

HeteroArch-GS: Aerial-Ground Mesh-guided Gaussian Splatting for Heterogeneous Architectural Landmarks with a Real-World Dataset

Junfan Wang^a, Han Hu^{a,*}, Zhihao Jia^a, Yang Jia^b, Bo Xiang^b, Jiwei Deng^c and Qing Zhu^a

^aFaculty of Geosciences and Engineering, Southwest Jiaotong University, Chengdu, 611756, Sichuan, China

^bSichuan Highway Planning, Survey, Design and Research Institute Ltd., Chengdu, 610041, Sichuan, China

^cChina Railway Design Corporation, Tianjin, 300308, China

ARTICLE INFO

Keywords:

Heterogeneous Architectural Landmarks

3D Gaussian Splatting

Aerial-Ground Rendering

Oblique Photogrammetric Mesh

ABSTRACT

Supplementing aerial photogrammetric models with ground-level imagery to recover fine detail has become standard practice for large-scale landmark reconstruction. Yet aerial-ground fusion under 3D Gaussian Splatting (3DGS) remains fundamentally challenged. Extreme viewpoint and scale disparities, exacerbated by the irregular topologies of heterogeneous architectures, provoke severe optimization conflicts that naive joint training cannot reconcile, yielding geometric collapse and pervasive blur.

In this paper, we propose HeteroArch-GS, a mesh-guided framework that uses oblique photogrammetric meshes as geometric priors to guide 3DGS optimization for aerial-ground reconstruction of heterogeneous landmarks. Specifically, we convert the aerial mesh into continuous differentiable geometric fields that jointly regularize the position, orientation, and anisotropic shape of Gaussian primitives throughout optimization. Together with mesh-guided anchor initialization and multi-directional pseudo-view supervision, these priors drive the Gaussian distribution toward surface-aligned representations, ensuring high-fidelity rendering across both aerial and ground viewpoints.

To support this, we construct AGC Landmarks, a novel real-world RGB dataset capturing diverse heterogeneous landmarks with aerial, ground, and object-centric perspectives. Extensive experiments on AGC Landmarks demonstrate that HeteroArch-GS matches the prior state-of-the-art on in-distribution rendering while clearly surpassing it on out-of-distribution object-centric views. Geometrically, our method achieves the highest F1-scores at 0.01 inlier threshold across all scenes. These results confirm that mesh-guided priors produce more compact and geometrically faithful reconstructions.

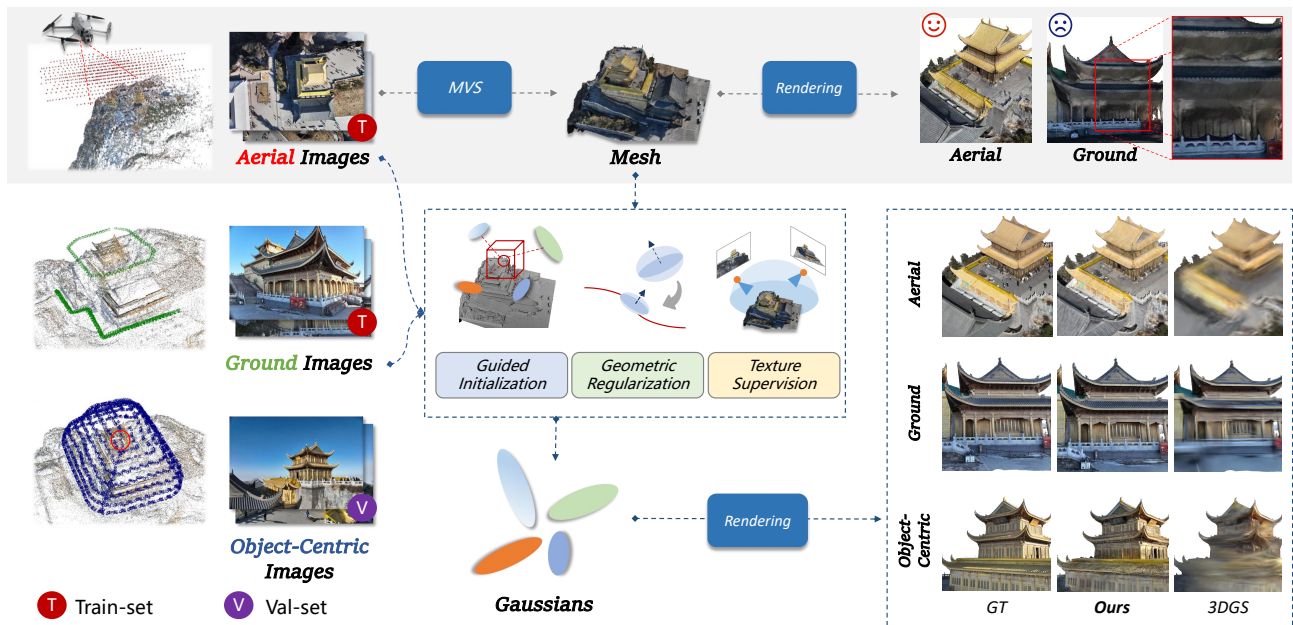


Fig. 1. City-scale meshes derived from aerial oblique photogrammetry are now widely available, yet landmark facades often lack the geometric and textural detail required for realistic ground-level rendering. HeteroArch-GS uses the existing aerial mesh to provide geometric and appearance priors, stabilizing aerial-ground 3D Gaussian Splatting optimization under severe cross-view conflicts. The resulting compact, surface-aligned representation recovers fine facade details and improves rendering fidelity across aerial, ground, and out-of-distribution object-centric views.

*Corresponding author

✉ wangjunfan@my.swjtu.edu.cn (J. Wang); han.hu@swjtu.edu.cn (H.

Hu)

1. Introduction

City-scale aerial oblique photogrammetry has already produced large-scale textured mesh reconstructions for major cities worldwide, as exemplified by platforms and initiatives such as Google Earth (Google, 2026) and China's National 3D Mapping Program (Chen et al., 2025b). These city-level assets are typically generated from aerial oblique images through Structure-from-Motion (SfM) (Schönberger and Frahm, 2016; Pan et al., 2024; Xu et al., 2025), Multi-View Stereo (MVS) (Schönberger et al., 2016; Zhu et al., 2026), and texture mapping (Wachter et al., 2014) for real-world urban modeling (Hu and Minner, 2023; Sefercik et al., 2025; Hu et al., 2026). However, landmark buildings demand higher fidelity than ordinary urban blocks, while aerial-derived meshes often represent them poorly. Lower facades, overhanging roofs, hollow components, and intricate decorative structures are frequently under-observed, distorted, or blurred, especially from ground-level viewpoints. Supplementing existing aerial mesh assets with close-range ground imagery is therefore a natural way to recover missing facade details and improve visual realism (Zhu et al., 2020; Chen et al., 2025c). Recent neural rendering methods (Xu et al., 2024; Shi et al., 2025), especially 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023), further shift this problem from mesh refinement toward photorealistic aerial-ground rendering in practical reality-capture workflows (Stamnes, 2026), opening the possibility of 3DGS-based reconstruction for heterogeneous architectural landmarks.

However, aerial-ground 3DGS is not a simple matter of adding more images (Zhang et al., 2024, 2025; Jiang et al., 2025). As shown in Fig. 2, aerial-only training yields faithful aerial renderings, and ground-only training preserves ground-level details. Yet when the two view groups are optimized together, both in-distribution domains can degrade. The reason is a severe cross-view conflict (Zhang et al., 2025; Jiang et al., 2025): aerial and ground images see the same structure at different scales, angles, occlusions, and visibility ranges. Heterogeneous landmarks make this conflict sharper, with overhangs, hollow spaces, thin components, and non-Lambertian materials producing ambiguous or inconsistent photometric gradients. Without explicit geometric constraints, standard 3DGS has no reliable mechanism to overcome these competing desires.

Furthermore, current benchmarks for aerial-ground Gaussian splatting still rely heavily on synthetic cities such as MatrixCity (Li et al., 2023). Synthetic rendering can improve visual realism, but aerial and street-level views are not governed by a single illumination regime: aerial images are shaped by wide-area sun-sky lighting and atmospheric effects, whereas street views are dominated by local occlusions, facade shadows, interreflections, and near-field material responses (Song and Qin, 2024; Kaleta et al., 2025). This gap becomes more critical in practical workflows: ground images are often supplementary captures taken after the aerial survey, so the observed object is recorded under a different temporal phase and illumination condition rather than under the same radiometric state as the simulated data.

The reconstruction target is also not an average urban block, but often a landmark building whose irregular geometry, curved structures (Fang et al., 2025), and complex materials directly challenge unconstrained 3DGS representations. As a result, real-world captures introduce not only harder geometry, but also noise, occlusion, and dynamic disturbances that synthetic benchmarks largely hide, leaving a substantial gap between benchmark performance and deployable aerial-ground reconstruction (Jiang et al., 2025).

Motivated by these challenges, we propose HeteroArch-GS for a practical aerial-ground reconstruction workflow: city-scale aerial surveys have already produced photogrammetric meshes, yet the landmark facades that matter most remain visually incomplete from the ground, making supplementary street-level capture essential. Instead of treating aerial and ground images as an unconstrained joint training set, HeteroArch-GS uses the existing aerial mesh as a geometric and textural prior to stabilize cross-view 3DGS optimization. To move beyond synthetic air-ground benchmarks, we further collect real landmark buildings across temples, historical architecture, and modern structures, with additional object-centric views dedicated to evaluating novel-view synthesis in real scenes.

The main contributions of this paper are summarized as follows:

- We construct *AGC Landmarks*, a real-world optical image dataset for heterogeneous architectural landmarks, explicitly covering aerial, ground, and object-centric perspectives. Beyond benchmarking neural rendering on complex topologies, it exposes real illumination changes across unconstrained outdoor captures and provides a concrete basis for studying lighting models in the wild.
- We propose *HeteroArch-GS*, a mesh-guided framework for aerial-ground 3DGS. Building on an existing 3DGS backbone, it turns the existing oblique photogrammetric mesh into a strong geometric and textural prior through anchor initialization, surface-aware regularization, and pseudo-view supervision. This keeps Gaussian primitives tied to plausible physical surfaces and substantially improves rendering from out-of-distribution object-centric views.
- We introduce an efficient lazy-loading strategy that alleviates the memory bottleneck of large aerial and ground image collections, enabling joint aerial-ground 3DGS optimization under limited computational resources.

2. Related Work

2.1. Multi-View Image Datasets

High-quality multi-view imagery is essential for accurate 3D reconstruction and photorealistic rendering. Since this work targets outdoor architectural landmarks, we compare representative datasets in Table 1 from three aspects: reconstruction level, data source, and viewpoint configuration.

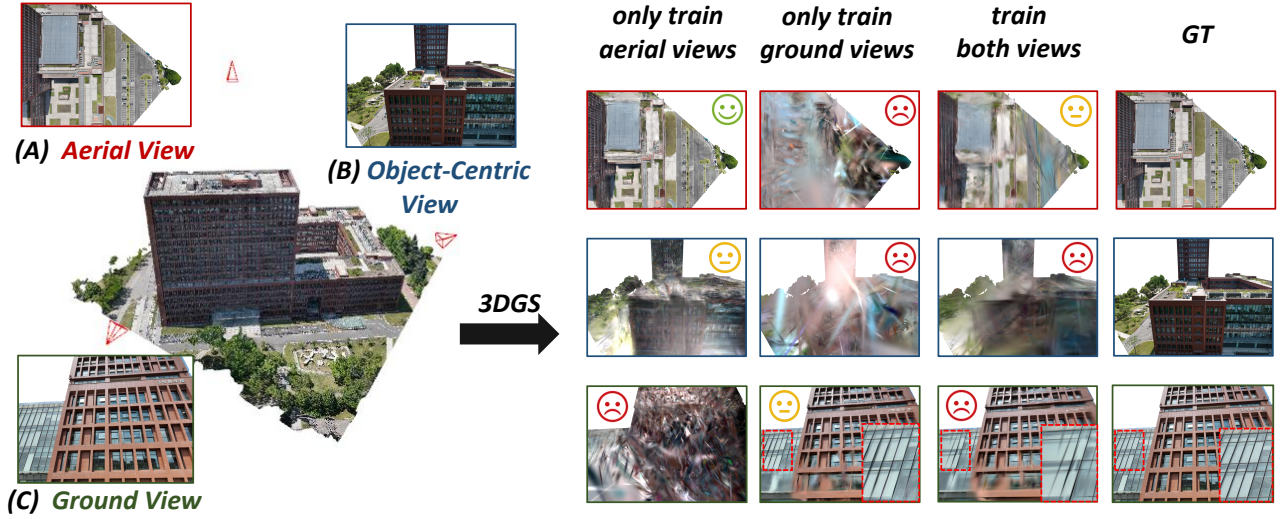


Fig. 2. Failure of naive aerial-ground joint training in 3DGS. We consider three viewpoint types for a target building: (A) aerial, (B) object-centric, and (C) ground views. Models trained only on aerial or ground images render their respective in-distribution views well, whereas naive joint training on aerial and ground images degrades even in-distribution results. All settings also struggle on out-of-distribution object-centric views.

Table 1

The comparison of multi-view image datasets for 3D reconstruction. For 'View', the '+' indicates that each scenario includes all these types of viewpoints, while the '/' indicates that each scenario in the dataset contains only one of these types of viewpoints.

Dataset	Level	Source	View	Image Size
Tanks and Temples (Knapitsch et al., 2017)	Instance	Real	Object-Centric	1920*1080
BlendedMVS (Yao et al., 2020)	Both	Both	Aerial/Ground/ Object-Centric	768*576 / 2048*1536
Waymo Open Dataset (Sun et al., 2020)	Scene	Real	Ground	1920*1080/1920*1040
SWJTU Dataset (Zhu et al., 2020)	Instance	Real	Aerial+Ground	6000*4000
UrbanScene3D (Lin et al., 2022)	Scene	Both	Aerial	6000*4000
MatrixCity (Li et al., 2023)	Scene	Synthetic	Aerial+Ground	1920*1080/1000*1000
GauU-Scene (Xiong et al., 2024)	Scene	Real	Aerial	5472*3648
Hier-GS (Kerbl et al., 2024)	Scene	Real	Ground	1024*690
UC-GS (Zhang et al., 2024)	Scene	Synthetic	Aerial+Ground	960*480/1280*720
Horizon-GS (Jiang et al., 2025)	Scene	Both	Aerial+Ground	1920*1080
AGC Landmarks	Instance	Real	Aerial+Ground+Object-Centric	5463*3956

Existing datasets can be roughly grouped into three categories. First, instance-level datasets such as Tanks and Temples (Knapitsch et al., 2017) mainly provide real object-centric captures from pedestrian viewpoints, while BlendedMVS (Yao et al., 2020) covers both instance- and scene-level targets but relies on heavily preprocessed semi-synthetic rendering. Second, aerial or ground-only scene datasets, including UrbanScene3D (Lin et al., 2022) and GauU-Scene (Xiong et al., 2024) for aerial imagery, and the Waymo Open Dataset (Sun et al., 2020) and Hier-GS (Kerbl et al., 2024) for vehicle-based street views, mainly describe large urban regions rather than individual landmarks. Third, aerial-ground datasets combine multiple acquisition platforms: SWJTU (Zhu et al., 2020) focuses on building instances with aerial and terrestrial subsets, whereas MatrixCity (Li

et al., 2023), UC-GS (Zhang et al., 2024), and Horizon-GS (Jiang et al., 2025) emphasize scene-scale air-to-ground reconstruction.

