

OpenFlyScan: A Quality-Guided Aerial Reconstruction System for Consumer Drones

Zhongrui You^{1,2}, Zhen Li^{2,3}, Junli Liu^{2,4}, Zhigang Wang², and Bin Zhao^{2,4}

Abstract—3D Gaussian Splatting (3DGS) provides high-fidelity scenes for large-scale embodied simulation, but constructing large-scale urban assets remains constrained by expensive equipment and delayed quality feedback. Preset surveys can leave complex surfaces insufficiently observed, with defects discovered only after reconstruction, requiring return visits and repeated processing. We present OpenFlyScan, a quality-guided aerial reconstruction system for consumer drones that integrates a GS quality model, a reacquisition planner, and a custom-designed mobile app. The model learns from GS rendering errors to predict regional reconstruction quality. Based on these predictions, the planner then generates complementary reacquisition strips to be executed through the app, which also supports automated oblique surveys and data transfer without additional hardware on board. Across real aerial scenes, the model effectively identifies regions that are likely to be poorly reconstructed. In the Expo West field experiment, targeted reacquisition improves PSNR at additional views by 10.95 dB. With consumer drones, OpenFlyScan integrates capture, targeted reacquisition, and reconstruction to support rapid, low-cost urban asset creation. We will open-source the code and mobile apps to support further research.

I. INTRODUCTION

3D Gaussian Splatting (3DGS) reconstructs real scenes from multi-view images and efficiently synthesizes realistic novel views [1]. It provides visual environments for robot simulation [2] and aerial vision-language navigation (VLN) [3]. Large-area navigation and multi-agent tasks motivate extending these environments to block-scale scenes covering building facades and the ground. However, constructing these large-scale scenes remains costly in equipment and labor.

Obtaining sufficient spatial coverage and multi-view observations requires coordinated flight planning and camera settings for multidirectional oblique surveys. Such automation is primarily available on enterprise surveying platforms from manufacturers such as DJI [4], [5], whose higher purchase costs and heavier equipment burden field deployment. Consumer drones are lighter and widely available, but need corresponding planning and execution capabilities for large-scale, multidirectional capture.

Moreover, covering the survey area with preset oblique flight strips does not ensure sufficient observations of complex surfaces for high-quality 3DGS reconstruction [6]. High-rise facades and irregular structures can still suffer from occlusion, large incidence angles, and limited viewing directions, while operators cannot readily judge the need for

reacquisition from transmitted images alone. Quality problems often become apparent only after full reconstruction, leading to return visits and repeated processing that increase time and labor costs [7].

Shortening this feedback cycle requires identifying regions worth reacquiring before full 3DGS reconstruction. Active capture methods provide geometric feedback through multi-view stereo (MVS) reconstruction confidence prediction [6] or mesh-quality assessment within incremental Structure-from-Motion (SfM), as in On-the-fly Feedback SfM [7] (hereafter On-the-fly), but leave two issues unresolved. First, missing geometry can result from insufficient observations or failed feature matching on weakly textured surfaces and across large viewpoint changes [8], [9]; geometry alone cannot distinguish these causes. Second, even geometrically supported surfaces may render poorly because of appearance and illumination changes. Feed-forward 3D models rapidly provide geometry and related features [10], [11], but these cues alone cannot fully reveal potential 3DGS reconstruction defects.

We propose OpenFlyScan, a quality-guided aerial reconstruction system for consumer drones that integrates a GS quality model, a reacquisition planner, and a custom-designed mobile app (Fig. 1). Building on pretrained feed-forward geometry, our GS quality model uses a Quality Predictor to fuse multi-view image features and relative camera geometry with geometric cues, learning their relationship to regional GS reconstruction quality. Trained across a broad collection of scenes, the model provides regional quality feedback before target-scene reconstruction. Regions predicted to reconstruct poorly are selected as reacquisition targets, for which the planner arranges complementary views into continuous flight strips. Additional images are combined with the initial-survey images for reconstruction. To make this feedback and reacquisition workflow practical on consumer drones, our mobile app coordinates automated oblique surveys, data transfer, and reacquisition without additional onboard computing or communication hardware, lowering the equipment barrier to automated capture.

