acquisition of the mesh from a synthetic dataset makes it impractical for real-world scenes. Our method leverages mesh to dominate the texture-rich area, effectively reducing the number of Gaussian primitives while maintaining a comparable rendering performance.

3 Methodology

Taking multi-view images with calibrated camera poses and sparse point clouds derived from the structure from motion (SFM) algorithm as input, we aim to combine the 3DGS and textured mesh to accurately represent indoor scenes while maintaining rendering quality and improve efficiency.

3.1 Preliminary

3D Gaussian Splatting 3DGS [Kerbl *et al.*, 2023] utilizes anisotropic 3D Gaussian primitives to represent the 3D scene, achieving state-of-the-art visual quality and rendering efficiency. Each explicit Gaussian primitive is characterized by the following contributions: position center $\boldsymbol{u} \in \mathbb{R}^3$, opacity α, orthogonal rotation matrix $\boldsymbol{R}$, diagonal scale matrix $\boldsymbol{S}$, and spherical harmonics (SH) coefficients. Each Gaussian is defined by $\boldsymbol{u}$ and covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ as:

$$G(\boldsymbol{x}) = exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{u})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{u})). \quad (1)$$

To guarantee the physical meaning of the covariance matrix during optimization, it is formulated as $\Sigma = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^{\boldsymbol{T}}\boldsymbol{R}^{\boldsymbol{T}}$ to keep a positive semi-definite character. To render images, 3D Gaussian primitives are projected onto the 2D image plane with the Jacobian affine approximation $\boldsymbol{J}$ of the project matrix and view transformation $\boldsymbol{W}$: $\Sigma' = \boldsymbol{J}\boldsymbol{W}\Sigma\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{J}^{\boldsymbol{T}}$. Then alpha blending is applied from front to back based on the sorted depth to render the color of each pixel as follows:

$$C(\boldsymbol{p}) = \sum_{i\in\mathcal{N}} c_i\sigma_i\prod_{j=1}^{i-1}(1-\sigma_j), \quad \sigma_i = \alpha_i G_i'(\boldsymbol{p}), \quad (2)$$

where N denotes the number of sorted Gaussians related to the pixel $\boldsymbol{p}$, c_i denotes the color of the projected 2D Gaussian G_i'. The $\mathcal{L}_1$ loss and D-SSIM term between the ground truth image $\boldsymbol{I}$ and the rendered image $\hat{\boldsymbol{I}}$ is utilized to optimize 3D Gaussians primitives:

$$\mathcal{L}_c(\boldsymbol{I},\hat{\boldsymbol{I}}) = (1-\lambda)\mathcal{L}_1(\boldsymbol{I},\hat{\boldsymbol{I}}) + \lambda\mathcal{L}_{D-SSIM}(\boldsymbol{I},\hat{\boldsymbol{I}}), \quad (3)$$

where λ is set to 0.2 as default.

Textured mesh A mesh is a collection of vertices and faces that provides the geometric description of 3D models. A texture map is an image that maps the detailed visual information of images onto the surface of a mesh. To render the triangle meshes with a texture map, barycentric interpolation is applied to accurately assign texture colors to individual pixels in the rendered image based on UV coordinates. In this way, textured meshes enable fast real-time rendering of flat regions with realistic visual results. However, it struggles to model intricate structures, which we rely on the Gaussians to represent.

3.2 Overview

Our hybrid representation depicts indoor scenes with triangular mesh surfaces and anisotropic Gaussian primitives. In this section, we first clear the flawed meshes at geometrically complex regions to make room for Gaussians and refine the remaining ones at the texture-rich region to optimize the texture map. Then, we propose a transmittance-aware texture loss equipped with a warm-up to optimize the hybrid representation jointly. The overall pipeline of our approach is shown in Fig. 2.

3.3 Mesh Pruning

Recent NeRF and Gaussian Splatting (GS) approaches extract meshes composed of numerous triangles from multi-view images. However, due to weak supervision in texture-less areas and challenges in representing complex geometries, the extracted meshes often exhibit significant geometric errors that adversely impact the hybrid representation. Furthermore, Gaussians outperform textured meshes in modeling geometrically complex and thin regions. To address these issues, we propose a pruning strategy that considers normal maps, adjacent angles, and mesh sizes to reduce the extracted mesh, thereby allocating space for Gaussian primitives to handle intricate regions.

We adopt the Planar-based Gaussian Splatting (PGSR) [Chen *et al.*, 2024] as the baseline for initial mesh extraction. In texture-less regions, Gaussian primitives exhibit large scales $\boldsymbol{S}$, resulting in numerous mesh bumps with incorrect geometry that occlude surrounding objects. To mitigate this, we first prune the mesh by removing Gaussians exceeding a predefined scale threshold and subsequently extract the mesh using the truncated signed distance function (TSDF). This process yields a coarse mesh with minor bumps and rough surfaces, leading to increased disk storage and slower rendering speeds.

