

Self-supervised Learning of Hybrid Part-aware 3D Representations of 2D Gaussians and Superquadrics

Zhirui Gao Renjiao Yi[†] Yuhang Huang Wei Chen Chenyang Zhu Kai Xu[†]
National University of Defense Technology
[zhirui-gao.github.io/PartGS](https://github.com/zhirui-gao/PartGS)

Abstract

Low-level 3D representations, such as point clouds, meshes, NeRFs and 3D Gaussians, are commonly used for modeling 3D objects and scenes. However, cognitive studies indicate that human perception operates at higher levels and interprets 3D environments by decomposing them into meaningful structural parts, rather than low-level elements like points or voxels. Structured geometric decomposition enhances scene interpretability and facilitates downstream tasks requiring component-level manipulation. In this work, we introduce **PartGS**, a self-supervised **part**-aware reconstruction framework that integrates 2D Gaussians and superquadrics to parse objects and scenes into an interpretable decomposition, leveraging multi-view image inputs to uncover 3D structural information. Our method jointly optimizes superquadric meshes and Gaussians by coupling their parameters within a hybrid representation. On one hand, superquadrics enable the representation of a wide range of shape primitives, facilitating flexible and meaningful decompositions. On the other hand, 2D Gaussians capture detailed texture and geometric details, ensuring high-fidelity appearance and geometry reconstruction. Operating in a self-supervised manner, our approach demonstrates superior performance compared to state-of-the-art methods across extensive experiments on the DTU, ShapeNet, and real-world datasets.

1. Introduction

3D reconstruction from multi-view images is a long-standing challenge in 3D vision and graphics [12, 17, 55]. Most reconstructed scenes are in low-level representations such as point clouds, voxels, or meshes. However, humans tend to understand 3D scenes as reasonable parts [35]. For instance, when observing a scene, we naturally construct high-level structural information, such as scene graphs, instead of focusing on low-level details like point clouds or voxels. Motivated by it, we propose a part-aware reconstruction framework

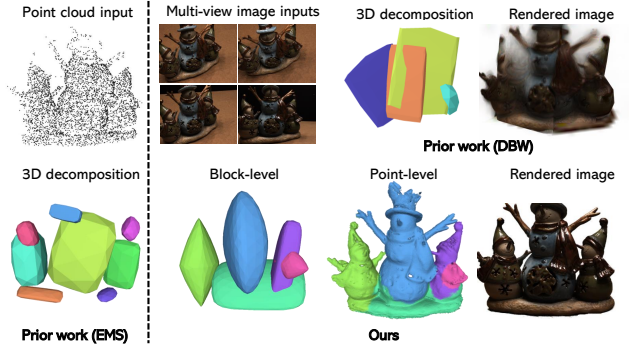


Figure 1. Prior works [32, 36] model the scene using primitives, while the proposed method can further model precise geometry details and textures. Here, EMS [32] takes point cloud inputs and reconstructs non-textured primitives.

that decomposes objects or scenes into meaningful shapes or parts, facilitating tasks such as physical simulation, editing, content generation and understanding [18, 28, 53, 54, 58].

Several prior works [29, 32, 33, 37, 56, 72] have explored part-aware reconstruction or 3D decomposition. However, most of them rely heavily on 3D supervision and often struggle to retain fine-grained geometric details, limiting their practicality in real-world scenarios. Recent advances [48, 68] have focused on extending the neural radiance field framework for part-aware reconstruction, building upon its remarkable success in 3D reconstruction from multi-view images. For example, PartNeRF [48] models objects using multiple neural radiance fields. Yet, the intricate composition of implicit fields complicates the learning process, leading to suboptimal rendering quality and inefficient decomposition. Recently, DBW [36] proposes a novel approach that decomposes scenes into block-based representations using superquadric primitives [2], optimizing both their parameters and UV texture maps through rendering loss minimization. While this method demonstrates effective scene decomposition into coherent components, it faces challenges in achieving accurate part-aware reconstruction of both geometry and appearance, as illustrated in Fig. 1.

[†] Corresponding authors

To address these limitations, we propose PartGS, a self-supervised part-aware hybrid representation that integrates 2D Gaussians [21] and superquadrics [2], to achieve both high-quality texture reconstruction and geometrically accurate part decomposition. There are previous approaches [14, 20, 50] combining mesh reconstruction and Gaussian splatting, where they first reconstruct the mesh and then attach Gaussians to it. In contrast to their primary focus on acquiring the representations, the proposed approach is to achieve part-aware decomposition reconstruction. Our method establishes a coupled optimization framework that simultaneously learns superquadric meshes and Gaussians through parameter sharing, with each part constrained to a single superquadric. The differentiable rendering of Gaussians drives this hybrid representation, leveraging the inherent convexity of superquadrics for qualitative 3D decomposition. The self-supervised training of part-aware reconstruction is built upon the assumption that each part of most objects should be a basic shape and can be represented by a superquadric. Meanwhile, Gaussian splatting, renowned for its superior rendering quality and efficient training, is incorporated to capture intricate texture details, speeding up reconstruction and improving rendering quality.

In the hybrid representation, 2D Gaussians are distributed on superquadric surfaces to form structured blocks. The pose and shape of Gaussians within each block are determined by their corresponding superquadrics rather than being optimized independently. The parameter space encompasses global controls for superquadric properties (shape, pose, and opacity) and local spherical harmonic coefficients for individual Gaussians. Compared to standard Gaussian splatting [21, 25], which populates the occupancy volume with independently parameterized Gaussians, the coupled representation is more compact and efficient.

During training, the parameters are optimized through rendering loss without additional supervision. However, we notice that image rendering loss often leads to local minima in superquadric shape optimization. To tackle this, we introduce several regularizers to maintain global consistency between the 3D representation and the input 2D information. This strategic implementation allows us to segment parts fully self-supervised, achieving block-level reconstruction. Finally, to better capture irregular shapes, we implement a point-level refinement step that frees 2D Gaussians to deviate from the surface, thereby enhancing geometric fidelity. Extensive experiments show that PartGS, at block-level and point-level, achieves 33.3% and 75.9% improvements in reconstruction accuracy, 3.18 and 16.13 increases in PSNR, and 4X and 3X speedups compared to the state-of-the-art baseline. Our contributions are summarized as follows:

- We introduce a novel hybrid representation for part-aware 3D reconstruction, combining the strengths of superquadrics and Gaussian splatting to achieve reasonable

decomposition and high-quality rendering.

- We propose several novel regularizers to enforce consistency between 3D decomposition and 2D observations, enabling self-supervised part decomposition.
- Compared to prior works, the method takes one step forward, by reconstructing both the block-level and point-level part-aware reconstructions, preserving both part segmentation and reconstruction precision.

2. Related Work

2.1. Shape Decomposition and Abstraction

Structured shape representation learning decomposes objects or scenes into coherent geometric primitives, facilitating shape understanding and generation [9, 15, 19, 47, 59]. Early approaches like Blocks World [43] and Generalized Cylinders [3] emphasized compact representations. Modern methods typically process 3D inputs (point clouds, meshes, voxels) by decomposing them into primitive ensembles, including cuboids [42, 49], superquadrics [32, 38, 39, 60], and convex shapes [8, 10]. For instance, MonteBoxFinder [42] integrates clustering [11, 51, 62], cuboid representations, and Monte Carlo Tree Search, while EMS [32] adopts a probabilistic approach for superquadric recovery. However, these methods are limited to coarse shape representations. Recent advances in shape abstraction [13, 22, 31, 40, 46] enable more detailed shape representations through flexible primitive deformation.

Some studies [1, 16, 36, 48, 68, 70] attempt to create structure-aware 3D representations directly from images. PartNeRF [48] introduces ellipsoid representations within NeRFs, its reliance on multiple implicit fields results in inefficient 3D decomposition. ISCO [1] and DBW [36] use 3D superquadrics for shape decomposition, which enables more meaningful structural separation. However, their simple shape parameters struggle to capture complex geometries, leading to poor geometry and appearance reconstruction. DPA-Net [68] has advanced 3D shape abstraction from sparse views but generates redundant parts and struggles with realistic texture rendering. A concurrent work, GaussianBlock [24], employs SAM [27] to guide superquadric splitting and fusion for 3D decomposition, yet its computational efficiency remains limited, typically requiring several hours per scene. In contrast, our approach accomplishes self-supervised [52, 61] part-aware scene reconstruction through an efficient hybrid representation that simultaneously maintains geometric fidelity and photorealistic rendering quality.