Across UAVFF3D-Real, Expo West, and Dishui Lake aerial surveys, our model identifies regions with high GS reconstruction error more accurately than On-the-fly. In real-world deployment, our system uses DJI Mini-series consumer drones for data capture and targeted reacquisition, completing high-quality 3DGS reconstruction of an approximately 1 km² area within 10 hours. Taking Expo West as an example, targeted reacquisition improves PSNR at additional views by 10.95 dB while improving initially weak regions.

¹Beihang University, Beijing, China.

²Shanghai Artificial Intelligence Laboratory, Shanghai, China.

³Shanghai Jiao Tong University, Shanghai, China.

⁴Northwestern Polytechnical University, Xi'an, China.

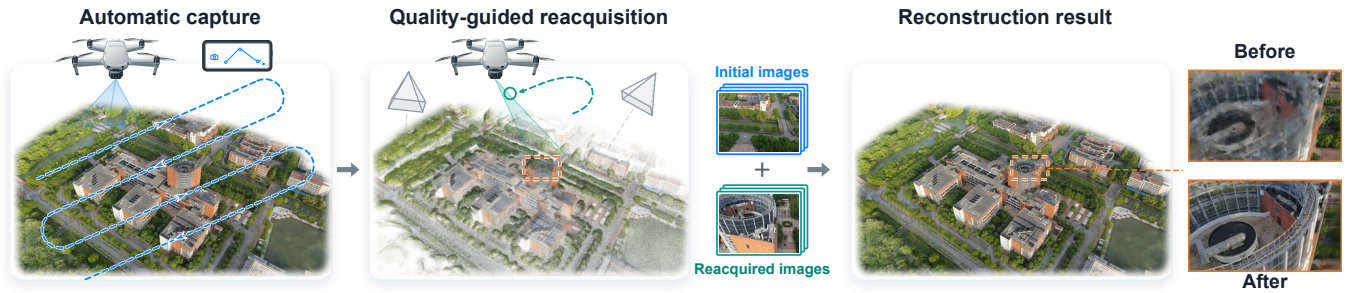


Fig. 1. OpenFlyScan workflow: automated capture with consumer drones, timely regional quality feedback, targeted reacquisition, and joint 3DGS reconstruction. Routes and viewing directions are schematic.

The main contributions of this paper are:

- **A lightweight capture and feedback system for consumer drones.** It enables automated oblique surveys, on-site feedback, and targeted reacquisition without additional onboard hardware.
- **A cross-modal regional quality prediction model for 3DGS reconstruction.** It combines image features, feed-forward geometry, and relative camera geometry to identify low-quality regions before target-scene 3DGS training.
- **A quality-guided reacquisition and reconstruction workflow.** It converts quality predictions into executable flight strips, with simulation and field experiments demonstrating improvements in poorly reconstructed regions through joint reconstruction.

II. RELATED WORK

A. Large-Scale Urban Reconstruction and Rendering

3DGS enables high-quality novel-view synthesis and real-time rendering [1]. Hierarchical 3DGS [12] and CityGaussianV2 [13] extend Gaussian reconstruction to large scenes through hierarchical representations and geometry-aware training, respectively. CityGaussianV2 reports road and facade reconstruction failures under occlusion and missing observations, while Hierarchical 3DGS identifies limited view coverage and calibration errors as sources of visual artifacts. Reliable multi-view capture therefore remains essential to constructing high-quality urban assets at scale.

B. UAV Acquisition and Active View Planning

Systematic oblique surveys coordinate flight-strip spacing, altitude, speed, and camera orientation to acquire dense multi-view observations. Enterprise platforms integrate survey planning and execution but require costly equipment: SkyLume uses a DJI M350 RTK with a C30 five-lens camera, while Praschl et al. use a DJI M30T for building reconstruction [4], [5] (Table I). DJI Mini drones can capture oblique views but lack an official automated oblique-survey workflow. DJI Mobile SDK exposes flight, gimbal, and camera controls, enabling automated surveys [14].