To further streamline the mesh, we apply a quadric error-based simplification method called QSlim to reduce the number of triangles to a desired count K, while preserving geometric topology. Next, we eliminate mesh segments representing complex geometries by analyzing the total variation of the normal map, angles between adjacent triangles, and triangle sizes. For each training view, we project the mesh onto 2D images to establish a mapping from pixels to triangles, enabling targeted pruning based on specific metrics.

Specifically, regions with high normal variation, indicating complex geometries, are identified using a prior normal map from the pre-trained StableNormal diffusion model. The top α_{normal} percent of pixels with the highest total variation are marked for triangle removal. Additionally, triangles with angles exceeding $45°$ between adjacent faces are pruned to eliminate boundaries of flat regions or complex geometries. To remove trivial geometries that hinder texture map optimization as occluders, we sort and prune the smallest α_{area} percent of triangles based on their area sizes.

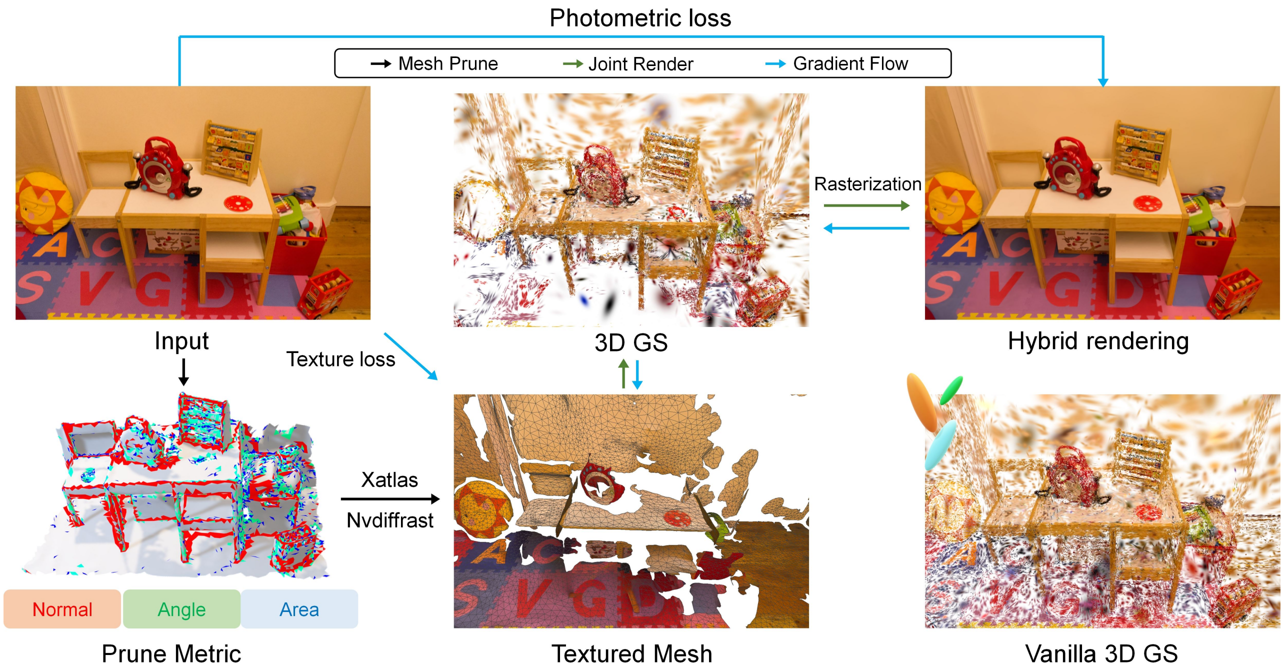


Figure 2: Overview of the proposed method pipeline. We first use normal, angle, and area size metrics to remove the meshes representing the geometrically complex region. Then, we use hybrid representation to induce Gaussians to fill the empty space and inhibit its densification in texture-rich flat regions under photometric and texture supervision.

This comprehensive pruning process may result in floating triangles and tiny holes. To resolve these issues, we remove isolated connected triangles with counts below a minimum threshold α_{group} and close any resultant holes in the mesh to prevent floating geometries. Finally, we apply the LS3 subdivision surface algorithm to smooth the mesh further. All pruning and refinement operations are automated with predefined hyperparameters, resulting in a smooth mesh that predominantly represents texture-rich flat areas.

