OUR-GS: ONLINE UAV RECONSTRUCTION WITH ADAPTIVE GAUSSIAN MAPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Online UAV reconstruction provides timely 3D spatial understanding during flight, supporting perception and decision-making as UAVs operate in previously unseen environments. Recent aerial reconstruction methods increasingly build on 3D Gaussian Splatting for high-quality, large-scale reconstruction, yet remain predominantly offline and rely on pre-collected imagery and extensive scene coverage. The online setting is substantially more challenging for these methods, as reconstruction must proceed from incomplete observations over wide depth ranges and uneven multi-view evidence within a limited optimization budget. This work presents OUR-GS, an online monocular Gaussian reconstruction framework for UAV streams that adapts Gaussian map construction and refinement to viewing distance and available multi-view evidence. OUR-GS combines multi-view depth reassessment, balanced Gaussian map expansion, and tangential-radial decomposed position updates to enhance cross-frame depth consistency, representation across depth ranges, and online refinement, thereby improving online reconstruction quality. Experiments on five outdoor UAV datasets demonstrate that OUR-GS consistently improves reconstruction quality over existing methods, with particularly pronounced gains on forward-facing large-scale sequences, while maintaining the lowest GPU memory requirement. Ablation studies further validate the contribution of each component in OUR-GS. The code will be released.

1 INTRODUCTION

UAVs increasingly operate across large and complex outdoor environments for inspection, exploration, surveying, and various other autonomous missions. Such missions require timely spatial understanding as new regions are progressively revealed during flight (Song et al., 2021; Matsuki et al., 2024). Online UAV reconstruction therefore aims to continuously build and update a 3D map from incoming visual observations, providing an evolving scene representation throughout the flight.

Recent advances in UAV reconstruction have substantially improved the quality and scalability of large-scale aerial scene modeling. In particular, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has emerged as a strong representation for aerial reconstruction. Existing methods scale reconstruction to large outdoor environments through scene partitioning and multi-scale representations, while improving reconstruction fidelity with depth constraints and multi-view priors (Jiang et al., 2025; Liu et al., 2025b; Tang et al., 2025). Despite this progress, UAV reconstruction remains predominantly offline, relying on pre-collected imagery and extensive scene coverage for reconstruction.

Online UAV reconstruction must incrementally construct the scene from incomplete observations under a limited optimization budget. Large depth variation and uneven multi-view evidence further complicate this process, especially when a forward-facing camera simultaneously captures both nearby structures and distant scenery. These challenges become particularly pronounced in recent 3DGS-based aerial reconstruction methods, spanning Gaussian initialization, map expansion, and refinement throughout successive map updates. Specifically, ❶ frame-wise depth estimates can remain inconsistent across observations (Zhou et al., 2026) (Fig. 1 (a)), causing inconsistent Gaussian placement and increasingly large 3D errors at greater distances; ❷ nearby regions may exhibit larger image-space motion and contribute more high-residual pixels (Fig. 1 (b)), potentially biasing residual-driven Gaussian allocation (Zhang et al., 2024) toward these regions while leaving distant structures underrepresented; and ❸ revising the same image-space misalignment during optimiza-

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.

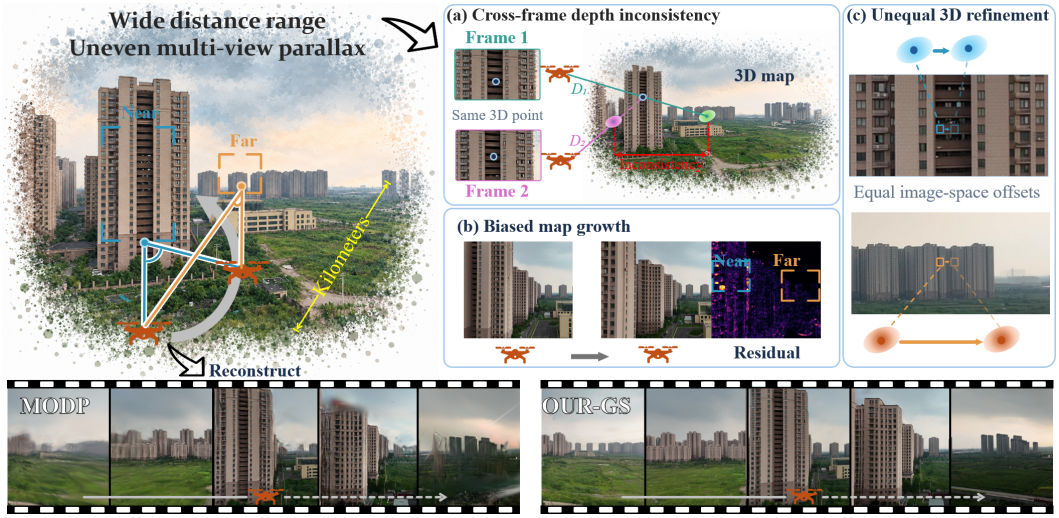

Figure 1: **Motivation.** Wide depth ranges and uneven parallax challenge online UAV reconstructions. (a) Inconsistent depth predictions across frames place the same scene point at different 3D locations. (b) Nearby regions may exhibit high residuals, biasing Gaussian allocation. (c) Distant Gaussians require larger displacements to correct equal image-space offsets. Consequently, existing methods (e.g., MODP (Wu et al., 2025b)) may struggle to recover high-fidelity scene appearance.

tion requires a larger 3D displacement as depth increases (Fig. 1 (c)), making drift in distant Gaussians difficult to correct in limited online iterations with weak parallax (Matsuki et al., 2024).

To address these challenges, this work presents OUR-GS, an online monocular Gaussian reconstruction framework for outdoor UAV streams. OUR-GS adapts 3DGS map construction and refinement to viewing distance and available multi-view evidence. It first applies Multi-view Depth Reassessment (§3.1), which uses multi-frame correspondences and target-excluded triangulation to reassess frame-wise depth estimates and revise existing Gaussians (for ❶). Building on the reassessed depths, Balanced Gaussian Map Expansion (§3.2) adaptively distributes Gaussian initialization across depth ranges while filtering redundant additions (for ❷). The expanded map is then refined by Tangential-Radial Decomposed Update (§3.3), which scales tangential displacements with viewing distance and suppresses unsupported radial motions using multi-view evidence (for ❸).

Experiments on five outdoor UAV datasets demonstrate consistent improvements in online reconstruction quality. OUR-GS achieves the best PSNR, SSIM and LPIPS on all five datasets. The gains are particularly pronounced on forward-facing UAV sequences with large depth variation, where OUR-GS improves PSNR on the in-the-wild Wild-UAV dataset from **24.25** dB to **28.80** dB, with better preservation of fine structures and scene details. OUR-GS also maintains the lowest GPU memory requirement across all datasets. Ablation studies further verify the contribution of each component to the overall reconstruction performance. The code will be released.

2 Related Work

UAV-Based Gaussian Reconstruction. Gaussian reconstruction from UAV imagery aims to recover the appearance and geometry of outdoor environments, providing detailed scene representations for aerial mapping and visualization. To accommodate large spatial extents, existing approaches employ scene partitioning and distributed optimization, improving data allocation and regional reconstruction (Lin et al., 2024; Liu et al., 2024; Chen & Lee, 2024; Su et al., 2025; Fan et al., 2025; Liu et al., 2025a). Multi-scale Gaussian representations and level-of-detail strategies further enable efficient rendering (Liu et al., 2024; Ren et al., 2025; Jiang et al., 2025; Yuan et al., 2025). Beyond scalability, geometric reconstruction is improved through depth constraints, multi-view priors, and geometry-aware density control, which help recover scene structures under challenging aerial observations (Liu et al., 2025b; Tang et al., 2025; Wang et al., 2026). More recent methods also extend aerial reconstruction to dynamic scenes with adaptive distractor masking or physical priors (Choi et al., 2026; Liu & Qin, 2026). Despite achieving high-quality reconstructions, these approaches

rely on offline optimization over pre-collected multi-view images with extensive scene coverage, failing to provide the timely 3D feedback required by time-sensitive applications during flight (Song et al., 2021; Matsuki et al., 2024). In contrast, OUR-GS operates on monocular UAV streams and incrementally updates a Gaussian map using current and historical observations.

Online Gaussian Mapping. Online Gaussian mapping incrementally constructs renderable scene representations from sequential observations (Matsuki et al., 2024; Li et al., 2025). Existing systems combine Gaussian mapping with direct or feature-based tracking (Yan et al., 2024; Huang et al., 2024; Keetha et al., 2024; Peng et al., 2024), and improve consistency through monocular depth priors, global optimization, and map deformation (Sandström et al., 2025; Zhang et al., 2025; Yugay et al., 2025; Homeyer et al., 2025; Wu et al., 2025a; Fan et al., 2026). Complementary approaches improve map construction through hierarchical occupancy management, structured initialization, and appearance modeling (Wu et al., 2025b; Wen et al., 2025; Deng et al., 2025; Xie et al., 2025; Hu & Han, 2026). More recent work handles dynamic distractors through uncertainty-aware masking (Zheng et al., 2025), or aggregates temporal features to predict camera poses and Gaussian attributes directly (Zhang et al., 2026). These methods tend to rely on dense, overlapping multi-view observations, as commonly obtained in close-range capture, to achieve high-quality reconstruction (Wu et al., 2025b; Matsuki et al., 2024). However, outdoor UAV streams can span much wider depth ranges, particularly when UAVs fly with forward-facing cameras, without consistently providing sufficient observations of individual surfaces. Their initialization and optimization strategies may face additional challenges under these conditions. Although recent efforts further support large-scale reconstruction through incremental Gaussian clustering and demonstrate real-world UAV deployment through globally optimized voxelized maps (Pu et al., 2026; Meuleman et al., 2025; Zhou et al., 2026), their Gaussian additions are primarily guided by image-space rendering discrepancies and may over-focus on nearby regions (Pu et al., 2026). Therefore, this work proposes OUR-GS to achieve more effective online Gaussian reconstruction for UAV streams.

