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]

[a]*Faculty of Geosciences and Engineering, Southwest Jiaotong University, Chengdu, 611756, Sichuan, China*
[b]*Sichuan Highway Planning, Survey, Design and Research Institute Ltd., Chengdu, Sichuan, China*
[c]*China Railway Design Corporation, China*

ARTICLE INFO

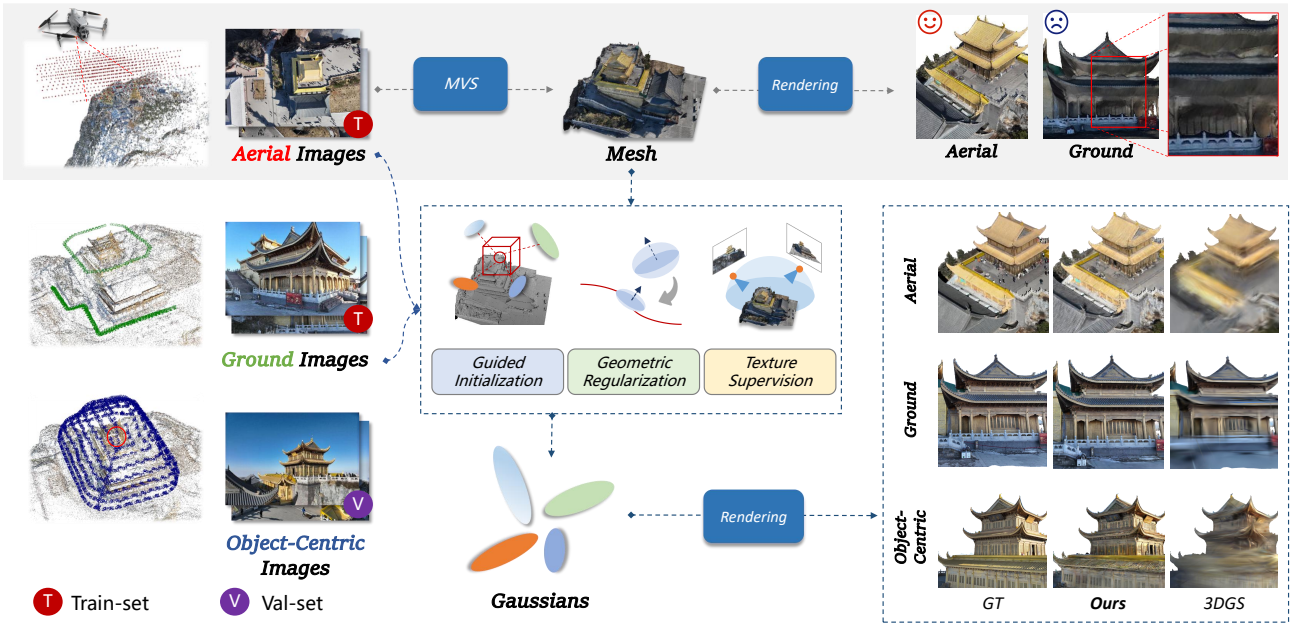
Keywords:
Heterogeneous Architectural Landmarks
3D Gaussian Splatting
Aerial-Ground Rendering
Oblique Photogrammetric Mesh

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 enables robustly converged aerial-ground 3DGS reconstruction of heterogeneous landmarks by leveraging oblique photogrammetric meshes as geometric priors. 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 across all scenes, with surface recall up to twice that of the strongest baseline. Notably, this advantage comes with fewer Gaussian primitives. These results confirm that mesh-guided priors produce more compact and geometrically faithful reconstructions.



*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) (Schonberger 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 (Waechter 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. 1, 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 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. It turns the existing oblique photogrammetric mesh into a strong geometric 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 removes the memory bottleneck of massive image sets, enabling scalable 3DGS training 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.

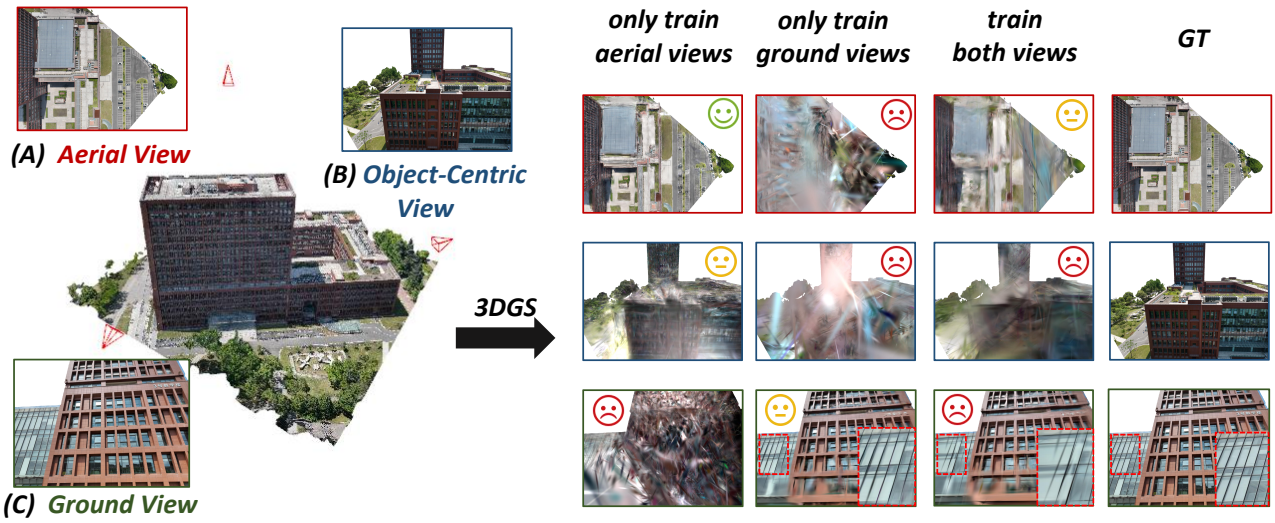


Fig. 1. Failure of naive aerial-ground joint training in 3DGS. We consider three viewpoint types for a target building: (A) aerial, (B) object-centric, and (C) ground views. Models trained only on aerial or ground images render their respective in-distribution views well, whereas naive joint training 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, City-Gaussian (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, 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. 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.

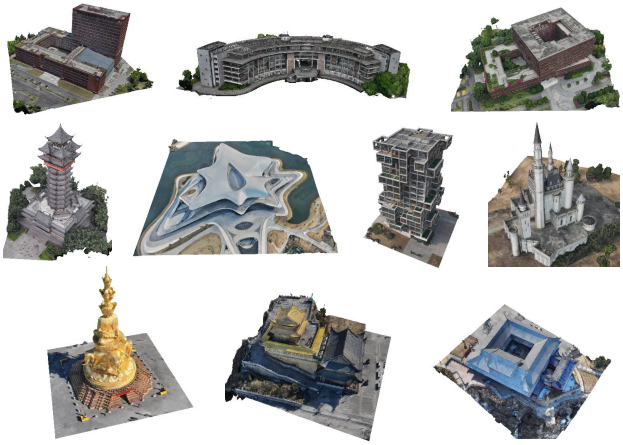


Fig. 2. Textured oblique photogrammetric mesh reconstructed from aerial images.

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. 2, 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, as visualized in Fig. 3.

- *Aerial.* We fly oblique photogrammetry routes to cover the full survey area. Depending on the target

Table 2
Metadata of the captured landmarks. The architectural type, scene feature, and number 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

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 collect low-altitude horizontal and slightly upward views to approximate pedestrian observations. 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.* Guided by the initial model, we use close-range photogrammetry mode to orbit the target at a distance of 10–20 m. This subset provides dense close-range observations for recovering fine geometry and texture details.