Preset survey coverage does not ensure sufficient observations of complex surfaces, motivating feedback-driven view planning. SCONE and MACARONS predict additional surface coverage to guide exploration [15], [16]. Mostegel

TABLE I
UAV ACQUISITION PLATFORMS

Model	Price (USD)	Weight (kg)	Official automated oblique survey
DJI M30T, <i>Praschl</i>	9,637	3.77	Yes
DJI M350 RTK, <i>SkyLume</i>	26,470	6.47	Yes
DJI Mini 2, <i>Ours</i>	449	<0.249	No
DJI Mini 4 Pro, <i>Ours</i>	759	<0.249	No

et al. predict MVS reconstruction confidence from acquired images to guide subsequent observations [6]. Learning Reconstructability predicts reconstructability from proxy geometry, viewpoints, and optional images to guide urban aerial path planning [17]. On-the-fly Feedback SfM combines incremental SfM, mesh-quality assessment, and predictive path planning to explore unknown areas and revisit poorly reconstructed regions [7].

Radiance-field methods also use the current model to guide acquisition. ActiveGS identifies under-reconstructed regions using Gaussian-map confidence [18]; FisherRF estimates candidate-view information gain through Fisher information [19]. Auto3R predicts uncertainty from current GS color and depth renderings to guide scanning [20]. These methods require recovered geometry or a current radiance field for feedback.

C. Feed-Forward Geometry and Quality Prediction

Feed-forward models such as VGGT and π^3 enable rapid scene perception by predicting camera poses and dense geometry directly from multi-view images [10], [11]. UAVFF3D benchmarks these capabilities on UAV imagery and provides aerial-adapted models [21], while GeoFF3D combines coordinate anchoring and spatial chunking for large-scale aerial reconstruction [22]. SwiftMap combines ORB-SLAM3 tracking and VGGT-based quality evaluation to provide on-site feedback for repeated flights [23]. However, geometric confidence and coverage do not directly measure GS rendering quality under available observations.

OpenFlyScan instead predicts regional GS reconstruction quality from captured images and feed-forward geometry, before optimizing the target scene's GS representation.