3 METHOD

OUR-GS reconstructs outdoor scenes online from monocular UAV streams. ORB-SLAM3 (Campos et al., 2021) is applied to estimate camera pose and sparse 3D points for incoming frames. Following MODP (Wu et al., 2025b), keyframes are selected based on sparse-point covisibility and temporal spacing. For each keyframe I_t at time t with pose P_t and sparse points S_t, OUR-GS updates the Gaussian map $\mathcal{G}_{t-1}$ to $\mathcal{G}_t$ using current and historical observations. Outdoor scenes pose two related difficulties: depth evidence varies across regions and observations, while large variations in viewing distance affect both map expansion and Gaussian refinement. OUR-GS addresses these challenges throughout the online reconstruction pipeline. Multi-view Depth Reassessment (MDR §3.1) estimates log-depth correction fields from multi-view evidence to revise incoming depths and existing Gaussians. Balanced Gaussian Map Expansion (BGME §3.2) then balances candidate sampling across log-depth clusters and filters redundant proposals in 3D or image space. During optimization, Tangential-Radial Decomposed Update (TRDU §3.3) decomposes optimizer displacements into tangential and radial components, scaling them by distance and attenuating radial motion according to parallax and depth uncertainty. The overall pipeline of OUR-GS is illustrated in Fig. 2.

3.1 MULTI-VIEW DEPTH REASSESSMENT

From I_t and sparse depths projected from S_t, DepthCov (Dexheimer & Davison, 2023) is applied to predict depth $D_t(\boldsymbol{u})$ and its uncertainty $\sigma_t(\boldsymbol{u})$ measured by the log-depth standard deviation at pixel $\boldsymbol{u}$. Only predictions with $\sigma_t(\boldsymbol{u}) < \sigma_{\max}$ are retained as valid depth, where $\sigma_{\max}$ is a fixed uncertainty threshold. However, depths inferred separately from sparse monocular observations can place the same structure at inconsistent 3D locations. Initializing Gaussians from these predictions transfers the disagreement into the map and may introduce persistent errors. To address this, MDR uses current and historical correspondences to correct incoming depths and revisit existing Gaussians.

Discrepancy Formulation. MDR maintains a sliding window of active correspondences spanning 24 frames. For each incoming keyframe I_t, we select two geometrically informative reference frames from the window based on camera poses and intrinsics. LoFTR (Sun et al., 2021) then establishes pixel correspondences between I_t and each reference frame, and DepthCov (Dexheimer & Davison, 2023) provides depth estimates at these matched pixels. LoFTR correspondences as-

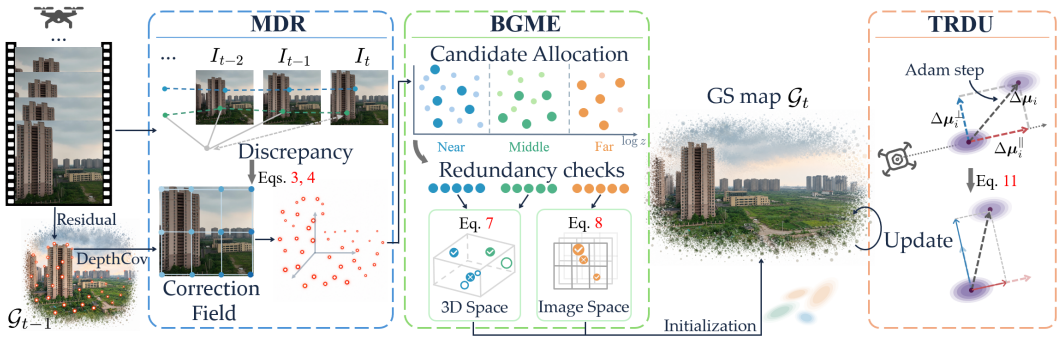

Figure 2: **Overview.** OUR-GS uses multi-view evidence and viewing distance to guide the map construction and refinement. MDR triangulates historical correspondences and estimates a spatial correction field (§3.1), enabling depth reassessment. BGME then balances candidate allocation across a wide range of distances and filters redundant proposals in 3D or image space (§3.2). TRDU scales position updates by reference distance and attenuates unsupported radial motion (§3.3).

sociated with the same scene point are linked across image pairs into a multi-view correspondence group, indexed by q. In this way, each group collects observations of a 3D point from different frames with valid depth estimates, recording the pixel coordinate $\boldsymbol{u}_{\tau q}$ of the point in frame I_τ, its predicted depth $D_\tau(\boldsymbol{u}_{\tau q})$ and uncertainty $\sigma_\tau(\boldsymbol{u}_{\tau q})$, and the corresponding camera pose. Here $\tau \le t$.

For a new observation added to group q at time t, we measure the discrepancy between its predicted depth and the multi-frame evidence from previous observations. When group q contains observations from at least two earlier frames, we triangulate all retained observations from those frames to obtain a reference depth $\widehat{D}_{tq}$ in the current camera. The target current observation at $\boldsymbol{u}_{tq}$ is excluded from triangulation. The discrepancy y_{tq} between $\widehat{D}_{tq}$ and the DepthCov prediction is expressed as:

$$y_{tq} = \log \frac{\widehat{D}_{tq}}{D_t(\boldsymbol{u}_{tq})}. \tag{1}$$

Correction Field. The discrepancy $y_{\tau q}$ for any observation in group q is computed as in Eq. 1, allowing historical observations to be reassessed as new views arrive. To extend these discrepancies beyond individual matched pixels, we use them to jointly estimate a spatially varying log-depth correction field for each frame. For I_τ, the field $\delta_\tau(\cdot)$ is parameterized by $\boldsymbol{a}_\tau \in \mathbb{R}^{12}$, which collects one coefficient for each node of a 3×4 grid placed over the image plane. The log-depth correction at pixel $\boldsymbol{u}$ is obtained by bilinear interpolation of the surrounding node coefficients, with interpolation weights $\boldsymbol{\phi}(\boldsymbol{u}) \in \mathbb{R}^{12}$. The field value $\delta_\tau(\boldsymbol{u})$ and the corrected depth $\widetilde{D}_\tau(\boldsymbol{u})$ are represented as:

$$\delta_\tau(\boldsymbol{u}) = \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{a}_\tau, \quad \widetilde{D}_\tau(\boldsymbol{u}) = D_\tau(\boldsymbol{u}) \exp\big(\delta_\tau(\boldsymbol{u})\big). \tag{2}$$

To encourage consistency across frames after depth correction, we couple the frame fields using pairs of observations from the same correspondence group. For a pair $e = (\tau, s; q)$, the observations belong to group q and come from two different frames I_τ and I_s. Their respective camera centers $\boldsymbol{o}_\tau, \boldsymbol{o}_s$ and viewing directions $\boldsymbol{d}_{\tau q}, \boldsymbol{d}_{sq}$ are both expressed in world coordinates. Their corrected depths are jointly constrained through a longitudinal consistency surrogate:

$$r_e = \boldsymbol{n}_e^\top \big[\big(\boldsymbol{o}_\tau + \widetilde{D}_\tau(\boldsymbol{u}_{\tau q})\boldsymbol{d}_{\tau q}\big) - \big(\boldsymbol{o}_s + \widetilde{D}_s(\boldsymbol{u}_{sq})\boldsymbol{d}_{sq}\big)\big], \tag{3}$$

where $\boldsymbol{n}_e = (\boldsymbol{d}_{\tau q} + \boldsymbol{d}_{sq})/\big\|\boldsymbol{d}_{\tau q} + \boldsymbol{d}_{sq}\big\|$ is the normalized mean viewing direction. Reducing r_e encourages the corrected observations to agree along the mean viewing direction. With $\mathcal{E}$ denoting the set of observation pairs within correspondence groups, we jointly solve for the field coefficients $\{\boldsymbol{a}_\tau\}$ using robust Gauss-Newton updates (Nocedal & Wright, 2006) as new observations arrive:

$$\min_{\{\boldsymbol{a}_\tau\}} \sum\nolimits_{\tau,q} c_{\tau q}\, h\Big(\frac{\delta_\tau(\boldsymbol{u}_{\tau q}) - y_{\tau q}}{\nu_{\tau q}}\Big) + \sum\nolimits_{e\in\mathcal{E}} c_e\, h\Big(\frac{r_e}{\nu_e}\Big) + \mathcal{R}\big(\{\boldsymbol{a}_\tau\}\big). \tag{4}$$

Here h is the Huber loss (Huber, 1992), $c_{\tau q}, c_e$ are weights from LoFTR's matching confidence and $c_e = \sqrt{c_{\tau q} c_{sq}}$. $\nu_{\tau q}, \nu_e$ are the corresponding residual standard deviations. The first term aligns

corrected depths with the triangulated references, while the second reduces the cross-frame disagreement measured by r_e. The regularizer $\mathcal{R}$ penalizes coefficient magnitudes and differences between neighboring grid coefficients to discourage large corrections and promote spatial smoothness.

With the estimated $\boldsymbol{a}_\tau$, $\delta_\tau(\boldsymbol{u})$ is evaluated at candidate pixels to obtain the revised depth $\widetilde{D}_\tau(\boldsymbol{u})$ before Gaussian initialization. To further correct depth errors already stored in the map, we revisit existing Gaussians whose initialization frames receive updated correction fields. Only DepthCov-initialized Gaussians admitted through redundancy checks (§3.2) are revisited. For Gaussian i with current mean $\boldsymbol{\mu}_i$, its initialization frame index t_i, source pixel $\boldsymbol{u}_i$, source camera center $\boldsymbol{o}_{t_i}$, and cumulative applied log correction β_i are retained. The updated field value $\delta_{t_i}(\boldsymbol{u}_i)$ specifies the total log-depth correction relative to the original depth prediction. Since the Gaussian may already have undergone earlier corrections, we subtract β_i from $\delta_{t_i}(\boldsymbol{u}_i)$ and apply the remaining increment:

$$\Delta_i = \texttt{clip}\big(\delta_{t_i}(\boldsymbol{u}_i) - \beta_i, -\epsilon, \epsilon\big), \quad \boldsymbol{\mu}_i \leftarrow \boldsymbol{o}_{t_i} + \exp(\Delta_i)(\boldsymbol{\mu}_i - \boldsymbol{o}_{t_i}). \tag{5}$$

Here the Gaussian mean $\boldsymbol{\mu}_i$ is updated to the adjusted position according to the log-depth correction, and ϵ limits each log-depth update. The cumulative log correction is updated as $\beta_i \leftarrow \beta_i + \Delta_i$, and stored for possible subsequent updates. More implementation details are provided in Appendix B.1.