To incorporate high-frequency visual information from multi-view images, we attach texture coordinates (UV maps) to the mesh using Xatlas. We then initialize the texture map using Nviffrast, a differentiable rasterization tool, under image supervision. Consequently, we obtain a smooth and discrete mesh that effectively represents texture-rich flat regions, allowing us to gradually densify Gaussian primitives using the textured mesh as a background with fixed depth values.

3.4 Joint optimization

Joint rasterization As shown in Fig. 2, we simultaneously optimized the 3DGS and the textured mesh in a differentiable way. For each training view, we first rendered the mesh to get a RGB image $\boldsymbol{I}_m$ and a depth map $\boldsymbol{D}_m$ and made the texture map optimizable. During the rasterization of GS, if the pixel is equipped with a triangle, we used the alpha-blending to produce an RGB value $\boldsymbol{I}_{gs}$ of Gaussian primitives before the triangle ($\boldsymbol{D}_{gs}<\boldsymbol{D}_m$) and got a transmittance coefficient $\boldsymbol{T}$ applied to the mesh. Otherwise, the rasterization performs as the vanilla GS. In short, we regard the mesh as an opaque background with a certain depth value. The final rasterization result is given as follows:

$$\boldsymbol{I}_h = \begin{cases} \boldsymbol{I}_{gs} + \boldsymbol{T} \times \boldsymbol{I}_m, & \text{if triangle available} \\ \boldsymbol{I}_{gs} + \boldsymbol{T} \times \boldsymbol{I}_{bg}, & \text{else} \end{cases}$$

where $\boldsymbol{I}_{bg}$ denotes the predefined background color. Following the training strategy of vanilla GS, we replaced the $\hat{\boldsymbol{I}}$ in eqn. 3 with the $\boldsymbol{I}_h$ to provide the supervision.

Transmittance-aware supervision Under the photometric supervision of the hybrid representation, the optimization process prefers to cover the front of the mesh with numerous Gaussian primitives. It will reduce the mesh's contribution to each pixel. To reduce the Gaussian primitives in front of the mesh, we adopt a $\|\widetilde{\boldsymbol{I}}, \widetilde{\boldsymbol{I}}_m\|_2^2$ norm loss between the masked ground truth $\widetilde{\boldsymbol{I}}$ and the masked texture image $\widetilde{\boldsymbol{I}}_m$. $\widetilde{\boldsymbol{I}}$ equals to $\boldsymbol{I}$ when triangle is available for this pixel, else $\widetilde{\boldsymbol{I}}$ equals to 0.

However, simply enforcing the rendered mesh to be close to the ground truth image will induce some ghosts on the mesh, shown in Fig. 5. When the explicit object in front of the back mesh is absent, in most cases in our settings, since we pruned the mesh with complex geometry. This supervision will project the object onto the back meshes along the ray from the camera center. When the mesh is viewed from other perspectives, we will see a ghost object that should not exist. First, we warm up the training only with the photometric loss for the first N iterations to make up the absent object with Gaussians. Then we proposed a transmittance-aware texture loss $\mathcal{L}_t$ to prevent the incorrect supervision hindering the optimization of the texture map as follows:

$$\mathcal{L}_t(\boldsymbol{I}, \boldsymbol{I}_m) = \mathcal{M}_{\boldsymbol{T}} \cdot \|\widetilde{\boldsymbol{I}}, \widetilde{\boldsymbol{I}}_m\|_2^2, \tag{4}$$

$\mathcal{M}_{\boldsymbol{T}}$ denotes a transmittance-aware mask for each pixel. If the $\boldsymbol{T}$ is larger than 0.5, we prefer to rely on the textured mesh to dominate the pixel to reduce the Gaussian primitives. Otherwise, we hope to use the hybrid representation. Therefore, we used a *sigmoid* function to get the mask for each pixel:

$$\mathcal{M}_{\boldsymbol{T}} = 1/(1 + e^{-k\cdot(\boldsymbol{T}-0.5)}). \tag{5}$$

Since we adopt the $\mathcal{L}_2$ to reduce the Gaussian primitives and use hybrid representation to render the final image, we set λ to zero after densification iteration (*i.e* 15k) of 3DGS training. Combined with the warm-up strategy, some Gaussian primitives will be created at the object's location, making the transmittance for the back mesh less than 0.5. Therefore, the object's color has a negligible impact on the optimization of back meshes. The joint optimization is supervised by:

$$\mathcal{L}(\boldsymbol{I}, \boldsymbol{I}_h, \boldsymbol{I}_m) = \mathcal{L}_c(\boldsymbol{I}, \boldsymbol{I}_h) + \lambda \cdot \mathcal{L}_t(\boldsymbol{I}, \boldsymbol{I}_m), \tag{6}$$

where λ controls the balance between the hybrid representation and the textured mesh.