Data Processing. Our processing pipeline converts the raw multi-view imagery into registered, foreground-focused inputs for Gaussian splatting. During acquisition, each image is tagged with RTK positioning at approximately 2 cm accuracy, which provides geo-referenced initial poses in a common world coordinate frame, as shown in Fig. 3. In the reconstruction stage, DJI Terra (DJI, 2026) further refines these observations through post-processed kinematic (PPK) correction and joint aerial-ground triangulation, producing aligned camera poses for all image subsets. We then reconstruct a textured oblique photogrammetric mesh from the aerial subset using DJI Terra, whose dense reconstruction, meshing, and texture mapping modules operate on the refined geo-referenced observations, as shown in Fig. 2. The mesh serves as a foreground proxy: we render depth maps from it and remove sky, distant background regions, and pixels with invalid or non-positive depth. Finally, to prepare the data for 3DGS training, we compute the scene center and radius from the aerial and ground camera distribution, transform all camera poses, sparse point clouds, and mesh geometry into a local coordinate system, and normalize each scene to a $[-1, 1]^3$ bounding box. The resulting registered 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. 4, *HeteroArch-GS* starts from an existing photogrammetric mesh and oriented oblique aerial images. To complement the aerial observations, we additionally capture ground images with RTK positioning. These ground images are further refined through PPK correction in DJI Terra (DJI, 2026) and integrated with the aerial images, yielding a unified aerial-ground triangulation result. 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 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 with anisotropic Gaussian primitives. Each Gaussian primitive is indexed by i, with center μ and covariance $\mathbf{\Sigma}$ factorized as $\mathbf{\Sigma} = \mathbf{RSS}^T\mathbf{R}^T$ for stable optimization. During rendering, the primitives covering pixel $\mathbf{x}'$ are sorted front-to-back in

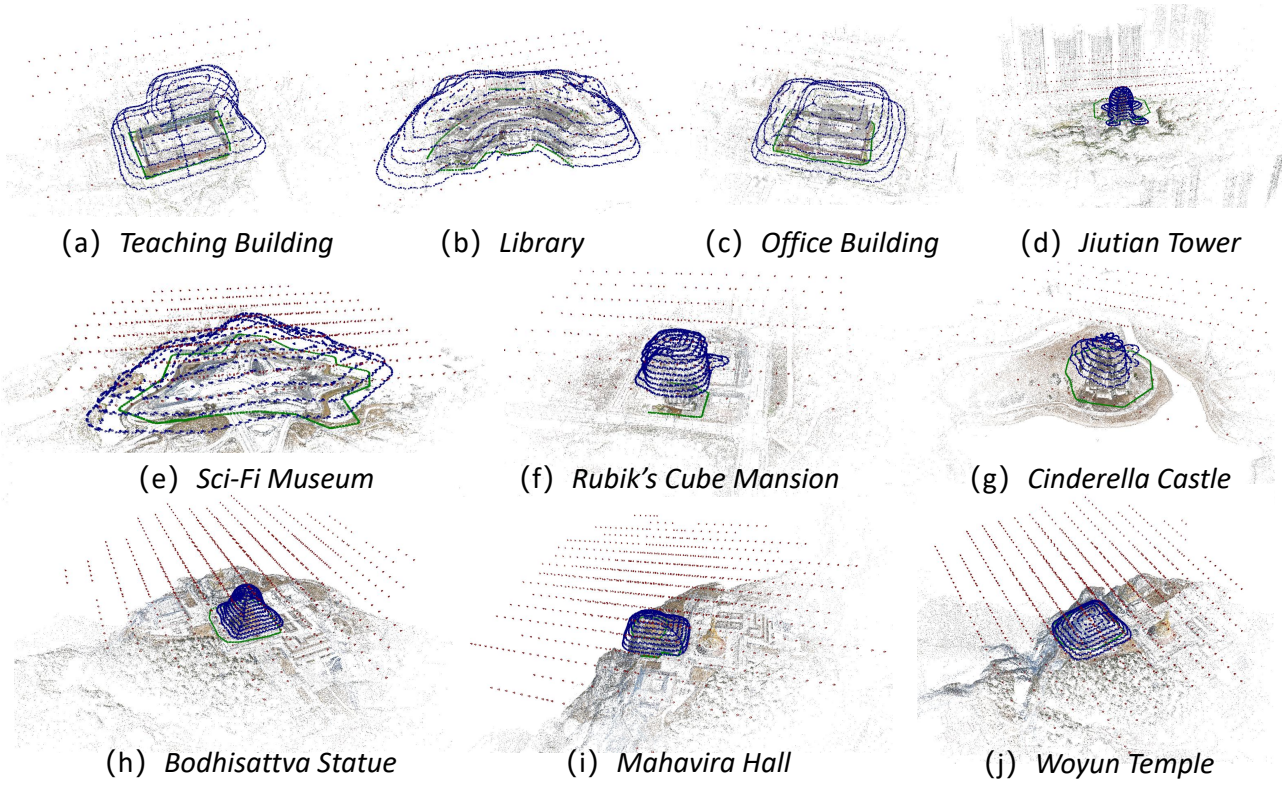


Fig. 3. Camera frustums and sparse point clouds of the AGC Landmarks dataset. Red, green, and blue frustums denote aerial, ground, and object-centric views, respectively.

Table 3
Nomenclature.

Symbol(s)	Meaning
$\mathbf{\Sigma}, \mathbf{R}, \mathbf{S}$	Gaussian primitive covariance.
μ, s, q, c, σ	Gaussian primitive attributes.
$\mathcal{P}$	Anchor point set.
$\mathbf{v}, \mathbf{f}, \mathbf{d}, \delta$	Anchor position, feature, offset direction and offset distance.
$\mathcal{S}$	Surface voxel set.
$\mathbf{O}, \mathbf{g}$	Surface occupancy indicator, voxel coordinate.
$U, \mathbf{G}$	Unsigned Distance Field (UDF) and Distance Gradient Field (DGF).
$\mathcal{L}$	Losses.
i, j	Indices for Gaussian primitives.
r	Indices for cameras.
k	Indices for anchors.

$\mathcal{N}$ and composited by point-based α-blending:

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

Following Scaffold-GS (Lu et al., 2024), we generate Gaussian primitives from sparse anchors $\mathcal{P}$ instead of optimizing all primitives independently. For the k-th anchor at $\mathbf{v}_k$ with feature $\mathbf{f}_k$, lightweight MLPs generate the parameters of its M associated Gaussian primitives:

$$\{\Delta\mu, \mathbf{s}, \mathbf{q}, \sigma, \mathbf{c}\}_i = \mathrm{MLP}(\mathbf{f}_k, \mathbf{d}_k, \delta_k) \tag{2}$$

where each primitive center is $\mu_i = \mathbf{v}_k + \Delta\mu_i$.

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 computed using the smoothed L_1 distance between the rendered RGB image and the target RGB image. The overall loss is defined as:

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_g \mathcal{L}_{geo}, \tag{3}$$