3.2 Balanced Gaussian Map Expansion

In online UAV reconstruction, nearby regions may contribute many pixels with large rendering errors and therefore dominate Gaussian initialization, probably leaving distant structures underrepresented. Meanwhile, repeated additions to already well-represented regions further consume the available budget. To address these problems, BGME balances initialization selection across depth ranges and filters redundant proposals through 3D coverage checks and image-space deduplication.

Candidate Allocation. Gaussian initialization candidates come from two sources: sparse points S_t provided by the tracking system and DepthCov estimates obtained at pixels with large rendering errors. The DepthCov estimates are reassessed by MDR (§3.1). Sparse tracked candidates retain their SLAM depths and bypass budget sampling, while directly undergoing 3D redundancy checks. For candidates derived from DepthCov, we adaptively form three depth clusters using the reassessed depths $\widetilde{D}_t(\boldsymbol{u}_i)$ and allocate a balanced sampling budget across them. The cluster centers are initialized at the $1/6$, $1/2$, and $5/6$ quantiles of their log depths. Candidates are then iteratively assigned to the nearest center in log-depth space, with each center updated to its cluster's mean log depth, until assignments remain unchanged or 32 iterations are reached. With a budget B for preselecting DepthCov candidates at each map expansion, each cluster receives an equal share unless it contains too few candidates, in which case its unused budget is redistributed equally among other clusters. For cluster $p \in \{1, 2, 3\}$ containing N_p candidates, the allocated budget b_p satisfies:

$$b_p = \min(N_p, \lambda), \quad \sum\nolimits_{p=1}^{3} b_p = \min\big(B, \sum\nolimits_{p=1}^{3} N_p\big), \tag{6}$$

where λ is the remaining budget divided equally among the unallocated clusters. This allocation prevents map expansion from overly favoring nearby regions and leaving other regions underrepresented. Then b_p candidates are uniformly sampled from each cluster p. The sampled candidates will undergo redundancy checks to determine whether they should be initialized as new Gaussians.

Redundancy Checks. Although MDR (§3.1) reduces depth inconsistencies across multiple frames, it cannot fully eliminate them, particularly for distant structures. A redundancy check in 3D space may then mistake depth errors for new structure and admit unnecessary Gaussians to the map. Therefore, we use image-space checks for distant candidates sampled from the farthest cluster that tend to lack sufficient depth support, denoted by $\chi_i = 1$. Meanwhile, the remaining sampled candidates and the sparse tracked points undergo 3D coverage checks, with $\chi_i = 0$.

For sampled candidates with $\chi_i = 0$, we compare their 3D positions $\boldsymbol{x}_i$ and RGB colors $\boldsymbol{c}_i$ with previously accepted samples through this branch, retrieved using 3D nearest-neighbor queries at $\boldsymbol{x}_i$. A spatial scale w_i is set for each candidate, proportional to the distance between its source pixel and an adjacent pixel after both are back-projected into 3D at a reference depth. Among the samples retrieved by each query, the closest sample j satisfying $w_j \le w_i$ is selected for comparison. The candidate is discarded as redundant if both the spatial and color conditions below are satisfied:

$$\big\|\boldsymbol{x}_i - \boldsymbol{x}_j\big\|_2 \le \min\big(\ell_j, \frac{w_i}{\sqrt{12}}\big), \qquad \big\|\boldsymbol{c}_i - \boldsymbol{c}_j\big\|_2 \le T_c, \tag{7}$$

where ℓ_j is the distance from j to its nearest other sample k retrieved by the same query and satisfying $w_k \leq w_j$. $w_i/\sqrt{12}$ is applied as the spatial distance cap, motivated by the standard deviation of uniform quantization error over an interval of scale w_i. The color tolerance T_c reflects the allowable RGB variation between neighboring accepted samples. More details are provided in Appendix B.2.

For distant sampled candidates with $\chi_i = 1$, we partition the image into cells and divide sampled candidates within each cell into log-depth bins, suppressing redundant proposals within each bin. For an image of size $W \times H$, the cell width is set to $\omega_u = \sqrt{WH/(2B)}$ and the log-depth bin width is set to $\omega_z = \sqrt{2}\,\sigma_{\max}$, where B is the sampling budget and $\sigma_{\max}$ is the DepthCov validity threshold. A candidate at pixel $\boldsymbol{u}_i = (u_{i,x}, u_{i,y})$ with depth $\widetilde{D}_t(\boldsymbol{u}_i)$ is assigned the bin index κ_i:

$$\kappa_i = \big(\lfloor \frac{u_{i,x}}{\omega_u} \rfloor, \lfloor \frac{u_{i,y}}{\omega_u} \rfloor, \lfloor \frac{\log \widetilde{D}_t(\boldsymbol{u}_i)}{\omega_z} \rfloor\big). \tag{8}$$

Among sampled candidates in the same bin, we retain only the one with the largest rendering residual and discard the others as redundant, prioritizing locations with greater reconstruction errors.

These redundancy checks promote compact representations of distant structures while admitting new samples to capture previously unresolved details as the UAV approaches. Accepted candidates from both branches are initialized as 3D Gaussians at location $\boldsymbol{\mu}_i^{\text{ini}}$ using depths reassessed by MDR §3.1 for DepthCov candidates and retained SLAM depths for sparse tracked points, with colors sampled from their source images and scales proportional to their depths divided by the mean focal length.

3.3 Tangential–Radial Decomposed Update

During online reconstruction, the incoming UAV frame requires Gaussian projections to be aligned with new observations. However, correcting the same image-space misalignment requires larger transverse position changes at greater depths, which may not be completed within limited online optimization iterations. TRDU is designed to compensate for this mismatch.

For Gaussian i initialized from frame t_i with location $\boldsymbol{\mu}_i^{\text{ini}}$ from BGME (§3.2), its source camera center $\boldsymbol{o}_{t_i}$ defines the reference distance $r_i = \left\|\boldsymbol{\mu}_i^{\text{ini}} - \boldsymbol{o}_{t_i}\right\|_2$ and viewing direction $\boldsymbol{v}_i = (\boldsymbol{\mu}_i^{\text{ini}} - \boldsymbol{o}_{t_i})/r_i$. Optimizing an L_1 and SSIM rendering loss (Kerbl et al., 2023) produces an Adam position displacement $\Delta\boldsymbol{\mu}_i$. We split it into radial and tangential components:

$$\Delta\boldsymbol{\mu}_i^{\parallel} = (\boldsymbol{v}_i^{\top}\Delta\boldsymbol{\mu}_i)\boldsymbol{v}_i, \qquad \Delta\boldsymbol{\mu}_i^{\perp} = \Delta\boldsymbol{\mu}_i - \Delta\boldsymbol{\mu}_i^{\parallel}. \tag{9}$$

At greater viewing distances, larger 3D position updates are needed to produce the same displacement in image space. This motivates scaling the Adam-proposed displacement by a scaling factor $\gamma_i = r_i/r_{\text{unit}}$ where r_{unit} is the fixed length scale for distance normalization. For Gaussians admitted through the image-space redundancy check that tend to lack parallax support ($\chi_i = 1$ in §3.2), we further attenuate their radial updates according to their depth uncertainty σ_i and parallax statistic η_i to restrict potentially unreliable radial movements. Here η_i is computed as the maximum squared sine of the angle between the source ray and rays from other available keyframe camera centers. An adaptive attenuation factor for radial updates is designed as:

$$g_i = \frac{\eta_i \sigma_i^2}{\eta_i \sigma_i^2 + (s_{i,\max}/r_i)^2}, \tag{10}$$

where $s_{i,\max}$ is the largest of Gaussian i's three current scale parameters, and $s_{i,\max}/r_i$ approximates how large Gaussian i appears in its initialization view. g_i encourages conservative radial updates when parallax is weak or initial depth uncertainty is low. For Gaussians with $\chi_i = 0$, we set $g_i = 1$. Finally, the position of Gaussian i is updated as:

$$\boldsymbol{\mu}_i \leftarrow \boldsymbol{\mu}_i + \gamma_i(\Delta\boldsymbol{\mu}_i^{\perp} + g_i\Delta\boldsymbol{\mu}_i^{\parallel}). \tag{11}$$

TRDU thus helps align Gaussian projections with incoming observations within limited online iterations while limiting radial displacements under weak parallax. More implementation details about the scaling and parallax statistic are provided in Appendix B.3.

4 EXPERIMENTS

In this section, OUR-GS is evaluated for online monocular reconstruction across diverse outdoor UAV scenes. The experiments assess the reconstruction quality and computational efficiency. Ablation studies further investigate how our proposed depth reassessment, Gaussian allocation, and Gaussian updates jointly contribute to the reconstruction performance.

4.1 DATASETS AND EVALUATION PROTOCOL

Datasets. OUR-GS is evaluated on four public data sources including GaussianPro (Cheng et al., 2024), HorizonGS (Jiang et al., 2025), 360DVO (Guo et al., 2026), PanoAir (Wu et al., 2026), together with our self-collected Wild-UAV dataset. GaussianPro includes four outdoor UAV video sequences sourced from YouTube. From HorizonGS, we select a long synthetic aerial sequence covering a continuous flight trajectory. 360DVO provides panoramic imagery, from which we extract forward-facing perspective streams and process each as a monocular sequence. PanoAir provides undistorted pinhole image sequences from real-world UAV flights in a single scene. Our Wild-UAV dataset comprises five kilometer-scale UAV flight recordings directly captured with DJI Mini 4 Pro in different environments, complementing the public data with practical acquisition conditions.

Evaluation Protocol. OUR-GS is compared with baselines including Splat-SLAM (Sandström et al., 2025), HI-SLAM2 (Zhang et al., 2025), GigaSLAM (Deng et al., 2025), MODP (Wu et al., 2025b) and MoonSplat (Pu et al., 2026). We reserve one in every ten frames for evaluation and exclude these frames from online mapping, while still using them for tracking. In addition to rendering quality metrics (PSNR, SSIM, and LPIPS), reconstruction speed (FPS) and peak GPU memory usage (VRAM) are reported. These metrics are first averaged within each sequence and then across sequences within each dataset. Reconstructions are performed entirely online, without any post-refinement. All experiments are conducted on RTX 6000 Ada GPUs with 48 GB VRAM.