4 Experiments

We first present the details of the proposed hybrid representation. We then assess performance for indoor scenes on the Deep blending dataset [Hedman *et al.*, 2018] and the challenging Scannet++ dataset [Yeshwanth *et al.*, 2023].

4.1 Implementation Details

We build our method upon the open-source 3DGS code. Following [Kerbl *et al.*, 2023], we train our models for 30K iterations across all scenes and use the same densification, schedule, and hyperparameters. We follow all the default settings of PGSR [Chen *et al.*, 2024] to extract the initial mesh. We use the Nvdiffrast [Laine *et al.*, 2020] to enable differentiable rendering of the mesh and the optimization of the textured map. Following [Papantonakis *et al.*, 2024], the FPS measurements isolate the runtime of the CUDA rasterizer routine only and exclude any graphics API overheads. The final mesh contains about a hundred thousand triangles with a texture map of size 3 × 2048 × 2048. We follow standard practice and report SSIM, PSNR, and LPIPS for rendering evaluation. All experiments are conducted on a single V100 GPU. More detailed information are provided in supplementary A.

Datasets. We verify the effectiveness of our approach using ten real-world indoor scenes from publicly available datasets: four scenes from the Deep blending [Hedman *et al.*, 2018] and six scenes from ScanNet++ [Yeshwanth *et al.*, 2023].

4.2 Quantitative Results

Rendering Performance. The quantitative results of various methods are presented in Tab. 1. We evaluated the standard metrics such as SSIM, PSNR, LPIPS, Gaussians number, and FPS, comparing against the baseline 3DGS and other Gaussians methods. Tab. 1 demonstrates that, in comparison to the 3DGS baseline, our hybrid representation effectively cuts down around 18% and 35% of Gaussian primitives used in 3DGS while maintaining comparable performance. We then incorporate the textured mesh into [Papantonakis *et al.*, 2024] with our joint optimization strategy to further decrease the number of the Gaussians by about 20% and 50% with a negligible impact on performance. Additionally, the real-time rendering speed is accelerated as a result of using fewer Gaussian primitives. We provide more detailed experimental results and analysis about integrating the mesh into GS methods in the supplementary C.

Since Mip-splatting performs similarly to 3DGS, except for anti-aliasing, we demonstrate the rendered images and Gaussian distributions of 3DGS and Reduced GS in Fig. 3. the The hybrid representation can effectively reduce the Gaussians of the texture-rich regions.

4.3 Qualitative Results

In addition to reducing the Gaussian primitives, the textured mesh demonstrates a sharper representation compared to the purely Gaussians. This is particularly evident in flat regions that are predominantly represented by the mesh. The proposed transmittance-aware supervision plays a crucial role by directly translating the high-frequency visual details from the images into the texture map. As the visual comparisons illustrated in Fig. 4, the mesh enhanced with a texture map is capable of rendering high-fidelity images, particularly in texture-rich flat regions. In contrast, rendering using Gaussian primitives tends to exhibit blurring and artifacts. This demonstrates the advantage of incorporating textured meshes in our hybrid approach to achieve superior visual quality in areas where detailed texture is essential. More visual results are provided in supplementary C.2.

4.4 Upper Bound

To evaluate the upper bound of the hybrid representation and the proposed joint optimization strategy, we conduct experiments on the Replica dataset [Straub *et al.*, 2019], a synthesis indoor scene dataset with ideal meshes.

We also apply the mesh prune onto the ideal mesh and then jointly optimize the GS and mesh to evaluate novel view renderings. Results in Tab. 2 show that if the mesh possesses an accurate geometry, the hybrid representation can achieve better performance, faster rendering with much fewer Gaussians.

4.5 Ablation

We conduct detailed ablation studies using four indoor scenes from the deep bleending dataset to validate the key components of the proposed approach.

Mesh prune & transmittance-aware supervision. We present the ablation study of our mesh prune strategy and transmittance-aware supervision in Tab. 3. For the raw mesh, we only use QSlim decimation to make it have the same number of faces as the mesh pruned by our method for a fare comparison. The results on deep bleeding demonstrate a marked improvement (around 1 dB) in rendering quality compared to the results of a naive combination of mesh and GS. For raw mesh without the transmittance-sensitive supervision, Gaussians were unable to compensate for errors caused by incorrect geometry in intricate parts (*e.g.,* telephone toy) and missing local objects (*e.g.,* wires) shown in Fig. 5. The object's color is wrongly projected onto the back mesh. The mesh