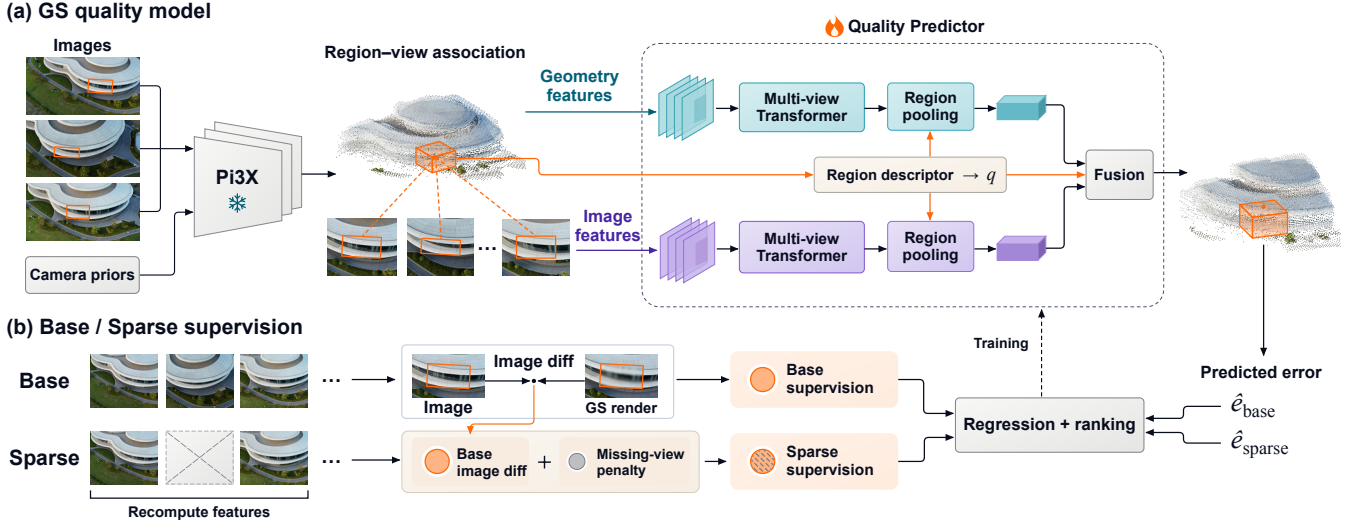


Fig. 2. GS quality model: (a) regional error prediction from geometry and image features; (b) Base/Sparse supervision. Snowflake/flame: frozen/trainable parameters. Region markers and image differences are schematic.

III. METHOD

OpenFlyScan connects initial capture with targeted reacquisition through on-site quality feedback. The mobile app manages data capture and mission execution on consumer drones, while a remote workstation predicts regional reconstruction quality before GS training and plans additional views. Initial-survey and additional images are jointly reconstructed.

A. Automatic Capture and Feedback

We develop Android and iOS applications using DJI MSDK [14] to enable automated multidirectional oblique surveys with consumer drones and integrate this capability with workstation-based quality assessment and reacquisition planning. We use the standard remote controller and the phone’s cellular or Wi-Fi connection to communicate with the workstation, without mounting additional computing or communication hardware on the drone, as required by SwiftMap [23]. The apps support 21 DJI models across the Mini, Air, Mavic, Spark, and Phantom series; our field surveys use the Mini 2 and Mini 4 Pro.

Capture planning. The operator specifies the survey area, flight altitude, viewing directions, and image overlap. The app projects the camera footprint onto a local horizontal plane using flight altitude, field of view, and gimbal pitch to set strip spacing and capture positions. To maintain the requested overlap, we constrain flight speed to $v \leq s_{\text{photo}}/T_{\text{min}}$, where s_{photo} is the capture spacing and T_{min} is the camera’s minimum capture interval.

On-site feedback. Uploading observations during flight lets data transfer overlap with capture, reducing the wait for image export after landing. After each capture trigger, the app asynchronously reads the current video frame and uploads it with timestamped GPS and camera metadata for analysis. High-resolution images remain on the drone for

final reconstruction. The workstation returns the point cloud, low-quality region markers, and proposed routes for operator review. In addition to regular survey paths, the app accepts specified camera positions, orientations, and capture actions from the workstation to execute local reacquisition.

B. Regional Reconstruction Quality Prediction

Our GS quality model consists of a pretrained Pi3X backbone and a trainable cross-modal Quality Predictor. The Quality Predictor outputs a normalized regional error $\hat{e}_r$ for each 3D region; larger values indicate poorer predicted quality and greater reacquisition priority. Fig. 2 shows the prediction pipeline and its shared Base/Sparse supervision.

Geometry and evaluation regions. We use UAVFF3D’s UAV-adapted Pi3X model [21], [24] to predict point clouds, confidence, and intermediate features from images, intrinsics, and camera pose priors derived from capture metadata. To control memory use and maintain spatial consistency, we follow GeoFF3D’s spatial large-scale reconstruction framework (SLRF) [22]: Pi3X processes overlapping image chunks defined by their footprints, and shared-image point correspondences provide Sim(3) alignment into a common frame.

Within each chunk’s point cloud, a region is a small spherical neighborhood whose radius follows the local sampled-point spacing. Region-view association in Fig. 2(a) uses point-to-image correspondences to find the image patches observing this neighborhood. Features at these corresponding locations describe the same local surface across views.

Training supervision. Fig. 2(b) shows the supervision construction. Training scenes provide both captured images and completed GS reconstructions. We compare rendered images with the captured images at the original camera poses and aggregate their exposure-corrected $\log_{10}$ -MSE into each 3D region. These regional errors, denoted $\bar{g}_{\text{base}}$, provide Base supervision for samples with the full local image set.