2.2. Mesh-based Gaussian Splatting

Gaussian splatting (GS) [25] has been rapidly adopted in multiple fields due to its remarkable rendering capability. Several studies [7, 14, 20, 50] attempt to align Gaussians with mesh surfaces for easier editing and animation. Among

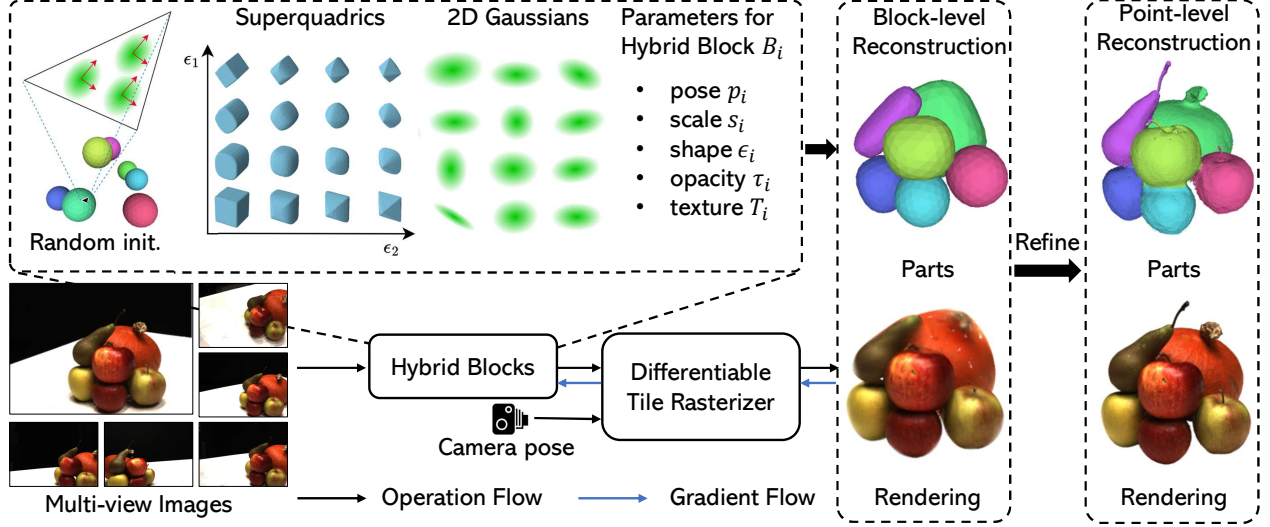


Figure 2. **Overview of our pipeline.** PartGS takes multi-view images to learn a parametric hybrid representation of superquadrics and 2D Gaussians. It initializes from random superquadrics and is gradually optimized during training to obtain a block-level reconstruction. Then, we free the constraints of Gaussians to model detailed geometry, achieving point-level reconstruction.

them, SuGaR [20] uses flat 3D Gaussians to enforce the alignment with the scene surface during optimization, minimizing the difference between the signed distance functions (SDF) of the desired Gaussians and actual Gaussians. GaMeS [50] introduces a hybrid representation of Gaussians and mesh, where Gaussians are attached to triangular facets of the mesh. Similarly, Gao *et al.* [14] proposes a mesh-based GS to achieve large-scale deformation effects on objects. Recently, 2DGS [21] proposes 2D Gaussians for surface modeling and significantly enhances the geometric quality. Inspired by them, we propose a representation that combines 2D Gaussians with superquadrics. A key distinction between our method and previous mesh-based GS approaches is that these methods require first reconstructing the scene’s mesh and then binding Gaussians to the mesh surface, which results in a non-continuous mesh. In contrast, our method directly optimizes the mesh through a rendering loss, enabling part-aware mesh reconstruction.

3. Method

Given a set of calibrated multi-view images and foreground masks, we aim to learn part-aware 3D representations, a meaningful decomposition of both geometry and appearance. Our approach adopts a two-stage optimization strategy: first, decomposing the object into basic shapes using a mixture of Gaussians and superquadrics at the block-level, followed by refining the decomposition at the point-level to achieve precise geometry. In Sec 3.1, we parameterize the hybrid representation. Sec. 3.2 then elaborates on leveraging this representation, enhanced by novel regularizers and an adaptive control strategy, to achieve self-supervised block-level

decomposition. Finally, Sec. 3.3 presents the process for obtaining detailed part-aware results.

3.1. Parametrizing the Hybrid Representation

As shown in Fig. 2, to leverage the strengths of both, we attach Gaussians to the surface of superquadric meshes. This representation retains the superquadric’s ability to parse a 3D scene into distinct parts. Meanwhile, spherical harmonics of Gaussians enable complex texture rendering via Gaussian splatting, addressing the texture learning limitations of prior work [32, 36, 42, 67]. Sharing pose parameters between superquadrics and 2D Gaussians further improves the representation’s efficiency.

The parametric representation is controlled by both primitive and Gaussian parameters, which are optimized simultaneously. Given a 3D scene $\mathcal{S}$, the proposed method decomposes it into multiple hybrid blocks, each consisting of a superquadric with associated Gaussians. Each scene is denoted by the hybrid representation: $\mathcal{S} = \mathcal{B}_1 \cup \dots \mathcal{B}_i \dots \cup \mathcal{B}_M$, where $\mathcal{B}_i$ is the i -th hybrid block, and M is the total number of blocks. Blocks are defined using manually designed parameters that control pose, opacity, scale, shape, and texture. These parameters are optimized via differentiable rendering to parse the target scene.

Shape and Scale Parameters. For each hybrid block $\mathcal{B}_i$, its geometry is controlled by superquadric parameters [2]. Specifically, there are two shape parameters ϵ_1, ϵ_2 that define its shape, along with three scale parameters s_1, s_2, s_3 , which scale the 3D axes. Analogous to icosphere vertex positioning, superquadric vertex coordinates are computed by:

$$\mathbf{v} = [s_1 \cos^{\epsilon_1}(\theta) \cos^{\epsilon_2}(\varphi); s_2 \sin^{\epsilon_1}(\theta); s_3 \cos^{\epsilon_1}(\theta) \sin^{\epsilon_2}(\varphi)], \quad (1)$$

where θ and φ represent the azimuth and elevation defined

in the spherical coordinate. The shape and scale parameters govern block deformation to learn part-aware geometry.

Pose Parameters. The pose of the i -th hybrid block is defined by its rotation $\mathbf{R}_i$ and translation $\mathbf{t}_i$. The vertex position from Eq. 1 is transformed from the local coordinate to world space as: $\hat{\mathbf{v}}_i^j = \mathbf{R}_i \mathbf{v}_i^j + \mathbf{t}_i$, where j indexes the vertices. Previous approaches, including SuGaR [20] and GaMeS [50], position Gaussians on reconstructed meshes with independent pose and shape parameters to enhance appearance modeling. In contrast, our method employs Gaussians to construct a differentiable rendering that bridges images and superquadrics. To achieve this, we couple the parameters of Gaussians with superquadrics.

For a posed superquadric, Gaussian centers are uniformly sampled on triangular faces, with their poses determined by face vertices. Following GaMeS [50], each Gaussian’s rotation matrix $\mathbf{R}_v = [r_1, r_2, r_3]$ and scaling $\mathbf{S}_v$ are computed from vertex positions. Given a triangular face V with vertices $v_1, v_2, v_3 \in \mathbb{R}^3$, orthonormal vectors are constructed such that r_1 aligns with the face normal, r_2 points from the centroid $m = \text{mean}(v_1, v_2, v_3)$ to v_1 . r_3 is obtained by orthonormalizing the vector from the center to the second vertex concerning the existing r_1 and r_2 :

$$r_3 = \frac{\text{ort}(v_2 - m; r_1, r_2)}{\|\text{ort}(v_2 - m; r_1, r_2)\|}, \quad (2)$$

where ort represents the orthonormalization in Gram-Schmidt [4]. For the scaling of 2D Gaussians, we use:

$$\mathbf{S}_v = \text{diag}(s_v^2, s_v^3), \quad (3)$$

where $s_v^2 = c\|m - v_1\|$, $s_v^3 = c < v_2, r_3 >$, and c is a size control hyperparameter. We place a fixed number of Gaussians on each triangular face. These designs eliminate the need to learn the geometry of Gaussians, thereby enhancing the efficiency of the representation.

Opacity Parameters. We define the total number of hybrid blocks as M . However, a typical scene does not contain exactly M blocks. Therefore, only the meaningful blocks are retained. To achieve this, we introduce a learnable parameter τ_i to represent each block’s opacity. During the optimization, only blocks with τ_i greater than a certain threshold are retained. Note that Gaussians within the same block share the same τ for opacity in rasterization.

Texture Parameters. The texture is modeled using 2D Gaussians positioned on the surface of the superquadrics. Spherical harmonics of each Gaussian control the texture and are optimized for rendering view-dependent images [25].

3.2. Block-level Decomposition

This section describes the optimization of the hybrid representation. We observed that minimizing the rendering loss across multi-view images alone led to instability in positioning hybrid blocks. Therefore, several regularization terms

are introduced to optimize the composition for maximal image formation consistency.

3.2.1. Optimization

In addition to the image rendering loss, we employ regularizers consisting of coverage, overlap, parsimony, and opacity entropy. The coverage loss encourages hybrid blocks to cover only meaningful regions; the overlap loss prevents blocks from overlapping; the parsimony loss regularizes the number of existing blocks; and the opacity entropy encourages block opacities close to binary.

Rendering Loss. The rendering loss is based on 3DGS [25], where it combines an L_1 term with a D-SSIM term:

$$\mathcal{L}_{\text{ren}} = (1 - \lambda) L_1 + \lambda L_{\text{D-SSIM}}. \quad (4)$$

Coverage Loss. The coverage loss ensures that the block set $\{\mathcal{B}_1 \dots \mathcal{B}_M\}$ covers the object while preventing it from extending beyond its boundaries. To determine the 3D occupancy field based on the blocks, we first define the approximate signed distance of a 3D point x to the i -th block, $D_i(x) = \Psi_i(x) - 1$. Here, Ψ_i denotes the superquadric inside-outside function [2] associated with the block i . Consequently, $D_i(x) \leq 0$ if the point x lies inside the i -th block, and $D_i(x) > 0$ if x is outside the block. Inspired by NeRF [34], we sample a ray set $\mathcal{R}$ using ray-casting based on camera poses. Given the object mask, we associate each ray $r \in \mathcal{R}$ with a binary label: $l_r = 1$ if the ray r is inside mask, otherwise $l_r = 0$. The coverage loss is defined as:

$$\mathcal{L}_{\text{cov}}(\mathcal{R}) = \sum_{r \in \mathcal{R}} l_r L_{\text{cross}}(r) + (1 - l_r) L_{\text{non-cross}}(r). \quad (5)$$

Here, L_{cross} encourages rays to intersect with blocks, while $L_{\text{non-cross}}$ discourages rays from intersecting with blocks:

$$L_{\text{cross}}(r) = \text{ReLU}\left(\min_{i \in \mathcal{M}} \min_{x_j^r \in \mathcal{X}_r} D_i(x_j^r)\right), \quad (6)$$

$$L_{\text{non-cross}}(r) = \text{ReLU}\left(\max_{i \in \mathcal{M}} \max_{x_j^r \in \mathcal{X}_r} -D_i(x_j^r)\right), \quad (7)$$

where x_j^r means the j -th sampled point along the ray r and $\mathcal{M} = \{1, \dots, M\}$. Intuitively, this implies that at least one sampled points $\mathcal{X}_r$ along the ray r inside the mask that lies within a certain block, while all points along the ray outside the mask should not belong to any block.