These datasets support important reconstruction studies, but they do not fully match the requirements of heterogeneous landmark modeling. Most real datasets either lack one of the aerial, ground, or object-centric viewpoint levels, or focus on regular urban blocks and geometrically simple structures. As a result, rendering quality is often evaluated within a limited viewpoint domain, which weakens validation under cross-level viewpoint shifts. In addition, synthetic or semi-synthetic datasets such as BlendedMVS (Yao et al., 2020), UrbanScene3D (Lin et al., 2022), MatrixCity (Li et al., 2023), UC-GS (Zhang et al., 2024), and Horizon-GS (Jiang et al., 2025) still face domain gaps in lighting, material appearance, and real capture disturbances. To address these limitations, we build a real outdoor image dataset for heterogeneous

architectural landmarks with aerial, ground, and object-centric viewpoints, enabling more comprehensive evaluation for feature matching, 3D reconstruction, novel view synthesis, and relighting.

2.2. Geometry-Enhanced Gaussian Splatting

Although 3DGS (Kerbl et al., 2023) enables real-time photorealistic rendering, its anisotropic Gaussian primitives lack explicit geometric organization. In large scenes, this can cause redundant growth, unstable geometry, and degraded surfaces under strong scale changes. Existing improvements mainly follow two lines: structured Gaussian organization and geometric regularization.

The first line organizes and selects Gaussian primitives more explicitly. Scaffold-GS (Lu et al., 2024) introduces anchors to bind feature-sharing Gaussian primitives and guide cloning and splitting. Octree-GS (Ren et al., 2025) extends this idea with multi-scale anchors and distance-aware Level-of-Detail (LOD) selection. For city-scale scenes, CityGaussian (Liu et al., 2025a,b) adopts divide-and-conquer spatial partitioning, while Hier-GS (Kerbl et al., 2024) builds an LOD hierarchy for efficient street-level roaming. These methods mainly improve scalability, efficiency, and primitive management.

The second line makes optimization more surface-aware with geometric priors. Monocular depth priors reduce floater artifacts (Chung et al., 2024), and ARSGaussian (Yao et al., 2026) uses LiDAR point clouds to guide Gaussian primitive growth and splitting. From image observations, 2DGS (Huang et al., 2024) and PGSR (Chen et al., 2025a) imposes planar constraints with multi-view geometric and photometric regularization, ULSR-GS (Li et al., 2025) combines multi-view guided densification with depth and normal consistency for aerial scenes, and GaussianCraft (Xiang et al., 2026) uses multi-view normal and scale priors for thin and hollow structures. Overall, these methods improve surface consistency through depth, point-cloud, planar, normal, or multi-view constraints.

Despite these advances, existing structures and priors remain limited for joint aerial-ground rendering of heterogeneous landmarks. Most regularizations assume relatively uniform camera trajectories or specific scene types, making it difficult to connect aerial topology with street-level details. For irregular buildings, severe occlusion, complex topology, and cross-level viewpoint shifts further hinder the alignment between Gaussian primitives and real surfaces.

2.3. Aerial-Ground Integrated Gaussian Splatting

Joint aerial-ground datasets introduce large resolution gaps and parallax, making direct co-training unstable. Existing methods mainly address this cross-view conflict in three ways. UC-GS (Zhang et al., 2024) uses cross-view uncertainty to weight aerial pixels according to vehicle-view rendering confidence, reducing collapse in road scenes. CrossView-GS (Zhang et al., 2025) decouples aerial and ground reconstruction with a dual-branch design, then fuses complementary cues through gradient-aware regularization.

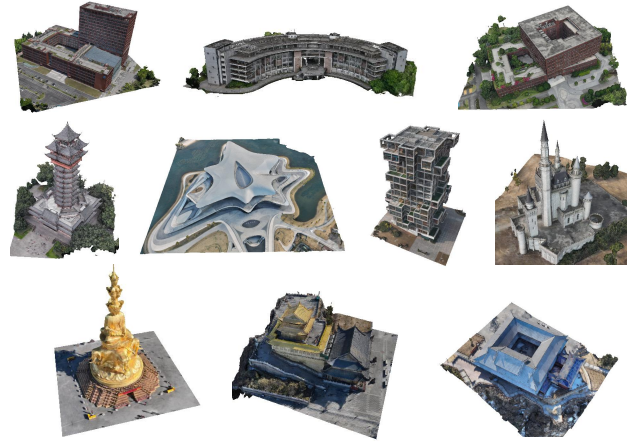


Fig. 3. Textured oblique photogrammetric mesh reconstructed from aerial images.

Horizon-GS (Jiang et al., 2025) adopts a coarse-to-fine LOD-anchor pipeline, using aerial images for global topology before refining street-level details. Overall, these methods improve aerial-ground integration through uncertainty weighting, view decoupling, or staged optimization.

However, they still provide limited global geometric consistency for heterogeneous landmarks. Although rendering quality improves along specific aerial or ground trajectories, Gaussian primitive positions, scales, and orientations may still deviate from real surfaces, causing artifacts under off-distribution viewpoints. To reduce this ambiguity, we use textured meshes from oblique photogrammetry as explicit geometric and appearance priors, supervising 3DGS to better bridge aerial, ground, and object-centric perspectives.

3. Aerial-Ground-Centric Landmarks Dataset

Scenes. We collected 10 building-scale landmarks for AGC Landmarks, with an emphasis on architectural diversity and structural heterogeneity. As summarized in Table 2, these targets cover six representative types: hollow buildings, large venues, irregular envelopes, ancient buildings, castle, and sculpture. Together, these scenes stress reconstruction in multiple ways. Their hollow layouts and dense decorative details create severe self-occlusion, while reflective glass, curved roofs, lakeside backgrounds, and cliff-side terrain further challenge robust geometry and appearance modeling. Textured meshes reconstructed from aerial images are shown in Fig. 3, highlighting the scale and structural variety of the collected landmarks.

Data Acquisition. All images are captured with a DJI Matrice 4E drone equipped with a camera of approximately 20 million effective pixels. Each image is recorded with synchronized RTK (Real-Time Kinematic) positioning to support cross-subset alignment and reconstruction. For each landmark, we acquire three complementary image subsets; their camera poses after bundle adjustment are visualized in Fig. 4.

Table 2

Metadata of the captured landmarks. The architectural type, scene feature, and count of aerial, ground, and object-centric images are listed for each landmark.

Name	Type	Scene Feature	Aerial	Ground	Object-centric
Teaching Building	Hollow building	Enclosed courtyard; multi-block facade	127	614	1900
Library	Large venue	Curved facade; broad open frontage	278	560	2277
Office Building	Hollow building	Interior courtyard; dense window grid	100	744	1612
Jiutian Tower	Ancient building	Multi-tier eaves; open galleries	277	234	2180
Sci-Fi Museum	Large venue	Lakeside setting; large-span curved roof	545	618	2067
Rubik's Cube Mansion	Irregular envelope	Stacked blocks; cantilevered terraces	193	168	2127
Cinderella Castle	Castle	Spire towers; enclosure walls	184	426	1306
Bodhisattva Statue	Sculpture	Gilded statue; stepped pedestal	631	251	2197
Mahavira Hall	Ancient building	Temple courtyard; layered tiled roofs	811	243	1449
Woyun Temple	Ancient building	Cliff-side terrain; overhanging roofs	757	131	1443

- *Aerial.* We fly oblique photogrammetry routes to cover the full survey area. Depending on the target height, the drone operates at 90–120 m with a 90% forward overlap, a 70% side overlap, and a 45° camera tilt. Five viewing directions are captured, following the standard configuration for oblique photogrammetry.
- *Ground.* We use the UAV to collect horizontal and upward-looking ground views under controlled acquisition conditions, with synchronized image timestamps and RTK positioning information that are more difficult to obtain consistently during handheld capture. This subset simulates supplementary ground-level photography, including the multiple viewing directions encountered in street-level and panoramic capture. The drone flies 2–5 m above the ground, keeps the camera oriented toward the target, and repeats passes at elevation angles of 0°, 15°, and 30°. These elevations simulate the changing human viewpoints encountered when walking around a landmark.
- *Object-centric.* After the aerial oblique survey, the DJI controller automatically generates a coarse point cloud online for flight planning. DJI's close-range photogrammetry mode uses this point cloud to generate flight paths at a prescribed distance from the target. This point cloud is distinct from the aerial oblique photogrammetric mesh used as a prior in HeteroArch-GS. The UAV then acquires dense object-centric images while orbiting the target at a distance of 10–20 m, providing close-range observations of fine geometric and texture details.

Data Processing. Our processing pipeline converts the raw multi-view imagery into geo-referenced and foreground-focused inputs for 3DGS optimization. During acquisition, all aerial, ground, and object-centric images are recorded with synchronized RTK positioning in a common geographic coordinate system. Aerial triangulation, including bundle adjustment using the RTK positioning information, is performed independently for the aerial, ground, and object-centric subsets in DJI Terra (DJI, 2026). The reprojection error is approximately one pixel or less for the aerial images and slightly larger for the ground and object-centric

images, but remains comparable across the subsets relative to their image resolutions. Detailed aerial triangulation reports will be provided with the dataset. The resulting bundle-adjusted camera poses are shown in Fig. 4 and used for subsequent processing and rendering of the captured images. To reproduce the practical scenario in which an aerial oblique photogrammetric asset is already available for a landmark, we reconstruct the textured mesh exclusively from the aerial images using DJI Terra (DJI, 2026), as shown in Fig. 3. This aerial mesh is then used as the geometric proxy for generating foreground masks for all image subsets. To prepare the data for 3DGS optimization, we compute the scene center and radius from the bundle-adjusted aerial and ground camera centers, transform all bundle-adjusted camera poses, sparse point clouds, and mesh geometry into a local coordinate system, and approximately normalize each scene to a $[-1, 1]^3$ bounding box. The resulting bundle-adjusted poses, foreground masks, sparse points, and normalized mesh provide a unified coordinate system, consistent scene scale, and geometry priors for downstream 3DGS optimization.

4. Mesh-Guided Aerial-Ground Gaussian Splatting for Heterogeneous Architectures

As illustrated in Fig. 5, *HeteroArch-GS* starts from an existing photogrammetric mesh and oblique aerial images with bundle-adjusted camera poses. To complement the aerial observations, we additionally capture ground images with RTK positioning. The camera poses of the aerial and ground subsets are obtained independently through aerial triangulation with bundle adjustment in DJI Terra (DJI, 2026), using the RTK positioning information. The resulting bundle-adjusted camera poses are transformed into the common local coordinate system described in Sec. 3 and used for aerial-ground 3DGS optimization. The mesh then serves as an explicit proxy for scene geometry and appearance, guiding three core components: Mesh-Guided Anchor Initialization (Sec. 4.2), Mesh-Guided Geometric Regularization (Sec. 4.3), and Mesh-Guided Pseudo-View Supervision (Sec. 4.4). During optimization, color supervision and geometric regularization jointly guide the Gaussian primitives to recover faithful appearance while staying close to the underlying scene structure. To train under limited hardware memory, we

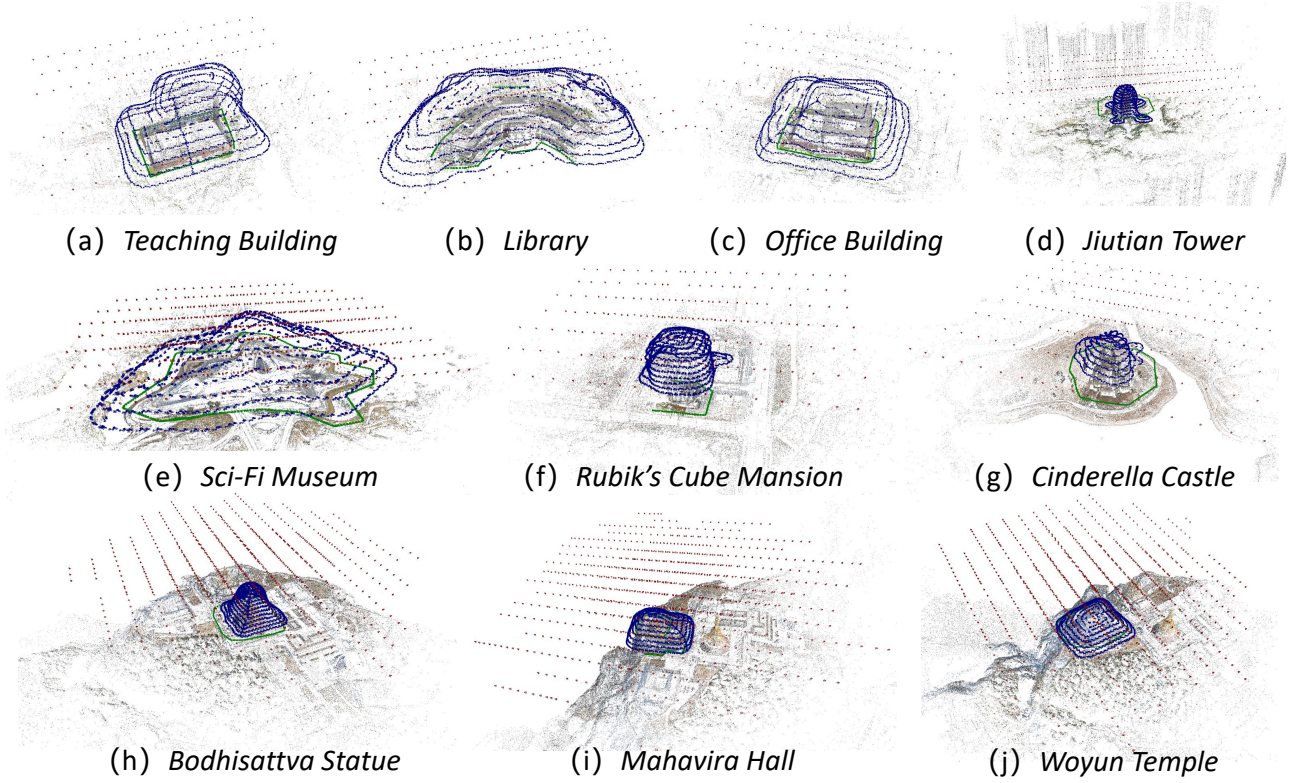


Fig. 4. Camera poses and sparse point clouds of the AGC Landmarks dataset after bundle adjustment in DJI Terra (DJI, 2026) using RTK positioning information. Red, green, and blue frustums denote the bundle-adjusted camera poses of the independently processed aerial, ground, and object-centric subsets, respectively. The sparse point clouds are obtained from the aerial and ground triangulations and used for training initialization.

introduce an on-demand data fetching mechanism (Sec. 4.5) that keeps only a fixed-capacity CPU cache and loads images as needed for fast retrieval.

4.1. Preliminaries

Our method is built on an anchor-based 3DGS backbone. We use two existing de-facto standard mechanisms as the base architecture: the anchor-driven primitive generation of Scaffold-GS (Lu et al., 2024), the LOD selection of Octree-GS (Ren et al., 2025). In addition, the staged aerial-ground optimization of Horizon-GS (Jiang et al., 2025) is also employed. We briefly summarize these components here, and our proposed strategies are built on top of this backbone. Table 3 summarizes the main notation used in our formulation.

3DGS (Kerbl et al., 2023) represents a scene using anisotropic Gaussian primitives. The i -th Gaussian primitive has center μ_i and covariance Σ_i , parameterized as $\Sigma_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^T \mathbf{R}_i^T$ for stable optimization. After projection onto the image plane, $\mathcal{N}$ Gaussian primitives covering pixel $\mathbf{x}'$ are sorted front-to-back and composited by α -blending:

$$\mathbf{c}(\mathbf{x}') = \sum_{i \in \mathcal{N}} \prod_{j \in \mathcal{N}, j < i} (1 - \sigma_j) \mathbf{c}_i \sigma_i, \quad \sigma_i = \alpha_i G'_i(\mathbf{x}'), \quad (1)$$

Table 3

Nomenclature.