Method	Scannet++					Deep Blending				
	SSIM ↑	PSNR ↑	LPIPS ↓	#Gaussian ↓	FPS ↑	SSIM ↑	PSNR ↑	LPIPS ↓	#Gaussian ↓	FPS ↑
3DGS [Kerbl *et al.*, 2023]	0.862	24.22	0.248	0.911 M	211	0.901	29.55	0.253	2.634 M	346
+ Mesh (Ours)	0.862	24.28	0.254	**0.742M**	**231**	0.893	29.48	0.262	**1.698 M**	**498**
Mip-splatting [Yu *et al.*, 2024b]	0.937	30.63	0.162	1.166 M	164	0.899	29.30	0.248	3.443M	129
+ Mesh (Ours)	0.934	30.46	0.168	**0.999 M**	**177**	0.893	29.23	0.259	**2.251 M**	**176**
Reduced GS [Papantonakis *et al.*, 2024]	0.862	24.09	0.248	0.427 M	239	0.900	29.66	0.255	1.333 M	223
+ Mesh (Ours)	0.861	24.09	0.251	**0.336 M**	**253**	0.896	29.51	0.260	**0.668 M**	**317**

Table 1: Quantitative evaluation comparing the proposed method with previous works on two indoor scenes. We report SSIM, PSNR , and LPIPS on test views, the number of Gaussians, and the FPS of the CUDA rasterizer routine.

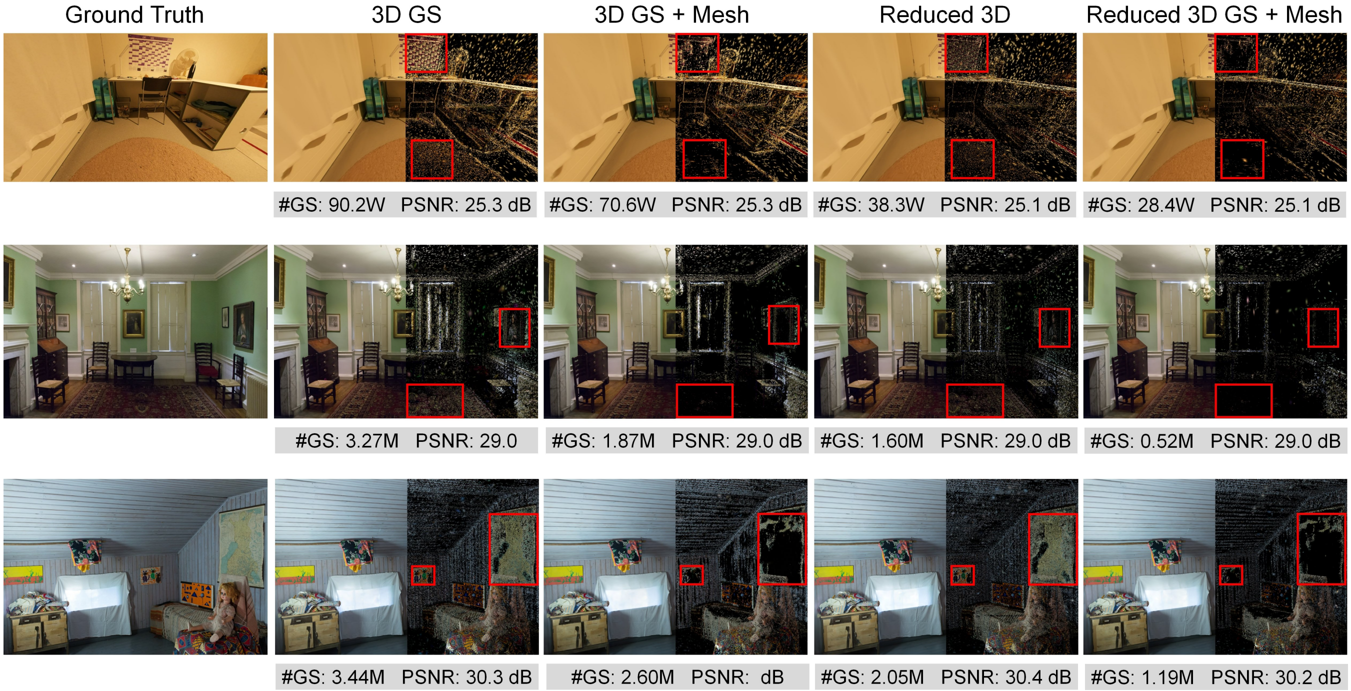


Figure 3: Visualization of the rendered images and Gaussian distribution of various methods. Our hybrid representation effectively reduces the Gaussians in texture-rich flat regions.

