# ARGUS-GS: SENSOR-GUIDED AERIAL-GROUND GAUSSIAN SPLATTING FOR 3D CITY MODELING WITH MODEL-TO-WORLD ALIGNMENT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

3D city modeling requires both broad scene coverage and accurate alignment with the physical world (*i.e.*, model-to-world alignment), for which image-based reconstruction techniques, such as Gaussian Splatting (GS), provide an effective solution. Vehicle and UAV imagery provide complementary coverage, but their large viewpoint gap complicates cross-stream fusion, while accumulated estimated pose errors along the trajectory can further distort the reconstruction away from its true physical location during coordinate alignment. We present ARGUS-GS, a sensor-guided ground-aerial collaborative 3D city modeling system that produces a unified Gaussian representation with model-to-world alignment. Standard onboard sensor measurements anchor each stream's estimated poses and sparse geometry into the shared physical-world metric frame, while revisit handling limits duplicate Gaussian insertion. We then initialize a shared Gaussian field using inverse-depth search with range-adaptive searching windows, maintaining consistent relative search coverage across viewing distances. Camera poses and Gaussian parameters are then jointly optimized using photometric losses and soft sensor-derived poses, with periodic reseeding of poorly reconstructed views. Simulated urban experiments show $5.8$ dB higher PSNR over baselines at ground-truth camera poses, with $66.8\,\%$ lower depth MAE and $87.5\,\%$ higher surface agreement.

## 1 INTRODUCTION

City-scale 3D models support applications such as augmented-reality street guidance and low-altitude delivery, requiring broad coverage and faithful scene reconstruction. Image-based reconstruction with 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) offers high visual quality and fast rendering (Lin et al., 2024; Liu et al., 2024). To obtain complementary urban coverage, vehicle-mounted cameras capture detailed street-level structures (Yan et al., 2024), while UAV imagery complements them with roofs and other regions poorly visible from the ground (Tang et al., 2025).

However, simply combining the two image streams for reconstruction does not guarantee spatial continuity in the reconstructed urban scene. The large differences in viewpoint and altitude between the two platforms can limit visual overlap, leaving too few reliable feature matches for joint 3D initialization (Figure 1, left). Registering each stream independently avoids this dependence on cross-stream matching, but produces reconstructions with arbitrary local origins, orientations, and scales. The first challenge is to establish a reliable common reference that brings the independently reconstructed streams into a unified metric coordinate system.

Beyond cross-stream alignment, practical applications also require *model-to-world* alignment, which further requires this reconstruction to preserve the scene's physical location, orientation, and scale in the physical world frame. External metric references, such as onboard sensor measurements, can globally transform both reconstructions into a common coordinate system, but only partially correct their misalignment. Errors in feature matching and incremental pose estimation can accumulate unevenly along a trajectory, producing local deviations that a single global transformation cannot generally remove. Consequently, geometry reconstructed from later observations may drift from its physical location. When a previously seen region is observed again (*i.e.*, a revisit), such inconsistent poses can reconstruct the same surfaces at different locations, producing duplicated or layered



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

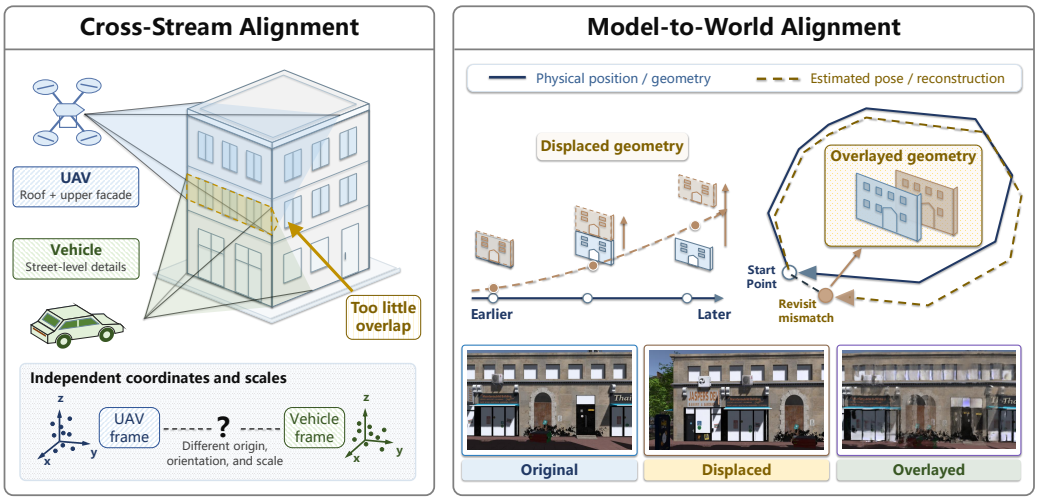


Figure 1: Challenges in aerial-ground collaborative 3D reconstruction.

geometry (Figure 1, right). Subsequent photometric and depth optimization hardly resolves these conflicting representations. The second challenge is therefore to maintain model-to-world alignment not only at the global frame level, but throughout the trajectory, preventing local pose errors from accumulating.

We present ARGUS-GS, a sensor-guided system for physically aligned Gaussian reconstruction from aerial-ground imagery with limited visual overlap. It uses standard onboard GNSS/IMU measurements already available on common platforms, without additional surveying equipment or annotated ground control points. First, we register the two image sequences separately under a shared sensor-derived reference and initialize a common Gaussian field using inverse-depth multi-view search with range-relative searching windows. Second, camera poses and Gaussian parameters are jointly refined using photometric and depth losses together with soft sensor constraints to limit departures from the physical reference. Revisit verification further guides pose correction and controls duplicate Gaussian insertion during 3D initialization, while error-guided reseeding periodically replenishes poorly reconstructed regions during subsequent optimization. In simulated urban scenes, evaluation on urban scenarios at ground-truth camera poses shows a $5.8\,\mathrm{dB}$ PSNR improvement, $66.8\%$ lower depth MAE, and $87.5\%$ higher surface agreement F-score over the baselines.

Our main contributions are:

- **Physically aligned aerial-ground Gaussian reconstruction.** We integrate complementary observations in a shared metric world frame using default onboard sensors, without cross-stream feature matching or annotated control points.
- **3D Initialization adapted to aerial-ground imagery.** We utilize rotation-aware keyframe selection and range-relative search windows to accommodate sharp camera turns and differing observation ranges when constructing the shared Gaussian field.
- **Joint refinement with sensor pose constraints.** We jointly refine camera poses and Gaussian parameters using photometric and depth criteria, with sensor-derived pose references acting as soft regularizers on camera position and orientation.
- **Revisit handling and error-guided reseeding.** We validate revisit regions and control Gaussian insertion to limit duplicated geometry, while periodically reseeding replenishes insufficiently reconstructed regions.