Symbol(s)	Meaning
$\mu, \mathbf{s}, \mathbf{q}$	Gaussian primitive center, scale, and rotation.
$\Sigma, \mathbf{R}, \mathbf{S}$	Gaussian primitive covariance, rotation matrix, and scaling matrix.
$\mathbf{c}, \alpha, \sigma$	Gaussian primitive color, opacity, and effective pixel-wise opacity.
$\mathcal{P}$	Anchor point set.
$\mathbf{v}, \mathbf{f}, \mathbf{O}$	Anchor position, feature, and Gaussian primitive offsets.
$\mathbf{d}, \delta$	Camera-anchor viewing direction and distance.
$\mathcal{S}$	Surface voxel set.
$\mathbf{g}$	Voxel coordinate.
$U, \mathbf{G}$	Unsigned Distance Field (UDF) and Distance Gradient Field (DGF).
$\mathcal{L}$	Loss functions.
i, j	Global indices of Gaussian primitives.
m	Local Gaussian primitive index within an anchor.
k	Anchor index.
r	Camera index.

where α_i denotes the opacity of the i -th Gaussian primitive, $G'_i(\mathbf{x}')$ is its projected 2D Gaussian response at pixel $\mathbf{x}'$, and σ_i is the resulting effective opacity contribution.

Following Scaffold-GS (Lu et al., 2024), Gaussian primitives are organized around sparse anchors $\mathcal{P}$. For the k -th

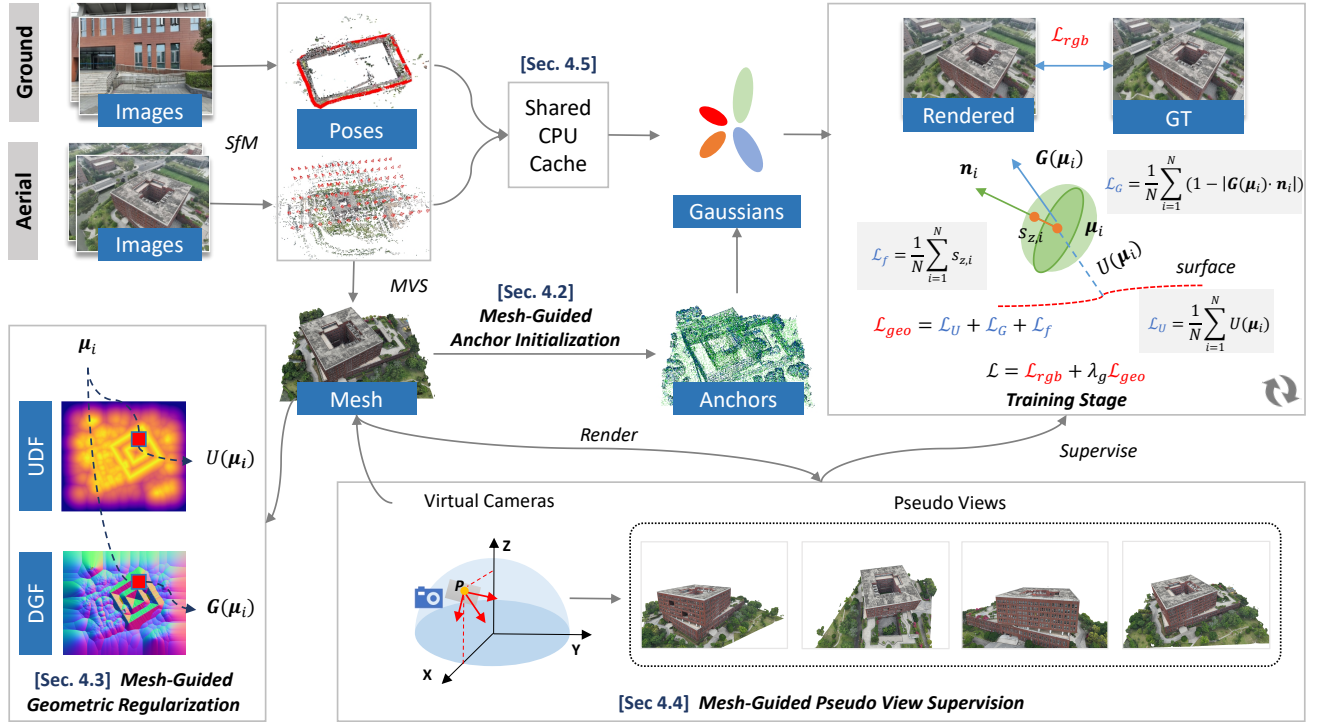


Fig. 5. The pipeline of proposed *HeteroArch-GS*. We start from an existing textured photogrammetric mesh, which serves as a geometric and photometric prior for the subsequent optimization of Gaussian primitives. The mesh guides the initialization of Scaffold-GS Gaussian anchors (Lu et al., 2024), provides continuous geometric supervision through an Unsigned Distance Field (UDF) and a Distance Gradient Field (DGF), and renders pseudo views to regularize 3DGS optimization. To handle large-scale datasets, we implement an efficient lazy loading mechanism that dynamically manages training images within a fixed-capacity CPU cache.

anchor located at $\mathbf{v}_k$, M Gaussian primitives are instantiated using learnable offsets, with the center of the m -th primitive given by $\boldsymbol{\mu}_{k,m} = \mathbf{v}_k + \mathbf{O}_{k,m}$, where $\mathbf{O}_{k,m}$ denotes the m -th learnable offset. The remaining Gaussian primitive attributes, including opacity, color, scale, and rotation, are decoded by lightweight MLPs from the anchor feature $\mathbf{f}_k$, the camera-anchor viewing direction $\mathbf{d}_{k,r}$, and distance $\delta_{k,r}$:

$$\{\alpha_{k,m}, \mathbf{c}_{k,m}, \mathbf{s}_{k,m}, \mathbf{q}_{k,m}\}_{m=1}^M = \text{MLP}(\mathbf{f}_k, \mathbf{d}_{k,r}, \delta_{k,r}). \quad (2)$$

Following Octree-GS (Ren et al., 2025), anchors are organized in an octree and selected by LOD to handle large scale changes in outdoor scenes. For the r -th camera, the active LOD threshold for the k -th anchor is determined by the distance between camera center and anchor position $\mathbf{v}_k$.

Following Horizon-GS (Jiang et al., 2025), we adopt a two-stage optimization schedule for aerial-ground imagery. The coarse stage prioritizes aerial views and aerial-specific anchor adaptation to establish a stable global structure, whereas the fine stage emphasizes ground views and ground-specific adaptation to recover close-range details. Our mesh-guided initialization, geometric regularization, and pseudo-view supervision are integrated into this backbone.

Following common 3DGS optimization practice (Kerbl et al., 2023; Lu et al., 2024; Jiang et al., 2025), our objective combines an RGB rendering loss and a geometric regularization loss. The rendering term $\mathcal{L}_{rgb}$ is a weighted combination

of the L_1 and D-SSIM terms between the rendered and target RGB images, following 3DGS (Kerbl et al., 2023). The overall loss is defined as:

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_g \mathcal{L}_{geo}, \quad (3)$$

where λ_g balances photometric rendering fidelity and geometric regularization. The geometric term $\mathcal{L}_{geo}$ is detailed in Sec. 4.3.

4.2. Mesh-Guided Anchor Initialization

Anchor-based 3DGS methods typically initialize Gaussian anchors from sparse SfM point clouds, whose distribution can be highly imbalanced under the resolution gap between aerial and ground imagery. Since our input already includes a mesh model, a direct strategy is to sample anchor points on the mesh surface. To cover both global surface regions and local structural details, we sample $|\mathcal{P}^f|$ anchor points on mesh faces and $|\mathcal{P}^e|$ anchor points along sharp edges, yielding the initialized anchor point set $\mathcal{P}$ with $|\mathcal{P}| = |\mathcal{P}^f| + |\mathcal{P}^e|$.

For the face-sampled anchor points $\mathcal{P}^f$, we perform area-weighted uniform sampling on the mesh model, where each triangular face is sampled with probability proportional to its area. This yields broad coverage of large planar structures, such as walls and roofs, and reduces initialization voids caused by SfM occlusions.

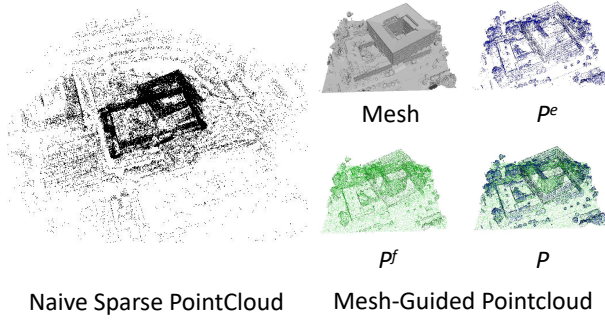


Fig. 6. Naive sparse point cloud and mesh-guided anchor initialization. Mesh-guided sampling places anchors on building surfaces and reduces noisy background points.

For the edge-sampled anchor points $\mathcal{P}^e$, we identify sharp mesh edges using a dihedral-angle threshold τ and sample $|\mathcal{P}^e|$ anchor points by linear interpolation along these edges. This increases anchor density near geometric discontinuities and improves boundary initialization.

The positions of initial Scaffold-GS Gaussian anchors (Lu et al., 2024) are assigned directly from the spatial coordinates of the sampled points. As shown in Fig. 6, the original SfM point cloud consists of 114,408 points, many of which are located in background regions outside the target building. In contrast, under the parameter settings of $|\mathcal{P}| = 80,000$, $|\mathcal{P}^f| = 40,000$, $|\mathcal{P}^e| = 40,000$, and $\tau = 30^\circ$, our strategy yields a point set that is more concentrated on the target building surface.

4.3. Mesh-Guided Geometric Regularization

Existing 3DGS geometric constraints often rely on image-view depth maps obtained from LiDAR or depth estimation (Yao et al., 2026; Chung et al., 2024; Turkulainen et al., 2025). During optimization, these constraints are applied after projection into the current camera view, so they are limited to visible pixels and can vary with viewpoint, occlusion, and depth-map quality.

In contrast, the aerial mesh provides a scene-level 3D prior: we regularize the point set formed by Gaussian primitive centers directly in 3D space rather than the projected image space. The mesh also provides reliable normal information, which constrains Gaussian ellipsoids to better align with the underlying surface.

4.3.1. Surface occupied field representation

To impose geometric constraints directly in 3D space, we need to repeatedly compute distances from Gaussian primitive centers to the mesh surface. However, exact point-to-mesh queries usually require nearest-face search or Bounding Volume Hierarchy (BVH) traversal, which involves irregular memory access and branching operations that are difficult to execute in real time within a GPU-based optimization loop.

Therefore, we convert the surface mesh into a surface occupancy voxel grid $\mathcal{S}$ parameterized by a spatial resolution N_{res} , and compute voxelized field representations, including

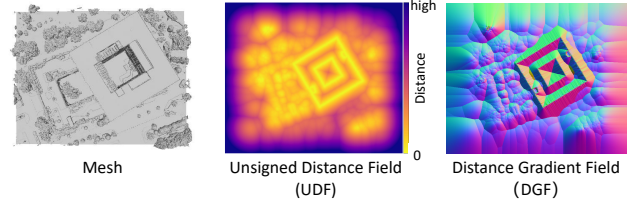


Fig. 7. Visualization of the mesh-derived UDF and DGF. The UDF magnitude map and the DGF direction vectors are shown on a cross-sectional slice extracted at the midpoint of the Z-axis.

the Unsigned Distance Field (UDF) and the Distance Gradient Field (DGF), denoted as U and $\mathbf{G}$, respectively. This representation replaces repeated point-to-mesh queries with GPU-friendly grid sampling, while still allowing continuous interpolation for arbitrary 3D coordinates.

Let p denote the voxel pitch, which is determined by the maximum extent of the mesh bounding box and the prescribed voxel resolution:

$$p = \frac{\max(\mathbf{b}_{\max} - \mathbf{b}_{\min})}{N_{res}}. \quad (4)$$

For each voxel $\mathbf{g}$, the nearest occupied surface voxel $\mathbf{g}^*$ is obtained using the Euclidean Distance Transform (EDT). The corresponding UDF value is computed as

$$\begin{aligned} \mathbf{g}^* &= \arg \min_{\mathbf{u} \in \mathcal{S}} \|\mathbf{g} - \mathbf{u}\|_2, \\ U(\mathbf{g}) &= p \|\mathbf{g} - \mathbf{g}^*\|_2. \end{aligned} \quad (5)$$

Here, multiplication by p converts the EDT distance from voxel-index units to the normalized scene coordinate system. Simultaneously, we construct the DGF to provide a nearest-surface directional prior for Gaussian primitive alignment. Defining the displacement vector as

$$\mathbf{d}(\mathbf{g}) = p (\mathbf{g} - \mathbf{g}^*), \quad (6)$$

the DGF is computed as

$$\mathbf{G}(\mathbf{g}) = \frac{\mathbf{d}(\mathbf{g})}{\|\mathbf{d}(\mathbf{g})\|_2 + \epsilon_G}, \quad \epsilon_G = 10^{-8}. \quad (7)$$

The stabilization term ϵ_G prevents numerical division by zero.

As illustrated in Fig. 7, we visualize the UDF and its corresponding DGF derived from the input mesh. During the optimization phase, both UDF and DGF fields are queried at Gaussian primitive centers through trilinear interpolation, enabling continuous field sampling and piecewise-differentiable gradients with respect to Gaussian primitive coordinates. The precomputed fields are fixed during optimization, and gradients are propagated only to Gaussian primitive parameters. For out-of-bound Gaussian primitive centers, we clamp the positions to the grid boundary, extend the UDF with the outside distance, and use the boundary DGF. This provides meaningful gradients for returning Gaussian primitives to the valid domain.

4.3.2. Geometric regularization with guided field

UDF regularization. For each Gaussian primitive center μ_i , we query the UDF through trilinear interpolation and use $U(\mu_i)$ as its surface-proximity penalty. Minimizing this term pulls the point set formed by Gaussian primitive centers toward the mesh surface in 3D space.

$$\mathcal{L}_U = \frac{1}{N} \sum_{i=1}^N U(\mu_i) \quad (8)$$

DGF regularization. Standard 3DGS lacks an explicit definition of surface normals, as the primitives are unconstrained volumetric ellipsoids. Extracting normal directions typically requires computationally expensive operations, such as performing Singular Value Decomposition (SVD) on the covariance matrix or dynamically searching for the axis with the minimum scale. To ensure efficient and differentiable surface alignment during optimization, we explicitly bind the surface normal to the local Z-axis of each Gaussian primitive's rotation matrix. For primitive i , the unit quaternion $\mathbf{q}_i = [w_i, x_i, y_i, z_i]^T$ gives the corresponding 3×3 rotation matrix $\mathbf{R}_i$, which defines the orientation of its local axes. By explicitly designating the local Z-axis as the normal, we can directly extract $\mathbf{n}_i$ from the third column of $\mathbf{R}_i$:

$$\mathbf{n}_i = \begin{bmatrix} 2(x_i z_i + y_i w_i) \\ 2(y_i z_i - x_i w_i) \\ 1 - 2(x_i^2 + y_i^2) \end{bmatrix} \quad (9)$$

To prevent numerical drift during optimization, the extracted normal vector is further L_2 -normalized as $\mathbf{n}_i \leftarrow \mathbf{n}_i / (\|\mathbf{n}_i\|_2 + \epsilon_n)$ with $\epsilon_n = 10^{-6}$. Given the primitive normal $\mathbf{n}_i$, the DGF regularization aligns it with the local distance-gradient direction queried at the primitive center:

$$\mathcal{L}_G = \frac{1}{N} \sum_{i=1}^N (1 - |\mathbf{G}(\mu_i) \cdot \mathbf{n}_i|) \quad (10)$$

Flattening regularization. Furthermore, simply assigning the Z-axis as the normal direction is insufficient if the primitive remains a volumetric ellipsoid. Therefore, we introduce a structural regularization constraint during optimization to enforce a disk-like geometry on the primitives. With explicit scaling factors $\mathbf{s}_i = [s_{x,i}, s_{y,i}, s_{z,i}]^T$, the flattening loss $\mathcal{L}_f$ penalizes the spatial extent of each primitive along its designated normal axis:

$$\mathcal{L}_f = \frac{1}{N} \sum_{i=1}^N s_{z,i} \quad (11)$$

By aggressively minimizing $s_{z,i}$, this photometric-independent geometric prior collapses the ellipsoids into thin, surface-aligned splats, ensuring that the extracted normal $\mathbf{n}_i$ accurately represents the underlying manifold geometry rather than arbitrary volumetric noise.