Overlap Loss. We introduce a regularization term to prevent overlap between individual blocks. Given the difficulty of directly calculating block overlap, we adopt a Monte Carlo method similar to [36]. Specifically, we sample multiple 3D points in space and penalize those that lie inside more than k blocks. Based on the superquadric inside-outside function, a soft occupancy function is defined as:

$$\mathcal{O}_i^x = \tau_i \text{sigmoid}\left(\frac{-D_i(x)}{\gamma}\right), \quad (8)$$

where γ is a temperature parameter. Thus, for a set of N sampled 3D points Ω , the overlap loss is expressed as:

$$\mathcal{L}_{\text{over}} = \frac{1}{N} \sum_{x \in \Omega} \text{ReLU} \left(\sum_{i \in \mathcal{M}} \mathcal{O}_i^x - k \right). \quad (9)$$

Parsimony Loss. To promote the use of the minimal number of blocks and achieve parsimony in decomposition, we introduce a regularization term that penalizes block opacity (τ). This loss is defined as:

$$\mathcal{L}_{\text{par}} = \frac{1}{M} \sum_{i \in \mathcal{M}} \sqrt{\tau_i}. \quad (10)$$

Opacity Entropy Loss. During optimization, the opacity of the block inside the object region tends to approach 1, while the opacity of the block outside the object region tends to approach 0. To facilitate this, we associate the opacities of blocks with the labeled rays and apply a cross-entropy loss L_{ce} between the block opacity and the mask labels, defined as follows:

$$\mathcal{L}_{\text{opa}}(\mathcal{R}) = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} L_{ce} \left(\max_{i \in \mathcal{M}} \tau_i(x^r), l_r \right). \quad (11)$$

Here, only points x^r inside the blocks are sampled, *e.g.*, $\{x^r \in \mathcal{X}_r \mid D_i(x^r) \leq 0\}$.

The total loss is the weighted sum of the loss terms described above:

$$\mathcal{L} = \mathcal{L}_{\text{ren}} + \lambda_{\text{cov}} \mathcal{L}_{\text{cov}} + \lambda_{\text{over}} \mathcal{L}_{\text{over}} + \lambda_{\text{par}} \mathcal{L}_{\text{par}} + \lambda_{\text{opa}} \mathcal{L}_{\text{opa}}. \quad (12)$$

3.2.2. Adaptive Number of Blocks.

Given that the number of parts may vary across different scenes, we allow the number of blocks to adjust during optimization adaptively. Specifically, when the opacity of an existing block falls below a threshold t , this block is removed immediately. Furthermore, we explore a block-adding mechanism to dynamically incorporate new components, ensuring comprehensive coverage of target objects. Specifically, DB-SCAN [45] is employed to cluster initial point clouds that are not covered by any blocks, where the point clouds are either randomly initialized or derived as by-products from COLMAP [44]. For each point cloud cluster, a new block is introduced at its center, with the remaining parameters of the block initialized randomly.

The final block-level reconstruction generates a textured 3D scene represented by multiple superquadrics, with surface details rendered through Gaussians, as illustrated in Fig. 2.

3.3. Point-level Decomposition

The primitive-based hybrid representation effectively decomposes the shape into parts but exhibits suboptimally for

complex objects. To address this limitation, we perform a refinement stage that enhances geometric fitting. In this stage, the constraint between Gaussians and superquadrics is decoupled, enabling independent Gaussian optimization. Furthermore, to prevent Gaussian components from one block passing through adjacent blocks and disrupting the continuity and plausibility of the part decomposition, we impose a new constraint to minimize the negative signed distance of each Gaussian entering other blocks:

$$\mathcal{L}_{\text{enter}} = \frac{1}{N} \sum_{x \in \Omega} \sum_{m \in \mathcal{M} \setminus \{\delta\}} \text{ReLU}(-D_m(x)), \quad (13)$$

where Ω is the sampled Gaussian set, and δ represents the identifier of the block to which the Gaussian x belongs. As seen in Fig. 2, the reconstruction becomes more accurate and aligns better with the target shape.

4. Experiments

We comprehensively evaluate our approach across four key aspects: 3D reconstruction quality, view synthesis performance, shape parsimony, and computational efficiency, comparing it with state-of-the-art methods in the field.

4.1. Datasets

Extensive experiments are conducted on two widely-used public datasets: DTU [23] and ShapeNet [6]. Specifically, 15 standard scenes from the DTU dataset that are widely adopted for reconstruction quality assessment are used for evaluation. Each scene contains either 49 or 64 images. Additionally, we construct a ShapeNet subset consisting of four categories: *Chair*, *Table*, *Gun*, and *Airplane*, with 15 objects selected per category. For each object, 50 rendered images are generated for training. We further validate real-world applicability on both the BlendedMVS dataset [63] and self-captured scenes, demonstrating robust performance across synthetic and natural environments.

4.2. Implementation Details

We build upon the 2DGS [21] architecture and add a non-learnable part attribute to each Gaussian component to enforce the constraint in Eq. 13. In the ShapeNet benchmark, we observe a common degradation in mesh reconstruction as shown in Fig. 7. To address this, we propose Gaussian scale regularization, as validated in Sec. 4.4. We set the initial number of primitives M to 8. The temperature parameter γ in Eq. 8 is 0.005 and the overlapping number k in Eq. 9 is 1.95. We train the hybrid representation for 30k iterations, followed by refinement optimization for another 30k iterations. During block-level optimization, the block-adding operation is carried out at the 5k-th and 10k-th iterations. The loss weights λ_{cov} , λ_{over} , λ_{par} , and λ_{opa} are set to 10, 1, 0.002, and 0.01, respectively. For more details about datasets, metrics, and baselines, please refer to *Supplementary*.

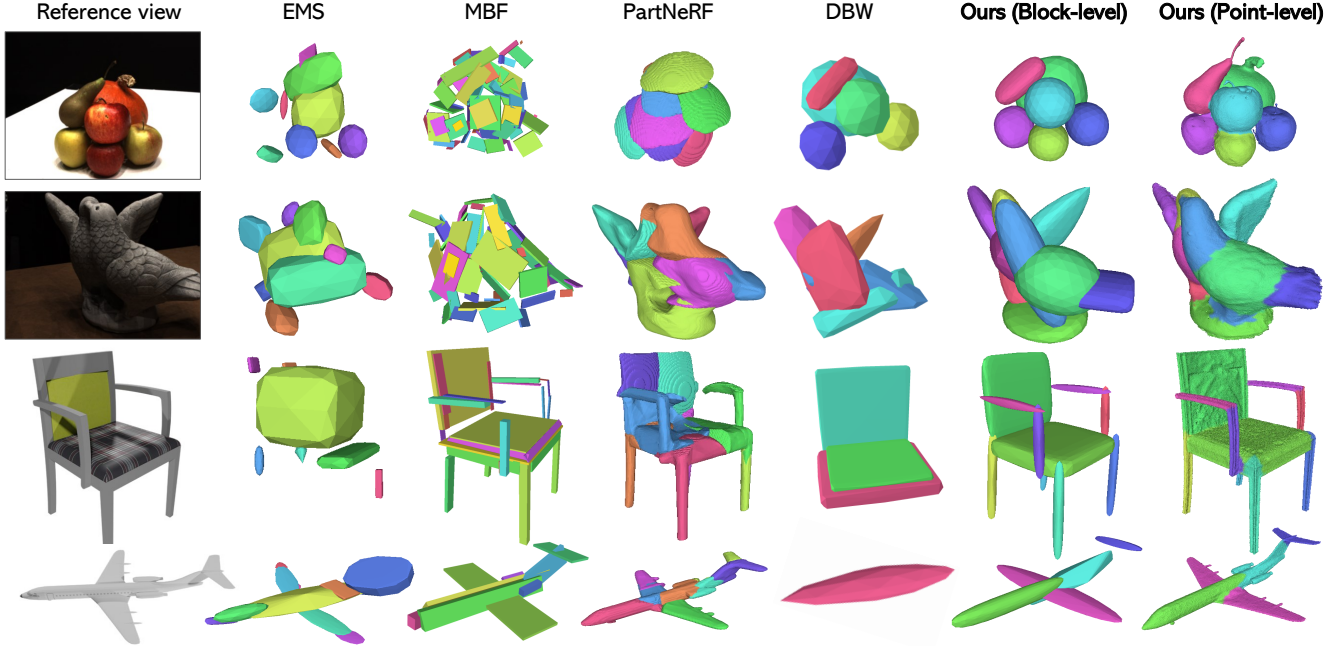


Figure 3. **Qualitative comparison on DTU [23] and ShapeNet [23]**. The first two rows are DTU examples, and the last two are ShapeNet examples, respectively. Our method is the only one that provides reasonable 3D part decomposition while capturing detailed geometry.