## 2 Preliminaries

**Neural representations for large-scale reconstruction.** Neural radiance fields (Mildenhall et al., 2020) and 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) provide powerful representations for photorealistic 3D reconstruction and novel-view synthesis. To scale these representations to large scenes, prior work improves computational efficiency through compact volumetric representations (Müller et al., 2022; Fridovich-Keil et al., 2022), scene decomposition (Tancik et al., 2022;

Turki et al., 2022; Mi & Xu, 2023), Gaussian partitioning (Lin et al., 2024; Liu et al., 2024), hierarchical level-of-detail structures (Kerbl et al., 2024; Ren et al., 2025), and distributed training (Zhao et al., 2025). Beyond rendering quality, geometry-aware methods improve surface accuracy through implicit surface representations and geometric regularization (Wang et al., 2021; Li et al., 2023b; Guédon & Lepetit, 2024; Huang et al., 2024; Yu et al., 2024; Chen et al., 2025). Recent large-scale Gaussian methods such as CityGaussianV2 (Liu et al., 2025) and UrbanGS (Li et al., 2026) further improve densification, depth estimation, pruning, and parallel training for geometrically accurate reconstruction. These methods primarily focus on scalability and geometric fidelity once camera poses are registered, whereas metric scale recovery and world-coordinate consistency under uncertain camera poses are neglected.

**Aerial-ground collaboration and georeferencing.** Aerial-ground datasets highlight the viewpoint and scale gaps that complicate cross-view reconstruction (Li et al., 2023a; Wang et al., 2026). DRAGON bridges viewpoint gaps through synthesized views (Ham et al., 2024), while UC-GS (Zhang et al., 2024), CrossView-GS (Zhang et al., 2026), and Horizon-GS (Jiang et al., 2025) improve fusion through uncertainty weighting, multi-branch reconstruction, and staged optimization, respectively. Georeferencing introduces an external metric reference, with similarity alignment correcting global translation, rotation, and scale differences (Umeyama, 1991). Photogrammetric workflows also use RTK observations and surveyed control points to establish georeferenced reconstructions (Atik, 2025). GeoRefGS jointly optimizes the Gaussian representation and a geographic transformation using camera-position, relative-geometry, and optional control-point constraints (Hou et al., 2026). However, workflows that depend on dedicated positioning or surveying equipments, and a single global similarity transformation cannot correct pose errors that vary along the trajectory.

# 3 ARGUS-GS FRAMEWORK

## 3.1 PROBLEM FORMULATION

We reconstruct a static urban scene from UAV $\mathcal{U}$ and vehicle $\mathcal{V}$ streams $\mathcal{D}^s = \{(\mathbf{I}_i^s, \mathbf{S}_i^s)\}_{i=1}^{N_s}$, $s \in \{\mathcal{U}, \mathcal{V}\}$, where $\mathbf{I}_i^s$ is an RGB image and $\mathbf{S}_i^s$ contains synchronized onboard GNSS/IMU readings. The goal is to recover a shared Gaussian field $\Theta$ and camera poses $\{\mathbf{T}_i^s\}$ aligned with the physical-world coordinate frame $\mathcal{W}$. The reconstruction must combine complementary observations while preserving the scene's physical scale, location, and orientation, rather than only consistency within an arbitrary local frame. Visual fidelity requires $\mathcal{R}(\Theta, \mathbf{T}_i^s) \approx \mathbf{I}_i^s$, where $\mathcal{R}$ denotes Gaussian rendering; geometric accuracy and physical alignment require reconstructed surfaces $\hat{\mathbf{x}}$ to approximate their physical counterparts $\mathbf{x}$ in $\mathcal{W}$. Since camera poses determine projection and visibility, their accurate recovery is essential to both image reconstruction and model-to-world alignment.

## 3.2 OVERVIEW

ARGUS-GS combines sensor-anchored initialization with joint refinement of camera poses and Gaussians (Figure 2). We respectively select informative keyframes from each stream and register them independently under a common sensor-derived reference, avoiding reliance on cross-stream feature matches (Section 3.3). The resulting sparse geometry and range-adaptive inverse-depth multi-view search initialize a shared Gaussian field in the world frame $\mathcal{W}$ (Section 3.4). The adaptive search window accommodates the different observation ranges of aerial and ground cameras without imposing one fixed absolute interval on both. Joint optimization then combines photometric and depth supervision with soft sensor constraints to refine Gaussian appearance, geometry, and camera poses while preserving physical alignment (Section 3.5). Revisit handling validates poses before inserting geometry, and error-guided reseeding replenishes poorly reconstructed regions during optimization (Section 3.6).

## 3.3 SENSOR-ANCHORED CAMERA REGISTRATION

We convert synchronized GNSS/IMU readings $\mathbf{S}_i^s$ into reference camera centres $\bar{\mathbf{c}}_i^s$ and orientations $\bar{\mathbf{R}}_i^s$ in $\mathcal{W}$ using calibrated sensor-to-camera transformations. These references supply metric scale and a common coordinate frame for both streams even with insufficient visual overlap for direct

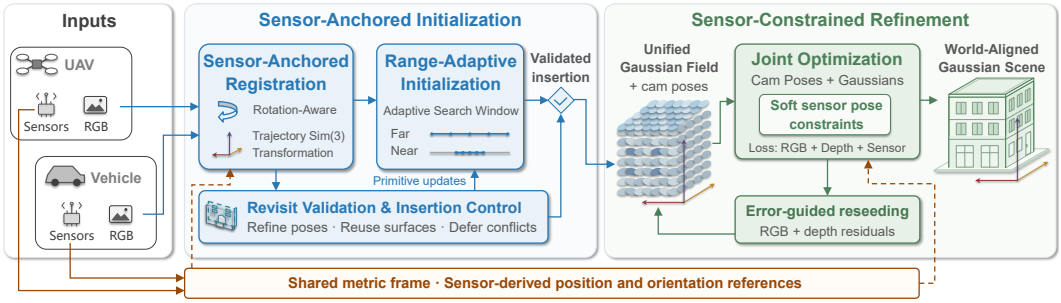


Figure 2: Overview of ARGUS-GS for sensor-guided ground-aerial Gaussian reconstruction.