In summary, the final geometric loss $\mathcal{L}_{geo}$ combines the UDF regularization $\mathcal{L}_U$, DGF regularization $\mathcal{L}_G$, and flattening regularization $\mathcal{L}_f$ equally:

$$\mathcal{L}_{geo} = \mathcal{L}_U + \mathcal{L}_G + \mathcal{L}_f \quad (12)$$

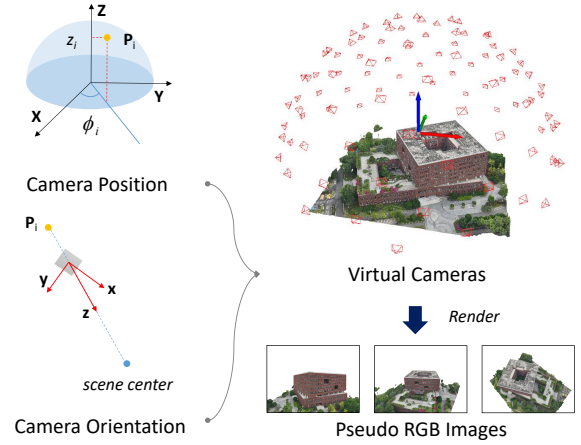


Fig. 8. The processing pipeline for generating RGB images of pseudo views. We sample a set of virtual camera positions uniformly around the textured mesh and render pseudo views at these positions.

4.4. Mesh-Guided Pseudo-View Supervision

While the refined initialization and regularization strategies provide stronger geometric priors, 3DGS optimization remains primarily driven by photometric loss from training views. This dependence often leads to view-dependent overfitting, where the model accurately reconstructs training images but its primitives collapse into geometric artifacts when viewed from novel perspectives. To mitigate this, we propose a supervision strategy using pseudo renderings from the existing reconstructed textured mesh to regularize the optimization process.

Virtual Camera Trajectory Construction. To provide comprehensive multi-view supervision, we generate virtual camera poses that uniformly cover the target scene. As shown in Fig. 8, we use Fibonacci Hemisphere Sampling (Keinert et al., 2015) to distribute the virtual views over a dome centered on the building. For each virtual-view index r , a look-at transformation orients the camera toward the scene center, producing pseudo views from diverse and approximately isotropic directions.

Training Strategy with Pseudo RGB Images. Using the generated virtual cameras, we render pseudo images I_{pse}^r from the textured mesh model. Although these images may contain geometric or textural biases inherent to the mesh reconstruction process, and thus lack photorealistic detail, they serve as a consistent global geometric anchor. Recognizing that the mesh model is reconstructed only from aerial photogrammetry, its rendering quality is inherently superior at altitudes consistent with the original aerial capture. To reflect this varying confidence, we assign a loss weight w_r to each pseudo image I_{pse}^r based on its camera altitude h_r . We define a linear mapping function to determine the importance of each pseudo image, and ensure it remains secondary to the ground-truth imagery:

$$w_r = \frac{\omega_{max} - \omega_{min}}{h_{max} - h_{min}} \cdot (h_r - h_{min}) + \omega_{min}, \quad (13)$$

The altitude boundaries h_{min} and h_{max} map to $w_r \in [\omega_{min}, \omega_{max}]$, ensuring that pseudo images provide geometric guidance without overwriting the fine-grained photometric details provided by the real-world images, which maintain a unit weight of 1.0.

4.5. Efficient Lazy Loading for Large-Scale Datasets

In large-scale reconstruction involving aerial and ground-level imagery, data loading and memory management become critical system bottlenecks. The original 3DGS implementation statically loads all training views into memory, exposing a capacity-throughput trade-off through two loading modes:

- **GPU mode.** All images are cached as fully decoded and normalized `float32` tensors in GPU VRAM. Although this maximizes optimization throughput, the dataset size is strictly bounded by available VRAM, making Out-Of-Memory (OOM) failures likely for large-scale scenes.
- **CPU mode.** The same `float32` tensors are stored in host RAM and transferred to the GPU on-the-fly. This shifts the capacity limit from GPU memory to system memory, while repeated transfers of uncompressed 32-bit tensors saturate PCIe bandwidth, causing GPU starvation and reduced optimization speed.

To reduce memory demands when processing large-scale datasets, we introduce a streaming architecture termed Efficient Lazy Loading. It separates image preprocessing from optimization and loads preprocessed views on demand through a capacity-bounded shared CPU cache. This avoids keeping all training images in RAM, while the supported dataset size and loading throughput remain subject to available storage, I/O bandwidth, and other system resources. Specifically, the architecture comprises the following key components:

Pre-Warmed Binary Cache. To avoid the CPU bottlenecks caused by on-the-fly image decoding and spatial resizing, we decouple data preprocessing from the optimization loop. Before optimization begins, all images and masks are processed in parallel and serialized to high-speed NVMe storage as uncompressed binary `uint8` tensors. We then initialize a capacity-bounded CPU hot cache with a memory quota M_{max} and pre-warm it by proactively loading these binary views into pinned host RAM until 90% of M_{max} is reached, reducing sequential I/O stalls during the initial optimization phase.

Dynamic Fetching and LRU Cache Management. As the optimization loop progresses, if a camera view undergoes its first sampling (cache miss), the system retrieves the preprocessed raw binary data directly from the SSD into the pinned host RAM. To manage the bounded capacity M_{max} , the CPU cache maintains a Least Recently Used (LRU) (Shi and Fan, 2025) eviction policy based on tensor byte-size to dynamically swap out stale frames.

Asynchronous JIT Materialization. When a view is explicitly requested for rasterization, the system leverages the

pinned memory to initiate an asynchronous Direct Memory Access (DMA) (Shirur et al., 2018) transfer to the GPU. Only upon arriving at the device is the `uint8` tensor converted to `float32` and normalized to the $[0, 1]$ range.

5. Experiment

We evaluate the proposed HeteroArch-GS from four complementary perspectives:

- **Visual quality.** Sec. 5.2 benchmarks HeteroArch-GS against state-of-the-art methods in terms of rendering fidelity, covering in-distribution aerial and ground views as well as out-of-distribution object-centric views.
- **Geometry quality.** Sec. 5.3 analysis measures how accurately the reconstruction captures scene geometry and suppresses redundant Gaussian primitives in invalid regions, which further validates the out-of-distribution rendering performance.
- **Ablation studies.** Sec. 5.4 isolates the contributions of the proposed Mesh-Guided Anchor Initialization (MAI), Mesh-Guided Geometric Regularization (MGR), Mesh-Guided Pseudo-View Supervision (MPS), and lazy loading strategy.
- **Hyperparameter sensitivity.** Sec. 5.5 analyzes the impact of key algorithmic configurations on model performance.

5.1. Experimental Setup

5.1.1. Dataset

Experiments are conducted on 10 landmarks from the proposed AGC Landmarks dataset. Consistent with existing mapping programs (Google, 2026; Chen et al., 2025b), the aerial photogrammetric mesh is reconstructed from the complete aerial image collection before the train/test split and is treated as a pre-existing external prior. The train/test split applies only to the subsequent 3DGS optimization stage. For the aerial and ground imagery, one out of every 8 images is held out from RGB supervision for testing, while the object-centric images are used entirely for testing. All images are downsampled by a factor of 4. To eliminate the influence of background regions during training, foreground masks for each viewpoint are pre-rendered using the mesh proxy, thereby masking out the backgrounds prior to the training phase. We report the experimental results on Teaching Building, Library, Jiutian Tower, and Mahavira Hall. These four scenes are selected to cover distinct architectural types and geometric characteristics, including modern institutional buildings, a multi-tier tower, and a traditional temple with layered roofs and complex eaves (Table 2). Quantitative results for the remaining six scenes are reported in Appendix A, with complete visual comparisons available on the project website¹.

¹<https://vrlab.org.cn/~hanhu/projects/heteroarch-gs>

Table 4

Rendering quantitative comparison to baselines on *Teaching Building*, *Library*, *Jiutian Tower*, and *Mahavira Hall*. The best and second-best results are highlighted in red and blue, respectively.

Method	In-Distribution						Out-of-Distribution		
	Aerial			Ground			Object-Centric		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
<i>Teaching Building</i>									
3DGS	16.2802	0.5018	0.4828	17.1639	0.6173	0.5005	14.9869	0.5034	0.5193
2DGS	16.3652	0.4933	0.5355	16.4739	0.5986	0.5281	14.4678	0.5130	0.5312
PGSR	12.5126	0.3825	0.5184	16.0390	0.5814	0.5336	12.0942	0.3622	0.5201
Scaffold-GS	18.1076	0.5437	0.4167	17.5677	0.6202	0.4741	15.0148	0.4831	0.4926
Octree-GS	18.1947	0.5446	0.3925	17.6069	0.6204	0.4386	14.9497	0.4708	0.4779
Horizon-GS	24.1769	0.8402	0.1361	20.8820	0.7053	0.3341	17.3014	0.5922	0.3509
Ours	24.0765	0.8434	0.1263	20.4808	0.6879	0.3923	17.4792	0.5964	0.3508
<i>Library</i>									
3DGS	18.0860	0.5346	0.5085	18.1483	0.5875	0.5041	17.0192	0.5201	0.5096
2DGS	17.7125	0.5352	0.5320	17.4860	0.5572	0.5367	16.4423	0.5236	0.5254
PGSR	13.8010	0.4291	0.5461	16.3412	0.5364	0.5439	14.1889	0.4294	0.5241
Scaffold-GS	18.1993	0.5345	0.4871	18.3401	0.5960	0.4817	16.9121	0.5061	0.4946
Octree-GS	18.1152	0.5283	0.4677	18.2433	0.5933	0.4671	16.9154	0.5011	0.4806
Horizon-GS	23.7575	0.7303	0.2779	20.6693	0.6629	0.3963	20.5166	0.6503	0.3407
Ours	23.8111	0.7370	0.2719	20.1780	0.6388	0.4551	20.5613	0.6519	0.3359
<i>Jiutian Tower</i>									
3DGS	20.5508	0.6013	0.3457	27.3534	0.8258	0.1875	14.3015	0.4517	0.5001
2DGS	18.0319	0.5774	0.4232	24.3595	0.7524	0.3010	13.9207	0.4553	0.5032
PGSR	16.2182	0.5313	0.3995	18.5947	0.6726	0.3213	10.6078	0.3613	0.5077
Scaffold-GS	20.5365	0.6088	0.3197	27.3367	0.8293	0.1758	14.1417	0.4310	0.4788
Octree-GS	20.4057	0.6006	0.3006	27.4963	0.8319	0.1466	15.0624	0.4392	0.4561
Horizon-GS	24.6944	0.8039	0.1752	26.4117	0.8457	0.1380	15.9219	0.4935	0.4199
Ours	24.6294	0.8089	0.1473	27.4766	0.8557	0.1312	16.5253	0.5055	0.4025
<i>Mahavira Hall</i>									
3DGS	18.4767	0.6111	0.3784	17.4933	0.5635	0.4740	13.2859	0.4596	0.5429
2DGS	17.5265	0.5844	0.4287	16.8919	0.5468	0.4833	12.5865	0.4516	0.5399
PGSR	17.5193	0.6426	0.3617	16.7678	0.5511	0.4744	12.5341	0.4210	0.5296
Scaffold-GS	19.2743	0.6418	0.3365	17.7109	0.5778	0.4341	13.2013	0.4488	0.5227
Octree-GS	18.5710	0.6298	0.3207	17.3450	0.5694	0.4157	13.3470	0.4395	0.5000
Horizon-GS	20.9907	0.7832	0.2057	21.6494	0.6936	0.2852	15.1602	0.4807	0.4361
Ours	20.4813	0.7759	0.1811	21.3986	0.6802	0.3115	15.0803	0.4766	0.4312

5.1.2. Metrics

The evaluation includes three categories:

- **Visual quality.** We evaluate rendering fidelity using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) (Wang et al., 2004), and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018). These metrics are computed exclusively within the minimum bounding box of the foreground region.
- **Geometric quality.** We measure the similarity between the point set formed by Gaussian primitive centers and a uniformly sampled reference point cloud derived from an independently reconstructed photogrammetric mesh using the object-centric image subset, which is not involved

in model training. Given the varying spatial coverage of 3DGS outputs across different methods, we first crop all point sets to the mesh bounding box for a fair comparison, and then calculate Precision (P), Recall (R), and F1-score (F1) at inlier thresholds of 0.001 and 0.01. These thresholds are defined in the normalized scene space, corresponding to 1 $\times$ and 10 $\times$ the coarsest voxel size, respectively. We also report the total number of Gaussian primitives and the ROI Ratio (the percentage of primitive centers within the bounding box) to evaluate how well the model concentrates its representation on foreground regions.

- **Runtime efficiency.** We benchmark the proposed data-loading strategy against the naive implementation to evaluate its runtime and memory efficiency on large-scale

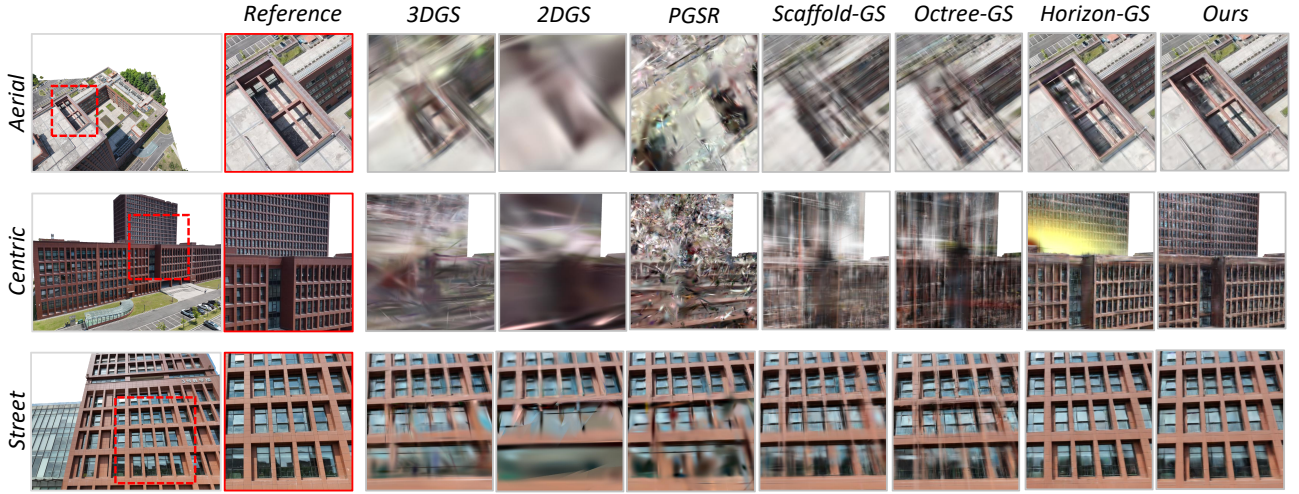


Fig. 9. Teaching Building. HeteroArch-GS preserves beam and facade-window structures more clearly in aerial and object-centric views, with fewer empty-space floaters. Horizon-GS remains slightly stronger on ground-level geometric details and complex glass reflections.