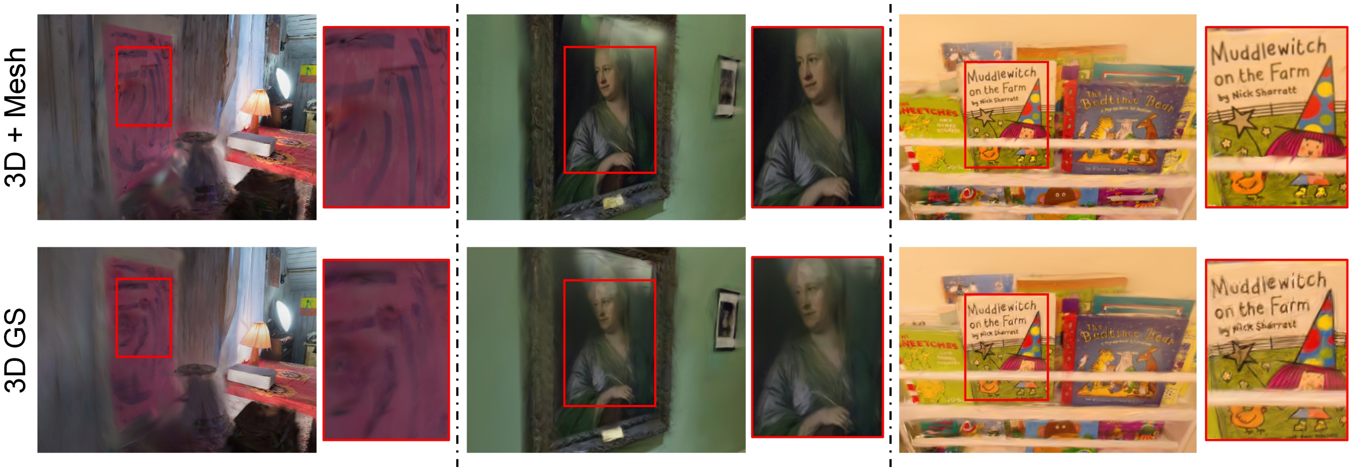
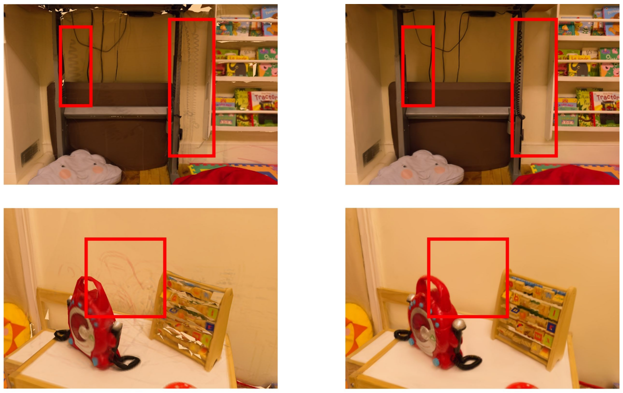


Figure 4: Comparison of the rendered images of 3DGS and our method. The hybrid representation demonstrates sharper renderings compared to the pure Gaussians, where the textured mesh delivers more precise background colors.



Raw Mesh + $M_T = 1$

Pruned Mesh + Sigmoid

Figure 5: Illustration of incorrect color projection and the final renderings of our method.

Method	SSIM ↑	PSNR ↑	LPIPS ↓	#Gaussian ↓	FPS ↑
3DGS	0.965	35.96	0.080	1.667	122
Ours	0.973	36.32	0.034	0.295	446

Table 2: Evaluation of hybrid representation on Replica.

pruning strategy can remove intricate parts, which leads to improved performance. We illustrate several visual comparisons of meshes before and after prune in supplementary B.

We evaluate the effectiveness of the proposed transmittance-aware supervision on the pruned mesh. If $\mathcal{M}_T$ is set to 1, there is an inability to distinguish between the foreground elements and the background components for each pixel of the training images. All pixel values are projected back to the mesh, resulting in inferior performance. The nonlinear sigmoid function performs better by acting as a soft indicator function.

Texture loss. As illustrated in Tab. 4, we assess the balance between the photometric loss and the texture loss by varying the coefficient λ. Without the supervision between the mesh and ground truth images ($\lambda = 0$), it will create more Gaussians in front of the mesh, thus weakening the representation of the textured mesh. Conversely, a high λ will lead to a high transmittance to the mesh, which will make the transmittance-aware mask degrade to a constant $\mathcal{M}_T = 1$ mask. Besides, it forces images rendered solely from the textured mesh to overfit the ground truth and potentially impact the rendering of the hybrid representation.

Mesh	$\mathcal{M}_T$	SSIM ↑	PSNR ↑	LPIPS ↓
Raw	0	0.8812	28.52	0.2803
Raw	1	0.8818	28.48	0.2806
Pruned	1	0.8913	29.31	0.2631
Pruned	T	0.8915	29.41	0.2627
Pruned	Sigmoid	0.8926	29.48	0.2627

Table 3: Comparison of raw and pruned mesh with different transmittance-aware mask.