4.2 RESULTS AND COMPARISONS

Reconstruction Quality. Table 1 presents the quantitative reconstruction results across all five datasets. OUR-GS achieves leading reconstruction performance across all evaluated metrics on the five datasets. Compared with the strong baseline MODP (Wu et al., 2025b), OUR-GS improves PSNR by 0.85–6.65 dB and reduces LPIPS by 0.038–0.456. Figure 3 further provides qualitative comparisons, where one scene is shown for each dataset. These comparisons assess the benefits of our reconstruction pipeline across different scene layouts and acquisition conditions.

The improvements are particularly pronounced on the forward-facing sequences from 360DVO, PanoAir and Wild-UAV, where OUR-GS improves PSNR over MoonSplat (Pu et al., 2026) by **2.54**, **4.18** and **4.84** dB, respectively. These observations often capture nearby structures and unbounded distant scenery simultaneously, exhibiting wide depth ranges and uneven parallax. Previous methods therefore tend to retain errors in under-observed regions and produce floating artifacts (e.g. the palace near the image boundary and distant buildings in Fig. 3). In contrast, our MDR helps OUR-GS reduce ghosting associated with cross-frame depth inconsistencies, while BGME and TRDU

Table 1: **Quantitative comparisons** (§4.2). OUR-GS is compared with Splat-SLAM (Sandström et al., 2025), HI-SLAM2 (Zhang et al., 2025), GigaSLAM (Deng et al., 2025), MODP (Wu et al., 2025b) and MoonSplat (Pu et al., 2026). Best results are highlighted as **1st** and <u>2nd</u>.

Method	GaussianPro			HorizonGS			360DVO			PanoAir			Wild-UAV		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Splat-SLAM	19.77	0.605	0.470	16.20	0.338	0.720	18.66	0.650	0.523	18.77	0.620	0.619	14.27	0.487	0.725
HI-SLAM2	21.68	0.618	0.510	19.96	0.504	0.579	22.96	0.783	0.403	20.97	<u>0.712</u>	0.572	17.56	0.659	0.635
GigaSLAM	21.24	0.567	0.570	14.67	0.294	0.832	25.70	0.839	<u>0.235</u>	23.46	0.674	0.418	21.23	0.671	0.450
MODP	<u>24.19</u>	<u>0.705</u>	<u>0.403</u>	<u>22.28</u>	<u>0.632</u>	<u>0.372</u>	<u>25.72</u>	<u>0.863</u>	0.240	21.22	0.690	0.645	<u>24.25</u>	<u>0.784</u>	<u>0.317</u>
MoonSplat	22.50	0.556	0.573	15.34	0.234	0.895	24.74	0.747	0.372	<u>23.69</u>	0.693	<u>0.331</u>	23.96	0.727	0.462
OUR-GS	**25.04**	**0.742**	**0.148**	**24.73**	**0.763**	**0.198**	**27.28**	**0.865**	**0.202**	**27.87**	**0.825**	**0.189**	**28.80**	**0.866**	**0.162**
Error Bars (±)	(0.13)	(0.015)	(0.005)	(0.10)	(0.019)	(0.007)	(0.18)	(0.021)	(0.008)	(0.09)	(0.011)	(0.010)	(0.19)	(0.015)	(0.003)

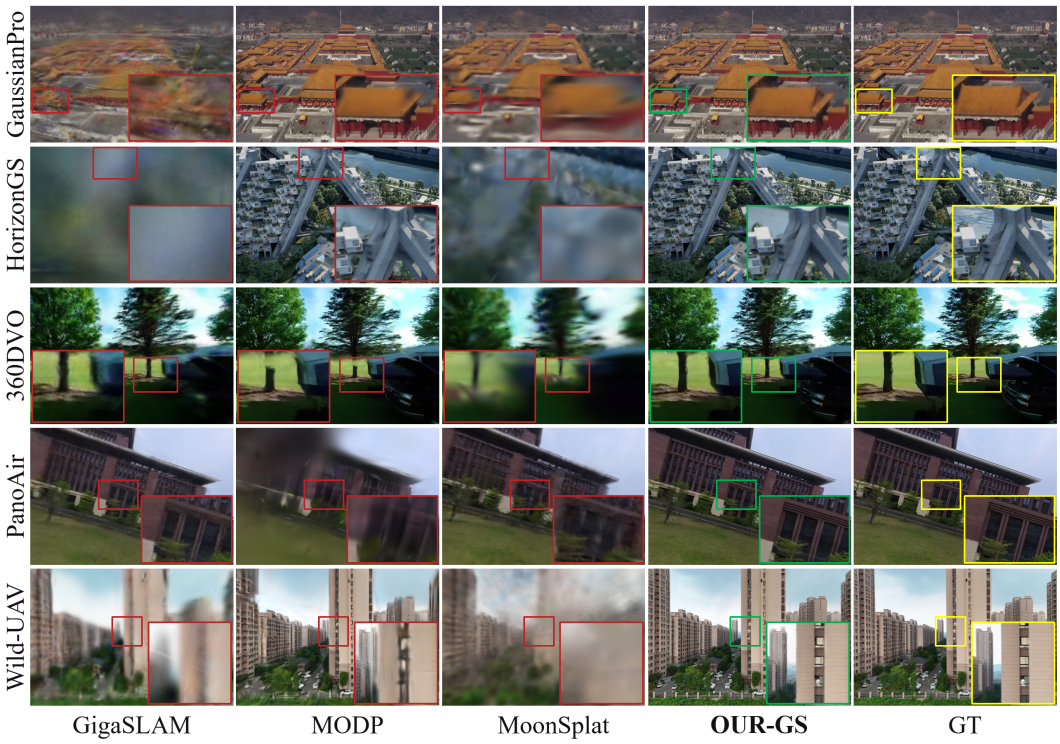

Figure 3: **Qualitative comparisons** (§4.2). Red rectangles highlight distorted or blurry reconstruction, while green rectangles indicate well-rendered results. GT stands for the ground truth.

help preserve sharper contours and more complete details across different depth ranges, as shown in Fig. 3. Improvements are also observed on the oblique views from GaussianPro and HorizonGS, where OUR-GS respectively exceeds the strongest baseline in PSNR by 0.85 and 2.45 dB. These results indicate that OUR-GS benefits reconstruction under different UAV observation conditions.

Computational Costs. Reconstruction speed (FPS) and peak GPU memory usage (VRAM) are compared in Table 2 to assess computational requirements. OUR-GS achieves competitive online reconstruction efficiency. Meanwhile, its BGME distributes new Gaussians across depth ranges and filters redundant additions, yielding the lowest GPU memory usage among compared methods while preserving the representational capacity. Even on Wild-UAV, where sequences span kilometers and contain thousands of frames, OUR-GS maintains memory friendly for consumer GPUs.

Tracking Accuracy. Although OUR-GS focuses on mapping quality, tracking accuracy is also compared to assess whether the reconstruction gains rely on more accurate tracking. Table 3 reports absolute trajectory error (ATE) on PanoAir, where ground-truth trajectories are available. OUR-GS has an ATE close to that of competing methods, yet it achieves better rendering quality even with slightly lower tracking accuracy (Table 1). These results highlight OUR-GS's ability to achieve higher reconstruction quality without relying on the most accurate tracking system.

Table 2: **Computational cost comparisons** of OUR-GS and baseline methods. FPS denotes reconstruction speed in frames per second and VRAM denotes peak GPU memory usage. See (§4.2) for more details.

Method	GaussianPro		HorizonGS		360DVO		PanoAir		Wild-UAV	
	FPS↑	VRAM↓	FPS↑	VRAM↓	FPS↑	VRAM↓	FPS↑	VRAM↓	FPS↑	VRAM↓
Splat-SLAM	**1.36**	8.92 G	0.80	14.95 G	1.53	8.99 G	1.08	46.38 G	1.55	18.97 G
HI-SLAM2	1.03	8.77 G	**1.23**	14.40 G	**1.65**	9.36 G	**1.87**	23.01 G	1.64	16.61 G
GigaSLAM	0.12	12.09 G	0.16	13.66 G	0.11	12.28 G	0.10	22.79 G	0.14	17.13 G
MODP	0.99	6.61 G	1.21	12.26 G	1.27	9.32 G	1.41	16.35 G	**2.03**	14.15 G
MoonSplat	0.56	12.54 G	1.06	16.94 G	0.73	19.70 G	0.96	38.80 G	0.69	34.63 G
OUR-GS	1.03	**5.48** G	1.12	**10.17** G	1.39	**7.05** G	1.34	**13.88** G	1.47	**10.77** G

Table 3: **Tracking accuracy** on PanoAir (§4.2).

Method	PanoAir
	ATE↓
Splat-SLAM	**0.41**
HI-SLAM2	0.44
GigaSLAM	3.29
MODP	0.51
MoonSplat	1.34
OUR-GS	0.50

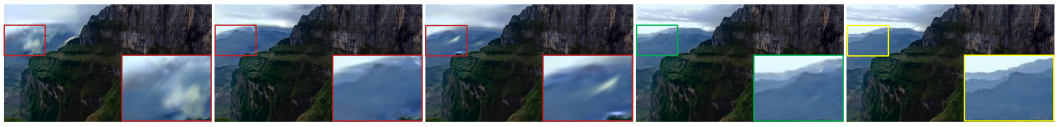

Figure 4: **Qualitative ablations** of the components in OUR-GS. See §4.3 for more details.

4.3 Ablation Study

In this section, ablation studies are conducted to evaluate the contribution of each component in OUR-GS. Additional ablations, visualizations and discussions are available in Appendix A.

Table 4: **Component ablations** on the 360DVO (Guo et al., 2026) dataset (§4.3).