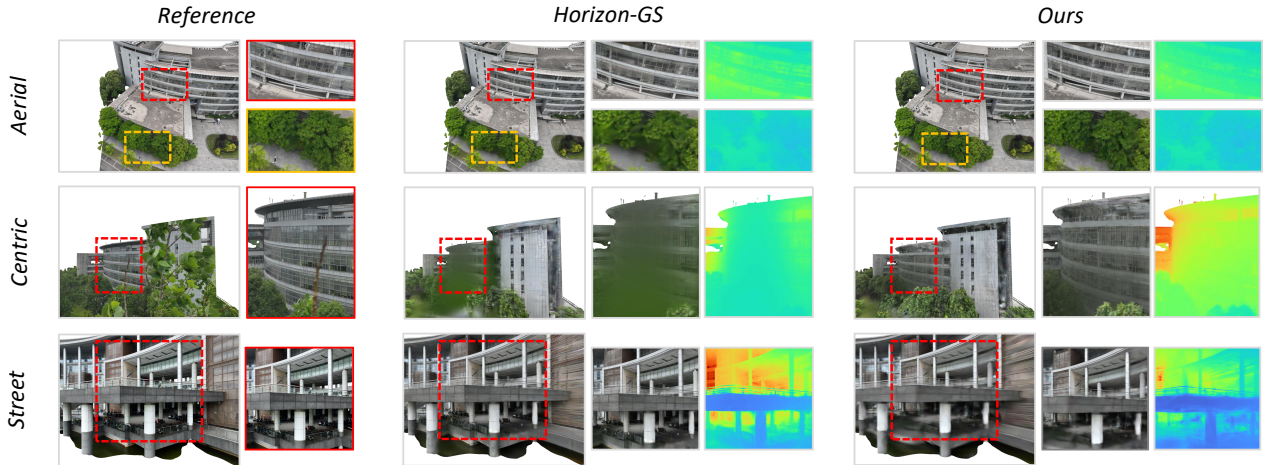


Fig. 10. Library. HeteroArch-GS produces denser and more coherent depth maps in aerial and object-centric views, preserving complex facade textures. Horizon-GS shows blurred texture regions in aerial views and foreground floaters in object-centric views, while our ground view is affected by mesh deficiencies.

datasets. We report the total end-to-end runtime from data preprocessing to the completion of training. Peak GPU VRAM, peak CPU RAM, and average GPU/CPU utilization are measured during the training stage.

5.1.3. Comparison Methods

We selected 3DGS (Kerbl et al., 2023), 2DGS (Huang et al., 2024), PGSR (Chen et al., 2025a), Scaffold-GS (Lu et al., 2024), Octree-GS (Ren et al., 2025), and Horizon-GS (Jiang et al., 2025) as the methods for comparison. We train 3DGS, 2DGS, PGSR, Scaffold-GS and Octree-GS for a total of 50k iterations and stopped the ADC strategy at 25k iterations. For Scaffold-GS, Octree-GS, and Horizon-GS, we set the number of offset Gaussian primitives per anchor to 10 and the coarsest voxel size to 0.001. For Horizon-GS, we

halve the training iterations for the aerial and ground stages to 30k and 20k, respectively, while simultaneously halving the stopping iteration of the ADC strategy to 15k and 10k, respectively.

5.1.4. Implementation Details

We set $|\mathcal{P}|$ to 80,000, with $|\mathcal{P}^f| = 40,000$ face anchor points and $|\mathcal{P}^e| = 40,000$ edge anchor points, and set the sharp-edge threshold τ to 30° ; use a voxel resolution of 512, and set the geometric regularization weight λ_g to 0.05; generate 100 virtual viewpoints, respectively. During training, images are randomly sampled at each iteration according to phase-specific probability ratios. Specifically, the aerial phase uses aerial:ground:pseudo images at 4:2:1, while the ground phase uses 2:2:1. Because the native data loaders of these

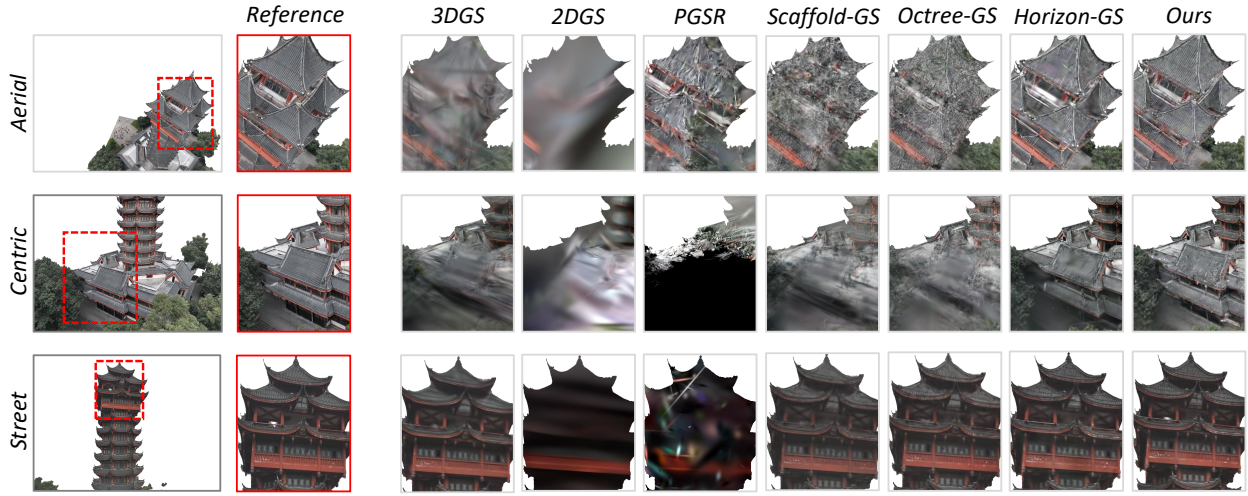


Fig. 11. Jiutian Tower. HeteroArch-GS recovers clear roof-tile textures on the eaves, whereas competing methods produce severely stretched Gaussian primitives and foggy floaters around the tower structure.

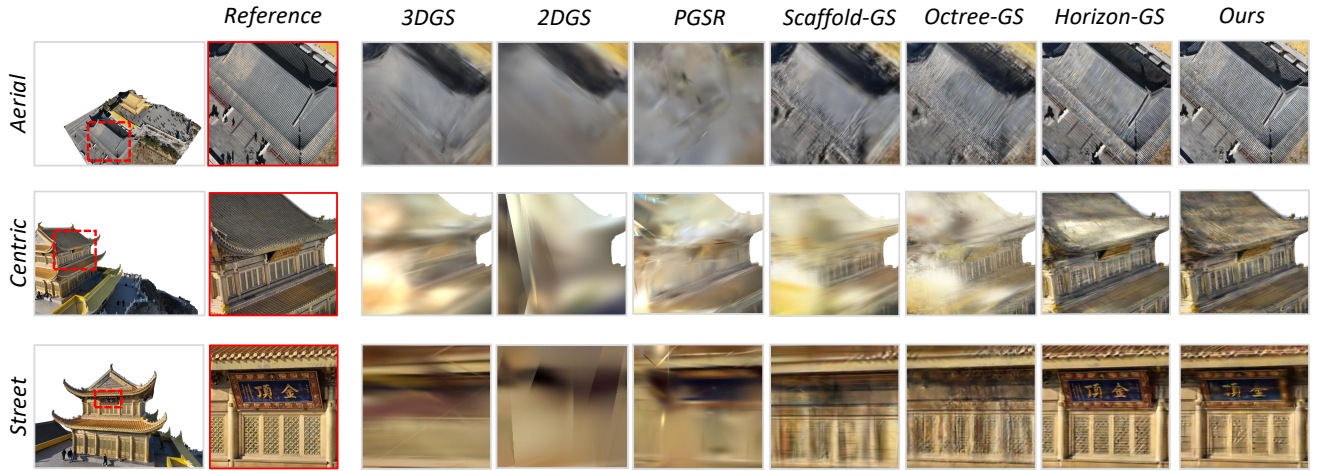


Fig. 12. Mahavira Hall. HeteroArch-GS better maintains eave and roof-tile structures in out-of-distribution object-centric views, reducing the blurry textures and floaters observed in baseline renderings.

baseline methods fail to efficiently process our experimental dataset, we replace them with our proposed strategy and allocate a shared CPU cache size of 8 GB. All experiments are conducted on an NVIDIA RTX 4090 GPU with 24GB of VRAM.

5.2. Visual Quality

As shown in Table 4, our method achieves the best overall rendering performance across the four evaluated scenes. For in-distribution aerial views, it consistently ranks first or second on all three metrics, achieving the lowest LPIPS on all four scenes and the highest SSIM on three of them. On ground views, our method remains highly competitive with Horizon-GS and achieves the best SSIM and LPIPS on *Jiutian Tower*. More importantly, its advantage becomes more pronounced under out-of-distribution object-centric viewpoints. Our method consistently achieves the lowest

LPIPS across all four scenes, while remaining competitive with Horizon-GS in terms of PSNR and SSIM, with each method attaining superior results on different scenes. These results indicate that our method not only preserves high rendering quality on aerial and ground views that follow the training distributions, but also generalizes more reliably to unseen viewpoints.

The qualitative results further show that these gains are closely tied to the structural characteristics of the targets. For facade-dominated buildings with beams, columns, windows, glass, and large planar or curved surfaces, our method preserves local architectural details more faithfully and produces substantially fewer floaters in empty regions. As shown in Fig. 9, this advantage is particularly visible on structural elements such as beams, facade windows, and object boundaries.

Table 5

Geometric quantitative comparison to baselines on *Teaching Building*, *Library*, *Jiutian Tower*, and *Mahavira Hall*. The best and second-best results are highlighted in **red** and **blue**, respectively.

Method	Primitive Count	ROI Ratio $\uparrow$	P@0.001 $\uparrow$	P@0.01 $\uparrow$	R@0.001 $\uparrow$	R@0.01 $\uparrow$	F1@0.001 $\uparrow$	F1@0.01 $\uparrow$
<i>Teaching Building</i>								
3DGS	212008	0.7345	0.0138	0.4650	0.0042	0.4228	0.0064	0.4429
2DGS	45988	0.9145	0.0135	0.4150	0.0011	0.1167	0.0020	0.1821
PGSR	5550919	0.1406	0.0081	0.2195	0.0116	0.3643	0.0095	0.2740
Scaffold-GS	2160370	0.8445	0.0230	0.6539	0.0559	0.8811	0.0326	0.7507
Octree-GS	8963010	0.8939	0.0308	0.7294	0.1293	0.8942	0.0498	0.8034
Horizon-GS	6687100	0.8747	0.0759	0.9577	0.1604	0.9427	0.1030	0.9501
Ours	6296870	0.8342	0.0705	0.9707	0.2728	0.9737	0.1121	0.9722
<i>Library</i>								
3DGS	311268	0.9620	0.0045	0.4534	0.0025	0.3305	0.0032	0.3823
2DGS	85266	0.9674	0.0052	0.4975	0.0009	0.1782	0.0015	0.2624
PGSR	5970252	0.3016	0.0014	0.1452	0.0053	0.3328	0.0023	0.2022
Scaffold-GS	2063090	0.9423	0.0150	0.7192	0.0401	0.7106	0.0218	0.7149
Octree-GS	5128410	0.9573	0.0197	0.7911	0.0731	0.7062	0.0310	0.7462
Horizon-GS	6040940	0.9714	0.0380	0.9524	0.1038	0.7580	0.0557	0.8442
Ours	5168050	0.9546	0.0290	0.9338	0.1308	0.8686	0.0474	0.9000
<i>Jiutian Tower</i>								
3DGS	102135	0.8278	0.0610	0.7863	0.0142	0.8141	0.0230	0.7999
2DGS	15577	0.8048	0.0501	0.6984	0.0017	0.3854	0.0032	0.4967
PGSR	1360679	0.2970	0.0004	0.0080	0.0005	0.0608	0.0005	0.0142
Scaffold-GS	934610	0.4507	0.1318	0.8882	0.1147	0.9600	0.1227	0.9227
Octree-GS	3922300	0.5279	0.1911	0.9579	0.2812	0.9228	0.2276	0.9400
Horizon-GS	1839840	0.4984	0.3032	0.9676	0.2011	0.9626	0.2418	0.9651
Ours	2079960	0.6183	0.2913	0.9838	0.3861	0.9815	0.3321	0.9827
<i>Mahavira Hall</i>								
3DGS	74457	0.5276	0.0498	0.5576	0.0058	0.7616	0.0103	0.6438
2DGS	15048	0.8455	0.0332	0.4637	0.0012	0.3827	0.0023	0.4194
PGSR	958616	0.4393	0.0746	0.4048	0.0797	0.9083	0.0771	0.5601
Scaffold-GS	2530440	0.3604	0.1616	0.8517	0.2132	0.9981	0.1838	0.9191
Octree-GS	10729720	0.3571	0.2662	0.9378	0.4178	0.9920	0.3252	0.9642
Horizon-GS	4045140	0.8221	0.3225	0.9724	0.3402	0.9927	0.3311	0.9824
Ours	3782650	0.8930	0.2971	0.9801	0.5387	0.9927	0.3830	0.9864

Fig. 10 further shows cleaner depth and texture reconstruction in complex facade regions, where baseline methods often blur details or introduce view-dependent artifacts. The main limitation appears in ground-level regions whose geometry is weakly represented by the aerial mesh proxy, as well as reflective facade materials with complex optical effects, which can be observed in the ground-view comparison in Fig. 10. In these cases, Horizon-GS can still retain a slight advantage in some ground views.

For landmarks with dense high-frequency geometry, such as eaves, roof tiles, railings, and layered roof structures, Fig. 11 shows that our method maintains sharper texture patterns and a more coherent arrangement of Gaussian primitive centers. In contrast, competing methods frequently produce blurry roof details, severely stretched Gaussian

primitives, or foggy floaters, especially when rendered from object-centric viewpoints outside the training distribution.

Fig. 12 further confirms this advantage on more complex layered roof structures. Overall, our method provides a stronger balance between rendering fidelity and geometric stability, particularly for complex structures under viewpoint extrapolation.

5.3. Scene Geometry Quality

As shown in Table 5, our method consistently obtains the best or second-best Recall at both thresholds, achieves the highest F1-score at 0.01 threshold on all four scenes, and remains comparable to Horizon-GS in Precision. These results indicate that our method reconstructs more complete scene geometry while maintaining high geometric accuracy.