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

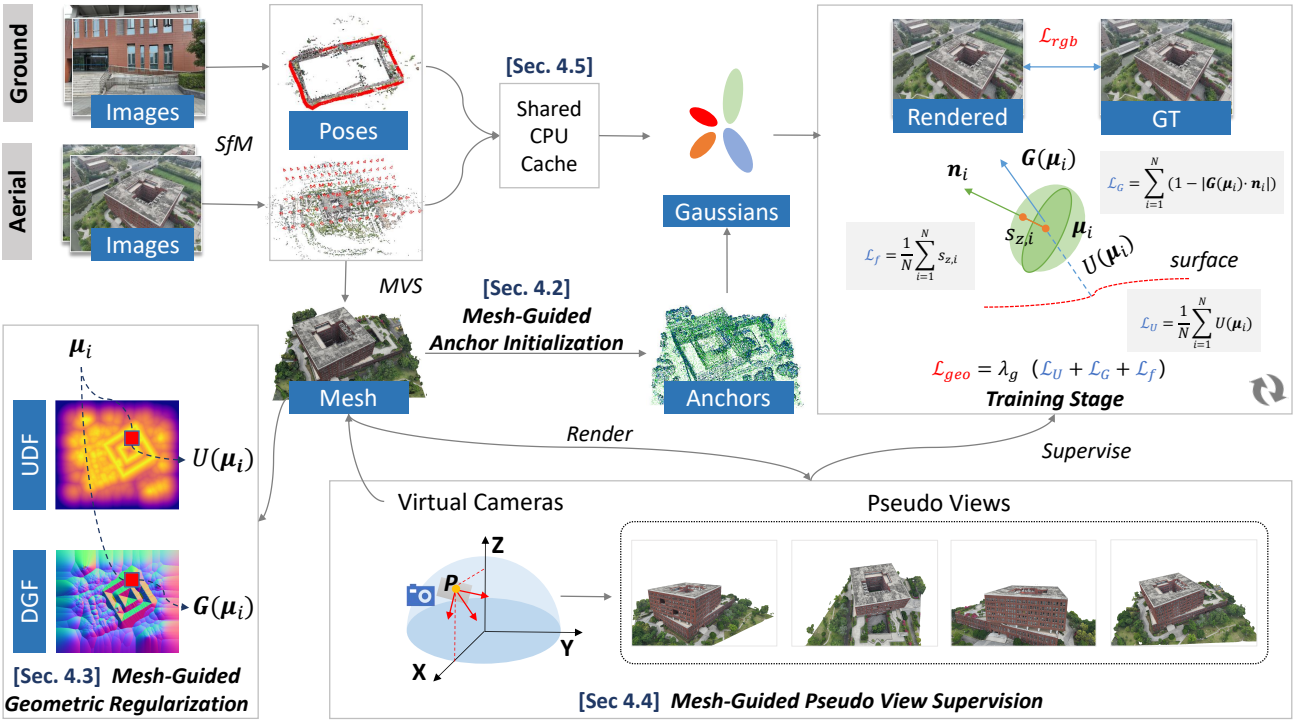


Fig. 4. The pipeline of proposed *HeteroArch-GS*. We start from an existing textured photogrammetric mesh, which serves as a geometric and photometric prior for the subsequent optimization of Gaussian primitives. The mesh guides the initialization of Scaffold-GS Gaussian anchors (Lu et al., 2024), provides 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.

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.

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. 5, the naive SfM point cloud consists of 114,408 points, many of which are distributed haphazardly in meaningless background areas. In contrast, under the parameter settings of $|\mathcal{P}| = 80{,}000$, $|\mathcal{P}^f| = 40{,}000$, $|\mathcal{P}^e| = 40{,}000$, and $\tau = 30°$, our strategy yields a point set that is significantly more concentrated on the target building surface.

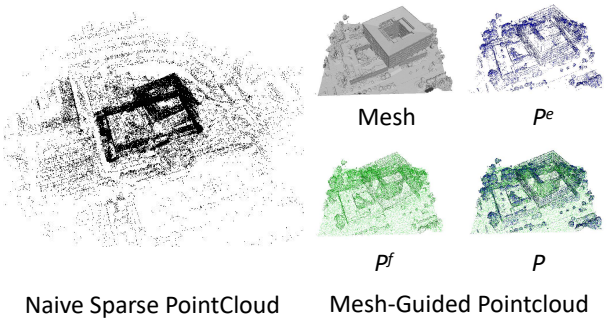


Fig. 5. Naive sparse point cloud and mesh-guided anchor initialization. Mesh-guided sampling places anchors on building surfaces and reduces noisy background points.

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 S parameterized by a spatial resolution N_{res}, and compute voxelized field representations, including 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 $\mathbf{O}(\mathbf{g})$ denote the surface occupancy indicator at voxel coordinate $\mathbf{g}$.

For any voxel $\mathbf{g}$, the nearest occupied surface voxel $\mathbf{g}^*$ is obtained by applying the Euclidean Distance Transform (EDT) over the occupancy grid, and the UDF value $U(\mathbf{g})$ is computed as:

$$\begin{aligned} \mathbf{g}^* &= \arg\min_{\mathbf{u}\in S} \|\mathbf{g}-\mathbf{u}\|_2, \\ U(\mathbf{g}) &= \|\mathbf{g}-\mathbf{g}^*\|_2. \end{aligned} \tag{4}$$

Simultaneously, we construct the DGF to guide surface alignment. The DGF points away from the nearest surface and is formulated as:

$$\mathbf{G}(\mathbf{g}) = \frac{\mathbf{g}-\mathbf{g}^*}{\|\mathbf{g}-\mathbf{g}^*\|_2} \tag{5}$$

As illustrated in Fig. 6, we visualize the continuous UDF and its corresponding DGF derived from the input mesh. During the optimization phase, the network can rapidly query the distance and distance-gradient direction for any arbitrary 3D coordinate through grid sampling, providing a continuous, differentiable, and efficient geometric prior without the computational burden of explicit mesh processing.

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.

$$\mathcal{L}_U = \frac{1}{N}\sum_{i=1}^{N} U(\mu_i) \tag{6}$$

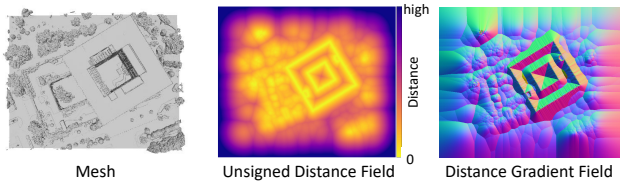


Fig. 6. The visualization of the conversion of a 3D mesh into continuous UDF and DGF. We display the distance and distance-gradient fields on a specific cross-sectional slice captured at the midpoint of the Z elevation.

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 bypass full matrix construction and directly extract $\mathbf{n}_i$ from the third column of $\mathbf{R}_i$:

$$\mathbf{n}_i = \begin{bmatrix} 2(x_i z_i + y_i w_i) \\ 2(y_i z_i - x_i w_i) \\ 1 - 2(x_i^2 + y_i^2) \end{bmatrix} \tag{7}$$

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:

$$\mathcal{L}_G = \frac{1}{N}\sum_{i=1}^{N}(1 - |\mathbf{G}(\mu_i)\cdot\mathbf{n}_i|) \tag{8}$$

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:

$$\mathcal{L}_f = \frac{1}{N}\sum_{i=1}^{N} s_{z,i} \tag{9}$$

The summation is taken over the current set of N active Gaussian primitives. By aggressively minimizing $s_{z,i}$, this photometric-independent geometric prior collapses the ellipsoids into thin, surface-aligned splats, ensuring that the

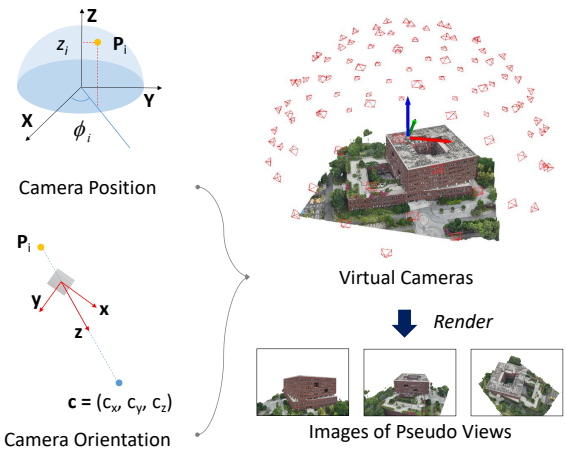


Fig. 7. The processing pipeline for generating RGB images of pseudo views. We sample a set of virtual camera positions uniformly around the textured mesh and render pseudo views at these positions.