Variant	PSNR↑	SSIM↑	LPIPS↓	FPS↑	VRAM↓
w/o MDR	25.89	0.854	0.237	**1.75**	**6.59** G
w/o BGME*	27.08	0.860	0.205	1.44	6.97 G
w/o TRDU	26.11	0.852	0.213	1.58	6.80 G
Full OUR-GS	**27.28**	**0.865**	**0.202**	1.39	7.05 G

Overall Design. We first analyze the contributions of the proposed depth reassessment, Gaussian allocation, and Gaussian update strategies through the following ablations in Table 4, including: (1) w/o MDR, which does not use MDR to align depth estimates across frames and instead uses points back-projected directly from per-frame DepthCov predictions as Gaussian candidates; (2) w/o BGME*, which omits the adaptive depth grouping and balanced Gaussian budget allocation proposed in BGME and instead performs random sampling over all candidate points, while retaining BGME's redundancy checks; (3) w/o TRDU, which disables TRDU and directly applies the Adam updates from the optimization, and (4) Full OUR-GS, which represents the complete proposed online reconstruction pipeline. Removing any of these components leads to a noticeable decline in reconstruction quality. Figure 4 further provides qualitative comparisons of the ablation variants. The highlighted regions show pronounced floaters and blurred ridge boundaries without MDR, which are caused by inconsistent depth estimates across frames that place Gaussians at misaligned 3D positions. Without BGME*, the distant ridges are smoother and less distinct than those of Full OUR-GS, illustrating the benefit of our balanced candidate allocation. Removing TRDU also leaves distortions, as errors in distant Gaussian positions are not sufficiently corrected within the limited online optimizations. These observations are consistent with their intended roles of depth reassessment, balanced candidate allocation, and position refinement, supporting the contribution of each component to online reconstruction quality.

Table 5: **Ablations of redundancy filtering strategy** on the 360DVO (Guo et al., 2026) dataset (§4.3). #GS denotes the final number of Gaussians.

Strategy	PSNR↑	SSIM↑	LPIPS↓	FPS↑	VRAM↓	#GS↓
3D filtering only	27.19	0.861	**0.201**	1.33	8.05 G	384 K
w/o RC	27.23	**0.876**	0.206	**1.41**	8.67 G	406 K
Full OUR-GS	**27.28**	0.865	0.202	1.39	**7.05** G	**349** K

Redundancy Handling. The influence of OUR-GS's redundancy handling in BGME is evaluated in Table 5, which presents three ablation variants: (1) 3D filtering only, which applies 3D redundancy filtering to all candidates, including weakly observed far candidates that are originally checked in image space by BGME; (2) w/o RC, which allows all sampled candidates to be initialized as Gaussians without redundancy checks; and (3) Full OUR-GS, which uses the redundancy checks as described in §3.2. Full OUR-GS achieves comparable reconstruction quality with the fewest Gaussians and the lowest VRAM usage, indicating that our Gaussian admission strategy effectively reduces Gaussian redundancy while preserving the representational ability of the Gaussian map.

5 Conclusion

This work presents OUR-GS to address the challenges of wide depth ranges and uneven parallax in online monocular UAV reconstruction. Within OUR-GS, MDR improves cross-frame depth consistency, BGME balances Gaussian allocation across depth ranges and reduces redundancy, and TRDU adapts position updates to viewing distance and multi-view support. Extensive experiments demonstrate that OUR-GS consistently improves online reconstruction quality over existing methods across five evaluated datasets and achieves the lowest GPU memory usage.

AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing, including grammar correction, wording refinement, and improvements to clarity and readability. We have not used generative AI tools for method development, experimental design, data analysis, result interpretation, or the creation of figures and citations. All AI-assisted revisions were manually reviewed by the authors to ensure technical accuracy and maintain consistency with the methods and experimental results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

Yu Chen and Gim Hee Lee. Dogs: Distributed-oriented gaussian splatting for large-scale 3d reconstruction via gaussian consensus. *Advances in Neural Information Processing Systems*, 37: 34487–34512, 2024.

Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first international conference on machine learning*, 2024.

Jaehoon Choi, Dongki Jung, Chris Maxey, Sungmin Eum, Yonghan Lee, Dinesh Manocha, and Heesung Kwon. Uav4d: Dynamic neural rendering of human-centric uav imagery using gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3372–3380, 2026.

Kai Deng, Yigong Zhang, Jian Yang, and Jin Xie. Gigaslam: Large-scale monocular slam with hierarchical gaussian splats. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–10, 2025.

Eric Dexheimer and Andrew J Davison. Learning a depth covariance function. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13122–13131. IEEE, 2023.

Chuanmao Fan, Chenxi Zhao, and Ye Duan. Gbg-slam: Key-frame centered monocular gaussian slam with scale consistent dense geometry boosting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 8257–8267, 2026.

Jixuan Fan, Wanhua Li, Yifei Han, Tianru Dai, and Yansong Tang. Momentum-gs: Momentum gaussian self-distillation for high-quality large scene reconstruction. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25250–25260. IEEE, 2025.

Xiaopeng Guo, Yinzhe Xu, Huajian Huang, and Sai-Kit Yeung. 360dvo: Deep visual odometry for monocular 360-degree camera. *IEEE Robotics and Automation Letters*, 2026.

Christian Homeyer, Leon Begiristain, and Christoph Schnörr. Droid-splat combining end-to-end slam with 3d gaussian splatting. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2788–2798. IEEE, 2025.

Pengchong Hu and Zhizhong Han. Sgad-slam: Splatting gaussians at adjusted depth for better radiance fields in rgbd slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18934–18945, 2026.

Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21584–21593, June 2024.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26789–26799. IEEE, 2025.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21357–21366. IEEE, 2024.

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

Yuetao Li, Zijia Kuang, Ting Li, Qun Hao, Zike Yan, Guyue Zhou, and Shaohui Zhang. Activesplat: High-fidelity scene reconstruction through active gaussian splatting. *IEEE Robotics and Automation Letters*, 10(8):8099–8106, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5166–5175, June 2024.

Hanyang Liu and Rongjun Qin. Aerodgs: Physically consistent dynamic gaussian splatting for single-sequence aerial 4d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19011–19021, June 2026.

Shiyong Liu, Xiao Tang, Zhihao Li, Yingfan He, Chongjie Ye, Jianzhuang Liu, Binxiao Huang, Shunbo Zhou, and Xiaofei Wu. Occlugaussian: Occlusion-aware gaussian splatting for large scene reconstruction and rendering. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26643–26652. IEEE, 2025a.

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*, pp. 265–282. Springer, 2024.

Yang Liu, Chuanchen Luo, Zhongkai Mao, Junran Peng, and Zhaoxiang Zhang. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL `https://openreview.net/forum?id=a3ptUbuzbW`.

Hidenobu Matsuki, Riku Murai, Paul H.J. Kelly, and Andrew J. Davison. Gaussian Splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18039–18048, June 2024.

Andreas Meuleman, Ishaan Shah, Alexandre Lanvin, Bernhard Kerbl, and George Drettakis. On-the-fly reconstruction for large-scale novel view synthesis from unposed images. *ACM Transactions on Graphics*, 44(4), 2025.

Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16695–16705. IEEE, 2025.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006. doi: 10.1007/978-0-387-40065-5.

Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024.

Guo Pu, Yixuan Han, Haofeng Li, Yao Zhang, Hui Zhou, and Zhouhui Lian. Moonsplat: Monocular online gaussian splatting with sim(3) global optimization. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2026.

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*, pp. 1–15, 2025. doi: 10.1109/TPAMI.2025.3568201.

Erik Sandström, Ganlin Zhang, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Youmin Zhang, Manthan Patel, Luc Van Gool, Martin Oswald, and Federico Tombari. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1686–1697, June 2025.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Soohwan Song, Daekyum Kim, and Sunghee Choi. View path planning via online multiview stereo for 3-d modeling of large-scale structures. *IEEE Transactions on Robotics*, 38(1):372–390, 2021.

Mai Su, Zhongtao Wang, Huishan Au, Yilong Li, Xizhe Cao, Chengwei Pan, Yisong Chen, and Guoping Wang. Hug: Hierarchical urban gaussian splatting with block-based reconstruction for large-scale aerial scenes. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28839–28848. IEEE, 2025.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 833–843, June 2025.

Meng Wang, Changqun Xia, Yuze Wang, Junyi Wang, Wantong Duan, Xinxiong Xie, and Yue Qi. Urban-gs: A unified 3d gaussian splatting framework for compact and high-fidelity aerial-to-street reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33207–33216, 2026.

Tianci Wen, Zhiang Liu, and Yongchun Fang. Segs-slam: Structure-enhanced 3d gaussian splatting slam with appearance embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28103–28113, October 2025.

Ke Wu, Zicheng Zhang, Muer Tie, Ziqing Ai, Zhongxue Gan, and Wenchao Ding. Vings-mono: Visual-inertial gaussian splatting monocular slam in large scenes. *IEEE Transactions on Robotics*, 2025a.

Songyin Wu, Zhaoyang Lv, Yufeng Zhu, Duncan Frost, Zhengqin Li, Ling-Qi Yan, Carl Ren, Richard Newcombe, and Zhao Dong. Monocular online reconstruction with enhanced detail preservation. In *ACM SIGGRAPH 2025 Conference Papers*, 2025b. doi: 10.1145/3721238.3730659.

Yiyang Wu, Xiaohu Zhang, Yanjin Du, Tongsu Zhang, Chujun Li, Siyang Chen, Guoyi Zhang, and Xiangpeng Xu. Panoair: A panoramic visual-inertial slam with cross-time real-world uav dataset. *arXiv preprint arXiv:2604.00852*, 2026.

Yusen Xie, Zhenmin Huang, Jin Wu, and Jun Ma. Gs-livm: Real-time photo-realistic lidar-inertial-visual mapping with gaussian splatting. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26869–26878. IEEE, 2025.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19595–19604. IEEE, 2024.

Zhensheng Yuan, Haozhi Huang, Zhen Xiong, Di Wang, and Guanghua Yang. Robust and efficient 3d gaussian splatting for urban scene reconstruction. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26209–26219. IEEE, 2025.

Vladimir Yugay, Theo Gevers, and Martin R Oswald. Magic-slam: Multi-agent gaussian globally consistent slam. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6741–6750. IEEE, 2025.

Wei Zhang, Qing Cheng, David Skuddis, Niclas Zeller, Daniel Cremers, and Norbert Haala. HI-SLAM2: Geometry-aware gaussian SLAM for fast monocular scene reconstruction. *IEEE Transactions on Robotics*, 41:6478–6493, 2025. doi: 10.1109/TRO.2025.3626627.

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European conference on computer vision*, pp. 326–342. Springer, 2024.

Zicheng Zhang, Ke Wu, Xiangting Meng, Keyu Liu, Jieru Zhao, and Wenchao Ding. Flash-mono: Feed-forward accelerated gaussian splatting monocular SLAM. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nv3q3crc5D.

Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11461–11471, June 2025.

Yan Zhou, Li Tang, Hongqi Jin, and Xiaoping Liu. Ousa-gs: Real-time structurally-consistent 3d reconstruction of outdoor unbounded scenes via adaptive gaussian splatting. *Knowledge-Based Systems*, 331:114834, 2026. doi: 10.1016/j.knosys.2025.114834.

A ADDITIONAL EXPERIMENTAL RESULTS

Visualizations of Ablation Studies.

Figure 5 compares 3D filtering only, w/o RC, and Full OUR-GS. These qualitative ablations, together with the results in Table 5, demonstrate that our redundancy handling effectively reduces Gaussian redundancy while preserving the representational ability of the Gaussian map.



3D filtering only | w/o RC | **Full OUR-GS** | GT

Figure 5: **Qualitative ablations of redundancy handling.**

Sensitivity to Tracking Systems. To assess whether our reconstruction gains are robust to different tracking systems, we perform online reconstruction on GaussianPro using camera poses and sparse points from different tracking systems, including ORB-SLAM3 (Campos et al., 2021), MASt3R (Murai et al., 2025) and COLMAP (Schonberger & Frahm, 2016). We compare OUR-GS with MODP (Wu et al., 2025b), as both methods support inputs from different tracking systems. Results averaged across scenes in Table 6 indicate that OUR-GS substantially improves reconstruction quality over MODP with all tracking systems, demonstrating the robustness of our reconstruction gains to the applied tracking system. Here COLMAP provides camera poses and sparse points estimated offline, which are supplied to both online mappers only to assess whether our mapping gains persist across different tracking inputs.

Table 6: **Sensitivity to tracking inputs** on GaussianPro. For each tracking system, both mappers receive identical camera poses and sparse points.

Tracker	Mapper	GaussianPro		
		PSNR↑	SSIM↑	LPIPS↓
ORB-SLAM3	MODP	24.19	0.705	0.403
	OUR-GS	**25.04**	**0.742**	**0.148**
MASt3R	MODP	23.93	0.703	0.357
	OUR-GS	**24.47**	**0.748**	**0.309**
COLMAP	MODP	27.42	0.863	0.162
	OUR-GS	**30.42**	**0.928**	**0.080**

Visualizations of More Comparisons. Figures 6, 7, and 8 present further reconstruction comparisons on more scenes spanning different datasets. The boxed insets enlarge details from regions at different viewing distances for easier inspection. Baselines may exhibit blurred structures or floating artifacts in these outdoor UAV views, where wide depth ranges and uneven parallax can limit reconstruction quality. In contrast, OUR-GS produces sharper reconstructions with fewer artifacts. Benefiting from BGME's balanced allocation across depth ranges, OUR-GS better preserves distant structures (buildings and mountain ridges in 360DVO). MDR corrects inconsistent depths, while TRDU adapts position updates to distance and geometric support, helping resolve finer details (tower structures in GaussianPro and facades and rooftops across Wild-UAV scenes). Additional video visualizations are provided in the supplementary material.

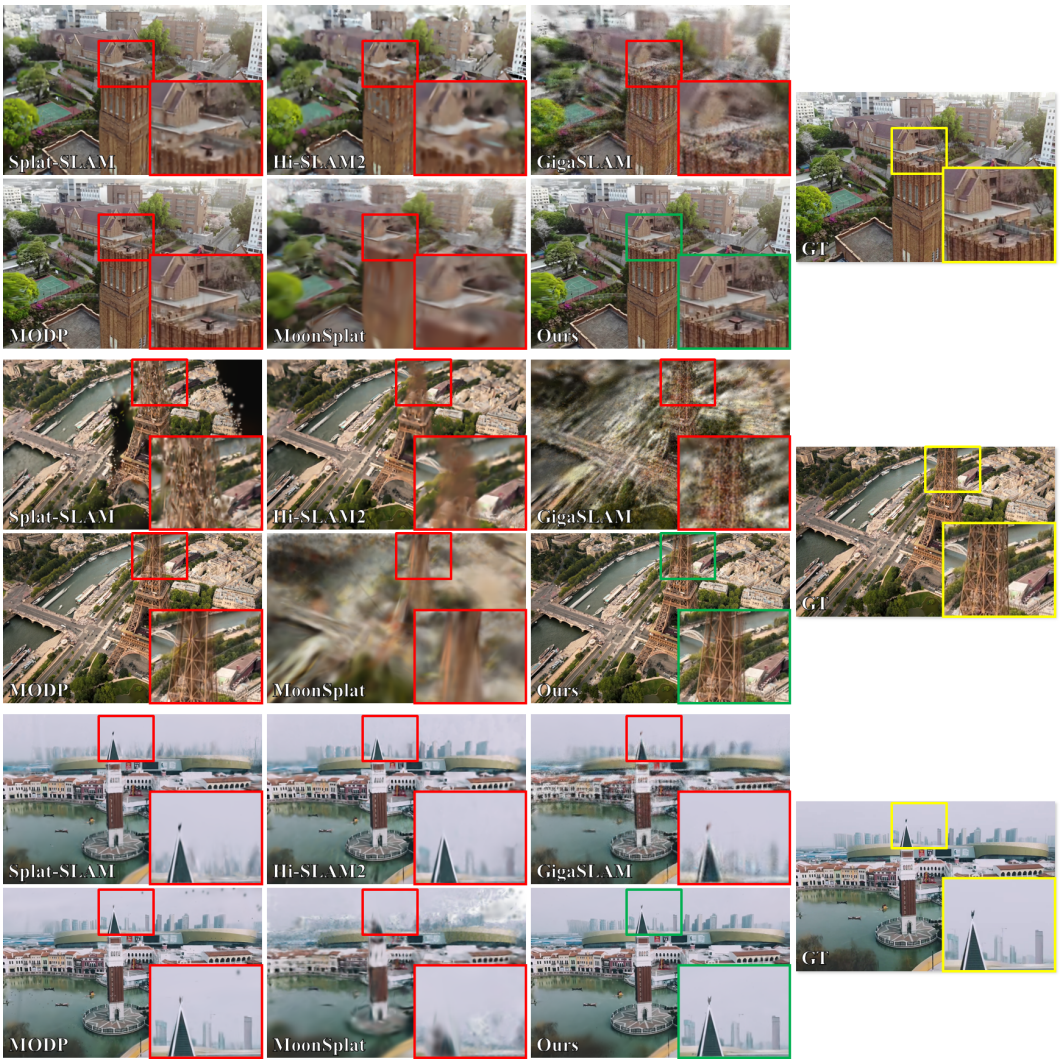

Figure 6: **Additional qualitative comparisons on GaussianPro.** Zoom in for a better visualization.

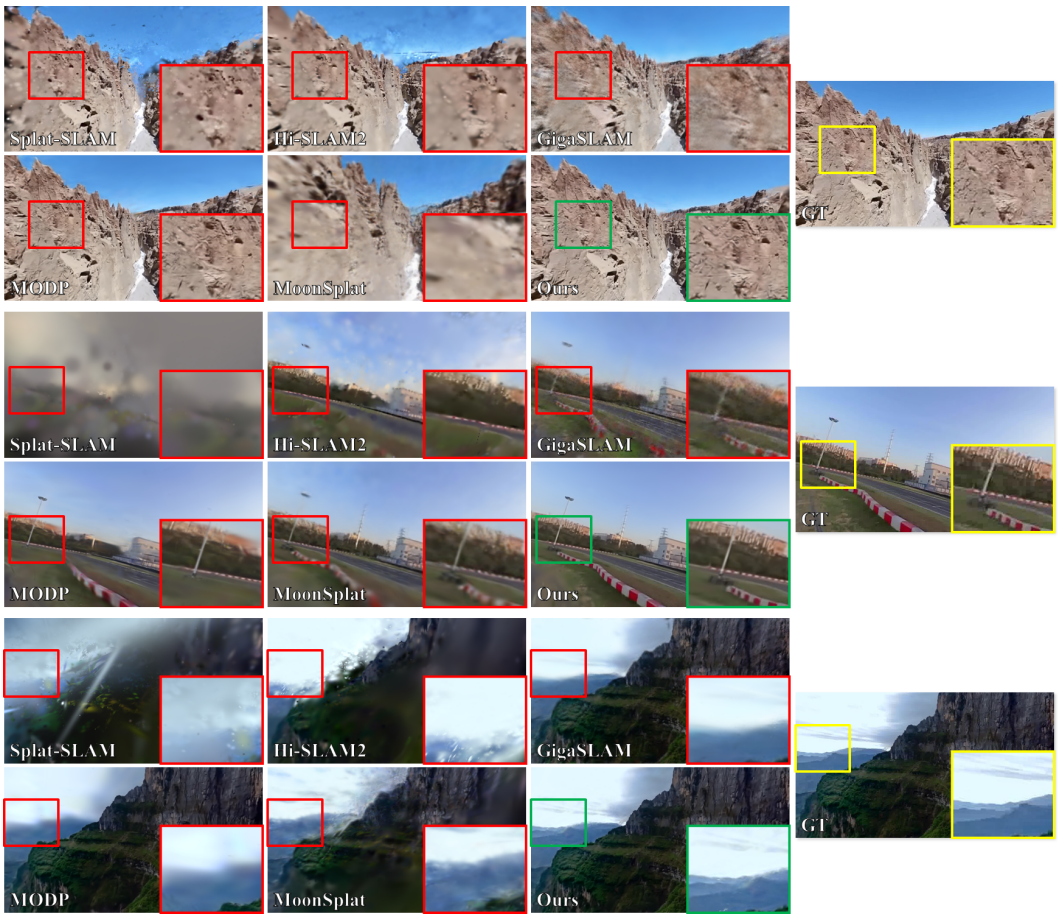

Figure 7: **Additional qualitative comparisons on 360DVO.** Zoom in for a better visualization.