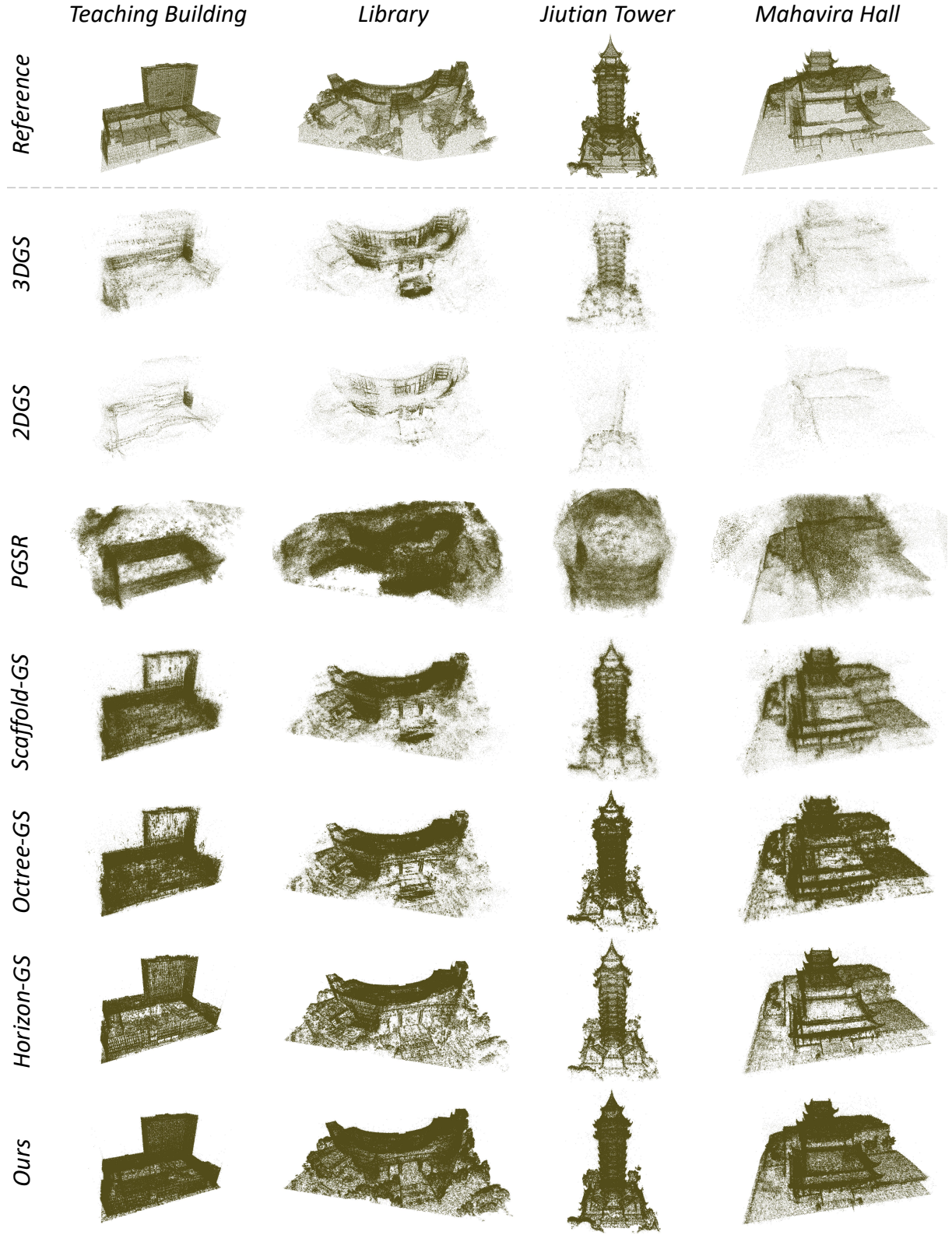


Fig. 13. The geometric qualitative comparison to baselines on Teaching Building, Library, Jiutian Tower, and Mahavira Hall. Existing methods either place unnecessary Gaussian primitives in blank areas or produce sparse primitive distributions in the target area. In contrast, with the aid of the geometric prior provided by oblique mesh, our Gaussian primitives concentrate more effectively on the target area to capture detailed structures while suppressing erroneous floaters.

Although 2DGS and PGSR are designed for geometric reconstruction, they do not consistently outperform 3DGS under joint aerial-ground training. This suggests that RGB observations and internal geometric constraints alone are insufficient to resolve the optimization conflicts caused by large cross-view differences in viewpoint, scale, and visible content. Anchor-based methods such as Scaffold-GS, Octree-GS, and Horizon-GS generally perform better, with Horizon-GS benefiting further from its staged aerial-ground optimization strategy. Our method further introduces the existing aerial photogrammetric mesh as an external geometric prior for Gaussian initialization and regularization, providing more explicit guidance for Gaussian placement, thus reducing local-view overfitting and geometric errors. As a result, our method achieves the most complete reconstruction, reflected by its high Recall and F1-scores. Its Precision is slightly lower than the best result, which may be partly due to noise in the aerial mesh prior. Meanwhile, the required number of Gaussian primitives remains moderate, with more than 60% located within the ROI in all scenes.

As shown in Fig. 13, 3DGS and 2DGS produce relatively sparse point sets and under-reconstructed regions, while PGSR generates many primitives in empty space, leading to background noise and floaters. Scaffold-GS, Octree-GS, and Horizon-GS better follow the target geometry but remain relatively sparse around complex structures such as facades and temple roofs. In contrast, our method distributes Gaussian primitives more consistently over the target surfaces and allocates denser primitives to high-frequency structures, providing more complete geometry and better support for detailed rendering.

5.4. Ablation Studies

5.4.1. Modular analyses

To investigate the individual contributions of the key modules in our proposed method, we evaluate the rendering and geometric accuracy on *Jiutian Tower*. As shown in Table 6 and Fig. 14, the full model achieves the best overall balance, with the highest scores for ground and object-centric views and geometric reconstruction. Specifically, omitting MAI leads to the largest performance degradation across all metrics, indicating that mesh-guided initialization supports the optimization and geometric accuracy of 3D Gaussian primitives. Furthermore, the removal of MPS diminishes the rendering quality in ground and object-centric perspectives, indicating that pseudo views are effective at constraining optimization in sparsely observed or challenging viewpoints. Interestingly, while excluding MGR yields a marginally higher PSNR for aerial views, it results in degraded geometric accuracy and poorer object-centric rendering. This phenomenon implies that MGR prevents the model from overfitting to the dominant aerial training views, trading a negligible decrease in aerial PSNR for a more accurate underlying geometry and superior OOD view synthesis performance.

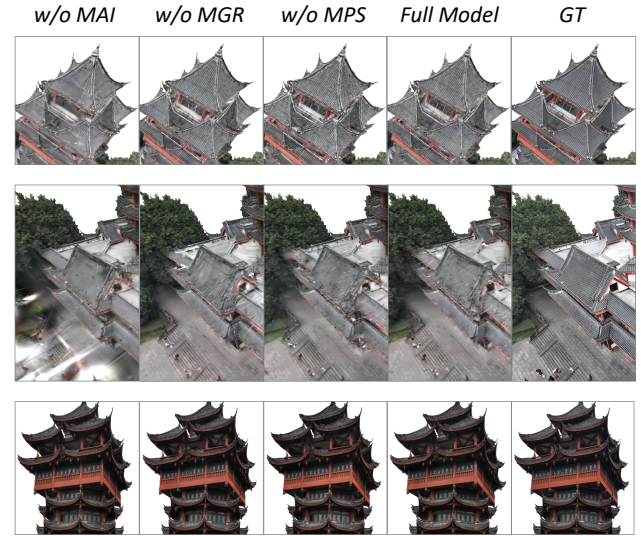


Fig. 14. The visualization of ablation study on key modules tested on *Jiutian Tower*. Rows 1, 2, and 3 show the aerial, object-centric, and ground views, respectively.

Table 6

Ablation study on key modules tested on *Jiutian Tower*. For the Full Model, we report the PSNR and F1-score at an inlier threshold of 0.01. For the ablated variants, we report the performance differences (Δ) relative to the Full Model across aerial, ground, and object-centric views. The most and second most beneficial modules (i.e., whose removal causes the greatest performance drop) are highlighted in red and blue, respectively. Negatively impactful terms (removal improves the metric) are shown in gray.

Method	PSNR $\uparrow$			F1 $\uparrow$
	Aerial	Ground	Object-Centric	
Full Model	24.6294	27.4766	16.5253	0.9827
w/o MAI	-0.4100	-0.7047	-0.6418	-0.0120
w/o MGR	+0.1436	-0.0465	-0.1065	-0.0067
w/o MPS	+0.1281	-0.2839	-0.0756	-0.0015

5.4.2. Geometric Regularization Term

Having established the effectiveness of the overall MGR module, we now isolate and evaluate the individual contributions of $\mathcal{L}_U$, $\mathcal{L}_G$, and $\mathcal{L}_f$ on *Teaching Building*. As shown in Table 7 and Fig. 15, the removal of individual regularization terms reveals a nuanced trade-off between overfitting to the training distribution and generalizing to novel perspectives.

Omitting $\mathcal{L}_U$ results in a slight decrease in overall geometric accuracy. While this omission marginally improves rendering fidelity in the training-dominated aerial and ground views, it reduces PSNR in the out-of-distribution object-centric view (-0.1226 dB). This suggests that without explicit spatial anchoring to the mesh surface, Gaussian primitives may overfit the training images, introducing geometric artifacts when evaluated from novel trajectories.

Interestingly, omitting $\mathcal{L}_G$ leads to the largest aerial-view rendering improvement (+0.1273 dB) and a small ground-view improvement (+0.0152 dB), alongside a slight increase in the F1-score (+0.0025). A similar trend is observed when removing $\mathcal{L}_f$. This phenomenon suggests that constraining 3D Gaussian primitives to flattened shapes aligned with the proxy mesh normal can restrict their volumetric capacity to model complex, view-dependent high-frequency details present in the training images. However, this increased optimization freedom reduces out-of-distribution rendering accuracy. The PSNR decrease in the object-centric view when removing $\mathcal{L}_G$ (-0.1250 dB) suggests that less constrained primitive shapes can introduce geometric ambiguity from unseen angles. Therefore, despite a minor sacrifice in the photometric accuracy of training views, integrating all regularization terms supports rendering quality in the evaluated object-centric views.

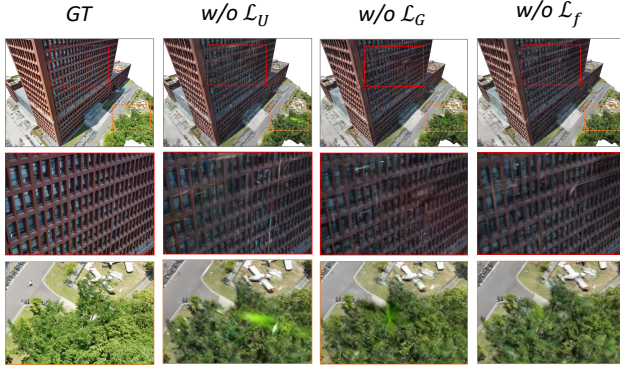


Fig. 15. The visualization of ablation study on geometric regularization terms tested on *Teaching Building*. We visualize the rendering results from object-centric views. The removal of $\mathcal{L}_U$ or $\mathcal{L}_G$ causes notable degradation of rendering quality.

Table 7

Ablation study on geometric regularization terms tested on *Teaching Building*. For the Full Model, we report the PSNR and F1-score at an inlier threshold of 0.01. For the ablated variants, we report the performance differences (Δ) relative to the Full Model across aerial, ground, and object-centric views. The most and second most beneficial modules (i.e., whose removal causes the greatest performance drop) are highlighted in red and blue, respectively. Negatively impactful terms (removal improves the metric) are shown in gray.

Method	PSNR $\uparrow$			F1 $\uparrow$
	Aerial	Ground	Object-Centric	
Full Model	24.0765	20.4808	17.4792	0.9722
w/o $\mathcal{L}_U$	+0.0057	+0.0279	-0.1226	-0.0048
w/o $\mathcal{L}_G$	+0.1273	+0.0152	-0.1250	+0.0025
w/o $\mathcal{L}_f$	-0.0100	+0.0049	-0.0124	+0.0002

5.4.3. Evaluation of Lazy Loading Strategy

We evaluate the contribution of our proposed lazy loading strategy by comparing it against the original GPU and CPU data loading strategies. Since this strategy is a general data-management mechanism applicable to 3DGS-based pipelines rather than a component specific to HeteroArch-GS, we conduct the evaluation on the original 3DGS pipeline to isolate its efficiency gains from other components of our framework. Specifically, 3DGS is optimized for 10k iterations on *Sci-Fi Museum*, which contains 1,016 training images. We report the total end-to-end runtime from data preprocessing to the completion of training, together with peak GPU VRAM, peak CPU RAM, and average GPU/CPU utilization measured during the training stage.

As shown in Fig. 16(a), the proposed lazy-loading strategy substantially reduces the runtime. The original GPU and CPU loading modes both require about 421 s for preprocessing. In contrast, our pipeline preprocesses the images into compact uint8 representations and serializes them to NVMe storage, reducing this stage to about 40 s. More importantly, this preprocessing is performed only once. For subsequent training runs on the same dataset, the serialized data can be reused directly, whereas the original loading schemes must reconstruct their in-memory representation for each run. For the 1,016 training views at a 4 $\times$ downsampled resolution of 1366 $\times$ 989, storing the preprocessed RGB images and foreground masks as uint8 tensors requires approximately 5.20 GB of additional NVMe storage. This storage overhead scales with the number and resolution of the training images. With a 4 GB cache, the full dataset cannot be retained in the host cache, so cache misses still trigger repeated data fetching during training. In contrast, the 8 GB cache is sufficient to hold the entire serialized dataset, effectively eliminating cache misses. This explains why the 8 GB configuration achieves a shorter training time than the 4 GB setting. Overall, the total runtime decreases from 682.8 s and 755.8 s for the original GPU and CPU modes to 303.4 s and 285.1 s for the 4 GB and 8 GB cache configurations, respectively.

As shown in Fig. 16(b), the original GPU mode keeps all the training images resident in GPU memory, which leads to a peak VRAM usage of 22.73 GB. Moving the image data to host memory in the original CPU mode reduces the VRAM requirement to 1.60 GB, but correspondingly shifts the storage burden to CPU RAM, which rises to 22.64 GB. Our lazy-loading strategy avoids keeping the entire dataset resident in either GPU or CPU memory. Instead, only a bounded set of views is maintained in the CPU cache and transferred on demand, which keeps the GPU memory at 1.64 GB and reduces the peak host-memory consumption to 5.97 GB and 7.27 GB for the 4 GB and 8 GB cache configurations, respectively. Our proposed strategy deliberately trades inexpensive secondary storage for lower runtime memory consumption, which is more suitable for large image collections where keeping all views resident in RAM or VRAM is inefficient.

The hardware utilization results in Fig. 16(c) are consistent with the above behavior. The original CPU mode shows

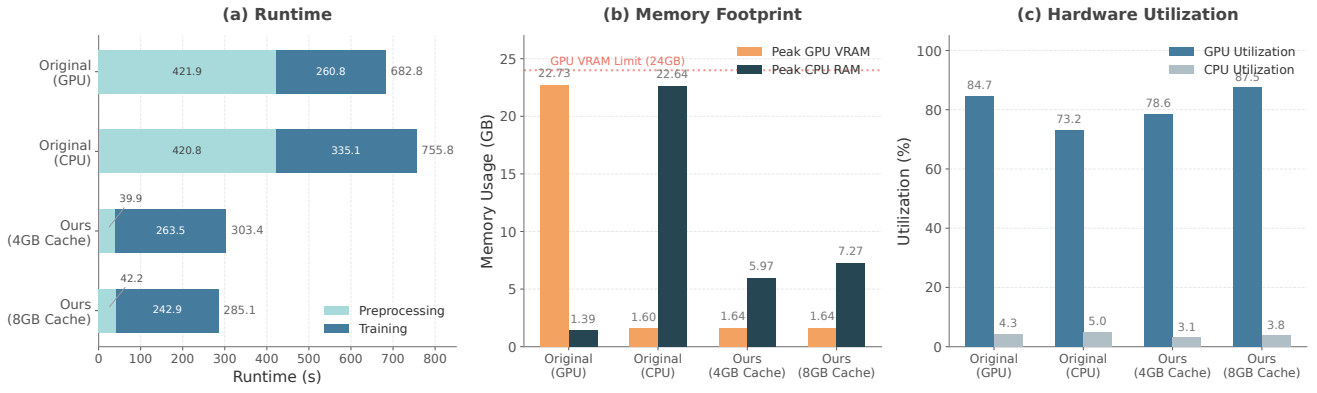


Fig. 16. Ablation study on the contribution of the lazy-loading strategy. We compare the proposed lazy-loading strategy with the original GPU and CPU data-loading strategies using 3DGS optimized for 10k iterations on *Sci-Fi Museum*, which contains 1,016 training images. The runtime is decomposed into preprocessing and training stages. Peak GPU VRAM, peak CPU RAM, and average GPU/CPU utilization are measured during the training stage.

the lowest GPU utilization, suggesting that the GPU is more frequently waiting for data to be supplied from host memory. With the proposed strategy, GPU utilization increases to 78.6% and 87.5% for the 4 GB and 8 GB cache settings, respectively, while CPU utilization remains low at 3.1% and 3.8%.