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:

$$\mathcal{L}_{geo} = \mathcal{L}_U + \mathcal{L}_G + \mathcal{L}_f \tag{10}$$

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. 7, 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 $\mathcal{I}^r_{pse}$ 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 $\mathcal{I}^r_{pse}$ 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:

$$w_r = \frac{\omega_{max} - \omega_{min}}{h_{max} - h_{min}} \cdot (h_r - h_{min}) + \omega_{min}, \tag{11}$$

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 break this capacity-throughput trade-off and efficiently handle unbounded dataset sizes, we introduce a streaming architecture termed Efficient Lazy Loading. Unlike existing implementations, this architecture completely decouples image preprocessing from the training loop. During optimization, it dynamically loads specified views on-demand while maintaining a capacity-bounded shared CPU cache to stage pending image data, thereby achieving an optimal balance between spatial memory footprint and I/O efficiency. 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 pre-processed 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), and Mesh-Guided Pseudo-View Supervision (MPS).
- **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. For the aerial and ground imagery, one out of every 8 images is allocated to the test set, whereas 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 as representative cases of distinct architectural types, as described in Table 2. The remaining six scenes and their corresponding results are included in the supplementary material[1].

[1] https://vrlab.org.cn/~hanhu/projects/heteroarch-gs/

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 the mesh proxy. 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 (corresponding to 1× and 10× 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 our proposed data loading strategy against the naive implementation to evaluate training speed and memory efficiency when processing large-scale datasets. We record the total training time, GPU peak VRAM, CPU peak RAM, and GPU and CPU utilization.

5.1.3. Comparison Methods

We selected 3DGS (Kerbl et al., 2023), 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, 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 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

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 ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Teaching Building									
3DGS	16.2802	0.5018	0.4828	17.1639	0.6173	0.5005	14.9869	0.5034	0.5193
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
Library									
3DGS	18.0860	0.5346	0.5085	18.1483	0.5875	0.5041	17.0192	0.5201	0.5096
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
Jiutian Tower									
3DGS	20.5508	0.6013	0.3457	27.3534	0.8258	0.1875	14.3015	0.4517	0.5001
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
Mahavira Hall									
3DGS	18.4767	0.6111	0.3784	17.4933	0.5635	0.4740	13.2859	0.4596	0.5429
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

experiments are conducted on an NVIDIA RTX 4090 GPU with 24GB of VRAM.

5.2. Visual Quality

As shown in Table 4, our method delivers a substantial improvement over 3DGS, Scaffold-GS, and Octree-GS in rendering quality. More importantly, it matches the state-of-the-art Horizon-GS on in-distribution aerial and ground views. In out-of-distribution object-centric scenarios, our method establishes a clear advantage over Horizon-GS. This advantage indicates that the proposed framework produces more robust and geometrically coherent reconstructions, with substantially fewer view-specific artifacts and floaters under viewpoint extrapolation.

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. 8, this advantage is particularly visible on structural elements such as beams, facade windows, and object boundaries.

Fig. 9 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. 9. 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. 10 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. 11 further confirms this advantage on more complex layered roof structures. Overall, our method provides a stronger balance between rendering fidelity and geometric

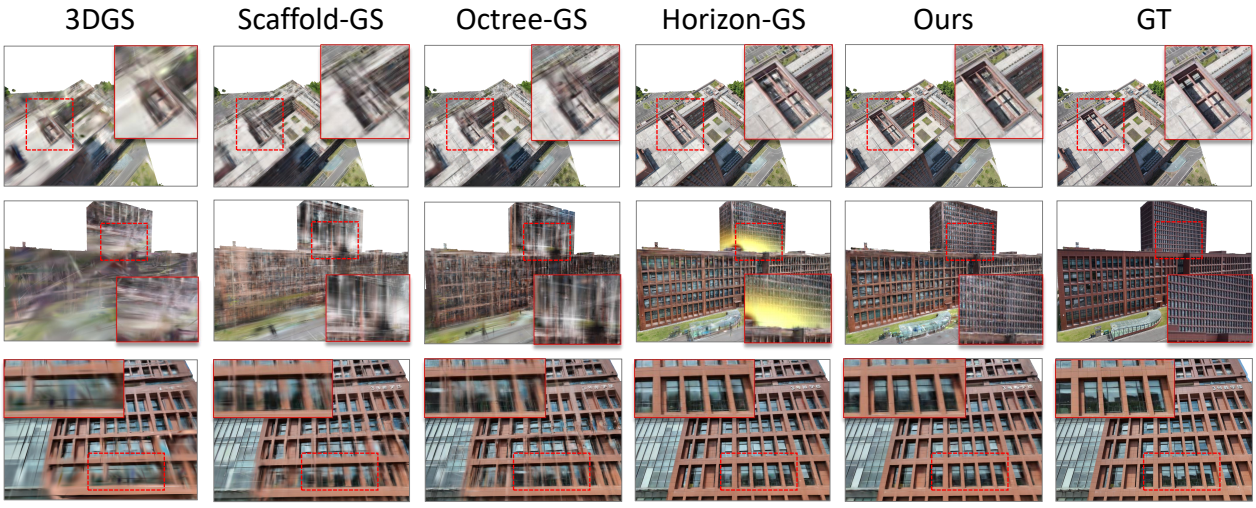


Fig. 8. Teaching Building. HeteroArch-GS preserves beam and facade-window structures more clearly in aerial and object-centric views, with fewer empty-space floaters. Horizon-GS remains slightly stronger on ground-level geometric details and complex glass reflections.

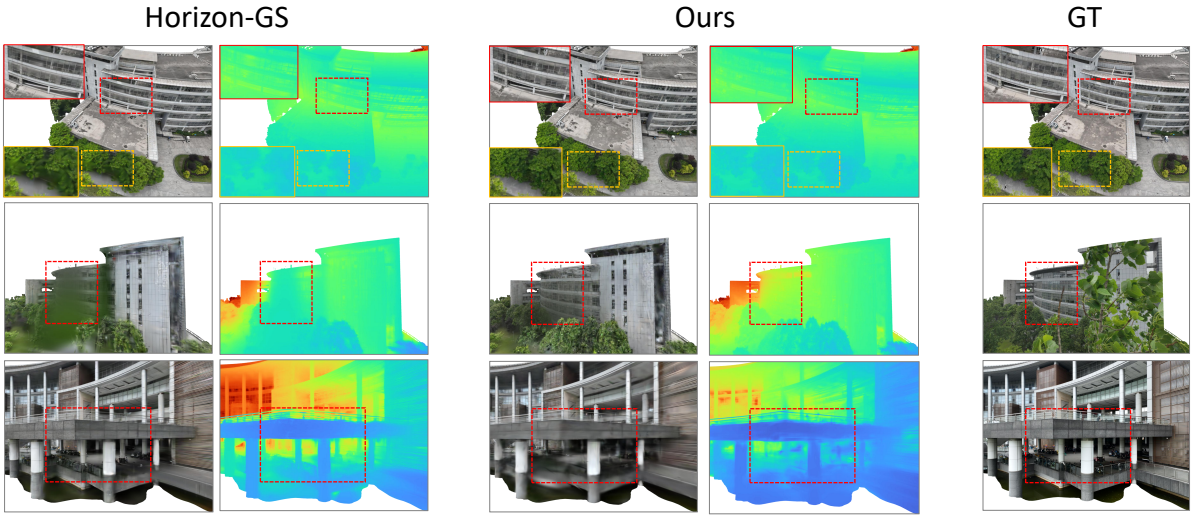


Fig. 9. Library. HeteroArch-GS produces denser and more coherent depth maps in aerial and object-centric views, preserving complex facade textures. Horizon-GS shows blurred texture regions in aerial views and foreground floaters in object-centric views, while our ground view is affected by mesh deficiencies.

stability, particularly for complex structures under viewpoint extrapolation.

5.3. Scene Geometry Quality

As shown in Table 5, our approach achieves the best overall geometric results across all evaluated scenes. It obtains the highest Precision, Recall, and F1-score under both inlier thresholds, with particularly clear gains at the stricter threshold of 0.001. For example, on Library, our R@0.001 reaches 0.4411, clearly higher than the second-best Octree-GS (0.1930). Our ROI Ratio is also consistently close to 1.0 (e.g., 0.9985 on Jiutian Tower and 0.9977 on Teaching Building), while using fewer Gaussian primitives than Octree-GS. These results show that our method improves geometric accuracy without relying on excessive primitive counts.