λ	SSIM ↑	PSNR ↑	LPIPS ↓	#Gaussian ↓
0	0.891	29.42	0.263	1.875 M
0.1	0.893	29.48	0.263	1.698 M
0.2	0.891	29.40	0.263	1.678 M
0.3	0.891	29.38	0.263	1.680 M

Table 4: Ablation study of the balance coefficient λ.

Method	PSNR ↑	#Gaussian ↓	Size ↓
0.5	29.24	1.715 M	4.513 MB
Xatlas	29.48	1.698 M	6.012 MB
Texrecon	29.50	1.715 M	73.42 MB

Table 5: Comparison of different approaches to initialize the texture map. Size refers to the storage requirement of the texture map.

Texture initialization. We compare three methods for initializing the texture map. The first sets all values in the texture map to 0.5. The second uses Xatlas to get UV coordinates and then optimizes the texture map with Nvdiffrast. The third method utilizes texrecon [Waechter *et al.*, 2014] to generate texture maps automatically. The first two methods allow setting the size of the texture map (set to 2024 × 2024 × 3 as default), while texrecon generates multiple texture maps that need to be merged manually. The results are shown in Tab. 5, we use the Xatlas as the default choice.

5 Conclusion

In this paper, we propose a hybrid representation that integrates textured meshes with 3D Gaussian Splatting (3DGS) to reduce the Gaussian primitives for indoor scenes. Based on the observation that texture-rich flat regions demand numerous Gaussian primitives to accurately capture significant color variations, we employ textured meshes to serve as background elements with specific depth values. This strategy substantially reduces the required number of Gaussians and accelerates rendering speeds. Additionally, we developed a mesh pruning method that considers normal maps, adjacent angles, and mesh sizes to eliminate meshes in geometrically intricate regions, thereby allocating space for Gaussian primitives. Our joint optimization technique ensures that both the textured mesh and Gaussian splats contribute effectively to the final representation, leveraging their respective strengths. Experimental results on the Deep Blending and ScanNet++ demonstrate that our hybrid approach maintains comparable rendering quality while significantly reducing the number of Gaussian primitives.

Limitations. While our method excels in scenes with prominently texture-rich flat regions, it is less effective in environments dominated by multiple intricate structures unless a high-quality mesh is provided. Furthermore, our current approach does not optimize vertex coordinates or account for view-dependent effects of meshes. Addressing these limitations will be the focus of our future research, aiming to extend the applicability and robustness of our hybrid representation across a wider range of complex scenes.

Acknowledgments

This work was supported in part by the Theme-based Research Scheme (TRS) project T45-701/22-R of the Research Grants Council (RGC), Hong Kong SAR.

References

[Bulò *et al.*, 2024] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in gaussian splatting. *CoRR*, abs/2404.06109, 2024.

[Chen *et al.*, 2023] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *CoRR*, abs/2312.00846, 2023.

[Chen *et al.*, 2024] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. PGSR: planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *CoRR*, abs/2406.06521, 2024.

[Choi *et al.*, 2024] Jaehoon Choi, Yonghan Lee, Hyungtae Lee, Heesung Kwon, and Dinesh Manocha. Meshgs: Adaptive mesh-aligned gaussian splatting for high-quality rendering, 2024.

[Dhamo *et al.*, 2024] Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Headgas: Real-time animatable head avatars via 3d gaussian splatting. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part II*, volume 15060 of *Lecture Notes in Computer Science*, pages 459–476. Springer, 2024.

[Fan *et al.*, 2023] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ FPS. *CoRR*, abs/2311.17245, 2023.

[Fan *et al.*, 2024] Lue Fan, Yuxue Yang, Minxing Li, Hongsheng Li, and Zhaoxiang Zhang. Trim 3d gaussian splatting for accurate geometry representation. *CoRR*, abs/2406.07499, 2024.

[Fang and Wang, 2024] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXVII*, volume 15135 of *Lecture Notes in Computer Science*, pages 165–181. Springer, 2024.

[Gao *et al.*, 2024] Lin Gao, Jie Yang, Bo-Tao Zhang, Jia-Mu Sun, Yu-Jie Yuan, Hongbo Fu, and Yu-Kun Lai. Mesh-based gaussian splatting for real-time large-scale deformation. *CoRR*, abs/2402.04796, 2024.

[Guédon and Lepetit, 2024] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 5354–5363. IEEE, 2024.