To represent insufficient observation coverage, we withhold a subset of input views and rerun Pi3X on the remaining images to obtain features under reduced observations (Sparse). Region matching links each reduced-view region to its Base counterpart. Sparse supervision combines the corresponding Base rendering error with a missing-view penalty based on the fraction of supporting views withheld:

$$\tilde{g}_{\text{sparse}} = \bar{g}_{\text{base}} + \Delta g_{\text{max}} \left[1 - \exp \left(-\frac{r_{\text{drop}}}{\tau} \right) \right]. \quad (1)$$

Here, r_{drop} is the fraction of supporting views withheld. Fitting to measured GS rendering errors under reduced observations gives $\Delta g_{\text{max}} = 0.22$ and $\tau \approx 0.06$. Both signals share the same training-derived normalization, mapping regional errors to $[0, 1]$ while preserving order.

Per-view features. The geometry and image branches in Fig. 2(a) first summarize observation geometry and surface appearance separately, then combine them to predict regional error. For every view observing a region, the geometry branch uses Pi3X point- and confidence-decoder features, relative camera geometry, and cross-view consistency measures to describe observation conditions and geometric agreement. The image branch uses features from the Pi3X image encoder [25] and local color statistics to describe appearance. Both branches use relative camera position, viewing direction, and distance, computed from aligned predicted poses and region positions.

Multi-view aggregation. A region descriptor summarizes the region’s position, shape, and observation statistics and is projected into query q . Each branch uses a one-layer, four-head Multi-view Transformer to combine information across supporting views. Region pooling uses q to guide attention over valid views and produce one vector per branch. Each pooled vector also produces an initial error estimate. A two-layer fusion MLP combines these vectors, estimates, and region statistics to output the predicted regional error $\hat{e}_r$.

Training. The Pi3X backbone remains frozen. We train the shared Quality Predictor on both sample types using weighted regional regression and pairwise ranking, with auxiliary view-level supervision from measured GS errors, as shown in Fig. 2(b). The overall loss is

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \lambda_{\text{rank}} \mathcal{L}_{\text{rank}} + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}. \quad (2)$$

Here, $\mathcal{L}_{\text{reg}}$ is a Smooth L1 loss between predicted regional errors and the normalized Base/Sparse targets, while $\mathcal{L}_{\text{rank}}$ encourages regions with larger target errors to receive higher predictions. The auxiliary loss $\mathcal{L}_{\text{aux}}$ applies regression and ranking to view-level error predictions before regional pooling; λ_{rank} and λ_{aux} control the relative contributions of the ranking and auxiliary terms.

C. Targeted Reacquisition and Joint Reconstruction

Reacquisition targets and candidate strips. We merge duplicate predicted low-quality regions identified across local chunks, retain the representatives with the largest predicted errors, and select a specified number as reacquisition targets. Nearby targets that can be captured together form a capture task. We jointly plan flight strips across targets rather than

selecting viewpoints independently, allowing shared observations and reducing redundant capture.

Scattered local targets use short parallel flight strips. When targets cover a larger contiguous area, the planner uses a local grid with nadir views and oblique views in four directions. Surface orientations and existing observations guide candidate strips, with capture positions determined by projected camera footprints and the requested overlap.

Coverage and complementary observations. We first select strips to cover the target regions and their principal viewing directions. Coverage is checked separately for each region–direction group. Strips are scored by their strongest usable view pair. A target is covered when its selected-strip score reaches 80% of its best candidate score; each group must cover 95% of the predicted-error weight among targets with valid candidates. Greedy selection favors coverage gained per additional image.

Coverage alone can leave support concentrated in similar directions. Candidate-view quality combines field-of-view coverage, confidence-weighted surface incidence, distance, and approximate occlusion. A strip must cover a surface sample from at least three new positions. Stereo support combines paired-view quality with an angular response over 8° – 60° . For each new view, we subtract existing nearby-direction support, estimated with a 35° Gaussian angular kernel, from its best paired support with an existing or another new view. The largest positive difference defines the sample’s added support. This favors complementary observations while retaining selected strips.