The qualitative comparison in Fig. 12 further explains where the improvement comes from. Baseline methods either leave many primitive centers in empty space, forming background noise and floaters, or distribute too few primitive centers over the target structure, leading to incomplete geometry. In contrast, our method concentrates the point set formed by Gaussian primitive centers around the foreground architecture. This allocation better follows building surfaces and high-frequency structures, such as eaves, facade elements, and roof details, while suppressing redundant primitives outside the target region.

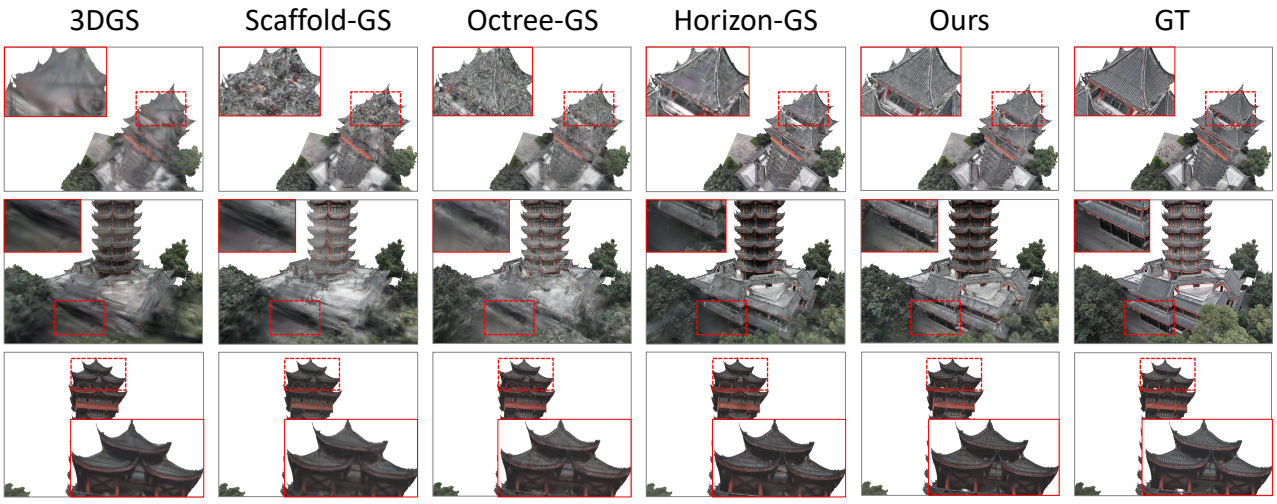


Fig. 10. Jiutian Tower. HeteroArch-GS recovers clear roof-tile textures on the eaves, whereas competing methods produce severely stretched Gaussian primitives and foggy floaters around the tower structure.

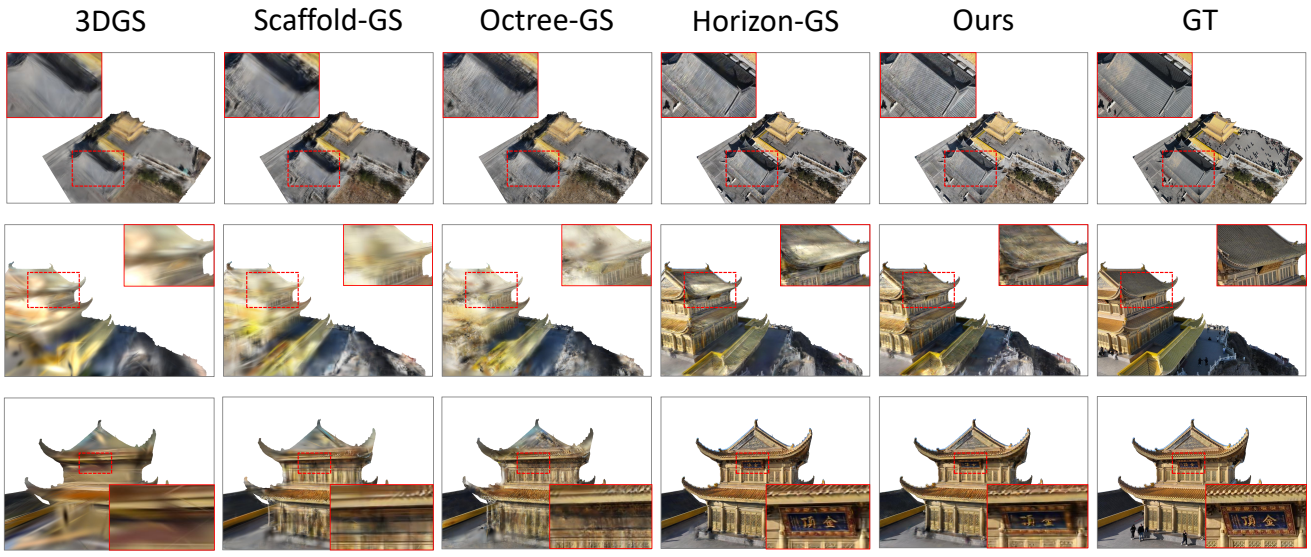


Fig. 11. Mahavira Hall. HeteroArch-GS better maintains eave and roof-tile structures in out-of-distribution object-centric views, reducing the blurry textures and floaters observed in baseline renderings.

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. 13, the full model achieves the best overall balance, particularly excelling in the rendering of ground and object-centric views and geometric reconstruction. Specifically, omitting MAI leads to the most severe performance degradation across all metrics, demonstrating that a robust geometric prior for initialization is critical for the proper convergence and structural integrity 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.

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 ↑	P@0.001 ↑	P@0.01 ↑	R@0.001 ↑	R@0.01 ↑	F1@0.001 ↑	F1@0.01 ↑
				Teaching Building				
3DGS	212,008	0.7786	0.0082	0.4082	0.0027	0.3206	0.0040	0.3591
Scaffold-GS	2,160,370	0.9189	0.0151	0.5875	0.0410	0.8477	0.0220	0.6940
Octree-GS	8,963,010	0.9547	0.0205	0.6682	0.0983	0.8717	0.0340	0.7565
Horizon-GS	6,687,100	0.9932	0.0528	0.9050	0.1516	0.9513	0.0783	0.9275
Ours	6,296,870	0.9977	0.0636	0.9605	0.2980	0.9999	0.1048	0.9798
				Library				
3DGS	102,135	0.9178	0.0350	0.7483	0.0080	0.6004	0.0131	0.6663
Scaffold-GS	934,610	0.5235	0.0769	0.8511	0.0697	0.8762	0.0731	0.8635
Octree-GS	3,922,300	0.6030	0.1181	0.9459	0.1930	0.8526	0.1465	0.8968
Horizon-GS	1,839,840	0.6322	0.1844	0.9517	0.1536	0.9213	0.1676	0.9363
Ours	2,079,960	0.9967	0.2231	0.9848	0.4411	1.0000	0.2963	0.9923
				Jiutian Tower				
3DGS	74,457	0.5853	0.0348	0.5224	0.0041	0.6670	0.0074	0.5859
Scaffold-GS	2,530,440	0.3965	0.1155	0.8249	0.1770	0.9984	0.1398	0.9034
Octree-GS	10,729,720	0.3896	0.1877	0.9234	0.3886	0.9958	0.2531	0.9582
Horizon-GS	4,045,140	0.8758	0.2190	0.9476	0.3070	0.9993	0.2556	0.9727
Ours	3,782,650	0.9985	0.2942	0.9785	0.6238	1.0000	0.3998	0.9891
				Mahavira Hall				
3DGS	311,268	0.9613	0.0055	0.4302	0.0031	0.3547	0.0040	0.3888
Scaffold-GS	2,063,090	0.9462	0.0175	0.6865	0.0473	0.7569	0.0255	0.7200
Octree-GS	5,128,410	0.9616	0.0226	0.7566	0.0873	0.7594	0.0359	0.7580
Horizon-GS	6,040,940	0.9845	0.0428	0.9195	0.1240	0.8452	0.0637	0.8808
Ours	5,168,050	0.9973	0.0394	0.9721	0.1874	0.9986	0.0651	0.9852