Method	Input	Renderable	Chamfer distance per scene															Mean	Mean
			S24	S37	S40	S55	S63	S65	S69	S83	S97	S105	S106	S110	S114	S118	S122	CD	#P
EMS [32]	3D GT	✗	6.32	3.54	2.99	4.30	4.16	4.01	3.75	3.24	4.97	4.34	4.16	7.62	7.58	4.46	4.03	4.65	7.7
MBF [42]	3D GT	✗	3.12	2.66	3.84	2.54	1.59	2.11	2.19	2.01	2.32	2.45	2.17	2.12	3.83	2.02	2.55	2.50	<u>34.1</u>
EMS [32] + Neus [55]	Image	✗	5.99	5.56	4.43	4.32	5.42	6.14	3.75	3.96	4.63	4.34	5.88	5.11	4.29	4.83	3.53	4.97	8.87
MBF [42] + Neus [55]	Image	✗	2.69	3.37	3.22	2.69	3.63	2.60	2.59	3.13	2.85	2.51	2.45	3.72	2.24	2.49	2.52	2.85	<u>46.7</u>
PartNeRF [48]	Image	✓	9.38	10.46	9.08	8.63	6.04	7.25	7.22	9.15	8.72	10.01	6.72	9.85	7.85	8.68	9.21	8.54	8.0
DBW [36]	Image	✓	5.41	8.35	1.57	3.08	3.40	4.15	7.46	3.94	6.63	4.85	4.38	4.65	6.29	4.34	3.04	4.76	4.8
Ours (Block-level)	Image	✓	5.68	4.91	1.85	2.61	3.75	4.66	3.75	7.57	4.27	4.38	3.49	4.48	3.61	4.21	3.70	4.19	5.9
Ours (Point-level)	Image	✓	0.70	1.17	0.55	0.65	1.06	1.23	1.10	1.36	1.37	0.78	0.92	1.41	0.69	1.05	0.71	0.98	5.9

Table 1. **Quantitative comparison on DTU [23]**. The Chamfer distance between the 3D reconstruction and the ground-truth is reported in 15 scenes. The best results are bolded, and the average numbers of primitives found (#P) that are greater than 10 are underlined.

Method	Part-aware	CD ↓	PSNR ↑	SSIM ↑	LIPPS ↓	Time ↓
Neuralangelo [30]	✗	0.61	33.84	-	-	> 10 h
2DGS [21]	✗	0.81	34.07	0.99	0.019	~ 10 m
PartNeRF [48]	✓	9.59	17.97	0.77	0.246	~ 8 h
DBW [36]	✓	4.73	16.44	0.75	0.201	~ 2 h
Ours (Block-level)	✓	<u>4.19</u>	<u>19.84</u>	<u>0.82</u>	<u>0.189</u>	~ 30 m
Ours (Point-level)	✓	0.98	35.04	0.99	0.015	~ 40 m

Table 2. **Quantitative results on DTU [23]**. Our method outperforms all part-aware approaches in image synthesis quality, reconstruction accuracy, and efficiency. Neuralangelo’s results are from the original paper, with all times measured on an RTX 3090 GPU.

Method	Input	Chamfer Distance ↓				Primitives (#P)				Mean	Mean
		Airplane	Table	Chair	Gun	Airplane	Table	Chair	Gun	CD	#P
EMS [32]	3D GT	3.40	6.92	19.0	2.02	9.4	7.88	10.3	8.4	7.84	9.0
MBF [42]	3D GT	2.83	2.18	<u>1.59</u>	2.32	10.85	13.9	13.4	14.3	2.21	13.1
PartNeRF [48]	Image	<u>2.29</u>	2.77	2.30	2.46	8.0	8.0	8.0	8.0	2.46	8.0
DBW [36]	Image	3.61	7.33	6.19	2.09	2.7	5.2	3.6	3.3	4.81	3.7
Ours (Block-level)	Image	2.47	<u>2.15</u>	2.32	<u>1.78</u>	3.9	6.6	7.6	5.0	<u>2.18</u>	5.8
Ours (Point-level)	Image	1.29	1.72	0.94	1.07	3.9	6.6	7.6	5.0	1.25	5.8

Table 3. **Quantitative comparison on ShapeNet [6]**. We report Chamfer distance and the number of parts. The best results are bolded, and the second-best results are underlined.

4.3. Evaluations

4.3.1. DTU and Shapenet Benchmark

Geometry Reconstruction. In Tab. 1 and Tab. 3, we compare our geometry reconstruction to SOTA shape decomposition methods on Chamfer distance and training time using DTU and Shapenet dataset. The Chamfer distance

metrics reported for ShapeNet are scaled by a factor of 100 for readability. Our method consistently outperforms in all scenes compared to prior works. As shown in Fig. 3, our approach consistently produces interpretable 3D decompositions with further refinement achieving more detailed geometry (the last two columns). MBF [42] achieves a low CD error but at the cost of using significantly more prim-

Method	Part-aware	Chamfer Distance ↓				PSNR ↑				SSIM ↑				LIPPS ↓			
		Airplane	Table	Chair	Gun	Airplane	Table	Chair	Gun	Airplane	Table	Chair	Gun	Airplane	Table	Chair	Gun
2DGS [21]	✗	1.47	2.37	0.50	1.03	40.89	39.80	39.05	41.74	0.994	0.990	0.990	0.994	0.009	0.027	0.017	0.009
PartNeRF [48]	✓	<u>2.29</u>	2.77	<u>2.30</u>	2.46	19.63	20.66	19.08	21.97	0.898	0.855	0.875	0.916	0.086	0.161	0.136	0.083
DBW [36]	✓	3.61	7.33	6.19	2.09	26.11	23.84	20.25	28.72	0.950	0.915	0.892	0.960	0.074	0.136	0.132	<u>0.042</u>
Ours (Block-level)	✓	2.47	<u>2.15</u>	2.32	<u>1.78</u>	<u>27.94</u>	<u>27.98</u>	<u>24.92</u>	<u>29.95</u>	<u>0.959</u>	<u>0.925</u>	<u>0.906</u>	<u>0.963</u>	<u>0.072</u>	<u>0.129</u>	<u>0.122</u>	0.047
Ours (Point-level)	✓	1.29	1.72	0.94	1.07	41.18	36.80	36.07	39.51	0.992	0.973	0.977	0.989	0.014	0.070	0.038	0.021

Table 4. **Quantitative results on ShapeNet [6].** We report the Chamfer distance and novel view synthesis results across four categories.



Figure 4. **Qualitative results on BlendedMVS [63] and self-captured data.** We present RGB renderings and decomposed parts from novel views. The top examples are from the BlendedMVS dataset, and the last example is from our captured scenes.

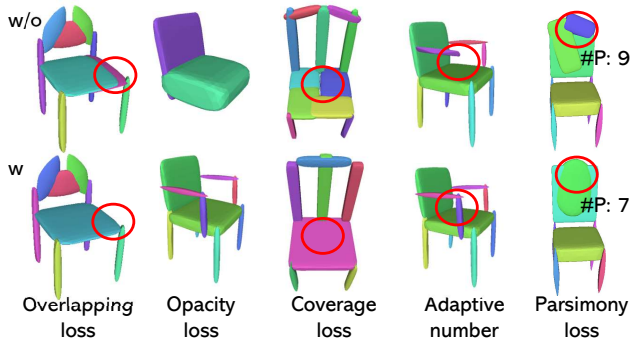


Figure 5. **Ablation studies on key strategies.** The block-level visual comparisons illustrate the impact of adopting our proposed strategy. The first row shows results without the strategy, and the second with the strategy implemented.

itives, leading to over-decomposition and the inclusion of meaningless parts. Moreover, as shown in Tab. 2 and Tab. 4, our approach achieves competitive results with the advanced Gaussian-based 2DGS [21] and surface reconstruction Neuralangelo [30], which are limited to producing unstructured meshes. Notably, our model demonstrates excellent efficiency, with a reconstruction speed approximately 10X faster than PartNeRF [48] and over 3X faster than DBW [36].

Appearance Reconstruction. In addition to part decomposition, our method enables high-fidelity image synthesis, attributed to integrating Gaussian splatting within hybrid

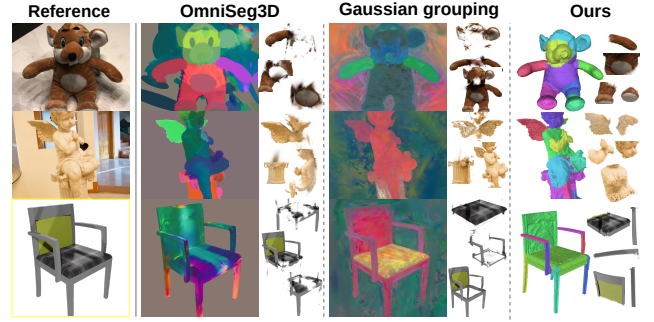


Figure 6. **Qualitative comparison with SAM-based methods.** Visual results demonstrate that our method produces more structurally coherent decompositions, whereas SAM-based approaches frequently exhibit spatial discontinuities.

Method	Block-level				Point-level				#P
	CD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	CD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	
Complete model	2.32	24.92	0.906	0.122	<u>0.94</u>	36.07	0.977	0.038	6.6
w/o L_{over}	<u>2.11</u>	25.17	<u>0.914</u>	<u>0.118</u>	1.34	36.04	<u>0.977</u>	<u>0.037</u>	7.6
w/o L_{opa}	5.72	21.12	0.853	0.174	1.06	35.60	0.975	0.040	3.8
w/o L_{cov}	3.21	22.96	0.896	0.134	1.34	36.00	0.976	0.038	8.53
w/o Adaptive	3.16	22.74	0.880	0.141	1.05	35.74	0.974	0.040	6.7
w/o L_{par}	1.91	25.48	0.918	0.115	0.93	36.18	0.977	0.037	10.1

Table 5. **Ablation studies on the ShapeNet [6].** We report Chamfer Distance, rendering metrics, and the number of primitives (#P).

representations. EMS [32] and MBF [42] operate directly on point clouds and consequently lack image rendering capabilities, while PartNeRF [48] and DBW [36] yield low-quality view synthesis. In contrast, our method achieves high-quality appearance rendering results, as demonstrated in Tab. 2 and Tab. 4.