registration. Within each stream, candidate frame $i$ is matched against the last accepted keyframe and selected when

$$\text{accept}(i) = (m_i > \tau_m) \ \wedge \ \big[(d_i > \tau_d) \ \vee \ (\Delta\phi_i > \tau_R)\big], \tag{1}$$

where $m_i$ is the match count, $d_i$ the median image displacement, and $\Delta\phi_i$ the change in sensor-derived camera orientation. The rotation trigger retains turning views with insufficient image displacement, while the minimum match requirement preserves visual support. For bootstrap keyframes $\mathcal{B}_s \subseteq \mathcal{K}_s$, where $\mathcal{K}_s$ is the selected keyframe set, let $(\widetilde{\mathbf{c}}_i^s, \widetilde{\mathbf{R}}_i^s)$ denote visually estimated local camera centres and orientations. We align them to the sensor references using

$$\mathbf{Q}_s = \text{proj}_{\text{SO(3)}}\left(\sum_{i\in\mathcal{B}_s} \bar{\mathbf{R}}_i^s (\widetilde{\mathbf{R}}_i^s)^\top\right), \quad (a_s, \mathbf{t}_s) = \arg\min_{a>0,\mathbf{t}} \sum_{i\in\mathcal{B}_s} \|a\mathbf{Q}_s\widetilde{\mathbf{c}}_i^s + \mathbf{t} - \bar{\mathbf{c}}_i^s\|_2^2 . \tag{2}$$

Here $\text{proj}_{\text{SO(3)}}$ yields the closest proper rotation $\mathbf{Q}_s$; fixing this orientation-derived rotation allows scale $a_s$ and translation $\mathbf{t}_s$ to be fitted even when nearly collinear camera positions provide weak rotational constraints. The resulting similarity transformation maps both local camera poses and their recovered sparse points into $\mathcal{W}$. Subsequent frames use perspective-$n$-point estimation with random sample consensus (PnP-RANSAC) and incremental bundle adjustment with historical points fixed. The recovered world-frame centres $\mathbf{c}_i^s$ and orientations $\mathbf{R}_i^s$ are projected toward their sensor references, with potential revisits receiving further refinement before Gaussian insertion (Section 3.6).

### 3.4 Range-Adaptive Gaussian Initialization

We triangulate matched features using registered camera poses, retaining points with positive depth and low reprojection error. We extend coverage beyond these sparse locations using multi-view stereo (MVS) with inverse-depth search, then back-project the recovered depths to initialize Gaussian centres. Its initial estimates $\rho_i(\mathbf{u})$ are monocular inverse depths aligned to the triangulated depths in the sensor-anchored coordinate frame. This alignment supplies a metric reference for each search centre; MVS then searches around that estimate to recover additional geometry.

For a fixed search width $\Delta_{\text{abs}}$, relative coverage $\Delta_{\text{abs}}/\rho$ grows with depth ($\rho = 1/z$): a fixed candidate budget poorly resolves distant surfaces in UAV views, while narrowing the shared window restricts depth correction for nearby surfaces observed by the vehicle camera. We instead use a depth-range-relative width with absolute bounds:

$$\Delta\rho_i(\mathbf{u}) = \min(\Delta_{\text{abs}}, \max(\Delta_{\min}, r\rho_i(\mathbf{u}))) . \tag{3}$$

The search interval is $[\rho_i(\mathbf{u}) - \Delta\rho_i(\mathbf{u}),\ \rho_i(\mathbf{u}) + \Delta\rho_i(\mathbf{u})]$, where $r$ controls relative width and $\Delta_{\min}$ and $\Delta_{\text{abs}}$ bound it. Away from these bounds, $\Delta\rho_i/\rho_i = r$ gives comparable relative coverage, concentrating candidates around distant surfaces while retaining a wider absolute correction range nearby. The lower bound prevents the interval from becoming arbitrarily narrow, and the upper bound limits expansion for large inverse depths. Both streams contribute primitives directly to the shared Gaussian field $\Theta$ aligned to $\mathcal{W}$.

### 3.5 Joint Gaussian Optimization with Sensor-Derived Poses

We jointly refine the shared Gaussian field $\Theta$ and keyframe camera poses by sampling equally from the two streams. For a keyframe $i$ from either stream, the loss combines appearance, depth, and

sensor pose constraints:

$$\mathcal{L}_i = \mathcal{L}_{\mathrm{rgb},i} + \lambda_{D,i}\mathcal{L}_{D,i} + \mathcal{L}_{\mathrm{pose},i}. \tag{4}$$

For depth supervision, let $M = \sum_g w_g z_g$ and $A = \sum_g w_g$ be the depth moment and accumulated opacity, where $w_g$ is Gaussian $g$'s alpha-compositing weight and $z_g$ its camera-$z$ coordinate. We use the opacity-normalized readout

$$\hat{z} = \frac{M}{A} \quad (A > 0), \qquad \hat{\rho} = \frac{A}{\max(M, \epsilon_z)}, \tag{5}$$

which removes opacity-dependent scaling from the depth signal. Unlike direct inverse-depth accumulation, this definition separates the estimated surface depth from the total compositing weight along the ray. Let $c_{D,i}(\mathbf{u})$ be depth confidence resampled to the rendering resolution and $m_i(\mathbf{u})$ the valid-observation mask; their product gives $\gamma_i(\mathbf{u}) = m_i(\mathbf{u})c_{D,i}(\mathbf{u})$. The mask is independent of rendered opacity, so low-opacity regions remain subject to depth supervision. The depth loss is

$$\mathcal{L}_{D,i} = \frac{\sum_{\mathbf{u}} \gamma_i(\mathbf{u})|\hat{\rho}_i(\mathbf{u}) - \rho_i(\mathbf{u})|}{\max(\sum_{\mathbf{u}} \gamma_i(\mathbf{u}), \epsilon_w)}. \tag{6}$$

Confidence weighting reduces unreliable pixels' influence, while weight normalization keeps the loss scale comparable across views with different valid coverage. The residual is evaluated against the aligned inverse-depth estimate used for initialization, maintaining a consistent depth convention between seeding and refinement.