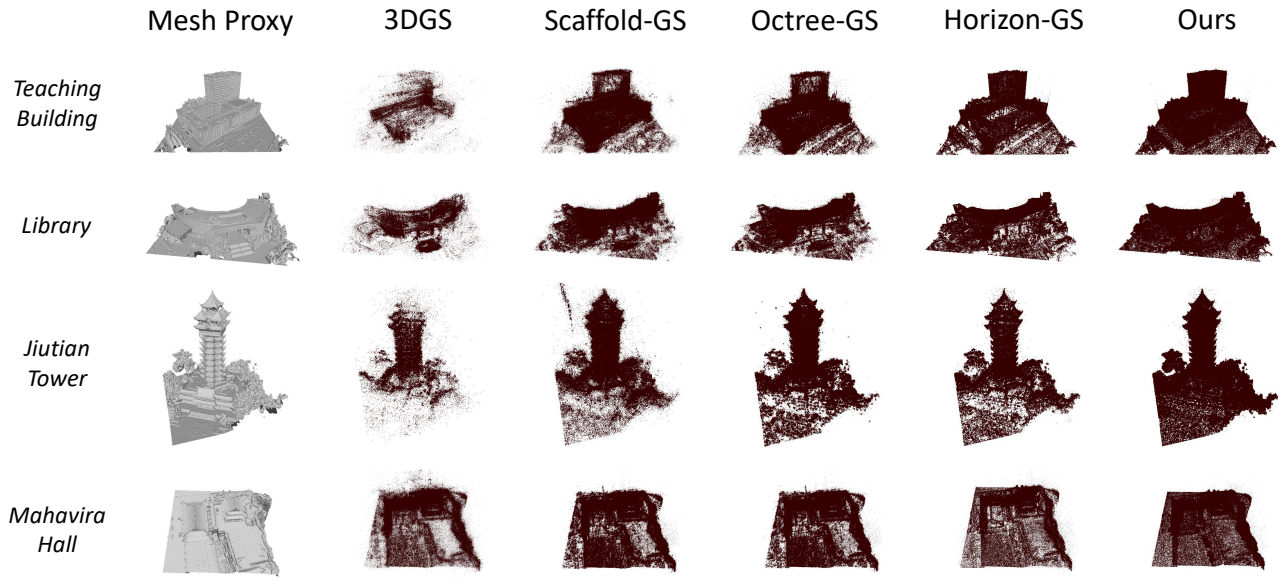


Fig. 12. The geometric qualitative comparison to baselines on Teaching Building, Library, Jiutian Tower, and Mahavira Hall. Existing methods either place unnecessary Gaussian primitives in blank areas or produce sparse primitive distributions in the target area. In contrast, with the aid of the geometric prior provided by oblique mesh, our Gaussian primitives concentrate more effectively on the target area to capture detailed structures while suppressing erroneous floaters.

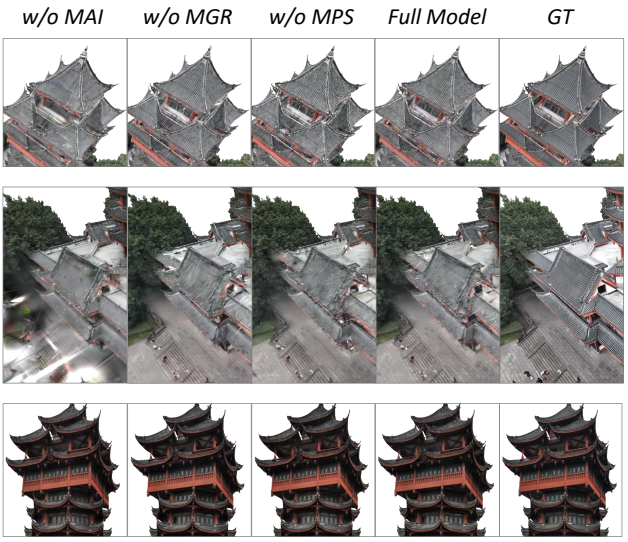


Fig. 13. The visualization of ablation study on key modules tested on *Jiutian Tower*. Rows 1, 2, and 3 show the aerial, object-centric, and ground views, respectively.

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 ↑			F1 ↑
	Aerial	Ground	Object-Centric	
Full Model	24.6294	27.4766	16.5253	0.9891
w/o MAI	−0.4100	−0.7047	−0.6418	−0.0363
w/o MGR	+0.1436	−0.0465	−0.1065	−0.0060
w/o MPS	+0.1281	−0.2839	−0.0756	+0.0008

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. 14, 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 severely penalizes the out-of-distribution object-centric view (−0.1226 dB). This indicates that without explicit spatial anchoring to the mesh surface, Gaussian primitives tend to float freely to perfectly memorize the training images, leading to structural collapse when evaluated from novel trajectories.

Interestingly, omitting $\mathcal{L}_G$ leads to the most significant rendering improvements in the aerial (+0.1273 dB) and ground views (+0.0152 dB), alongside a slight increase in the F1-score (+0.0038). A similar trend is observed when removing $\mathcal{L}_f$. This phenomenon suggests that strictly forcing 3D Gaussian primitives into flattened splats aligned with the proxy mesh normal inherently restricts their volumetric capacity to model complex, view-dependent high-frequency details present in the training images. However, this increased optimization freedom comes at a severe cost to out-of-distribution rendering. The notable performance drops in the object-centric view when removing $\mathcal{L}_G$ (−0.1250 dB) confirm that unconstrained volumetric ellipsoids introduce severe geometric ambiguity from unseen angles. Therefore, despite a minor sacrifice in the photometric accuracy of training views, integrating all regularization terms is essential for maintaining robust structural integrity and high-fidelity rendering across divergent, unseen viewpoints.

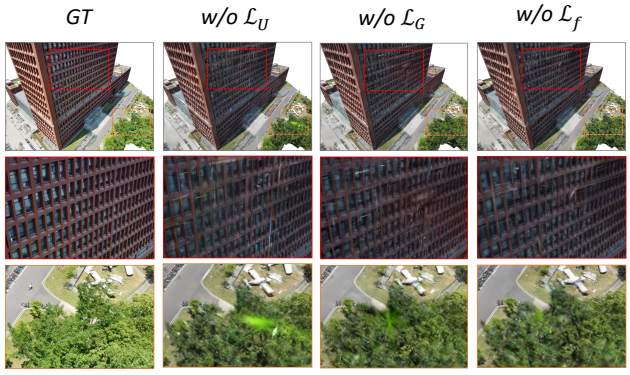


Fig. 14. The visualization of ablation study on geometric regularization terms tested on *Teaching Building*. We visualize the rendering results from object-centric views. The removal of $\mathcal{L}_U$ or $\mathcal{L}_G$ causes notable degradation of rendering quality.