4.3.2. Real-life Data

To further demonstrate the applicability of our method for learning shape decomposition, we test our model on the real-life images from the BlendedMVS dataset [63] and a self-captured dataset. As shown in Fig. 12, our approach can robustly produce both realistic appearances and reasonable 3D decompositions across a variety of data types. More results are provided in the supplementary material.

4.3.3. Compared to SAM-based Methods

We also conduct comparisons to SAM-based [27] methods for this task, as shown in Fig. 6. Despite impressive advances in 3D segmentation and editing [5, 26, 65, 66] by

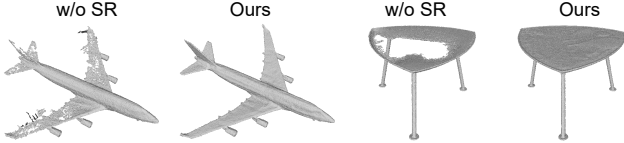


Figure 7. **Impact of Gaussian scale regularization (SR).** The degraded mesh produced by 2DGS [21] is effectively improved.

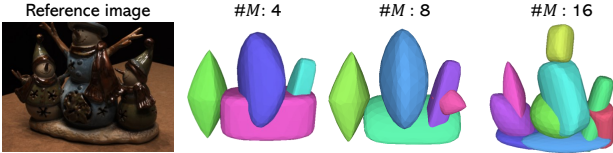


Figure 8. **Impact of the initial number of primitives (M).** Increasing M yields a finer-grained decomposition, while decreasing it produces a coarser decomposition.

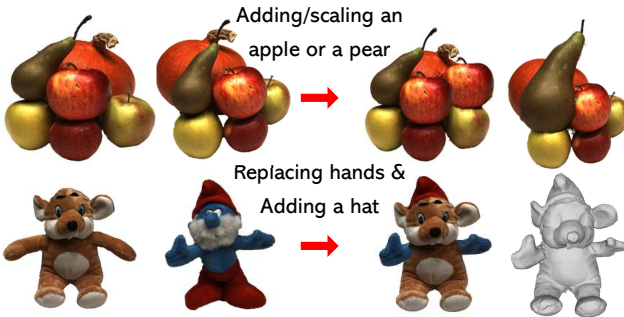


Figure 9. **Applications. Part-Aware Editing (top):** After optimization, we can easily edit the scene by adding, scaling, or moving specific parts. **3D Content Generation (bottom):** By combining parts of different objects, we can create new 3D content.

Method	Block-level				Point-level				#P
	CD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	CD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	
$\lambda_{\text{par}} = 0.1$	6.46	16.10	0.721	0.255	1.08	34.30	0.987	0.018	2.7
$\lambda_{\text{par}} = 0.01$ (*)	4.19	19.84	0.820	0.189	1.05	35.04	0.988	0.015	5.9
$\lambda_{\text{par}} = 0.001$	4.01	20.18	0.835	0.180	1.03	35.04	0.988	0.015	7.1
$M = 4$	4.97	19.06	0.794	0.213	1.06	34.93	0.987	0.016	5.0
$M = 8$ (*)	4.19	19.84	0.820	0.189	1.05	35.04	0.988	0.015	5.9
$M = 16$	3.99	20.87	0.83	0.176	1.07	35.05	0.988	0.015	8.3

Table 6. **Effect of parsimony weight (λ_{par}) and initial primitives count (M) on DTU [23].** We report Chamfer distance, rendering metrics, and primitives count (#P). * denotes the default setting.

distilling 2D information, achieving semantic disentanglement but remains challenging due to indistinct textures and severe cross-view 2D inconsistencies. Lifting 2D segmentation features to 3D segmentation will also lead to feature misalignment, leading to visual artifacts. In contrast, our method performs direct 3D space segmentation, yielding more reasonable and structurally coherent decompositions.

4.4. Ablations

In this section, we first analyze the design choices by isolating each strategy to assess its impact. We report the averaged

quantitative performance on 15 instances of the chair category in Tab. 5 and provide visual comparisons in Fig. 5. Removing overlapping loss provides better reconstruction accuracy but more overlapping parts. The opacity loss drastically affects the number of primitives and doubles the CD, highlighting its role in controlling primitive presence and reconstruction quality. Coverage loss is crucial in reconstruction accuracy, ensuring primitives align correctly with the target. The adaptive primitive strategy fills in missing parts and improves the reconstruction metric. Without parsimony loss, CD and rendering metrics are optimal but result in over-decomposition. Note that the full model did not yield the highest accuracy because the part-aware reconstruction trades off the accuracy and rationality of the part decomposition. The objective is to achieve high accuracy while preserving reasonable parts.

We also validate the effectiveness of Gaussian scale regularization in Fig. 7. 2DGS [21] tends to use large-scale Gaussians in texture-less areas and create holes in the constructed mesh by TSDF integration [71]. The presence of holes stems from the inability of large-scale Gaussians to maintain view-consistent depth. Scale regularization effectively addresses this issue by suppressing large-scale Gaussians.

Lastly, in Tab. 6, we analyze the influence of two key hyperparameters on decomposition granularity: the parsimony loss weight λ_{par} and the initial primitive count M . Stronger parsimony regularization reduces the number of primitives, while weaker regularization increases it. As M rises, both reconstruction and view synthesis performance slightly improve. This demonstrates that adjusting M and λ_{par} enables our method to effectively control the granularity of object or scene decomposition. Fig. 8 visually illustrates this impact.

4.5. Applications

Fig. 9 illustrates two applications of our method that the original 3DGS and NeRF-based approaches do not support. Firstly, after optimization, we obtain the part decomposition, facilitating easy editing of specific objects or scene components, e.g., adding, moving, removing, or scaling. Secondly, by combining parts of different objects, our method enables the creation of new high-quality 3D content.

5. Conclusions

We introduce PartGS, a hybrid representation of superquadrics and 2D Gaussians, to learn 3D scenes in a part-aware representation. Compared to prior works, the proposed method retains geometry details, supporting high-quality image rendering. It obtains state-of-the-art performance in comprehensive evaluations. One limitation is that it takes background-free scene images as inputs, leveraging segmentation tools such as SAM [27]. In the future, we aim to explore how to model backgrounds and extend to larger and more complex scenes.

Acknowledgements

This work is supported in part by the NSFC (62325211, 62132021, 62372457), the Major Program of Xiangjiang Laboratory (23XJ01009), Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), the Natural Science Foundation of Hunan Province of China (2022RC1104).

References

- [1] Stephan Alaniz, Massimiliano Mancini, and Zeynep Akata. Iterative superquadric recomposition of 3d objects from multiple views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18013–18023, 2023. 2
- [2] Alan H. Barr. Superquadrics and Angle-Preserving Transformations. *IEEE Computer Graphics and Applications*, 1981. 1, 2, 3, 4
- [3] Thomas Binford. Visual Perception by Computer. In *IEEE Conference on Systems and Control*, 1971. 2
- [4] Åke Björck. Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications*, 197:297–316, 1994. 4
- [5] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. *arXiv preprint arXiv:2312.00860*, 2023. 7
- [6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv:1512.03012 [cs.CV]*, 2015. 5, 6, 7, 1
- [7] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023. 2
- [8] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [9] Zhiqin Chen, Qimin Chen, Hang Zhou, and Hao Zhang. Dae-net: Deforming auto-encoder for fine-grained shape co-segmentation. *arXiv preprint arXiv:2311.13125*, 2023. 2
- [10] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [11] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 1981. 2
- [12] Huachen Gao, Shihe Shen, Zhe Zhang, Kaiqiang Xiong, Rui Peng, Zhirui Gao, Qi Wang, Yugui Xie, and Ronggang Wang. Fdc-nerf: learning pose-free neural radiance fields with flow-depth consistency. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3615–3619. IEEE, 2024. 1
- [13] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 2
- [14] 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. *arXiv preprint arXiv:2402.04796*, 2024. 2, 3
- [15] Zhirui Gao, Renjiao Yi, Zheng Qin, Yunfan Ye, Chenyang Zhu, and Kai Xu. Learning accurate template matching with differentiable coarse-to-fine correspondence refinement. *Computational Visual Media*, 10(2):309–330, 2024. 2
- [16] Zhirui Gao, Renjiao Yi, Yaqiao Dai, Xuening Zhu, Wei Chen, Chenyang Zhu, and Kai Xu. Curve-aware gaussian splatting for 3d parametric curve reconstruction, 2025. 2
- [17] Zhirui Gao, Renjiao Yi, Chenyang Zhu, Ke Zhuang, Wei Chen, and Kai Xu. Generic objects as pose probes for few-shot view synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 1
- [18] Zeqi Gu, Yin Cui, Zhaoshuo Li, Fangyin Wei, Yunhao Ge, Jinwei Gu, Ming-Yu Liu, Abe Davis, and Yifan Ding. Artiscene: Language-driven artistic 3d scene generation through image intermediary. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 2891–2901, 2025. 1
- [19] Yanran Guan, Han Liu, Kun Liu, Kangxue Yin, Ruizhen Hu, Oliver van Kaick, Yan Zhang, Ersin Yumer, Nathan Carr, Radomir Mech, et al. Fame: 3d shape generation via functionality-aware model evolution. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1758–1772, 2020. 2
- [20] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *arXiv preprint arXiv:2311.12775*, 2023. 2, 3, 4
- [21] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. *SIGGRAPH*, 2024. 2, 3, 5, 6, 7, 8, 1, 4
- [22] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural template: Topology-aware reconstruction and disentangled generation of 3d meshes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18572–18582, 2022. 2
- [23] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanaes. Large Scale Multi-view Stereopsis Evaluation. In *CVPR*, 2014. 5, 6, 8, 1, 2, 3
- [24] Shuyi Jiang, Qihao Zhao, Hossein Rahmani, De Wen Soh, Jun Liu, and Na Zhao. Gaussianblock: Building part-aware compositional and editable 3d scene by primitives and gaussians. *arXiv preprint arXiv:2410.01535*, 2024. 2
- [25] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 2, 4
- [26] Chung Min* Kim, Mingxuan* Wu, Justin* Kerr, Matthew Tancik, Ken Goldberg, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 7