To limit pose drift under image and depth fitting, we retain soft sensor constraints on camera position and orientation:

$$\mathcal{L}_{\mathrm{pose},i} = \eta_i \Big[ \frac{\lambda_c}{3} \sum_{d=1}^{3} h_{\beta_c}(c_{i,d} - \bar{c}_{i,d}) + \frac{\lambda_R}{9} \|\mathbf{R}_i - \bar{\mathbf{R}}_i\|_F^2 + \lambda_\theta h_{\beta_\theta}(\theta_i - \bar{\theta}_i) \Big]. \tag{7}$$

Here $h_\beta$ is the smooth L1 penalty, $\eta_i$ ramps the constraint weight, and the additional optical-axis elevation term applies only to the UAV. The angles $\theta_i$ and $\bar{\theta}_i$ denote the optical-axis elevations computed from the optimized and sensor-derived camera orientations, respectively. The sensor references remain soft constraints, allowing image and depth evidence to correct camera poses without freely shifting the model away from its physical reference. Penalty definitions, normalization, and weight schedules are detailed in Appendix C.2.

### 3.6 Revisit handling and error-guided reseeding

**Revisit detection and deferred insertion.** Each stream retains reliable starting keyframes and enables revisit detection after its sensor-derived camera centre exceeds a departure distance from the initial position. Sensor-predicted viewing-frustum overlap identifies candidate revisits, which enter a short keyframe buffer with Gaussian insertion deferred while feature tracking continues.

We match buffered observations to historical landmarks and validate PnP-RANSAC poses using spatially distributed reprojection inliers and consistency across reference views. Confirmed correspondences extend existing landmark tracks, allowing different viewpoints to share the same physical points. Revisit verification therefore depends on shared scene content, without requiring the platform to return to the same camera position or orientation. Starting from sensor-projected poses, local refinement uses reliable historical landmarks, robust reprojection residuals, and soft sensor position and orientation constraints to update buffered poses and new sparse points. We then update triangulation and monocular-depth alignment, using geometrically consistent historical and buffered views for dense search. Insertion requires both a validated pose and multi-view evidence of newly visible or insufficiently represented geometry; adequately represented surfaces reuse existing geometry. Depth conflicts and uncertain correspondences remain pending for rechecking as observations arrive and before offline optimization, rather than triggering insertion solely on frustum overlap.

**Reseeding with appearance and depth residuals.** We jointly rank eligible keyframes using moving-average RGB L1 errors $E_{\mathrm{rgb},i}$ and confidence-weighted log-depth penalties $E_{\mathrm{D},i}$:

$$S_i = E_{\mathrm{rgb},i} + \lambda_{\mathrm{seed}} E_{\mathrm{D},i}. \tag{8}$$

Here $\lambda_{\text{seed}}$ balances appearance and geometry, and the depth penalty compares rendered depth with the aligned monocular reference in log space (Appendix C.3). Depth residuals expose geometric disagreement even when appearance errors are small. Within each selected view, we define the relative depth error and confidence gate as

$$e_i(\mathbf{u}) = |\hat{z}_i(\mathbf{u})\rho_i^0(\mathbf{u}) - 1|, \qquad b_i(\mathbf{u}) = \mathbf{1}[c_{D,i}(\mathbf{u}) \geq \tau_c], \tag{9}$$

where $\rho_i^0$ is the fixed, metrically aligned monocular inverse-depth reference and $\tau_c$ is the depth-confidence threshold. Image-detail deficits increase sampling priority, while large relative depth errors raise it further where $b_i = 1$. A confidence-gated sampling floor also gives texture-poor regions with depth disagreement a chance to receive new primitives. We periodically repeat dense initialization for the highest-scoring views using cached features and depths; all new seeds undergo the same insertion check against updated poses and surviving geometry. Large residuals increase inspection priority without overriding geometric validation. A short cooldown protects new Gaussians from opacity pruning while their parameters adapt; sampling and numerical details are provided in Appendix C.3.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets.** Existing aerial-ground 3D reconstruction benchmarks, including real captures from Horizon-GS (Jiang et al., 2025) and synthetic data from MatrixCity (Li et al., 2023a), do not provide synchronized streams for both platforms, as required by our sensor-guided evaluation. We therefore collect exposure-synchronized RGB and GNSS/IMU data along different routes in urban scenes with varied appearances using a PX4-Gazebo-Unity simulator to study accumulated drift and revisit consistency. Ground-truth camera poses and corresponding depth maps are reserved for evaluation. Our dataset comprises 20 paired UAV-vehicle sequences collected across 12 urban scenes with 60 fps, covering a total area of over $4.9\,\text{km}^2$. In each stream, odd-numbered frames are used for reconstruction and training, while even-numbered frames are reserved as the test set and excluded from the reconstruction pipeline along with their sensor measurements. For evaluation, RGB images and depth maps are rendered at the Unity ground-truth camera poses of the even-numbered frames. Comparing these renderings with the reference RGB and depth therefore evaluates both reconstruction quality and model-to-world alignment.

**Implementation details.** We implement ARGUS-GS in PyTorch and use pretrained XFeat (Potje et al., 2024) for visual feature extraction, and Depth Anything V2 (Yang et al., 2024) for monocular depth estimation. Experiments run on an NVIDIA RTX 4090 GPU, with 100 iterations of joint optimization per keyframe within one dataset.

**Baselines.** We compare ARGUS-GS against original 3DGS (Kerbl et al., 2023), VastGaussian (Lin et al., 2024), Horizon-GS (Jiang et al., 2025), CityGaussianV2 (Liu et al., 2025), UrbanGS (Li et al., 2026), and GeoRefGS (Hou et al., 2026) using the same RGB recordings and sensor measurements for all methods. For methods defined in local coordinates, we align the separately initialized UAV and vehicle camera poses and Gaussian fields to the common world frame using a sensor-derived $\text{Sim}(3)$ transformation before optimization. We evaluate GeoRefGS without annotated ground control points (GCPs) for a fair comparison against ARGUS-GS setting that does not rely on additional surveying equipment.

**Metrics.** PSNR and SSIM measure RGB fidelity and structural similarity at ground-truth test poses, while depth MAE measures absolute camera-$z$ depth error in metres on valid pixel pairs.

For camera-pose accuracy, let $\mathcal{J}$ contain the training frames registered by all compared methods across both streams. For each method, we denote its final camera centres and orientations by $(\mathbf{c}_i, \mathbf{R}_i)$. We compare the final poses after joint optimization directly with simulator ground truth $(\mathbf{c}_i^\star, \mathbf{R}_i^\star)$ in the world frame $\mathcal{W}$:

$$\text{MAE}_{\text{pos}} = \frac{1}{|\mathcal{J}|}\sum_{i\in\mathcal{J}} \|\mathbf{c}_i - \mathbf{c}_i^\star\|_2, \qquad \text{MAE}_{\text{rot}} = \frac{180}{\pi|\mathcal{J}|}\sum_{i\in\mathcal{J}} e_{R,i}, \tag{10}$$

where $e_{R,i} = \arccos\big(\mathrm{clip}\big((\mathrm{tr}[(\mathbf{R}_i^\star)^\top \mathbf{R}_i] - 1)/2, -1, 1\big)\big)$ is the geodesic rotation error in radians, with both orientations using the same optical-camera convention. Position and rotation MAE are reported in metres and degrees, respectively.