Overall, the proposed loading strategy enables efficient processing and training of large-scale image datasets under constrained CPU/GPU memory resources by combining reusable NVMe-based serialization with bounded host-memory caching.

5.5. Hyperparameter Sensitivity

To assess the robustness of HeteroArch-GS to hyperparameter choices, we conduct sensitivity analyses on several key settings. These experiments are designed to characterize performance variations under different parameter configurations rather than to identify a globally optimal setting.

5.5.1. Number of Face and Edge Anchors in MAI

We investigate the specific impacts of face-sampled and edge-sampled anchors within the MAI module on both rendering quality and geometric accuracy. By keeping $|\mathcal{P}| = 80,000$ fixed, we evaluate three face-to-edge anchor ratios, $|\mathcal{P}^f| : |\mathcal{P}^e| = 1 : 3$, $1 : 1$, and $3 : 1$, on *Library*.

Table 8 demonstrates that the incorporation of the MAI module, regardless of the specific face-to-edge anchor ratio, improves the overall geometric accuracy of the scene. The F1-score increases from 0.8848 without MAI to above 0.8970 for all three ratios, with only small variations among different configurations. The rendering metrics also remain relatively stable. Increasing the proportion of face anchors slightly improves the PSNR of aerial and ground views, whereas the object-centric performance fluctuates mildly without showing a clear monotonic correlation with the ratio. These results indicate that the proposed method is not highly sensitive to the face-to-edge ratio. As illustrated in Fig. 17, the balanced configuration $|\mathcal{P}^f| : |\mathcal{P}^e| = 1 : 1$ provides a favorable visual trade-off: compared with the edge-dominant setting,

it better represents large planar and curved facade regions, while compared with the face-dominant setting, it preserves fine structural edge details more effectively. We therefore use $1 : 1$ as a simple and balanced default setting, rather than as a configuration selected to maximize a particular metric.

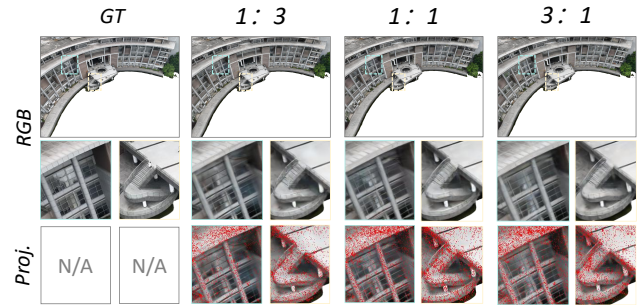


Fig. 17. The visualization of parameter sensitivity study on face-to-edge anchor ratio on *Library*. We visualize the rendering results from aerial views. Proj. denotes the overlay of the rendering result and the image-plane projections of Gaussian primitive centers within the current field of view (red dots). The balanced $1 : 1$ configuration provides a favorable visual trade-off between representing large planar and curved surfaces and preserving fine structural edge details on staircases and columns.

Table 8

Parameter sensitivity study on face-to-edge anchor ratio tested on *Library*. We report the PSNR across aerial, ground, and object-centric views and F1-score at an inlier threshold of 0.01.

Ratio	PSNR $\uparrow$			F1 $\uparrow$
	Aerial	Ground	Object-Centric	
w/o MAI	23.6375	20.5005	20.5062	0.8848
$ \mathcal{P}^f : \mathcal{P}^e = 1 : 3$	23.7631	20.1753	20.6437	0.9001
$ \mathcal{P}^f : \mathcal{P}^e = 1 : 1$	23.8111	20.1780	20.5613	0.8999
$ \mathcal{P}^f : \mathcal{P}^e = 3 : 1$	23.8424	20.2325	20.6591	0.8973

5.5.2. UDF Grid Resolution

We evaluate how the voxel grid resolution of the UDF affects rendering quality and geometric accuracy. In our implementation, the UDF accelerates spatial queries for mesh-guided geometric regularization, and its resolution controls the strictness of the geometric constraints: a higher resolution imposes tighter surface constraints, whereas a lower resolution provides larger spatial tolerance. We conduct this ablation study on *Library* with four grid resolutions: 64, 128, 256, and 512.

As shown in Table 9, the rendering performance across different viewpoints exhibits distinct sensitivities to the UDF grid resolution. For the training-in-distribution aerial view, the PSNR variation across all tested resolutions is extremely marginal, fluctuating within a negligible magnitude of approximately 0.02 dB. This observation indicates that when the proxy mesh’s macroscopic geometry is inherently correct and highly consistent with the observation angle, as the mesh is derived strictly from aerial images, the strictness of the UDF resolution exerts almost no influence on the aerial rendering outcome. Similarly, for the out-of-distribution object-centric views, no distinct correlation pattern emerges. Given that the geometric quality of the proxy mesh in these highly occluded or complex facade regions varies unpredictably, the rendering quality fluctuates randomly within a tight bound of less than 0.1 dB. These negligible variations collectively demonstrate that our model is robust to this hyperparameter.

Ground-view rendering also changes only slightly with UDF resolution. As shown in Fig. 18, reducing the resolution from 512 to 64 increases PSNR from 20.1780 dB to 20.2573 dB. Across the tested field resolutions, PSNR varies by less than 0.1 dB for each view subset, indicating that rendering quality is relatively insensitive to field resolution. This stability suggests that the voxelized field representation with continuous interpolation provides spatial tolerance during geometric regularization and can mitigate the influence of local mesh noise to some extent. Therefore, the results do not indicate a single resolution that simultaneously optimizes both rendering quality and geometric accuracy. We use 512³ as a simple high-resolution default to retain fine geometric representation, rather than because it maximizes every rendering metric.

Table 9

Parameter sensitivity study on UDF grid resolution tested on *Library*. We report the PSNR across aerial, ground, and object-centric views and F1-score at an inlier threshold of 0.01.

Resolution	PSNR $\uparrow$			F1 $\uparrow$
	Aerial	Ground	Object-Centric	
64	23.7892	20.2573	20.6145	0.8583
128	23.8136	20.2459	20.5232	0.8953
256	23.8128	20.2458	20.6101	0.9036
512	23.8111	20.1780	20.5613	0.8999

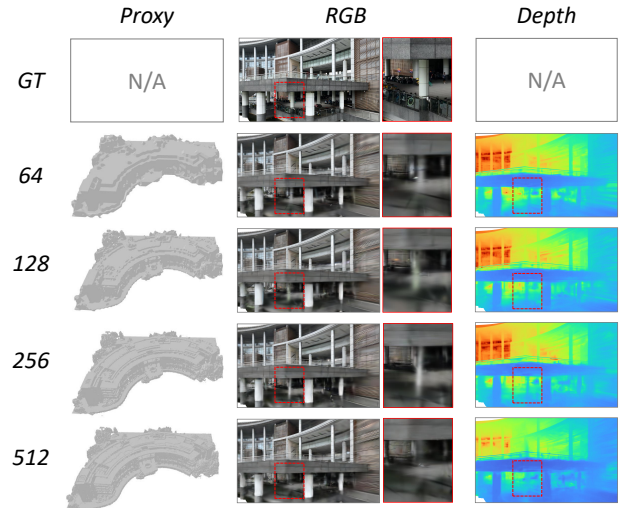


Fig. 18. Parameter sensitivity study on UDF grid resolution tested on *Library*. We visualize the rendering RGB and depth results from ground views. A high-resolution UDF imposes more stringent constraints, coercing 3D Gaussian primitives to overfit noisy surface priors and degrading rendering quality.

5.5.3. LOD Strategy on Aerial-to-Ground Fly-through Rendering

Aerial-to-ground fly-through rendering is an important application scenario for our model jointly optimized with aerial and ground-level imagery. To assess how sensitive this task is to the LOD strategy, we compare the rendering quality along the same continuous camera trajectories with and without LOD enabled. For each landmark in the *AGC Landmarks* dataset, we first specify several key camera poses from ground-level to aerial viewpoints, and then use mesh-guided occlusion relationships and spline interpolation to generate smooth exterior trajectories around the buildings. Each frame is rendered along the trajectory and compiled into a fly-through video, with all video results provided on our project webpage. We use *Mahavira Hall* as a representative case study.

As shown in Fig. 19, enabling the LOD strategy produces cleaner and more stable rendering during the aerial-to-ground transition. Although a few noisy 3D Gaussian primitives remain in extrapolated views outside the training distribution, the rendered frames are generally clean and preserve rich structural details across the trajectory. In contrast, disabling LOD introduces noticeable noise artifacts at several viewpoints, indicating that fly-through rendering quality is sensitive to whether LOD is used. This result demonstrates that dynamically adjusting the level of detail of 3D Gaussian primitives according to the camera-to-scene distance can filter out scale-inappropriate primitives for the current frame, thereby improving visual quality in aerial-to-ground navigation.

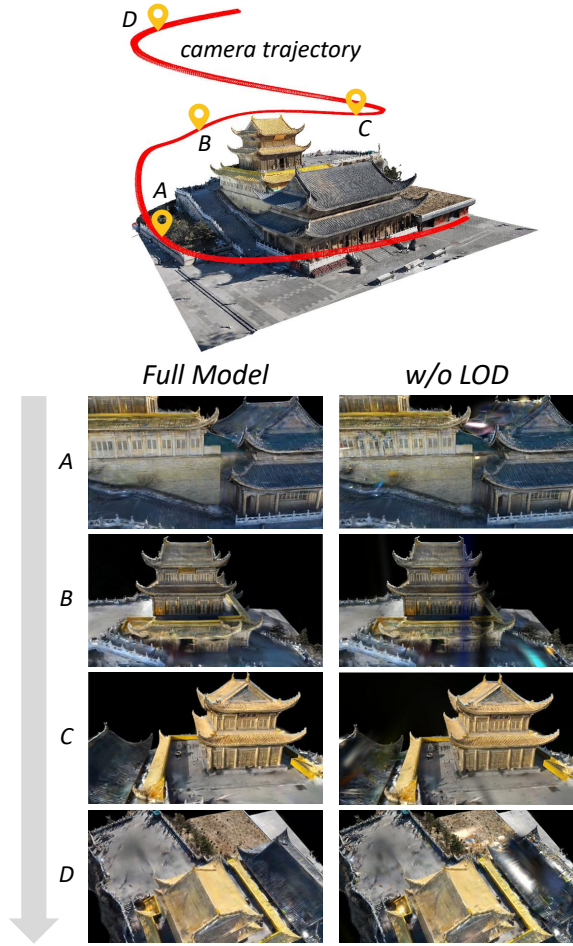


Fig. 19. Parameter sensitivity study on the LOD strategy in aerial-to-ground fly-through rendering tested on Mahavira Hall. We visualize the rendering results along the planned camera trajectory with and without LOD enabled. The LOD strategy effectively filters out noisy 3D Gaussian primitives for each frame, leading to cleaner and more detailed rendering.

6. Conclusion

In this paper, we present HeteroArch-GS, a mesh-guided framework that fuses aerial imagery with supplementary ground-level captures into converged 3D Gaussian Splatting of heterogeneous architectural landmarks. Our target is a practical workflow where oblique photogrammetric meshes already exist from aerial surveys, yet landmark facades remain visually incomplete from the ground—making street-level capture indispensable, but also introducing severe cross-view optimization conflicts. To resolve this, HeteroArch-GS harnesses the aerial mesh as a strong geometric and visual prior. Through mesh-guided anchor initialization, surface-aware regularization, and pseudo-view supervision, Gaussian primitives are constrained to physically plausible manifolds, enabling robust aerial-ground convergence where naive joint training collapses. Furthermore, we construct AGC Landmarks, a real-world benchmark that captures diverse heterogeneous landmarks—ranging from cliffside

temples and lakeside pavilions to modern architectural structures—from aerial, ground, and object-centric viewpoints, explicitly designed to stress-test cross-view rendering in the wild.

Limitations and Future Work. Although our strategies enhance generalization, existing mesh-guided priors still cannot fully guarantee photorealistic rendering in out-of-distribution viewpoints where training information is severely sparse. This suggests a fundamental limit to pure reconstruction-based approaches. In the future, we believe that integrating advanced diffusion generative models (Rom-bach et al., 2022) to augment the extrapolation capabilities of radiance fields may offer a more potent solution for synthesizing photorealistic details in unobserved perspectives, moving beyond the constraints of existing priors. Additionally, the AGC Landmarks dataset also provides geolocation metadata and UTC acquisition timestamps for each image, offering a foundation for incorporating real-world solar illumination models to drive material decomposition and enable physically based photorealistic rendering of out-of-distribution viewpoints.

Acknowledgement

This study was supported in part by the National Natural Science Foundation of China (Project No. U25A20772, 42230102) and the Natural Science Foundation of Sichuan Province under Grant 2026NSFSCZY0054.

A. More Comparison Results

Table 10 reports object-centric rendering metrics on the remaining six scenes. HeteroArch-GS consistently ranks first or second across the three metrics, demonstrating strong generalization to unseen viewpoints. Figure 20 further compares the geometric reconstruction accuracy of all methods. At an inlier threshold of 0.01, HeteroArch-GS achieves the highest F1-score on all six remaining scenes, demonstrating consistently superior geometric accuracy across diverse landmark structures. We further provide additional qualitative comparisons on object-centric views in Fig. 21. Our method produces rendering quality comparable to Horizon-GS while preserving fine structural details and exhibiting fewer noticeable visual artifacts. These qualitative results are consistent with the quantitative comparison above and further demonstrate the effectiveness of our method for cross-view novel-view synthesis. Additional experimental results are available on our project webpage².

References

- Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., Wang, N., Liu, H., Bao, H., Zhang, G., 2025a. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 31, 6100–6111. doi:10.1109/TVCG.2024.3494046.

²<https://vrlab.org.cn/~hanhu/projects/heteroarch-gs>

Table 10

Quantitative comparison across the other six scenes. PSNR, SSIM, and LPIPS are evaluated on the object-centric views. The best and second-best results are highlighted in red and blue, respectively.