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 ↑			F1 ↑
	Aerial	Ground	Object-Centric	
Full Model	24.0765	20.4808	17.4792	0.9797
w/o $\mathcal{L}_U$	+0.0057	+0.0279	−0.1226	−0.0039
w/o $\mathcal{L}_G$	+0.1273	+0.0152	−0.1250	+0.0038
w/o $\mathcal{L}_f$	−0.0100	+0.0049	−0.0124	−0.0006

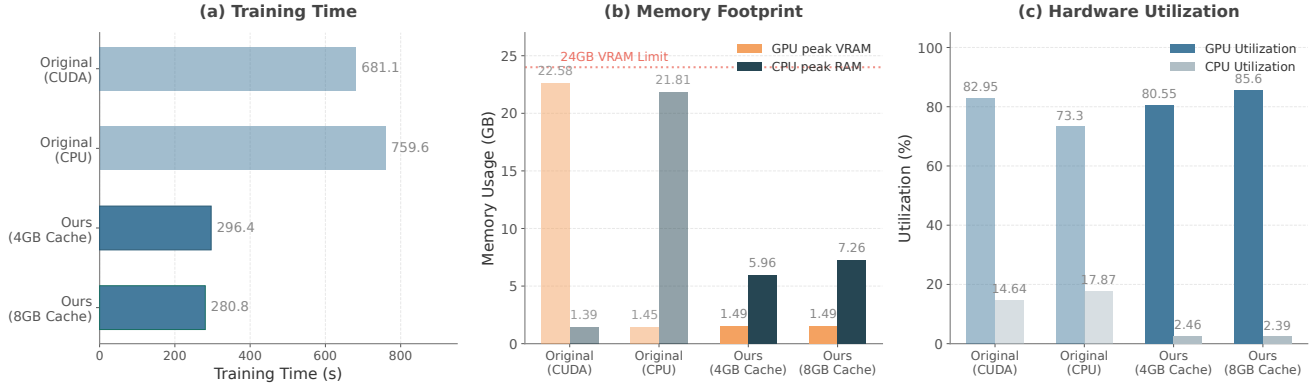


Fig. 15. Ablation study on the contribution of the lazy loading strategy. We evaluate the proposed lazy loading data loading strategy against the original `CUDA` and `CPU` data loading strategies using the 3DGS model, optimized for 10k iterations on the *Sci-Fi Museum* dataset which contains 1016 training images. The training time, GPU peak VRAM, CPU peak RAM, and the utilization of GPU and CPU are reported.

5.4.3. Evaluation of Lazy Loading Strategy

We evaluate the contribution of our proposed lazy loading strategy by comparing it against the original `CUDA` and `CPU` data loading strategies. The experiment is conducted on the 3DGS model, optimized for 10k iterations on *Sci-Fi Museum* which contains 1,016 training images. We report the training time, GPU peak VRAM, CPU peak RAM, and the utilization of GPU and CPU.

As illustrated in Fig. 15(b), the original `CUDA` strategy statically anchors all tensors in the VRAM. At a scale of 1,016 images, the GPU VRAM consumption surges to 22.58 GB, perilously close to the 24 GB hardware limit. Conversely, while the original `CPU` strategy successfully reduces the VRAM footprint to a safe 1.45 GB, it comes at the steep cost of inflating the host RAM usage to 21.81 GB. In contrast, when adopting our proposed lazy loading data loading strategy, the model demonstrates overwhelming spatial efficiency. Under both the 4 GB and 8 GB shared CPU cache configurations, this strategy bounds the VRAM footprint to a minimal 1.49 GB. More crucially, benefiting from `uint8` precision compression and strict capacity quota control, the CPU RAM peaks are drastically compressed to 5.96 GB and 7.26 GB, respectively. This substantiates that the proposed lazy loading data loading strategy successfully decouples the dataset scale from the runtime memory consumption.

As shown in Fig. 15(a), while the original `CUDA` mode requires 681.1 seconds to complete the training, when adopting our proposed lazy loading data loading strategy with an 8 GB cache, the training process consumes only 280.8 seconds, achieving an astonishing 2.4× speedup. As shown in Fig. 15(c), when adopting the proposed lazy loading data loading strategy, the CPU utilization plummets to approximately 2.4%. By completely offloading the CPU and removing it as a bottleneck, the GPU compute units are fully saturated, reaching an 85.6% utilization rate, ultimately leading to a reduction in training time.

To investigate the resilience under stress, we compare the performance when adopting our proposed lazy loading data loading strategy with 8 GB and 4 GB cache capacities. When the cache budget is restricted from 8 GB to 4 GB, the system inevitably experiences more LRU cache misses, thereby triggering disk I/O. However, as observed in Fig. 15(a), the training time incurs only a marginal penalty, slightly increasing from 280.8 seconds to 296.4 seconds. It maintains peak throughput that far exceeds the original implementation while operating on an ultra-compact host RAM footprint of less than 6 GB. This compellingly demonstrates that even under deprived memory conditions, the proposed lazy loading data loading strategy averts crashes and bypasses the prolonged computational stalls typical of original implementation.

5.5. Hyperparameter Sensitivity

5.5.1. Number of Face and Edge Anchors in MAI

We further 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, significantly elevates the overall geometric accuracy of the scene. The F1-score surges from 0.9241 in the baseline without MAI to over 0.9820 across all three ratios. Furthermore, the geometric quality is largely insensitive to the exact ratio, maintaining stably high F1-scores with negligible variance. In terms of rendering fidelity, increasing the number of face anchor points improves the PSNR for both aerial and ground views. In contrast, the rendering performance in object-centric views fluctuates slightly and does not reveal an obvious correlation with the ratio. As clearly demonstrated in Fig. 16, the balanced configuration $|\mathcal{P}^f| : |\mathcal{P}^e| = 1:1$ strikes an effective visual balance. Compared to the edge-dominant ratio of $1:3$, it yields superior rendering quality for expansive planar and curved facade regions. Conversely, compared to the face-dominant ratio of $3:1$, this balanced

configuration preserves structural edge details more meticulously.

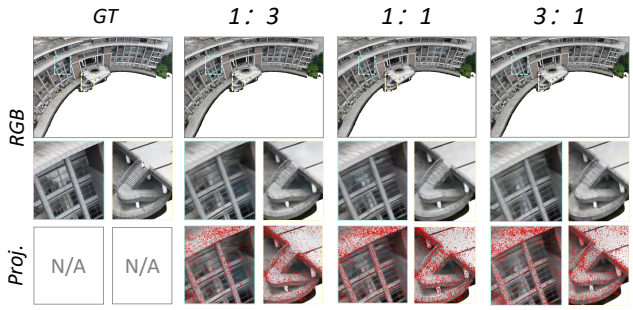


Fig. 16. The visualization of parameter sensitivity study on face-to-edge anchor ratio tested on *Library*. We visualize the rendering results from aerial views. Proj. refers to the overlaid visualization of the rendering result and the image-plane projections of Gaussian primitive centers within the current field of view (red dots). The balanced configuration $|\mathcal{P}^f| : |\mathcal{P}^e| = 1 : 1$ optimizes rendering quality. Unlike skewed ratios, it improves the fidelity of large planar and curved surfaces while simultaneously retaining 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. The best and second-best results are highlighted in red and blue, respectively.

Ratio	PSNR ↑			F1 ↑
	Aerial	Ground	Object-Centric	
w/o MAI	23.6375	20.5005	20.5062	0.9241
$\|\mathcal{P}^f\| : \|\mathcal{P}^e\| = 1 : 3$	23.7631	20.1753	20.6437	0.9824
$\|\mathcal{P}^f\| : \|\mathcal{P}^e\| = 1 : 1$	23.8111	20.1780	20.5613	0.9852
$\|\mathcal{P}^f\| : \|\mathcal{P}^e\| = 3 : 1$	23.8424	20.2325	20.6591	0.9849

5.5.2. UDF Grid Resolution

We evaluate how the voxel grid resolution of the Unsigned Distance Field (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 final 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 the model is robust to this hyperparameter.

However, a noticeable trend emerges in the ground-level view. As shown in Fig. 17, lower grid resolutions, such as 64 or 128, generally yield superior rendering fidelity compared to the highest tested resolution of 512, improving the PSNR from 20.1780 dB to 20.2573 dB. Because the proxy mesh lacks ground-level observational data, it inevitably suffers from topological inaccuracies and severe high-frequency noise at lower building facades. A high-resolution UDF imposes an excessively rigid constraint, coercing the 3D Gaussian primitives to overfit these erroneous surface priors and restricting their spatial optimization freedom. In contrast, decreasing the resolution inherently thickens the zero-level set of the UDF, providing a vital spatial tolerance. Therefore, appropriately reducing the UDF resolution mitigates the adverse effects of noisy geometric priors, leading to visible improvements in ground-level rendering.