- [27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2, 7, 8, 1
- [28] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 1
- [29] Lingxiao Li, Minhuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. Supervised Fitting of Geometric Primitives to 3D Point Clouds. In *CVPR*, 2019. 1
- [30] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 6, 7, 1
- [31] Di Liu, Xiang Yu, Meng Ye, Qilong Zhangli, Zhuowei Li, Zhixing Zhang, and Dimitris N Metaxas. Deformer: Integrating transformers with deformable models for 3d shape abstraction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14236–14246, 2023. 2
- [32] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Robust and Accurate Superquadric Recovery: a Probabilistic Approach. In *CVPR*, 2022. 1, 2, 3, 6, 7
- [33] Romain Loiseau, Elliot Vincent, Mathieu Aubry, and Loic Landrieu. Learnable Earth Parser: Discovering 3D Prototypes in Aerial Scans. *arXiv:2304.09704 [cs.CV]*, 2023. 1
- [34] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020. 4
- [35] Niloy Mitra, Michael Wand, Hao (Richard) Zhang, Daniel Cohen-Or, Vladimir Kim, and Qi-Xing Huang. Structure-aware shape processing. In *SIGGRAPH Asia 2013 Courses*, 2013. 1
- [36] Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei A. Efros, and Mathieu Aubry. Differentiable Blocks World: Qualitative 3D Decomposition by Rendering Primitives. In *NeurIPS*, 2023. 1, 2, 3, 4, 6, 7
- [37] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single rgb image. *Cornell University - arXiv, Cornell University - arXiv*, 2018. 1
- [38] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics Revisited: Learning 3D Shape Parsing Beyond Cuboids. In *CVPR*, 2019. 2
- [39] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning Unsupervised Hierarchical Part Decomposition of 3D Objects from a Single RGB Image. In *CVPR*, 2020. 2
- [40] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3204–3215, 2021. 2
- [41] Thomas Porter and Tom Duff. Compositing Digital Images. In *SIGGRAPH*, 1984. 4
- [42] Michaël Ramamonjisoa, Sinisa Stekovic, and Vincent Lepetit. MonteBoxFinder: Detecting and Filtering Primitives to Fit a Noisy Point Cloud. In *ECCV*, 2022. 2, 3, 6, 7, 1
- [43] Lawrence G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 2
- [44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 1
- [45] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DbSCAN revisited, revisited: why and how you should (still) use dbSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017. 5
- [46] Qingyao Shuai, Chi Zhang, Kaizhi Yang, and Xuejin Chen. Dpf-net: combining explicit shape priors in deformable primitive field for unsupervised structural reconstruction of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14321–14329, 2023. 2
- [47] Chunyu Sun, Yuqi Yang, Haoxiang Guo, Pengshuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Semi-supervised 3d shape segmentation with multilevel consistency and part substitution. *Computational Visual Media*, 2022. 2
- [48] Konstantinos Tertikas, Despoina Paschalidou, Boxiao Pan, Jeong Joon Park, Mikaela Angelina Uy, Ioannis Emiris, Yanis Avrithis, and Leonidas Guibas. PartNeRF: Generating Part-Aware Editable 3D Shapes without 3D Supervision. In *CVPR*, 2023. 1, 2, 6, 7
- [49] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning Shape Abstractions by Assembling Volumetric Primitives. In *CVPR*, 2017. 2
- [50] Joanna Waczyńska, Piotr Borycki, Sławomir Tadeja, Jacek Tabor, and Przemysław Spurek. Games: Mesh-based adapting and modification of gaussian splatting. *arXiv preprint arXiv:2402.01459*, 2024. 2, 3, 4
- [51] Xinhang Wan, Jiyuan Liu, Xinbiao Gan, Xinwang Liu, Siwei Wang, Yi Wen, Tianjiao Wan, and En Zhu. One-step multi-view clustering with diverse representation. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2024. 2
- [52] Xinhang Wan, Jiyuan Liu, Hao Yu, Qian Qu, Ao Li, Xinwang Liu, Ke Liang, Zhibin Dong, and En Zhu. Contrastive continual multiview clustering with filtered structural fusion. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 2
- [53] Fengxiang Wang, Mingshuo Chen, Yueying Li, Di Wang, Haotian Wang, Zonghao Guo, Zefan Wang, Boqi Shan, Long Lan, Yulin Wang, et al. Geollava-8k: Scaling remote-sensing multimodal large language models to 8k resolution. *arXiv preprint arXiv:2505.21375*, 2025. 1
- [54] Fengxiang Wang, Hongzhen Wang, Zonghao Guo, Di Wang, Yulin Wang, Mingshuo Chen, Qiang Ma, Long Lan, Wenjing Yang, Jing Zhang, et al. Xlrs-bench: Could your multimodal llms understand extremely large ultra-high-resolution remote sensing imagery? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14325–14336, 2025. 1
- [55] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit

- Surfaces by Volume Rendering for Multi-view Reconstruction. In *NeurIPS*, 2021. [1](#), [6](#)
- [56] Xiaogang Wang, Yuelang Xu, Kai Xu, Andrea Tagliasacchi, Bin Zhou, Ali Mahdavi-Amiri, and Hao Zhang. Pie-net: Parametric inference of point cloud edges. *Advances in neural information processing systems*, 33:20167–20178, 2020. [1](#)
- [57] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. [1](#)
- [58] Zhifeng Wang, Renjiao Yi, Xin Wen, Chenyang Zhu, and Kai Xu. Vastsd: Learning 3d vascular tree-state space diffusion model for angiography synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 15693–15702, 2025. [1](#)
- [59] Tong Wu, Lin Gao, Ling-Xiao Zhang, Yu-Kun Lai, and Hao Zhang. Star-tm: Structure aware reconstruction of textured mesh from single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2021. [2](#)
- [60] Yuwei Wu, Weixiao Liu, Sipu Ruan, and Gregory S. Chirikjian. Primitive-based Shape Abstraction via Nonparametric Bayesian Inference. In *ECCV*, 2022. [2](#)
- [61] Xihong Yang, Xiaochang Hu, Sihang Zhou, Xinwang Liu, and En Zhu. Interpolation-based contrastive learning for few-label semi-supervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(2):2054–2065, 2022. [2](#)
- [62] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, pages 10834–10842, 2023. [2](#)
- [63] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A Large-scale Dataset for Generalized Multi-view Stereo Networks. In *CVPR*, 2020. [5](#), [7](#), [1](#), [4](#)
- [64] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *NeurIPS*, 2020. [1](#)
- [65] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024. [7](#)
- [66] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. *arXiv preprint arXiv:2311.11666*, 2023. [7](#)
- [67] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9491–9500, 2019. [3](#)
- [68] Fenggen Yu, Yimin Qian, Xu Zhang, Francisca Gil-Ureta, Brian Jackson, Eric Bennett, and Hao Zhang. Dpa-net: Structured 3d abstraction from sparse views via differentiable primitive assembly. *arXiv preprint arXiv:2404.00875*, 2024. [1](#), [2](#)
- [69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. [1](#)
- [70] Hongyi Zhou, Xiaogang Wang, Yulan Guo, and Kai Xu. Monomobility: Zero-shot 3d mobility analysis from monocular videos, 2025. [2](#)
- [71] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. [8](#), [3](#)
- [72] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J Guibas, and Hao Zhang. Adacoseg: Adaptive shape co-segmentation with group consistency loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8543–8552, 2020. [1](#)

Self-supervised Learning of Hybrid Part-aware 3D Representations of 2D Gaussians and Superquadrics

Supplementary Material

At first, this supplementary material provides detailed experimental settings, including data processing procedures, implementation details for comparison baseline methods, and evaluation metrics (Sec.6). Subsequently, we provide further implementation specifics, including the training configurations and rendering process (Sec.8).

In Fig. 10 and Fig. 11, we provide additional visual comparisons of our method against state-of-the-art baselines on DTU [23] and ShapeNet [6] datasets. Moreover, we include visual results on several scenes from the BlendedMVS [63] and self-captured data in Fig. 12. It can be observed that the decomposed reconstructions by our method are more reasonable and precise, which is believed to be beneficial for downstream tasks. Our code and data are available at <https://github.com/zhirui-gao/PartGS>.

6. More Details on Experiments

6.1. Datasets

We conduct evaluations on two public datasets: DTU [23] and ShapeNet [6]. DTU is a multi-view stereo (MVS) dataset comprising 80 forward-facing scenes, each captured with 49 to 64 images. In experiments, we use 15 publicly recognized scenes commonly adopted in previous studies [21, 30]. We downsample the images to a resolution of 400×300 for computational efficiency.

To validate the effectiveness of our method in the decomposition of man-made objects, we construct a subset of the ShapeNet dataset comprising four categories: *Chair*, *Table*, *Gun*, and *Airplane*. Each category includes 15 distinct objects, providing diverse instances. For each object, we randomly sample 100 camera poses on a sphere and render images at a resolution of 400×400 . The rendered images are equally split into training and testing sets.