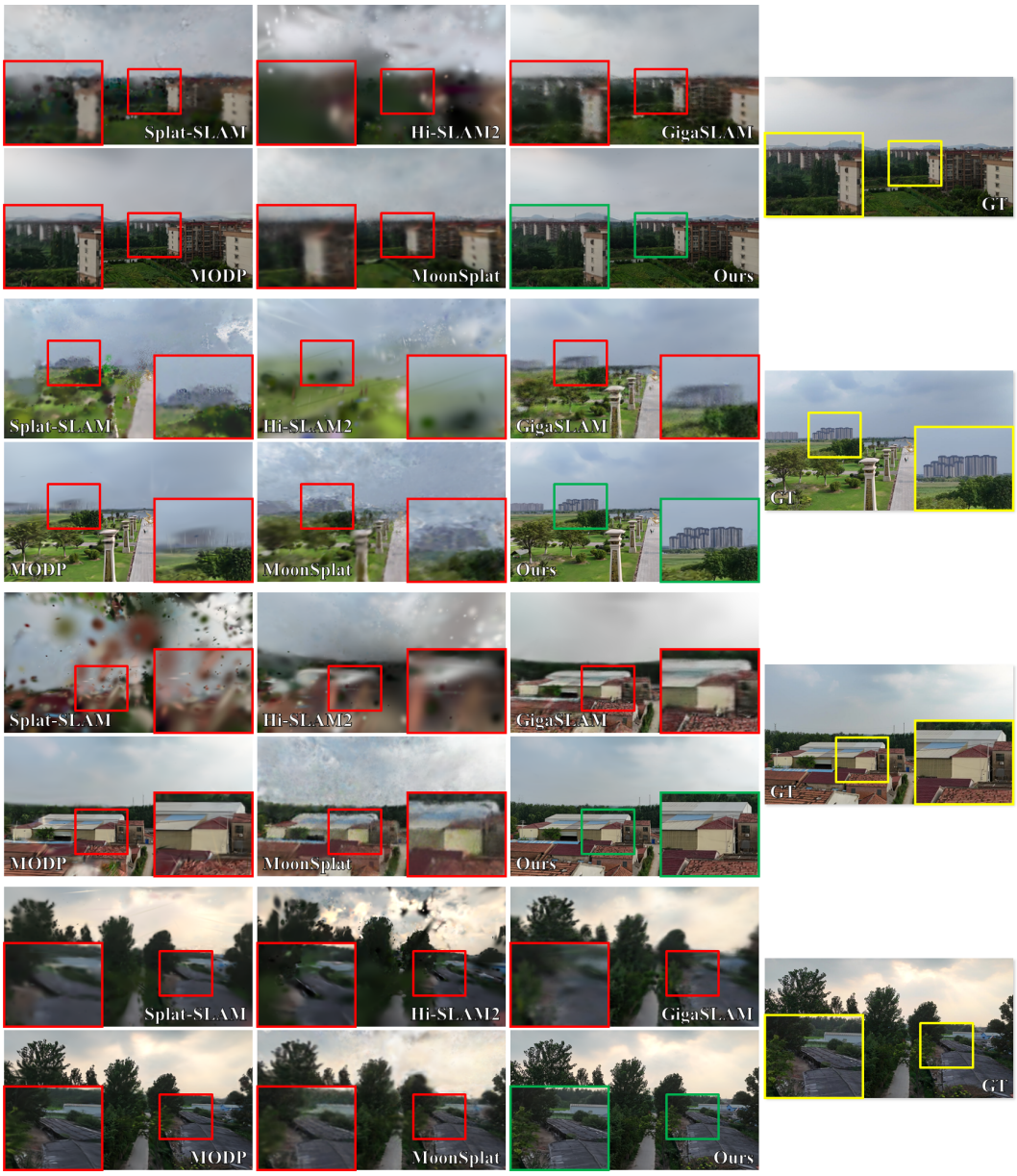

Figure 8: **Additional qualitative comparisons on Wild-UAV.** Zoom in for a better visualization.

B MORE METHOD DETAILS

Implementation Details. In online reconstructions, we set $\sigma_{\max} = 0.06$ and $\epsilon = \log(1.08)$, which are described in §3. LoFTR uses outdoor pretrained weights with a confidence threshold of 0.5 and a Sampson-error threshold of 1.5 pixels. The sampling budget B for DepthCov candidates is set to 1800. Gaussians are initialized with opacity 0.5, identity rotation, and colors sampled from their source images. For DepthCov-initialized Gaussians, the isotropic scale is $\exp(0.5)\widetilde{D}_{t_i}(\boldsymbol{u}_i)/(2f_{t_i})$, where $f_{t_i} = (f_{x,t_i} + f_{y,t_i})/2$ is the mean focal length of the source frame in pixels. Gaussians initialized from sparse points follow the same rule using their tracked depths. Keyframe online optimization uses 10 Adam iterations to minimize the rendering loss $\mathcal{L} = 0.8L_1 + 0.2(1 - \mathrm{SSIM})$.

Gaussian State. In addition to standard Gaussian parameters, including position, color, and scale, each Gaussian i initialized from DepthCov predictions retains the auxiliary state:

$$\mathcal{A}_i = (t_i, \mathbf{u}_i, \mathbf{o}_{t_i}, \beta_i, \chi_i, \sigma_i, \eta_i)\,. \tag{12}$$

As described in §3, t_i, $\mathbf{u}_i$, and $\mathbf{o}_{t_i}$ denote its source frame index, source pixel, and source camera center, respectively. The source frame and pixel identify its original viewing ray. The remaining attributes are the cumulative applied log-depth correction β_i, redundancy branch label χ_i (0 for 3D checks and 1 for image-space checks), depth uncertainty σ_i from DepthCov, and parallax statistic η_i. These attributes support depth corrections by MDR and Gaussian position updates by TRDU.

B.1 MULTI-VIEW DEPTH REASSESSMENT

Reference-frame Selection. For each incoming image I_t, MDR selects two reference frames from its 24-frame active window before image matching, according to the camera poses and intrinsics. The selection favors camera translations that constrain depth across the image. Let $\boldsymbol{R}_\tau$ be the world-to-camera rotation of frame τ, and let π_τ denote its perspective projection with intrinsics. For a pixel $\boldsymbol{u}$ in current frame I_t, let $\widetilde{\boldsymbol{u}} = (u_x, u_y, 1)^\top$ and let ϱ denote its inverse depth. Its projection into a historical frame I_τ can be expressed as:

$$\boldsymbol{p}_{\tau t}(\varrho, \boldsymbol{u}) = \pi_\tau\big(\boldsymbol{R}_\tau \boldsymbol{R}_t^\top \boldsymbol{K}_t^{-1}\widetilde{\boldsymbol{u}} + \varrho \boldsymbol{R}_\tau(\boldsymbol{o}_t - \boldsymbol{o}_\tau)\big), \tag{13}$$

where $\boldsymbol{K}_t$ is the current camera intrinsic matrix. We divide the current image I_t into a 3×3 grid and denote the nine grid centers by $\mathcal{U}$. Each grid center pixel $\boldsymbol{u}$ in I_t defines a viewing ray from the current camera center. Points at different depths along this ray project to the same pixel $\boldsymbol{u}$ in I_t, but generally to different pixels in a historical image I_τ. For each candidate frame I_τ, we use the camera poses and intrinsics to measure how this historical projection changes with inverse depth ϱ. This sensitivity gives the score $\mathcal{I}_\tau(\boldsymbol{u})$: a larger value indicates that the historical view can better distinguish different depths along the current viewing ray. $\mathcal{I}_\tau(\boldsymbol{u})$ is expressed as:

$$\mathcal{I}_\tau(\boldsymbol{u}) = \frac{1}{\xi_{\mathrm{pix}}^2}\left\| \frac{\partial \boldsymbol{p}_{\tau t}(\varrho, \boldsymbol{u})}{\partial \varrho}\Big|_{\varrho=0} \right\|_2^2, \qquad \xi_{\mathrm{pix}} = 0.75 \text{ pixels}. \tag{14}$$

Here ξ_{pix} is the assumed standard deviation of pixel localization noise. The derivative is evaluated at the infinite-depth limit, i.e. $\varrho = 0$, so reference selection does not require an initial depth estimate. We set $\mathcal{I}_\tau(\boldsymbol{u}) = 0$ if the projected direction lies behind the reference camera or outside its image. For each candidate frame I_τ, we compute

$$\mathcal{S}_\tau = \sum_{\boldsymbol{u}\in\mathcal{U}} \log\Big(1 + \frac{\mathcal{I}_\tau(\boldsymbol{u}) + \mathcal{I}_{\mathrm{ref}}(\boldsymbol{u})}{m_{\mathcal{I}}}\Big), \tag{15}$$

where $m_{\mathcal{I}}$ is the median positive score across all candidate frames and grid centers. If all scores are zero, the most recent available frames are selected. Initially, $\mathcal{I}_{\mathrm{ref}}(\boldsymbol{u}) = 0$, and the candidate with the largest $\mathcal{S}_\tau$ is selected as the first reference I_{τ_1}. We then set $\mathcal{I}_{\mathrm{ref}}(\boldsymbol{u}) = \mathcal{I}_{\tau_1}(\boldsymbol{u})$, recompute the scores for the remaining candidates, and select the highest-scoring frame as the second reference. This encourages the second reference to strengthen depth constraints in image regions less supported by the first.

Triangulation. For a target observation $\boldsymbol{u}_{\tau q}$, MDR triangulates the other retained observations in correspondence group q, requiring observations from at least two other frames. An initial 3D point is obtained by least-squares intersection of their viewing rays and refined by minimizing reprojection

error with fixed camera poses. Its depth in frame I_τ gives the reference depth $\widehat{D}_{\tau q}$ used in § 3.1. The target observation is excluded from triangulation and used to check the point by reprojection. Only observations available at the current time are used, allowing retained historical observations to be reassessed as new views arrive. As the window advances, correspondence groups incorporate new matches and retain a bounded observation history for triangulation. Groups receiving no new observations for more than 96 frames are removed.