Surface agreement compares predicted and reference point clouds obtained by back-projecting their respective test-view depth maps using ground-truth camera poses and intrinsics. For a distance tolerance $\tau$, precision $p_\tau$ is the fraction of predicted points within $\tau$ of a reference point, and recall $r_\tau$ is the fraction of reference points within $\tau$ of a predicted point. Their harmonic mean, $F_\tau = 2p_\tau r_\tau/(p_\tau + r_\tau)$, measures surface agreement. Reference points remain included where predicted depth is missing, meaning missing surfaces reduce recall. We report depth MAE and surface agreement over cumulative camera-depth ranges of 0–20, 0–50, and 0–100 m. Within each range, we evaluate $F_\tau$ at matching tolerances of $0.05$, $0.10$, and $0.25$ m. Surface agreement is reported on $[0, 1]$, with the subscripts of $F_\tau$ indicating tolerance in metres. We combine observations from both streams within each sequence and average sequence-level scores equally; details of surface construction and matching are provided in Appendix B.

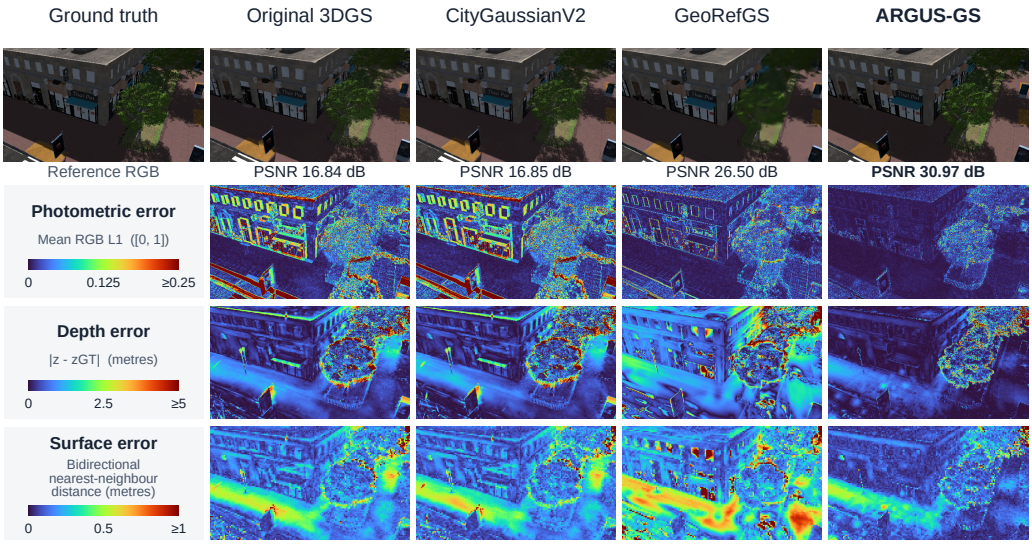


Figure 3: Visualized comparison of evaluation results.

Table 1: Rendering quality, world-frame pose accuracy, and model size.

| Method | PSNR (dB)↑ | SSIM↑ | Position MAE (m)↓ | Rotation MAE (°)↓ | Gaussians (millions)↓ | Model size (MB)↓ |
|---|---|---|---|---|---|---|
| 3DGS | 16.909 | 0.533 | 0.501 | 1.125 | 2.10 | 519 |
| VastGaussian | 17.159 | 0.564 | 0.532 | 1.094 | 6.46 | 1255 |
| Horizon-GS | 17.623 | 0.537 | 0.561 | 1.047 | 5.37 | 1296 |
| CityGaussianV2 | 17.688 | 0.541 | 0.554 | 1.036 | 4.03 | 912 |
| UrbanGS | 17.912 | 0.553 | 0.526 | 0.935 | 3.60 | 950 |
| GeoRefGS | 25.114 | 0.752 | 0.227 | 0.323 | **0.34** | **84** |
| ARGUS-GS | **31.340** | **0.919** | **0.013** | **0.012** | 2.57 | 692 |

Table 2: Geometric accuracy across depth ranges and matching tolerances.

| Method | 0–20 m | | | | 0–50 m | | | | 0–100 m | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Depth MAE (m)↓ | Surface agreement | | | Depth MAE (m)↓ | Surface agreement | | | Depth MAE (m)↓ | Surface agreement | | |
| | | $F_{0.05}$ ↑ | $F_{0.10}$ ↑ | $F_{0.25}$ ↑ | | $F_{0.05}$ ↑ | $F_{0.10}$ ↑ | $F_{0.25}$ ↑ | | $F_{0.05}$ ↑ | $F_{0.10}$ ↑ | $F_{0.25}$ ↑ |
| 3DGS | 0.686 | 0.325 | 0.568 | 0.891 | 1.036 | 0.098 | 0.324 | 0.702 | 1.266 | 0.088 | 0.299 | 0.654 |
| VastGaussian | 0.450 | 0.314 | 0.574 | 0.865 | 1.102 | 0.086 | 0.305 | 0.677 | 1.298 | 0.091 | 0.296 | 0.632 |
| Horizon-GS | 0.473 | 0.309 | 0.583 | 0.864 | 1.053 | 0.116 | 0.340 | 0.726 | 1.336 | 0.073 | 0.303 | 0.669 |
| CityGaussianV2 | 0.465 | 0.336 | 0.568 | 0.873 | 1.017 | 0.124 | 0.337 | 0.696 | 1.327 | 0.096 | 0.326 | 0.627 |
| UrbanGS | 0.436 | 0.297 | 0.614 | 0.823 | 1.001 | 0.127 | 0.360 | 0.683 | 1.315 | 0.103 | 0.314 | 0.683 |
| GeoRefGS | 1.251 | 0.087 | 0.223 | 0.612 | 1.522 | 0.063 | 0.223 | 0.598 | 1.791 | 0.056 | 0.202 | 0.545 |
| ARGUS-GS | **0.213** | **0.541** | **0.815** | **0.977** | **0.357** | **0.337** | **0.675** | **0.911** | **0.438** | **0.306** | **0.628** | **0.863** |

### 4.2 Main Results

Table 1 compares rendering quality and world-frame camera-pose accuracy, while Table 2 reports geometric accuracy across depth ranges and matching tolerances. The low PSNR and SSIM of the non-georeferencing baselines do not necessarily indicate poor local appearance reconstruction, since evaluation at ground-truth camera poses also penalizes spatial misalignment. Even when local textures remain recognizable, displaced scene structures produce large pixelwise discrepancies against the reference images. Figure 3 illustrates this effect, where recognizable buildings coexist with pronounced errors. GeoRefGS achieves lower world-frame position and orientation errors than the other non-georeferencing baselines. However, without GCP annotations that require extra surveying and localization equipment, its overly sparse Gaussian representation limits both visual fidelity and surface reconstruction accuracy. ARGUS-GS combines informative keyframe selection with joint refinement of camera poses and Gaussian field under sensor constraints, improving both physical alignment and geometric support.