We iteratively select the feasible strip package maximizing predicted-error-weighted support gain per additional image:

$$A^* = \arg \max_{A \in \mathcal{F}(H)} \frac{\sum_p w_p g_p(A | H)}{n(A)}. \quad (3)$$

Here, H contains initial-survey observations and expected support from selected strips; $g_p(A | H)$ is the added directional support at surface sample p , and w_p is its weight derived from predicted error. A candidate package A includes flight strips and any intermediate rows needed for continuous scan coverage, and $n(A)$ counts all added images. The feasible set $\mathcal{F}(H)$ enforces image-count, path-length, estimated flight-time, and height limits. After accepting A^* , we update expected support on fixed initial-survey geometry.

Mission execution and joint reconstruction. Selected strips are ordered to reduce transit and executed through the mobile app. Images are captured continuously along strips, not during transit. Initial camera poses remain fixed as the spatial reference. We match the additional images to feature tracks from initial-survey images and estimate their poses from 2D–3D correspondences using PnP/RANSAC [9], [26]. We then combine the observations, retriangulate the sparse geometry, and use OpenMVS [27] to produce a dense initialization for GS training with all images.

To reduce color-parameter storage, we retain the three base-color coefficients of degree-three spherical harmonics (SH3) [1] and encode the 45 higher-order coefficients with an eight-dimensional code and a shared 45×8 linear decoder.

This reduces per-Gaussian color storage from 48 to 11 parameters. The codes and decoder are optimized jointly with the scene. The resulting GS scenes support UAV simulation in AirSim, built on Unreal Engine (UE) [28]. Large-scene rendering uses hierarchical level of detail (LOD) and on-demand loading.

IV. EXPERIMENTS

We evaluate OpenFlyScan on public datasets and real aerial scenes to answer three questions: (Q1) can regional quality prediction identify poorly reconstructed GS regions more accurately than geometric quality and coverage scores; (Q2) what information do image appearance, feed-forward geometry, and relative camera geometry provide for quality prediction; and (Q3) can quality feedback guide on-site reacquisition and improve the final reconstruction?

A. Experimental Setup

We train the Quality Predictor on 17 scene groups containing 44,398 images from four datasets: UAVFF3D [21], GauU-Scene V2 [29], SkyLume [4], and UrbanScene3D [30]. Evaluation uses three UAVFF3D-Real validation sequences and our aerial captures at Expo West and Dishui Lake, all excluded from predictor training. The Quality Predictor and its auxiliary branch are trained for 1,200 steps on 48 GPUs. The Quality Predictor has 1.20M parameters.

Field surveys use DJI Mini 2 and Mini 4 Pro drones with standard controllers and smartphones. Feed-forward geometry processing and experiments with our reconstruction backend use an RTX 4090 workstation, with eight CPU workers for cross-chunk alignment. We evaluate rendered image quality using PSNR.

B. Q1: Regional Reconstruction Quality Prediction

Scene-wide quality ranking. We evaluate predicted regional errors against GS rendering errors measured after reconstruction. The reference errors are computed by comparing rendered images with captured images at the original camera poses, with exposure correction. Recall@20 measures how many of the actual highest-error 20% of regions are recovered when selecting the predicted highest-error 20%; Spearman’s ρ measures agreement across the full ranking. We compare On-the-fly [7], MACARONS [16], and SwiftMap-Adapt [23]. We adapt SwiftMap’s publicly available modules for regional quality evaluation, forming SwiftMap-Adapt, which combines VGGT confidence with local-window pose alignment and regional aggregation. Ours (confidence-only) replaces GS quality model predictions with aggregated Pi3X geometric confidence.

As shown in Table II, our method achieves the highest Recall@20 and rank correlation in all three evaluation groups. Its average Recall@20 is 49.88%, compared with 23.81% for On-the-fly, an improvement of 26.07 percentage points. The predicted error thus prioritizes high-error regions more accurately than the geometric and coverage scores.