Pixel localization uncertainty is propagated through the triangulation fit to estimate the log-depth standard deviation $\widehat{\sigma}_{\tau q}$ of the reference depth $\widehat{D}_{\tau q}$. The discrepancy $y_{\tau q}$ compares this reference with the DepthCov prediction in log depth. We compute its variance as:

$$\nu_{\tau q}^2 = \widehat{\sigma}_{\tau q}^2 + \sigma_\tau(\boldsymbol{u}_{\tau q})^2. \tag{16}$$

For a pair $e = (\tau, s, q)$, the residual r_e measures disagreement along $\boldsymbol{n}_e$, as described in Eq. 3. To first order, $D_\tau(\boldsymbol{u}_{\tau q})\sigma_\tau(\boldsymbol{u}_{\tau q})$ approximates the depth standard deviation in frame τ. Its contribution along $\boldsymbol{n}_e$ is scaled by $\boldsymbol{n}_e^\top \boldsymbol{d}_{\tau q}$, with the corresponding contribution from frame s computed in the same way. Adding the two variance contributions gives:

$$\nu_e^2 = [(\boldsymbol{n}_e^\top \boldsymbol{d}_{\tau q})\, D_\tau(\boldsymbol{u}_{\tau q})\, \sigma_\tau(\boldsymbol{u}_{\tau q})]^2 + [(\boldsymbol{n}_e^\top \boldsymbol{d}_{sq})\, D_s(\boldsymbol{u}_{sq})\, \sigma_s(\boldsymbol{u}_{sq})]^2. \tag{17}$$

Both variances are evaluated before applying depth corrections and held fixed during each field optimization. The corresponding standard deviations, $\nu_{\tau q}$ and ν_e, are used in the joint optimization of the correction fields, as described in Eq. 4.

Field Optimization. Each frame has a 3×4 grid of correction coefficients $\boldsymbol{a}_\tau$, as defined in § 3.1. We regularize both their magnitudes and the differences between neighboring nodes:

$$\mathcal{R}(\{\boldsymbol{a}_\tau\}) = \frac{1}{2\xi_{\text{coef}}^2} \sum_\tau \sum_{m=1}^{12} a_{\tau m}^2 + \frac{1}{2\xi_{\text{smooth}}^2} \sum_\tau \sum_{(m,n)\in\mathcal{E}_{\text{grid}}} (a_{\tau m} - a_{\tau n})^2, \tag{18}$$

where $a_{\tau m}$ is the m-th coefficient of $\boldsymbol{a}_\tau$, and $\mathcal{E}_{\text{grid}}$ contains horizontally and vertically adjacent node pairs. We set $\xi_{\text{coef}} = 0.20$ and $\xi_{\text{smooth}} = 0.08$. The first term penalizes large correction coefficients. A zero correction field leaves the predicted depths unchanged. The second encourages neighboring image regions to receive similar corrections, producing a smoothly varying field. This regularizer is added to the triangulated-depth and pairwise consistency terms in Eq. 4. We jointly optimize the frame coefficients using four robust Gauss-Newton iterations with Huber threshold 1.345, and constrain each coefficient to $[-\log 1.25, \log 1.25]$. For each incoming keyframe, MDR updates the correction fields of frames involved in the corresponding observation pairs $\mathcal{E}$. Once a frame leaves the active window, its correction field is held fixed at its last estimate.

B.2 Balanced Gaussian Map Expansion

The spatial scale w_i and color tolerance T_c described in §3.2 help determine whether a new 3D candidate is already sufficiently covered by previously accepted samples and can therefore be omitted.

Spatial Scale. The spatial scale w_i is a coverage width proportional to the 3D distance between adjacent source-image pixels back-projected at a reference depth. For the source frame I_{t_i}, we use the median positive depth of the sparse points S_{t_i} from the tracking system as the reference depth $\bar{z}_{t_i}$. With the mean focal length f_{t_i} in pixels, the 3D distance between adjacent back-projected pixels is approximately $\bar{z}_{t_i}/f_{t_i}$. We set the spatial scale $w_i = \frac{\bar{z}_{t_i}}{8 f_{t_i}}$, using one eighth of this distance to keep the coverage width smaller than the spacing associated with adjacent pixels. Candidates from the same source image initially share this width, and an accepted MDR correction will rescale it together with the candidate's depth. The median depth reduces the influence of isolated sparse points at extreme distances.

Among the samples returned by a nearest-neighbor query, we select the closest sample j satisfying $w_j \le w_i$. This condition allows only samples at the same or finer spatial coverage to cover the candidate. Consequently, a coarse sample cannot prevent the addition of a finer sample when the UAV obtains a closer view. As described in Eq. 7, we then check whether the candidate lies sufficiently close to j through:

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \le \min\Big(\ell_j, \frac{w_i}{\sqrt{12}}\Big). \tag{19}$$

ℓ_j is the distance from j to its nearest other sample k returned by the same query with $w_k \leq w_j$. $w_i/\sqrt{12}$ is applied as the spatial distance cap, motivated by the standard deviation of uniform quantization error over an interval of width w_i. This prevents a sample from covering a large region simply because its neighbors are far away.

Color Tolerance. The color tolerance T_c sets the allowed RGB difference between a candidate and its reference sample. This difference is used as a reference for deciding whether a candidate and its selected sample j are similar enough in appearance to be considered redundant. We estimate T_c from the RGB differences between neighboring accepted samples.

For each query, we compute the color differences between j and its eligible neighbors k, defined above. We collect the reference-neighbor pairs (j, k) obtained during candidate redundancy checks and denote this collection by $\mathcal{P}$. With RGB values normalized to $[0, 1]$, we compute T_c as the root mean square of the color differences:

$$T_c = \sqrt{\frac{1}{|\mathcal{P}|} \sum_{(j,k)\in\mathcal{P}} \|\boldsymbol{c}_j - \boldsymbol{c}_k\|_2^2}. \tag{20}$$

Here, $|\mathcal{P}|$ is the number of collected comparisons. Two spatially close samples may lie on opposite sides of a texture boundary. We therefore reject a candidate only when the spatial condition in Eq. 19 holds and $\|\boldsymbol{c}_i - \boldsymbol{c}_j\|_2 \leq T_c$, as described in Eq. 7. A candidate with a larger color difference is not rejected by this comparison, allowing nearby samples to contribute distinct appearance details.

B.3 Tangential–Radial Decomposed Update

Distance Scaling. A tangential displacement of magnitude $d_i^{\perp}$ at reference distance r_i changes the viewing direction by $\Delta\theta_i$, satisfying

$$d_i^{\perp} = r_i \tan \Delta\theta_i. \tag{21}$$

Thus, producing the same change in viewing direction requires a tangential displacement proportional to r_i. For a radial displacement $d_i^{\parallel}$ from the reference position, the log-range change satisfies

$$\log \frac{r_i + d_i^{\parallel}}{r_i} = \log\left(1 + \frac{d_i^{\parallel}}{r_i}\right) \approx \frac{d_i^{\parallel}}{r_i}, \qquad |d_i^{\parallel}| \ll r_i. \tag{22}$$

These relations motivate scaling both components by $\gamma_i = r_i/r_{\text{unit}}$ in Eq. 11. r_{unit} is a fixed length for distance normalization and is set to 1 in map coordinates across all sequences. g_i further controls the radial step according to depth uncertainty and parallax.

Computation of η_i. The statistic η_i measures the parallax between the source ray and viewing rays at other available observations. We consider up to four most recent historical keyframes. For Gaussian i with initialized position $\boldsymbol{\mu}_i^{\text{ini}}$ from BGME (§ 3.2), $\mathcal{V}_i^0$ contains the indices of historical frames in which its initialization position has positive depth and projects inside the image. For each $\tau \in \mathcal{V}_i^0$, the camera center $\boldsymbol{o}_\tau$ defines a unit viewing direction:

$$\boldsymbol{v}_{i\tau} = \frac{\boldsymbol{\mu}_i^{\text{ini}} - \boldsymbol{o}_\tau}{\|\boldsymbol{\mu}_i^{\text{ini}} - \boldsymbol{o}_\tau\|_2}. \tag{23}$$

The initial parallax statistic relative to the source viewing direction $\boldsymbol{v}_i$ is given by:

$$\eta_i = \max_{\tau\in\mathcal{V}_i^0} \left[1 - (\boldsymbol{v}_i^\top \boldsymbol{v}_{i\tau})^2\right]. \tag{24}$$

The term $1 - (\boldsymbol{v}_i^\top \boldsymbol{v}_{i\tau})^2$ equals $\sin^2\theta_{i\tau}$, where $\theta_{i\tau}$ is the angle between $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i\tau}$. Taking its maximum uses the strongest parallax among the eligible historical views. If $\mathcal{V}_i^0$ is empty, we set $\eta_i = 0$. For far Gaussians with $\chi_i = 1$, we subsequently update η_i once a new keyframe I_t arrives. The projection results from its mapping render identify Gaussians whose current means have positive camera-space depth and project inside the image. For these eligible Gaussians, $\boldsymbol{v}_{it}$ is evaluated as in Eq. 23, and the statistic is updated as

$$\eta_i \leftarrow \max\left\{\eta_i,\, 1 - (\boldsymbol{v}_i^\top \boldsymbol{v}_{it})^2\right\}. \tag{25}$$

This statistic η_i is retained for computing g_i in Eq. 10 during optimization.

C DISCUSSIONS

Impact. Offline aerial reconstruction typically produces a final 3D model only after complete image collection and extensive post-flight optimization. In contrast, embodied UAVs operating in inspection, exploration, or search-and-rescue missions cannot postpone spatial understanding until the flight ends. They require an evolving 3D representation of the surrounding environment while observations are still arriving, so that downstream planners or operators can identify insufficiently observed regions, select additional viewpoints, adjust flight routes, and revisit important areas before leaving them. By integrating observations over time, such a representation also provides persistent spatial context beyond what can be inferred from individual incoming frames. Online reconstruction therefore turns the 3D map from a post-flight product into a continuously updated spatial state within the UAV's perception-action loop, enabling scene understanding to influence data acquisition and subsequent actions during the mission.

OUR-GS supports this role by maintaining a compact, renderable Gaussian map under incomplete observations, wide depth ranges, and uneven multi-view evidence. It can serve as an online perception backend for active inspection, viewpoint planning, and operator-in-the-loop decision-making.

Limitations. Despite the strong online reconstruction performance, OUR-GS still has limitations. OUR-GS does not yet explicitly handle dynamic objects in UAV observations. Modeling dynamic scenes during large-scale online reconstruction remains challenging, as object motion must be distinguished from camera motion under uneven parallax and limited computation. In addition, the effects of extreme lighting and adverse weather conditions on reconstruction have not yet been considered. Extending OUR-GS to reconstruct dynamic environments and exploring its application under these challenging conditions are important directions for our future work.