Revisited regions expose a further consequence of pose inconsistency. Figure 4 shows the underlying pose discrepancies along one of the looped routes. The baseline trajectories exhibit larger, spatially varying position errors, especially in the revisiting region. When a platform revisits a region, pose errors can place newly reconstructed surfaces at locations inconsistent with the existing geometry. Adding Gaussians at these locations produces duplicated or layered surfaces. Such duplicate geometry can persist during Gaussian refinement because image fitting alone can not resolve conflicting representations of the same physical surface. Figure 5 further compares RGB renderings and their photometric, depth, and surface error maps in a revisited region. The baselines clearly fail to isolate the overlaying effect, even with georeferencing. Our revisit handling addresses this mechanism by validating and refining incoming poses against historical landmarks before inserting new geometry.

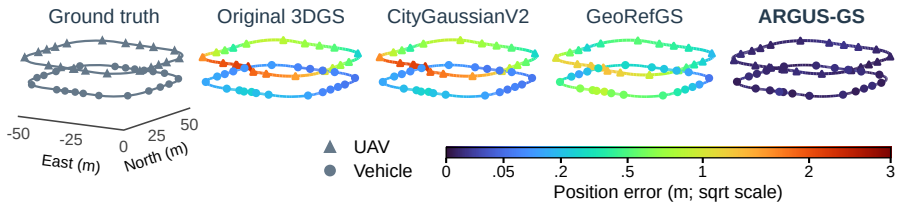


Figure 4: World-frame trajectory errors.

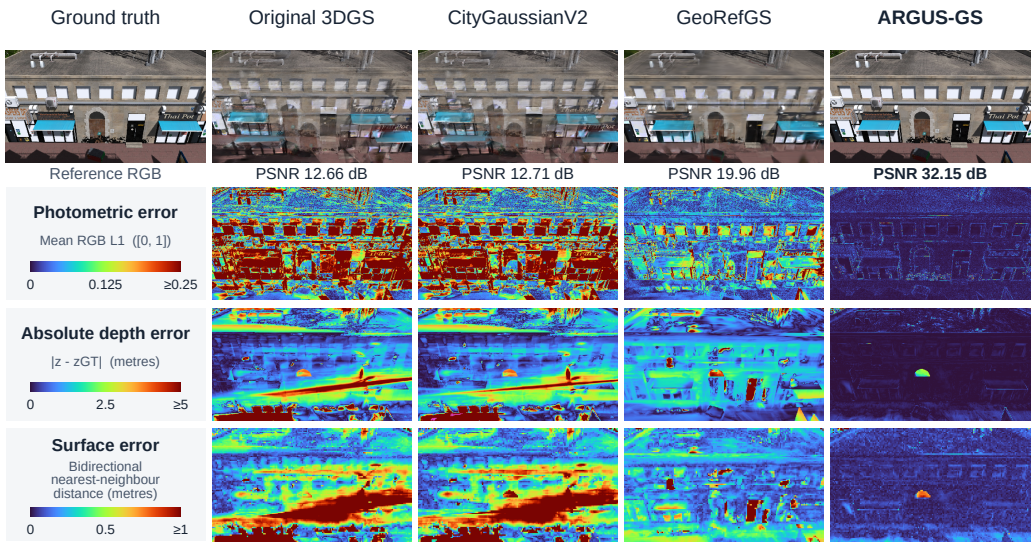


Figure 5: Visualized comparison within a revisited region.

### 4.3 Ablation Studies

We selectively disable or remove key components to assess their contributions to pose accuracy and reconstruction quality, with the results summarized in Table 3.

Across the ablations, removing the sensor pose loss produces both the largest pose errors and the greatest decline in rendering quality. This joint deterioration supports the observation that errors in camera poses reflect on both appearance quality and physical alignment: displaced structures can produce large image errors even when their local appearance is well reconstructed. Without the sensor constraint, image and depth fitting can jointly shift camera poses and scene geometry away from the physical reference, propagating pose errors into the reconstruction. The degradation with a fixed inverse-depth window reflects a different limitation: the same absolute interval provides uneven relative search coverage across aerial and street-level viewing distances, making Gaussian initialization less reliable and degrading the final reconstruction.

Without the revisit insertion check, residual registration discrepancies can introduce duplicate or conflicting geometry when the same surfaces are observed again. The increase in photometric error is concentrated in revisited regions, where PSNR drops by more than 3 dB relative to the full workflow, while rendering quality elsewhere remains comparable. This localized degradation highlights the importance of controlling insertion before geometric inconsistencies enter the Gaussian field. Removing reseeding primarily degrades geometry and rendering while leaving pose accuracy relatively close to the full workflow, suggesting that accurate poses alone do not compensate for insufficient geometric support. Targeted reseeding supplies additional primitives in poorly reconstructed regions, complementing the refinement of existing geometry.

Table 3: Component ablation results; geometry uses 0–50 m and $\tau = 0.10$ m.

| Variant | PSNR (dB)↑ | SSIM↑ | Position MAE (m)↓ | Rotation MAE (°)↓ | Depth MAE (m)↓ | Surface $F_\tau$↑ |
|---|---|---|---|---|---|---|
| Full workflow | 31.340 | 0.919 | 0.013 | 0.012 | 0.357 | 0.675 |
| Fixed inverse-depth window | 30.493 | 0.883 | 0.047 | 0.035 | 0.578 | 0.586 |
| w/o sensor pose loss | 27.244 | 0.794 | 0.148 | 0.215 | 0.819 | 0.474 |
| w/o revisit insertion check | 30.029 | 0.827 | 0.038 | 0.026 | 0.583 | 0.613 |
| w/o reseeding | 29.965 | 0.836 | 0.018 | 0.017 | 0.466 | 0.621 |

### 4.4 Model Size

Together with Figure 3, the Gaussian counts and model sizes reported in Table 1 support our earlier observation: denser baseline models can preserve local detail while remaining substantially misaligned, leading to low rendering scores at ground-truth camera poses. GeoRefGS's overly sparse representation is consistent with poor initialization without GCP annotations, while its better alignment explains its relatively higher visual scores. ARGUS-GS achieves high visual fidelity and accurate model-to-world alignment with a moderate Gaussian count, highlighting the importance of geometric consistency over density alone.