In Dishui Lake, Ours (confidence-only) has nearly zero rank correlation with actual GS error ($\rho = -0.001$), whereas

TABLE II
REGIONAL QUALITY RANKING

Method	Metric	UAVFF3D-Real (3)	Expo West	Dishui Lake	All
SwiftMap-Adapt	R@20↑	38.82	23.12	26.45	30.27
	ρ ↑	0.389	0.136	0.226	—
On-the-fly	R@20↑	26.68	25.36	20.01	23.81
	ρ ↑	0.122	0.096	0.010	—
MACARONS	R@20↑	34.60	23.30	22.84	27.66
	ρ ↑	0.210	0.018	0.036	—
Ours (confidence-only)	R@20↑	43.03	34.22	30.16	36.13
	ρ ↑	0.448	0.204	-0.001	—
Ours	R@20↑	51.88	60.51	40.41	49.88
	ρ ↑	0.596	0.680	0.540	—

R@20 (%). UAVFF3D-Real: three-sequence mean.

our model reaches $\rho = 0.540$. Regions judged unreliable by the geometry model are not necessarily those with the poorest rendered images. In the same scene, reconstruction and alignment failures prevent On-the-fly from assigning scores to all but three evaluation regions. For evaluation, we prioritize unscored regions as potential reconstruction failures and assign them equal rank. When this tied group crosses the top-20% cutoff, we compute Recall@20 as the expected recall under random tie-breaking.

Identifying severely degraded surfaces. For target-level comparison in Dishui Lake, we evaluate Ours, Ours (confidence-only), SwiftMap-Adapt, SCONE [15], and MACARONS, each selecting 24 reacquisition targets. Seventeen of ours fall within the highest-error 20% of regions, compared with 2–9 for the baselines. The median initial GS PSNR of our selected regions is 17.81 dB, compared with 21.08–22.00 dB for the baselines. Lower PSNR shows that our targets concentrate on more severely degraded surfaces.

Fig. 3(a) shows a severely degraded facade selected by our method. Facade lines and signage are clear in the captured image but heavily blurred in the rendered image. Panel (b) shows a target-selection miss. Our model ranks this residential facade within the predicted highest-error 5.5%, but spatial aggregation and budget allocation omit it from our final target set; SwiftMap-Adapt selects it. This case highlights a limitation of our target-selection strategy: spatial aggregation and budget allocation can leave severely degraded surfaces without targeted reacquisition, even when their reconstruction errors are correctly identified.

Target distribution and geometric support. Fig. 4(a–d) compares the reacquisition locations selected in Expo West. Our targets concentrate on the central architectural complex, while baseline targets are more dispersed across peripheral roads, vegetation, and terrain.

Fig. 4(e,f) further illustrates how missing geometric support complicates target selection. In (e), parts of the building lack SfM points, leaving geometry-based quality assessment without support on those surfaces. Yet the open water in (f) also contains point-cloud gaps; treating every unsupported region as a defect to repair would produce unnecessary reacquisition targets. Missing geometry alone therefore does

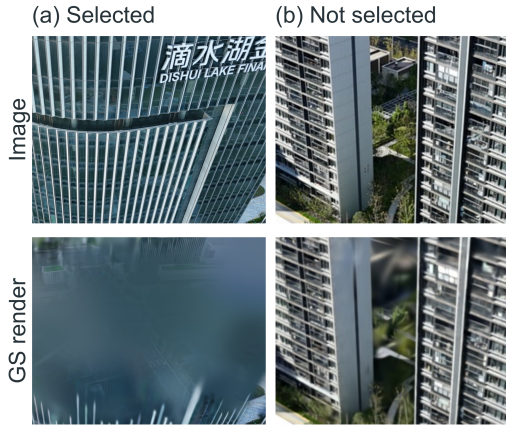


Fig. 3. Dishui Lake: captured images (top) and initial GS renders (bottom). (a) Ours-selected facade; (b) high-scoring facade omitted by Ours but selected by SwiftMap-Adapt.

not determine which surfaces warrant reacquisition.

Fig. 4(g,h) compares the regional priorities produced by the confidence-only baseline and the GS quality model. Dots mark the top 5% of regions under each variant’s ranking; for the GS quality model, violet-to-magenta indicates increasing predicted regional error. The selected regions differ, showing how learned quality prediction changes reacquisition priorities. Table II confirms that the GS quality model more accurately identifies high-error regions.