Method	Office Building			Sci-Fi Museum			Rubik's Cube Mansion		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS	16.9383	0.5050	0.5554	17.7189	0.6605	0.5150	12.6274	0.4859	0.5122
2DGS	15.6643	0.5013	0.5641	16.9397	0.6432	0.5306	11.6768	0.5090	0.5208
PGSR	13.5054	0.3619	0.5462	14.2552	0.5052	0.5490	11.8169	0.3708	0.5281
Scaffold-GS	16.8888	0.4928	0.5395	17.6271	0.6515	0.5089	12.9867	0.4613	0.4956
Octree-GS	16.6839	0.4827	0.5280	17.4358	0.6464	0.4991	13.0555	0.4415	0.4965
Horizon-GS	19.4573	0.5911	0.3742	20.2518	0.7423	0.3824	16.1388	0.5785	0.3931
Ours	19.1336	0.5946	0.3678	20.0707	0.7320	0.4008	15.3116	0.5779	0.4074

Method	Cinderella Castle			Bodhisattva Statue			Woyun Temple		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS	17.7008	0.5996	0.4250	14.4460	0.5313	0.4427	12.1361	0.3729	0.5779
2DGS	16.0354	0.5916	0.4437	13.3196	0.5234	0.4583	11.5548	0.3677	0.5735
PGSR	14.2198	0.5137	0.4415	14.3178	0.5299	0.4232	11.6120	0.3462	0.5630
Scaffold-GS	17.3965	0.5946	0.4165	14.2293	0.5208	0.4456	12.2880	0.3663	0.5719
Octree-GS	17.4212	0.5855	0.4031	14.5518	0.5164	0.4243	12.8576	0.3691	0.5408
Horizon-GS	21.5066	0.6921	0.3142	16.8466	0.5756	0.3665	14.6888	0.4013	0.4574
Ours	20.8856	0.6905	0.3081	16.8947	0.5742	0.3583	14.7647	0.4047	0.4570

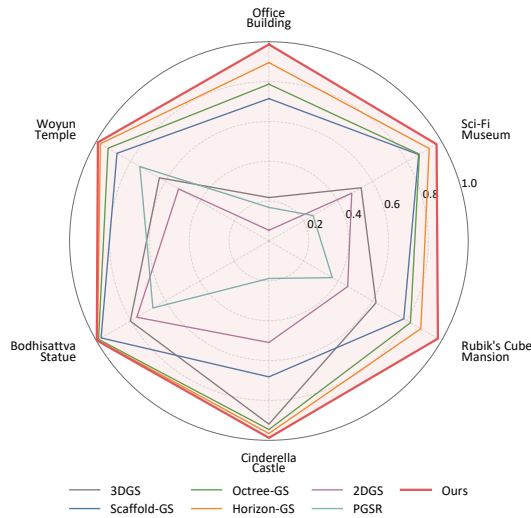


Fig. 20. Geometric F1-score comparison across the remaining six scenes. We visualize the geometric F1-scores of all methods at an inlier threshold of 0.01. Higher values indicate better geometric reconstruction accuracy.

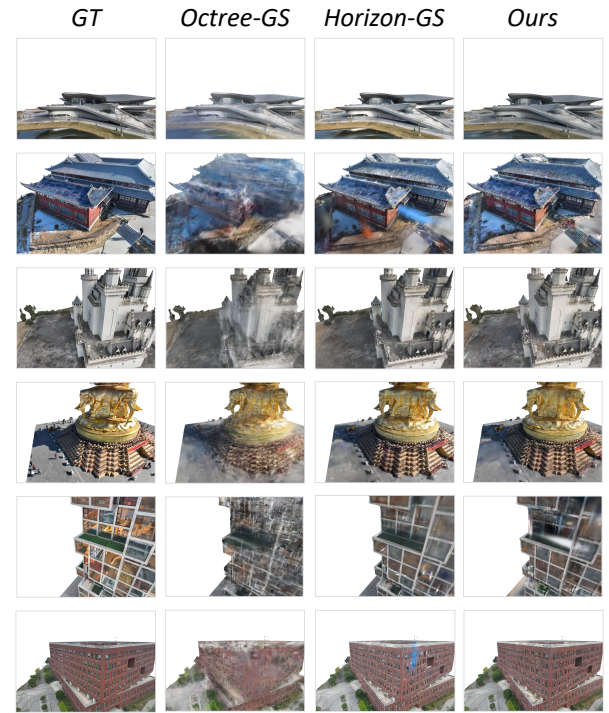


Fig. 21. More rendering visualizations. We show the rendering results from object-centric views which are out of training distribution. Our method is competitive in rendering high-fidelity details with fewer visual artifacts.

- Chen, J., Tian, H., Gao, Y., Zhang, Y., Liu, W., Wu, H., Zhang, H., Huang, W., Liu, J., 2025b. China's national 3d mapping program (3drglm): overall architecture and key technological issues. *Acta Geodaetica et Cartographica Sinica* 54, 636–649. doi:10.11947/j.AGCS.2025.20240115.
- Chen, M., Zhang, Z., Liu, K., Guo, W., Fang, T., Li, W., Zhu, Q., Ge, X., Xu, B., Hu, H., Gong, J., Rao, Y., Wang, Y., 2025c. Robust hierarchical point matching between aerial and ground imagery through depth map-based partitioned attention aggregation. *The Photogrammetric Record* 40, e70008. doi:https://doi.org/10.1111/phor.70008.
- Chung, J., Oh, J., Lee, K.M., 2024. Depth-regularized optimization for 3d gaussian splatting in few-shot images, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 811–820. doi:10.1109/CVPRW63382.2024.00086.
- DJI, 2026. DJI Terra: Make the world your digital asset. DJI Enterprise product page. URL: <https://enterprise.dji.com/dji-terra>. accessed: 2026-05-28.

- Fang, T., Chen, M., Li, W., Ge, X., Hu, H., Zhu, Q., Xu, B., Ouyang, W., 2025. A novel depth information-guided multi-view 3d curve reconstruction method. *The Photogrammetric Record* 40, e70003. doi:https://doi.org/10.1111/phor.70003.
- Google, 2026. Photorealistic 3d tiles. Google Maps Tile API documentation. URL: <https://developers.google.com/maps/documentation/tile/3d-tiles>. last updated: 2026-05-21.
- Hu, D., Minner, J., 2023. Uavs and 3d city modeling to aid urban planning and historic preservation: A systematic review. *Remote Sensing* 15, 5507. doi:https://doi.org/10.3390/rs15235507.

- Hu, Z., Li, W., Yu, J., Liu, M., Ye, J., Chen, P., Huang, H., 2026. 3d building reconstruction from monocular remote sensing imagery via diffusion models and geometric priors. *ISPRS Journal of Photogrammetry and Remote Sensing* 232, 124–137. doi:<https://doi.org/10.1016/j.isprsjprs.2025.11.018>.
- Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S., 2024. 2d gaussian splatting for geometrically accurate radiance fields, in: *ACM SIGGRAPH 2024 Conference Papers*, Association for Computing Machinery, New York, NY, USA. doi:[10.1145/3641519.3657428](https://doi.org/10.1145/3641519.3657428).
- Jiang, L., Ren, K., Yu, M., Xu, L., Dong, J., Lu, T., Zhao, F., Lin, D., Dai, B., 2025. Horizon-gs: Unified 3d gaussian splatting for large-scale aerial-to-ground scenes, in: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26789–26799. doi:[10.1109/CVPR52734.2025.02495](https://doi.org/10.1109/CVPR52734.2025.02495).
- Kaleta, J., Kania, K., Trzcinski, T., Kowalski, M., 2025. Lumigauss: Relightable gaussian splatting in the wild, in: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–10. doi:[10.1109/WACV61041.2025.00011](https://doi.org/10.1109/WACV61041.2025.00011).
- Keinert, B., Innmann, M., Sanger, M., Stamminger, M., 2015. Spherical fibonacci mapping. *ACM Trans. Graph.* 34. doi:[10.1145/2816795.2818131](https://doi.org/10.1145/2816795.2818131).
- Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G., 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 42. doi:[10.1145/3592433](https://doi.org/10.1145/3592433).
- Kerbl, B., Meuleman, A., Kopanas, G., Wimmer, M., Lanvin, A., Drettakis, G., 2024. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Trans. Graph.* 43. doi:[10.1145/3658160](https://doi.org/10.1145/3658160).
- Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V., 2017. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.* 36. doi:[10.1145/3072959.3073599](https://doi.org/10.1145/3072959.3073599).
- Li, Y., Jiang, L., Xu, L., Xiangli, Y., Wang, Z., Lin, D., Dai, B., 2023. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond, in: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3182–3192. doi:[10.1109/ICCV51070.2023.00297](https://doi.org/10.1109/ICCV51070.2023.00297).
- Li, Z., Yao, S., Wu, T., Yue, Y., Zhao, W., Qin, R., ngel F. Garca-Fernandez, Levers, A., Ralph, J., Zhu, X., 2025. Ulsr-gs: Urban large-scale surface reconstruction gaussian splatting with multi-view geometric consistency. *ISPRS Journal of Photogrammetry and Remote Sensing* 230, 861–880. doi:<https://doi.org/10.1016/j.isprsjprs.2025.10.008>.
- Lin, L., Liu, Y., Hu, Y., Yan, X., Xie, K., Huang, H., 2022. Capturing, reconstructing, and simulating: the urbanscene3d dataset, in: *European Conference on Computer Vision*, Springer. pp. 93–109. doi:https://doi.org/10.1007/978-3-031-20074-8_6.
- Liu, Y., Luo, C., Fan, L., Wang, N., Peng, J., Zhang, Z., 2025a. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians, in: *European Conference on Computer Vision*, Springer. pp. 265–282. doi:https://doi.org/10.1007/978-3-031-72640-8_15.
- Liu, Y., Luo, C., Mao, Z., Peng, J., Zhang, Z., 2025b. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes, in: *The Thirteenth International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=a3ptUbuZbW>.
- Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B., 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering, in: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20654–20664. doi:[10.1109/CVPR52733.2024.01952](https://doi.org/10.1109/CVPR52733.2024.01952).
- Pan, L., Barath, D., Pollefeys, M., Schonberger, J.L., 2024. Global structure-from-motion revisited, in: *European Conference on Computer Vision*, Springer. pp. 58–77. doi:https://doi.org/10.1007/978-3-031-73661-2_4.
- Ren, K., Jiang, L., Lu, T., Yu, M., Xu, L., Ni, Z., Dai, B., 2025. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–15. doi:[10.1109/TPAMI.2025.3568201](https://doi.org/10.1109/TPAMI.2025.3568201).
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models, in: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685. doi:[10.1109/CVPR52688.2022.01042](https://doi.org/10.1109/CVPR52688.2022.01042).
- Schonberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M., 2016. Pixel-wise view selection for unstructured multi-view stereo, in: *European conference on computer vision*, Springer. pp. 501–518. doi:https://doi.org/10.1007/978-3-319-46487-9_31.
- Schonberger, J.L., Frahm, J.M., 2016. Structure-from-motion revisited, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113. doi:[10.1109/CVPR.2016.445](https://doi.org/10.1109/CVPR.2016.445).
- Sefercik, U.G., Aydin, I., Nazar, M., 2025. Generation of precise 3d building models for digital twin projects using multi-source data fusion and integration into virtual tours. *ISPRS Open Journal of Photogrammetry and Remote Sensing* 18, 100108. doi:<https://doi.org/10.1016/j.ophoto.2025.100108>.
- Shi, C., Tang, F., Wu, Y., Ji, H., Duan, H., 2025. Accurate and complete neural implicit surface reconstruction in street scenes using images and lidar point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 220, 295–306. doi:<https://doi.org/10.1016/j.isprsjprs.2024.12.012>.
- Shi, Z., Fan, Z., 2025. Optimized caching strategy: A hybrid of least recently used and least frequently used methods, in: *Proceedings of the 2025 5th International Conference on Computer Network Security and Software Engineering*, Association for Computing Machinery, New York, NY, USA. p. 133–140. doi:[10.1145/3732365.3732389](https://doi.org/10.1145/3732365.3732389).
- Shirur, Y.J.M., Sharma, K.M., A., A., 2018. Design and implementation of efficient direct memory access (dma) controller in multiprocessor soc, in: *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, pp. 1–6. doi:[10.1109/ICNEWS.2018.8903991](https://doi.org/10.1109/ICNEWS.2018.8903991).
- Song, S., Qin, R., 2024. A general albedo recovery approach for aerial photogrammetric images through inverse rendering. *ISPRS Journal of Photogrammetry and Remote Sensing* 218, 101–119. doi:<https://doi.org/10.1016/j.isprsjprs.2024.09.001>.
- Stamnes, G., 2026. 3d gaussian splatting in reality capture workflows. *Photogrammetric Engineering & Remote Sensing* 92, 11–15. doi:<https://doi.org/10.14358/PERS.92.1.11>.
- Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D., 2020. Scalability in perception for autonomous driving: Waymo open dataset, in: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451. doi:[10.1109/CVPR42600.2020.00252](https://doi.org/10.1109/CVPR42600.2020.00252).
- Turkulainen, M., Ren, X., Melekhov, I., Seiskari, O., Rahtu, E., Kannala, J., 2025. Dn-splatter: Depth and normal priors for gaussian splatting and meshing, in: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2421–2431. doi:[10.1109/WACV61041.2025.00241](https://doi.org/10.1109/WACV61041.2025.00241).
- Wachter, M., Moehle, N., Goesele, M., 2014. Let there be color! large-scale texturing of 3d reconstructions, in: *European conference on computer vision*, Springer. pp. 836–850. doi:https://doi.org/10.1007/978-3-319-10602-1_54.
- Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 600–612. doi:[10.1109/TIP.2003.819861](https://doi.org/10.1109/TIP.2003.819861).
- Xiang, H., Zhang, F., Li, X., Yang, C., Gao, Y., Liu, W., Zhao, L., Li, D., Huang, X., 2026. Gaussianraft: Fine-grained 3d gaussians for efficient large-scale surface reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing* 235, 651–667. doi:<https://doi.org/10.1016/j.isprsjprs.2026.03.020>.
- Xiong, B., Zheng, N., Liu, J., Li, Z., 2024. Gauu-scene v2: Assessing the reliability of image-based metrics with expansive lidar image dataset using 3dgs and nerf. *arXiv:2404.04880*.
- Xu, N., Qin, R., Huang, D., Remondino, F., 2024. Multi-tiling neural radiance field (nerf)—geometric assessment on large-scale aerial datasets. *The Photogrammetric Record* 39, 718–740. doi:<https://doi.org/10.1111/phor.12498>.
- Xu, Z., Niu, Y., Jiang, J., Qin, R., Cui, X., 2025. Pose-graph optimization for efficient tie-point matching and 3d scene reconstruction from oblique uav images. *ISPRS Journal of Photogrammetry and Remote Sensing* 225, 461–491. doi:<https://doi.org/10.1016/j.isprsjprs.2025.04.013>.
- Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L., 2020. Blendedmvs: A large-scale dataset for generalized multi-view

- stereo networks, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1787–1796. doi:[10.1109/CVPR42600.2020.00186](https://doi.org/10.1109/CVPR42600.2020.00186).
- Yao, Y., Zhang, B., Zhang, W., Gao, L., Peng, D., Li, B., Wang, Y., Wang, B., 2026. Arsgaussian: 3d gaussian splatting with lidar for aerial remote sensing novel view synthesis. *ISPRS Journal of Photogrammetry and Remote Sensing* 231, 288–306. doi:<https://doi.org/10.1016/j.isprsjprs.2025.10.022>.
- Zhang, C., Cao, Y., Zhang, L., 2025. Crossview-gs: Gaussian splatting for cross-view scene reconstruction. *Computational Visual Media*.
- Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 586–595. doi:[10.1109/CVPR.2018.00068](https://doi.org/10.1109/CVPR.2018.00068).
- Zhang, S., Ye, B., Chen, X., Chen, Y., Zhang, Z., Peng, C., Shi, Y., Zhao, H., 2024. Drone-assisted road gaussian splatting with cross-view uncertainty, in: 35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25–28, 2024, BMVA. URL: <https://bmvc2024.org/proceedings/452>.
- Zhu, M., Yan, J., Zhong, J., Elfadaly, A., Dai, L., 2026. A comparative review of representative techniques in image-based 3d dense reconstruction. *Photogrammetric Engineering & Remote Sensing* 92, 351–362. doi:<https://doi.org/10.14358/PERS.25-00073R4>.
- Zhu, Q., Wang, Z., Hu, H., Xie, L., Ge, X., Zhang, Y., 2020. Leveraging photogrammetric mesh models for aerial-ground feature point matching toward integrated 3d reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing* 166, 26–40. doi:<https://doi.org/10.1016/j.isprsjprs.2020.05.024>.