Additionally, to explore the potential of PartGS in handling real-life data, we present qualitative results on the BlendedMVS dataset [63] and self-captured scenes. For BlendedMVS, we use official camera poses, while for self-captured scenes, camera poses are estimated by COLMAP [44] and normalized using IDR [64]. All images are resized to 400×300 pixels. For the DTU, ShapeNet, and BlendedMVS datasets, we utilize the ground-truth foreground masks provided within the datasets. For self-captured real-world scenes, we employ the Segment Anything Model (SAM) [27] to segment foreground objects.

6.2. Implementation Details on Baselines

We compare our method with four state-of-the-art works on 3D shape decomposition: EMS [32], MonteboxFinder (MBF) [42], PartNeRF [48] and DBW [36]. MBF and EMS are applied to point clouds, utilizing cuboids and superquadrics, respectively, to fit the 3D points. The input point clouds are sampled from either ground truth 3D shapes (GT) or meshes reconstructed by state-of-the-art MVS method Neus [55]. We adhere to the testing procedure outlined by DBW [36] and randomly sample 5K and 200K points from the GT meshes as the input for EMS and MBF, respectively. Similar to our problem setting, PartNeRF and DBW use images as input to construct structural 3D representations. To ensure fair comparisons, we discard the pre-defined ground plane in DBW, as the test scenario consists exclusively of foreground objects. For PartNeRF, we perform instance-specific training and set the same number of parts as in our method, while retaining all other default parameter settings. Additionally, we compare with SOTA Gaussian Splatting reconstruction method 2DGS [21], and surface reconstruction approach Neuralangelo [30], to provide an intuitive evaluation of the reconstruction quality achieved by our method.

6.3. Evaluation Metrics

We evaluate from 3D reconstruction quality, view synthesis, shape parsimony, and reconstruction time.

- The 3D reconstruction quality is measured by the official Chamfer distance (CD) evaluation [23] between the recovered geometry and GT, reflecting the accuracy of 3D reconstructions.
- View synthesis uses three standard metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [57], and Learned Perceptual Image Patch Similarity (LPIPS) [69].
- Shape parsimony is quantified by the number of parts, while reconstruction time refers to the running time measured on the same device. While shape parsimony does not directly indicate the quality of the decomposition, a smaller number of decomposed components, given consistent reconstruction quality, suggests a more concise and reasonable decomposition. This is because fitting different parts of an object with fewer blocks is inherently more challenging, whereas utilizing more blocks simplifies the fitting process.

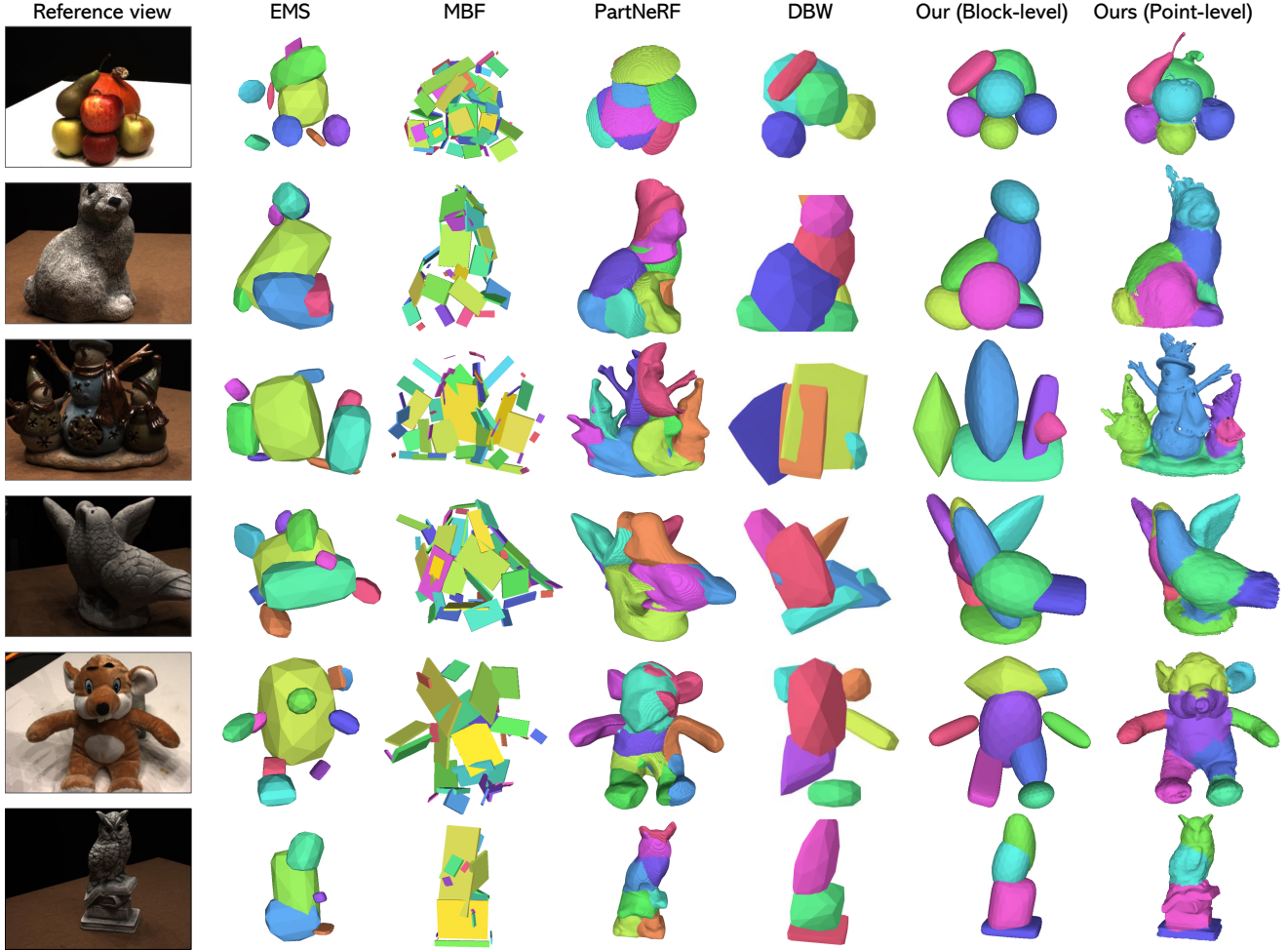


Figure 10. **Qualitative comparisons on DTU [23].** We compare our approach with state-of-the-art baselines on the DTU dataset with the background removed. The last two columns show our block-level and point-level reconstructions, respectively. Our method is the only one that provides reasonable 3D part decomposition while capturing detailed geometry.

7. Additional Comparative Experiments

Very recently, DPA-Net [68] and GaussianBlock [24] have achieved advanced 3D shape abstraction. DPA-Net enables part-aware reconstruction in a feedforward manner, requiring approximately 3 days of GPU training on a pre-collected dataset and 2 hours per object for inference. GaussianBlock utilizes SAM [27] to guide superquadric splitting and fusion for 3D decomposition, with a processing time of 6 hours per object. We compare our method on DTU and BlendedMVS scenes, with results presented in Fig. 13. The proposed approach achieves comparable performance while demonstrating superior efficiency. It is important to clarify the supervision requirements: DPA-Net is a supervised method that relies on SAM-generated segmentation masks for training, whereas GaussianBlock, while not strictly 3D supervised, depends on high-quality pre-trained datasets that

typically require days of training. Our method distinguishes itself by being completely self-supervised, requiring neither 3D supervision nor pre-trained components, yet achieving significantly better computational efficiency.

8. Implementation Details

8.1. Training Configurations

To ensure uniform distributions of 2D Gaussians across surfaces of the mesh, random barycentric coordinates are generated directly within each triangular face. Specifically, barycentric weights are computed as $u = \sqrt{rand}$ and $v = rand$. These weights are transformed to obtain $\alpha = [1 - u, u(1 - v), uv]$, $rand \in (0, 1)$. The sampled position on triangular face is calculated as $o = \alpha_0 v_0 + \alpha_1 v_1 + \alpha_2 v_2$, where v_0, v_1, v_2 are vertices of the triangle. This method compensates for the non-uniformity caused by the triangle's