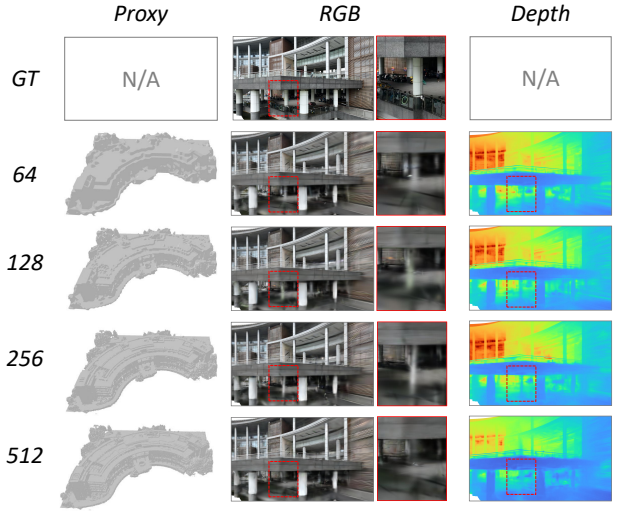


Fig. 17. Parameter sensitivity study on UDF grid resolution tested on *Library*. We visualize the rendering RGB and depth results from ground views. A high-resolution UDF imposes more stringent constraints, coercing 3D Gaussian primitives to overfit noisy surface priors and degrading rendering quality.

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

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. The best and second-best results are highlighted in red and blue, respectively.

Resolution	PSNR ↑			F1 ↑
	Aerial	Ground	Object-Centric	
64	23.7892	20.2573	20.6145	0.9031
128	23.8136	20.2459	20.5232	0.9571
256	23.8128	20.2458	20.6101	0.9762
512	23.8111	20.1780	20.5613	0.9851

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. 18, 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.

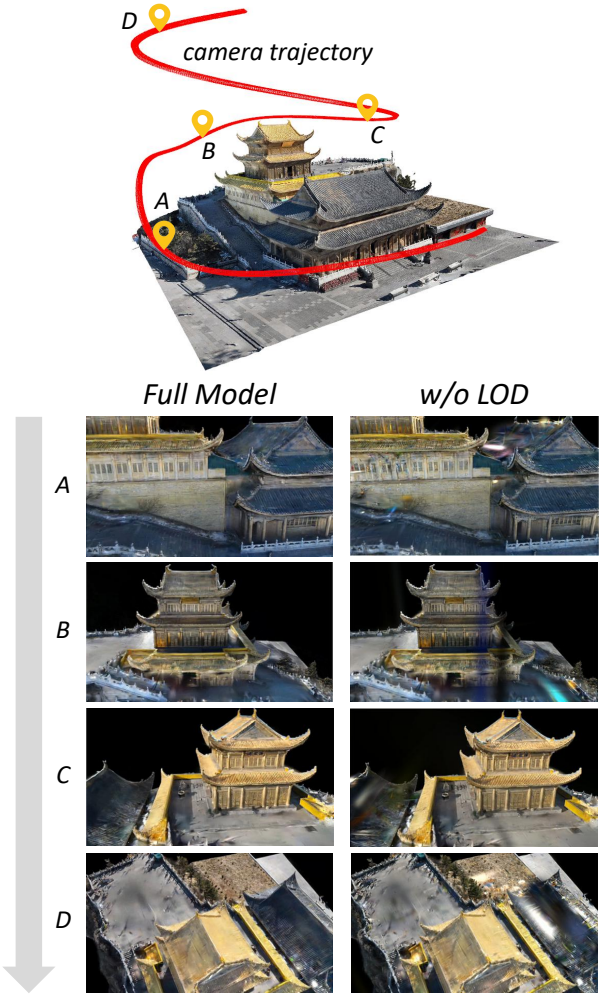


Fig. 18. Parameter sensitivity study on the LOD strategy in aerial-to-ground fly-through rendering tested on *Mahavira Hall*. We visualize the rendering results along the planned camera trajectory with and without LOD enabled. The LOD strategy effectively filters out noisy 3D Gaussian primitives for 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 (Rombach 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.

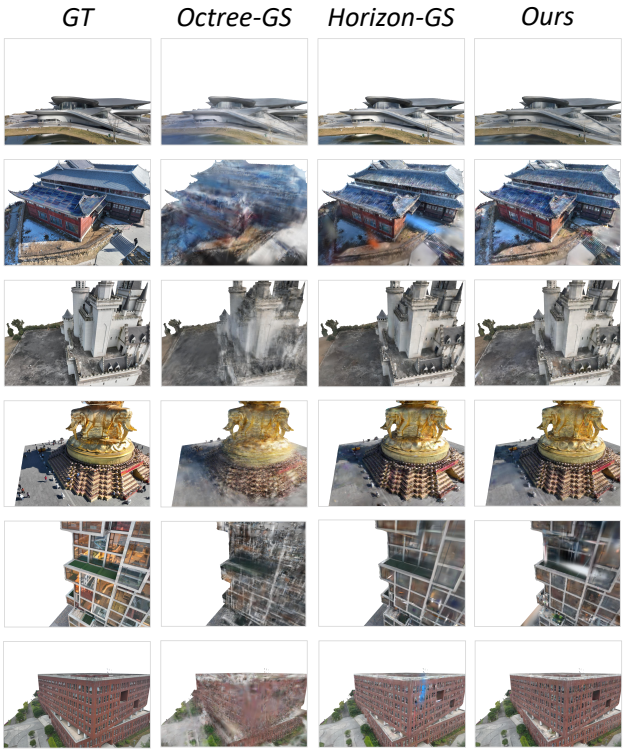


Fig. 19. More rendering visualizations. We show the rendering results from object-centric views which are out of training distribution. Our method is competitive in rendering high-fidelity details with fewer visual artifacts.

Additionally, the AGC Landmarks dataset 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.

A. More Comparison Results

As illustrated in Fig. 19, We show more comparison results on AGC Landmarks. Our method achieves an overall rendering fidelity on par with the state-of-the-art Horizon-GS, while exhibiting noticeably fewer visual artifacts. The other experimental results are available at our project webpage[2].

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.

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.

Chung, J., Oh, J., Lee, K.M., 2024. Depth-regularized optimization for 3d gaussian splatting in few-shot images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 811–820.

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.

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.

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.

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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 26789–26799.

Kaleta, J., Kania, K., Trzciński, T., Kowalski, M., 2025. LumiGauss: Relightable Gaussian Splatting in the Wild, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV).

Keinert, B., Innmann, M., Sänger, M., Stamminger, M., 2015. Spherical fibonacci mapping. ACM Trans. Graph. 34.

Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., 2023. 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics 42.

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 Transactions on Graphics 43.

Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V., 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) 36, 1–13.

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: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3205–3215.

Li, Z., Yao, S., Wu, T., Yue, Y., Zhao, W., Qin, R., Ángel F. García-Fernández, 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.

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.

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.

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.

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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20654–20664.

Pan, L., Baráth, D., Pollefeys, M., Schönberger, J.L., 2024. Global structure-from-motion revisited, in: European Conference on Computer Vision, Springer. pp. 58–77.

[2] https://vrlab.org.cn/~hanhu/projects/heteroarch-gs

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.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684–10695.

Schonberger, J.L., Frahm, J.M., 2016. Structure-from-motion revisited, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4104–4113.

Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M., 2016. Pixelwise view selection for unstructured multi-view stereo, in: European conference on computer vision, Springer. pp. 501–518.

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 , 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.

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. p. 133–140.

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.

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.

Stamnes, G., 2026. 3d gaussian splatting in reality capture workflows. Photogrammetric Engineering & Remote Sensing 92, 11–15.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al., 2020. Scalability in perception for autonomous driving: Waymo open dataset, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2446–2454.

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.

Waechter, M., Moehrle, N., Goesele, M., 2014. Let there be color! large-scale texturing of 3d reconstructions, in: European conference on computer vision, Springer. pp. 836–850.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, 600–612.

Xiang, H., Zhang, F., Li, X., Yang, C., Gao, Y., Liu, W., Zhao, L., Li, D., Huang, X., 2026. Gaussiancraft: Fine-grained 3d gaussians for efficient large-scene surface reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing 235, 651–667.

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.

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.

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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1790–1799.

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.

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: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 586–595.

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.

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.

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.