## 5 Conclusion

We presented ARGUS-GS, a sensor-guided framework for reconstructing a shared Gaussian field from complementary UAV and vehicle imagery with model-to-world alignment. ARGUS-GS uses default onboard GNSS/IMU sensor measurements to provide a common physical reference for 3D initialization and subsequent joint refinement of camera poses and Gaussian parameters, without requiring cross-stream feature matches or annotated ground control points. Range-adaptive 3D initialization and error-guided reseeding improve geometric support across viewing distances, while revisit handling refines incoming poses and controls Gaussian insertion to limit duplicated surfaces. Evaluation in simulated urban environments demonstrates that ARGUS-GS achieves higher rendering fidelity and more accurate world-frame camera poses and surface geometry than the evaluated baselines. These results highlight the value of maintaining both physical alignment and geometric support in aerial-ground city reconstruction.

### AI USE STATEMENT

In this work, we used generative AI tools for assisting with translation and interpreting results. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, cleaning and reformatting dataset and supporting qualitative and thematic data analysis, and assisting in the writing of proofs is not applicable to this work. Additionally, we used generative AI tools for editing a research paper to improve readability and sourcing/searching for information. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES

Muhammed Enes Atik. Comparative assessment of neural radiance fields and 3D gaussian splatting for point cloud generation from UAV imagery. *Sensors*, 25(10):2995, 2025. doi: 10.3390/s25102995. URL https://www.mdpi.com/1424-8220/25/10/2995.

Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):6100–6111, 2025. doi: 10.1109/TVCG.2024.3494046. URL https://doi.org/10.1109/TVCG.2024.3494046.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5501–5510, 2022.

Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yujin Ham, Mateusz Michalkiewicz, and Guha Balakrishnan. DRAGON: Drone and ground gaussian splatting for 3D building reconstruction. In *IEEE International Conference on Computational Photography (ICCP)*, 2024.

Jiahang Hou, Xinsheng Zhang, Hao Li, and Siyuan Cui. GeoRefGS: Towards georeferenced 3D gaussian splatting from unmanned aerial vehicle platforms. *Drones*, 10(3):195, 2026.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH Conference Papers*, 2024.

Lihan Jiang, Kerui Ren, Mulin Yu, Linning Xu, Junting Dong, Tao Lu, Feng Zhao, Dahua Lin, and Bo Dai. Horizon-GS: Unified 3D gaussian splatting for large-scale aerial-to-ground scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3D gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4), 2024.

Changbai Li, Haodong Zhu, Hanlin Chen, Xiuping Liang, Tongfei Chen, Shuwei Shao, Linlin Yang, Huobin Tan, and Baochang Zhang. UrbanGS: Efficient and Scalable Architecture for Geometrically Accurate Large-Scene Reconstruction. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=L3utaw6SD9.

Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. MatrixCity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3205–3215, 2023a.

Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H. Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8456–8465, 2023b.

Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. VastGaussian: Vast 3D gaussians for large scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. CityGaussian: Real-time high-quality large-scale scene rendering with gaussians. In *European Conference on Computer Vision (ECCV)*, 2024.

Yang Liu, Chuanchen Luo, Zhongkai Mao, Junran Peng, and Zhaoxiang Zhang. CityGaussianV2: Efficient and geometrically accurate reconstruction for large-scale scenes. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhenxing Mi and Dan Xu. Switch-NeRF: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *International Conference on Learning Representations (ICLR)*, 2023.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):102:1–102:15, 2022. doi: 10.1145/3528223.3530127.

Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R. Nascimento. XFeat: Accelerated features for lightweight image matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-GS: Towards consistent real-time rendering with LOD-structured 3D gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. doi: 10.1109/TPAMI.2025.3568201. URL https://doi.org/10.1109/TPAMI.2025.3568201.

Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Jiadong Tang, Yu Gao, Dianyi Yang, Liqi Yan, Yufeng Yue, and Yi Yang. DroneSplat: 3D gaussian splatting for robust 3D reconstruction from in-the-wild drone imagery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable construction of large-scale NeRFs for virtual fly-throughs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Zengyan Wang, Sirshapan Mitra, Rajat Modi, Hui Lim, and Yogesh Rawat. Sky2Ground: A benchmark for site modeling under varying altitude. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12227–12236, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_Sky2Ground_A_Benchmark_for_Site_Modeling_under_Varying_Altitude_CVPR_2026_paper.html.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street Gaussians for Modeling Dynamic Urban Scenes. In *European Conference on Computer Vision (ECCV)*, 2024. URL https://zju3dv.github.io/street_gaussians/.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024. URL https://arxiv.org/abs/2404.10772.

Chenhao Zhang, Yuanping Cao, and Lei Zhang. CrossView-GS: Gaussian splatting for cross-view scene reconstruction. *Computational Visual Media*, 2026. URL https://github.com/zachzhang07/CrossView-GS.

Saining Zhang, Baijun Ye, Xiaoxue Chen, Yuantao Chen, Zongzheng Zhang, Cheng Peng, Yongliang Shi, and Hao Zhao. Drone-assisted road gaussian splatting with cross-view uncertainty. In *British Machine Vision Conference (BMVC)*, 2024.

Hexu Zhao, Haoyang Weng, Daohan Lu, Ang Li, Jinyang Li, Aurojit Panda, and Saining Xie. On scaling up 3D gaussian splatting training. In *International Conference on Learning Representations (ICLR)*, 2025.

# A Experimental Setup

**Simulator.** The capture platform is a closed loop of PX4 SITL (v1.17) for flight control, Gazebo for vehicle and airframe dynamics, and a Unity 2022.3 player that renders all three cameras from the physical state, connected through ROS 2.

**Urban scenes.** We generate and edit the urban scene models in Blender using the City Generator V2 and iCity add-ons, then import them into Unity for rendering.

**Vehicle following.** The UAV carries a main camera and a downward-facing camera stabilized toward the vehicle roof. The downward-facing camera detects AprilTag markers mounted on the vehicle roof to guide the UAV in following the vehicle, while the main camera records the surrounding urban scene.