[Guo *et al.*, 2023] Yuan-Chen Guo, Yan-Pei Cao, Chen Wang, Yu He, Ying Shan, and Song-Hai Zhang. Vmesh: Hybrid volume-mesh representation for efficient view synthesis. In June Kim, Ming C. Lin, and Bernd Bickel, editors, *SIGGRAPH Asia 2023 Conference Papers, SA 2023, Sydney, NSW, Australia, December 12-15, 2023*, pages 17:1–17:11. ACM, 2023.

[Hedman *et al.*, 2018] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel J. Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.*, 37(6):257, 2018.

[Kerbl *et al.*, 2023] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023.

[Kim and Lim, 2024] Jiyeop Kim and Jongwoo Lim. Integrating meshes and 3d gaussians for indoor scene reconstruction with SAM mask guidance. *CoRR*, abs/2407.16173, 2024.

[Laine *et al.*, 2020] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020.

[Li *et al.*, 2024] Haoran Li, Haolin Shi, Wenli Zhang, Wenjun Wu, Yong Liao, Lin Wang, Lik-Hang Lee, and Pengyuan Zhou. Dreamscene: 3d gaussian-based text-to-3d scene generation via formation pattern sampling. *CoRR*, abs/2404.03575, 2024.

[Lin *et al.*, 2024] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 5166–5175. IEEE, 2024.

[Lin *et al.*, 2025] Jiaqi Lin, Zhihao Li, Binxiao Huang, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Xiaofei Wu, Fenglong Song, and Wenming Yang. Decoupling appearance variations with 3d consistent features in gaussian splatting. *arXiv preprint arXiv:2501.10788*, 2025.

[Liu *et al.*, 2024] Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. Humangaussian: Text-driven 3d human generation with gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 6646–6657. IEEE, 2024.

[Luo *et al.*, 2024] Haimin Luo, Min Ouyang, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. Gaussianhair: Hair modeling and rendering with light-aware gaussians. *CoRR*, abs/2402.10483, 2024.

[Mallick *et al.*, 2024] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Francisco Vicente Carrasco, Markus Steinberger, and Fernando De la Torre. Taming 3dgs: High-quality radiance fields with limited resources. *CoRR*, abs/2406.15643, 2024.

[Mildenhall *et al.*, 2020] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020.

[Papantonakis *et al.*, 2024] Panagiotis Papantonakis, Georgios Kopanas, Bernhard Kerbl, Alexandre Lanvin, and George Drettakis. Reducing the memory footprint of 3d gaussian splatting. *Proc. ACM Comput. Graph. Interact. Tech.*, 7(1):16:1–16:17, 2024.

[Qian *et al.*, 2024] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 20299–20309. IEEE, 2024.

[Shao *et al.*, 2024] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 1606–1616. IEEE, 2024.

[Straub *et al.*, 2019] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Yuheng Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard A. Newcombe. The replica dataset: A digital replica of indoor spaces. *CoRR*, abs/1906.05797, 2019.

[Svitov *et al.*, 2024] David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. HAHA: highly articulated gaussian human avatars with textured mesh prior. *CoRR*, abs/2404.01053, 2024.

[Turki *et al.*, 2024] Haithem Turki, Vasu Agrawal, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Deva Ramanan, Michael Zollhöfer, and Christian Richardt. Hybridnerf: Efficient neural rendering via adaptive volumetric surfaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 19647–19656. IEEE, 2024.

[Waczynska *et al.*, 2024] Joanna Waczynska, Piotr Borycki, Slawomir Konrad Tadeja, Jacek Tabor, and Przemyslaw Spurek. Games: Mesh-based adapting and modification of gaussian splatting. *CoRR*, abs/2402.01459, 2024.

[Waechter *et al.*, 2014] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! — Large-scale texturing of 3D reconstructions. In *Proceedings of the European Conference on Computer Vision*. Springer, 2014.

[Wen *et al.*, 2024] Jing Wen, Xiaoming Zhao, Zhongzheng Ren, Alexander G. Schwing, and Shenlong Wang. Gomavatar: Efficient animatable human modeling from monocular video using gaussians-on-mesh. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 2059–2069. IEEE, 2024.

[Wu and Tuytelaars, 2024] Minye Wu and Tinne Tuytelaars. Implicit gaussian splatting with efficient multi-level triplane representation. *CoRR*, abs/2408.10041, 2024.

[Wu *et al.*, 2024] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 20310–20320. IEEE, 2024.

[Yeshwanth *et al.*, 2023] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 12–22. IEEE, 2023.

[Yu *et al.*, 2024a] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. GSDF: 3dgs meets SDF for improved rendering and reconstruction. *CoRR*, abs/2403.16964, 2024.

[Yu *et al.*, 2024b] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 19447–19456. IEEE, 2024.

[Zhou *et al.*, 2024] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VI*, volume 15064 of *Lecture Notes in Computer Science*, pages 324–342. Springer, 2024.