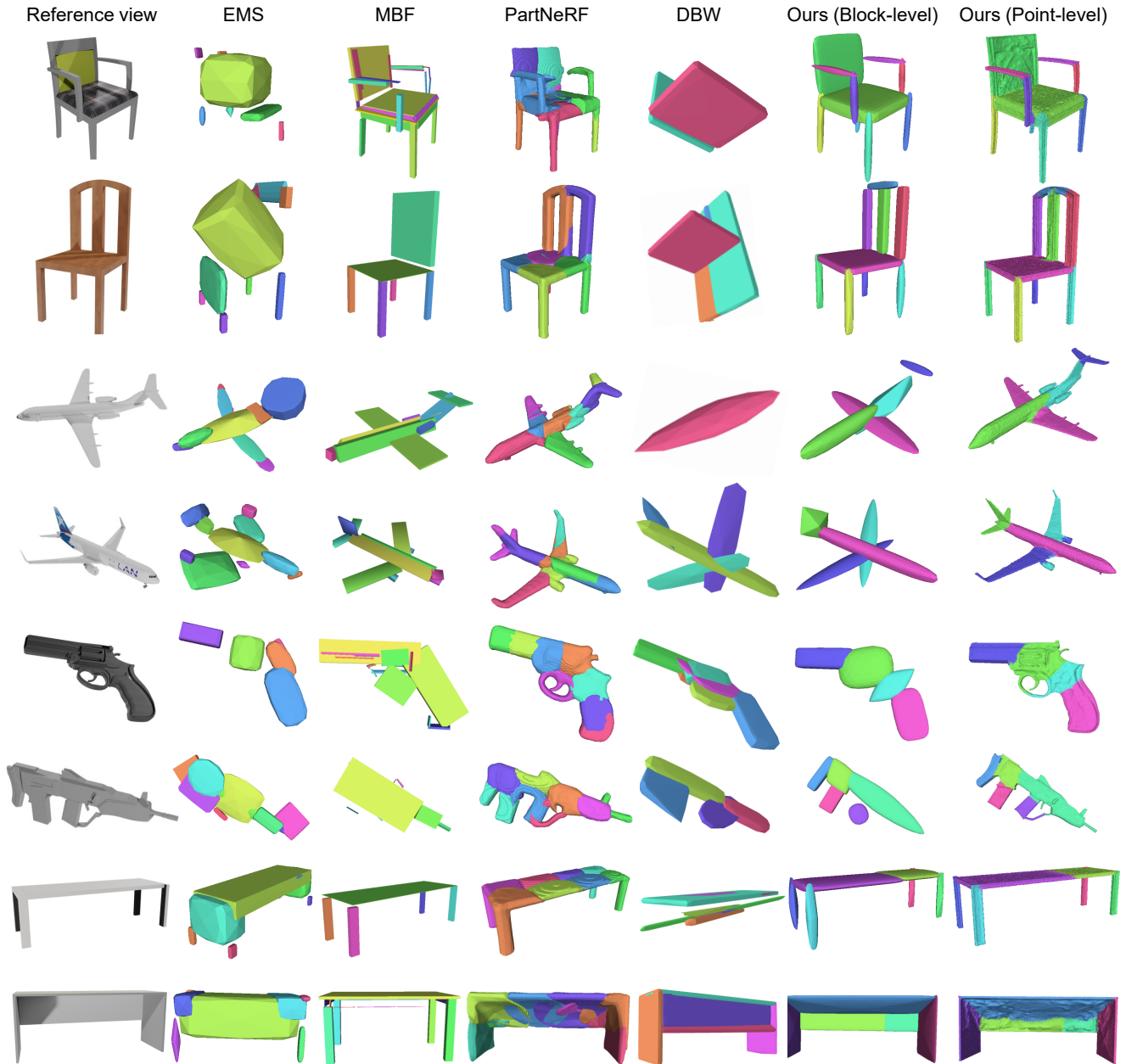


Figure 11. **Qualitative comparisons on ShapeNet [23].** We compared our approach with state-of-the-art baselines across four categories. The last two columns display our block-level and point-level reconstructions, respectively. Our method uniquely provides reasonable 3D part decomposition while simultaneously capturing detailed geometry.

geometry. Unlike naive random sampling which results in uneven distributions, it ensures that the sampled points are evenly distributed across mesh surfaces.

The same hyperparameters are used for all experiments. We set the initial number of primitives M to 8. In the hybrid representation, each superquadric mesh is a level-2 icosphere (320 triangular face). Each triangular face contains 100 Gaussians, with a scaling parameter c of 0.1. The number of

sampled points in each ray is 2048.

During refinement, we employ regularization in 2DGS to achieve better geometric reconstruction, including depth distortion maps, depth maps, and normal maps. Additionally, we introduce a mask cross-entropy to filter out extra noise Gaussians. To extract the meshes from 2D Gaussians, we use truncated signed distance fusion (TSDF) to fuse rendered depth maps, utilizing Open3D [71].

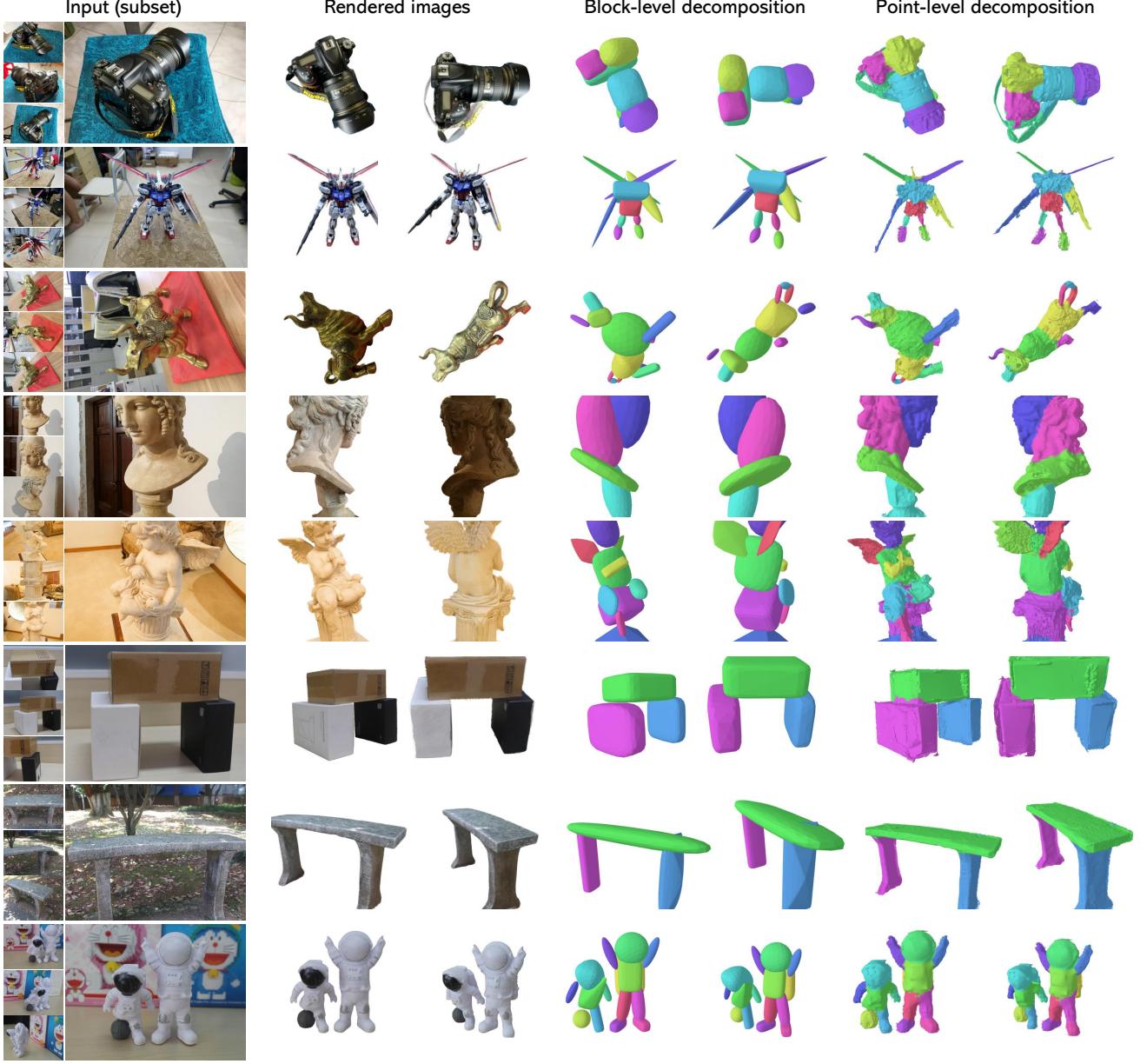


Figure 12. **Qualitative results on BlendedMVS [63] and self-captured data.** We demonstrate the RGB renderings and decomposed parts from novel views across a variety of objects. The first five examples are from the BlendedMVS dataset, and the remaining examples are from our own captured scenes.

8.2. Rendering

With Gaussians attached to the surface of each block, we achieve view-dependent rendering through tile-based rasterization as Gaussian Splatting [21, 25]. Given a view, Gaussians on the superquadric surface are projected onto the image space, forming an RGB image. Initially, the screen space determines the bounding box for each Gaussian. Subsequently, these Gaussian ellipses are sorted according to their depths of center to the image plane. Finally, volu-

metric alpha compositing [41] is utilized to integrate the alpha-weighted RGB values for each pixel.

To formulate the process, considering a viewing transformation W , the covariance matrix Σ'_i of i -th Gaussian in the camera coordinate system is calculated by:

$$\Sigma'_i = JW\Sigma W^T J^T, \quad (14)$$

where J is the Jacobian of the affine approximation of the projective transformation, and Σ is the covariance matrix

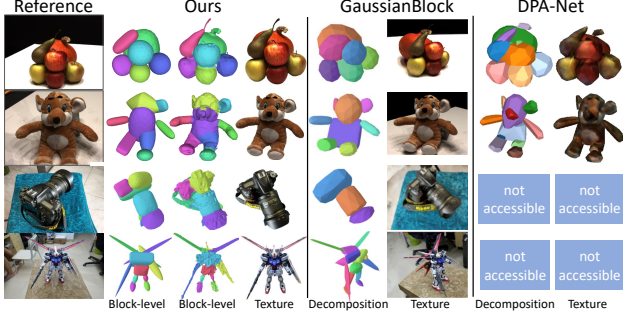


Figure 13. **Qualitative comparisons to DPA-Net and Gaussian-Block.** The first two examples are from the DTU dataset, and the last two examples are from the BlendedMVS dataset.

of the Gaussian ellipse. Note that the last row and column of Σ are omitted since we adopt 2D Gaussians. Following alpha compositing, we first calculate an alpha value for each Gaussian ellipse:

$$\alpha_i = \tau_i \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{u}_i)^T \Sigma'_{i-1} (\mathbf{x} - \mathbf{u}_i)\right). \quad (15)$$

Here, $\mathbf{u}_i$ is the center coordinate of the projected Gaussian ellipse, and τ_i is the opacity of the block where the i -th Gaussian is located. The calculated alphas are sorted according to their depths from the image plane. Meanwhile, we can acquire the color value c_i from the spherical harmonics of $\mathcal{N}$ ordered points, thus obtaining the rendering RGB value:

$$C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (16)$$

This process is differentiable and can optimize the hybrid representation through gradient descent.