# B Metrics

This section supplements the metric definitions in Section 4.1 with surface construction, matching, and aggregation details.

**Surface construction.** For each test view, predicted depth uses the opacity-normalized camera-$z$ readout in Equation 5. Predicted and reference depths are independently filtered to retain finite values in $(0, d_{\max}]$, where $d_{\max}$ is the upper limit of the range in Table 2; predictions additionally require accumulated opacity of at least $0.5$. Each valid depth $z$ at pixel $\mathbf{u}$ is back-projected as

$$\mathbf{b}_v(\mathbf{u}, z) = \mathbf{c}_v^\star + \mathbf{R}_v^\star\big(z\mathbf{K}_v^{-1}\widetilde{\mathbf{u}}\big), \tag{11}$$

where $\widetilde{\mathbf{u}} = (u_x, u_y, 1)^\top$, $\mathbf{K}_v$ contains the camera intrinsics, and $(\mathbf{c}_v^\star, \mathbf{R}_v^\star)$ is the ground-truth camera pose. Samples from both streams are pooled into predicted and reference clouds $\mathcal{P}$ and $\mathcal{Q}$ separately, preserving reference samples where predictions are missing. All methods use the same pixel-grid sampling and point budgets, with subsampling applied after range filtering; no additional cloud alignment is performed.

**Surface matching.** For nearest-neighbour distance $d(\mathbf{x}, \mathcal{Y}) = \min_{\mathbf{y}\in\mathcal{Y}} \|\mathbf{x} - \mathbf{y}\|_2$, the precision and recall used to compute $F_\tau$ in Section 4.1 are

$$p_\tau = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{p}\in\mathcal{P}} \mathbf{1}[d(\mathbf{p}, \mathcal{Q}) \le \tau], \qquad r_\tau = \frac{1}{|\mathcal{Q}|}\sum_{\mathbf{q}\in\mathcal{Q}} \mathbf{1}[d(\mathbf{q}, \mathcal{P}) \le \tau], \tag{12}$$

where $\mathbf{1}[\cdot]$ is the indicator function. We set $F_\tau = 0$ if $p_\tau + r_\tau = 0$ or the predicted cloud is empty while the reference is nonempty; cases without a valid reference are reported as unavailable.

# C Additional Training Details

## C.1 Depth supervision

After each update of keyframe $i$, its depth-loss weight follows

$$\lambda_{D,i}^{(n+1)} = \max\Big(\lambda_{D,\min}, \kappa_D \lambda_{D,i}^{(n)}\Big), \tag{13}$$

where $n$ counts keyframe updates, $\kappa_D \in (0, 1]$ is the decay factor, and $\lambda_{D,\min} > 0$ preserves geometric supervision during offline refinement.

## C.2 Pose penalties and weight schedules

The smooth L1 penalty in Equation 7 is $h_\beta(x) = x^2/(2\beta)$ for $|x| < \beta$ and $h_\beta(x) = |x| - \beta/2$ otherwise. Its pose-constraint multiplier is

$$\eta_i = s_{\text{pose}} \min\left(1, \frac{n_i + 1}{N_{\text{ramp}}}\right), \tag{14}$$

where $n_i$ counts completed updates of keyframe $i$, $N_{\text{ramp}}$ controls the ramp duration, and $s_{\text{pose}}$ is the shared pose-loss multiplier. For optical-forward direction $\mathbf{f}_i = \mathbf{R}_i \mathbf{e}_z$, the UAV elevation is $\theta_i = \operatorname{atan2}(f_{i,z}, \|\mathbf{f}_{i,xy}\|_2)$; $\bar{\theta}_i$ is computed from $\bar{\mathbf{R}}_i$ in the same way. The vehicle has no separate elevation penalty ($\lambda_\theta = 0$).

### C.3 Error-guided reseeding

The depth score in Equation 8 uses a fixed metric inverse-depth reference $\rho_i^0$ and confidence weights $\omega_i$, masked to retain finite, positive reference values and confidence:

$$\ell_{D,i} = \frac{\sum_{\mathbf{u}} \omega_i(\mathbf{u}) h_{\beta_{\log}}\big(\log \max(\hat{\rho}_i(\mathbf{u}), \epsilon_\rho) - \log \rho_i^0(\mathbf{u})\big)}{\max(\sum_{\mathbf{u}} \omega_i(\mathbf{u}), \epsilon_\omega)}. \tag{15}$$

Only valid reference pixels enter the sum; $\epsilon_\rho, \epsilon_\omega > 0$ are numerical floors. For $k \in \{\text{rgb}, D\}$, the scores are updated as

$$E_{k,i}^{(n+1)} = (1-\mu) E_{k,i}^{(n)} + \mu \ell_{k,i}^{(n+1)}, \tag{16}$$

where $\ell_{\text{rgb},i}$ is the mean RGB L1 error and $\mu \in (0, 1]$ controls averaging; each average is initialized from its first observation. Every $T_{\text{seed}}$ optimization steps, the $K_{\text{seed}}$ highest-scoring eligible keyframes across both streams are selected.

Within a selected view, let $a_i = \text{clip}(\kappa_a[\mathcal{D}(\mathbf{I}_i) - \mathcal{D}(\hat{\mathbf{I}}_i)], 0, 1)$ be the scaled image-detail deficit, where $\mathcal{D}$ is the Laplacian-based detail response evaluated at the geometry resolution. Using $e_i$ and $b_i$ from Equation 9, we define the repair weight $r_i = \text{clip}((e_i - \delta_D)/s_D, 0, 1)$. Pixels are independently sampled with probability

$$\pi_i(\mathbf{u}) = m_i^{\text{seed}}(\mathbf{u}) \,\text{clip}(\max(a_i(\mathbf{u}), p_{\min} r_i(\mathbf{u}) b_i(\mathbf{u})) + \lambda_{\text{repair}} r_i(\mathbf{u}) b_i(\mathbf{u}), 0, 1)\,, \tag{17}$$

where $m_i^{\text{seed}}$ masks unobserved pixels, $p_{\min}$ supplies a sampling floor in supported regions needing repair, and $\lambda_{\text{repair}}$ controls the depth-driven increase. The thresholds $\delta_D, s_D, \tau_c$ control error activation, scaling, and confidence gating; $r_i = 1$ where the rendered depth is unavailable. Sampled pixels undergo dense initialization and the insertion checks in Section 3.6; newly inserted Gaussians remain trainable while protected from opacity pruning for $T_{\text{cool}}$ steps.