C. Q2: Information Supporting Quality Prediction

Cumulative feature ablation. We cumulatively ablate the region-specific information in each feature group to examine what the Quality Predictor relies on. Model parameters remain fixed, and each stage retains all preceding ablations.

With full input, mean Recall@20 is 49.88%, as in Table II. In Fig. 5, recall falls to 40.23% after ablating image features, 35.47% after additionally ablating Pi3X point- and confidence-decoder features, and 30.40% after also ablating relative camera geometry. The decline indicates that all three feature groups contribute to regional quality ranking.

D. Q3: Reacquisition Gains and Use in UAV Simulation

Processing time during field operations. For a real outdoor survey of 1,115 images, the on-site feedback workflow from available observations to a reacquisition plan takes approximately 5 min on a single RTX 4090 workstation. Geometry processing accounts for 304.4 s, including data preparation, footprint estimation, feed-forward prediction, cross-chunk alignment, and export.

Joint reconstruction quality and memory. We evaluate reconstruction quality and GPU memory on SkyLume iPark using 1,548 training images and 222 evaluation views. As shown in Table III, under the same 10M-Gaussian cap, our backend produces fewer Gaussians than the original 3DGS with a different densification policy and contribution-based pruning, partly accounting for its lower memory use even with ordinary SH3. In our backend, the compact SH3

configuration reduces memory from 3.564 to 1.820 GiB (48.94%) without reducing PSNR.

TABLE III
RECONSTRUCTION QUALITY AND MEMORY

Configuration	Gaussians (M)	PSNR (dB)↑	Memory (GiB)↓
3DGS	28.93	22.30	43.51
3DGS (10M cap)	9.97	22.75	15.54
Ours (SH3)	1.32	22.71	3.56
Ours (compact)	1.38	22.94	1.82

Peak allocated memory. GPU: A800 80GB (uncapped 3DGS); RTX 4090 (others).

Reacquisition in simulation. We refer to the SfM-based reacquisition baselines as SfM-Q in simulation and SfM in real flight. SfM-Q combines On-the-fly quality scoring [7] with the shared route generator. We compare our method with SfM-Q in NYC and CitySample. Both methods start from the same initial survey and select the same number of targets within each scene: eight in NYC and 18 in CitySample. We evaluate all initial-survey views and the worst 20% of these views before reacquisition, keeping this subset fixed.

TABLE IV
REACQUISITION QUALITY (PSNR, dB)

Scene	Method	All views↑	Worst 20%↑
NYC	Before	26.420	21.753
	SfM-Q	26.605	22.929
	Ours	26.867	24.088
CitySample	Before	28.152	21.320
	SfM-Q	28.336	22.513
	Ours	28.604	23.433

As shown in Table IV, our method improves PSNR on these weak views by 2.335 dB in NYC and 2.113 dB in CitySample, exceeding SfM-Q by 1.159 and 0.921 dB, respectively. More than 75% of the initially weak views improve in both scenes, showing that the gains are not confined to a few locations.

Planning quality and acquisition cost. To examine the trade-off between target-local quality and acquisition cost, we evaluate complementary-strip selection in NYC and CitySample against five-direction scanning, On-the-fly-Adapt [7], and SCONE-Adapt [15]. On-the-fly-Adapt plans for local targets with an added two-view filter; SCONE-Adapt retains the pretrained occupancy and visibility models and coverage scoring but uses fixed geometry with greedy batch selection and route reordering. Table V reports fixed-target crop quality and planned acquisition cost.

Compared with SCONE-Adapt, our method achieves higher target-local quality with shorter planned paths in both scenes. Compared with five-direction scanning, we use 40.7% and 67.9% fewer images with only 0.73 and 0.21 dB lower crop PSNR. Complementary strips approach dense-scanning target-local quality with substantially fewer images.

Real-world reacquisition. In the Expo West field experiment, all methods are given the same image budget for

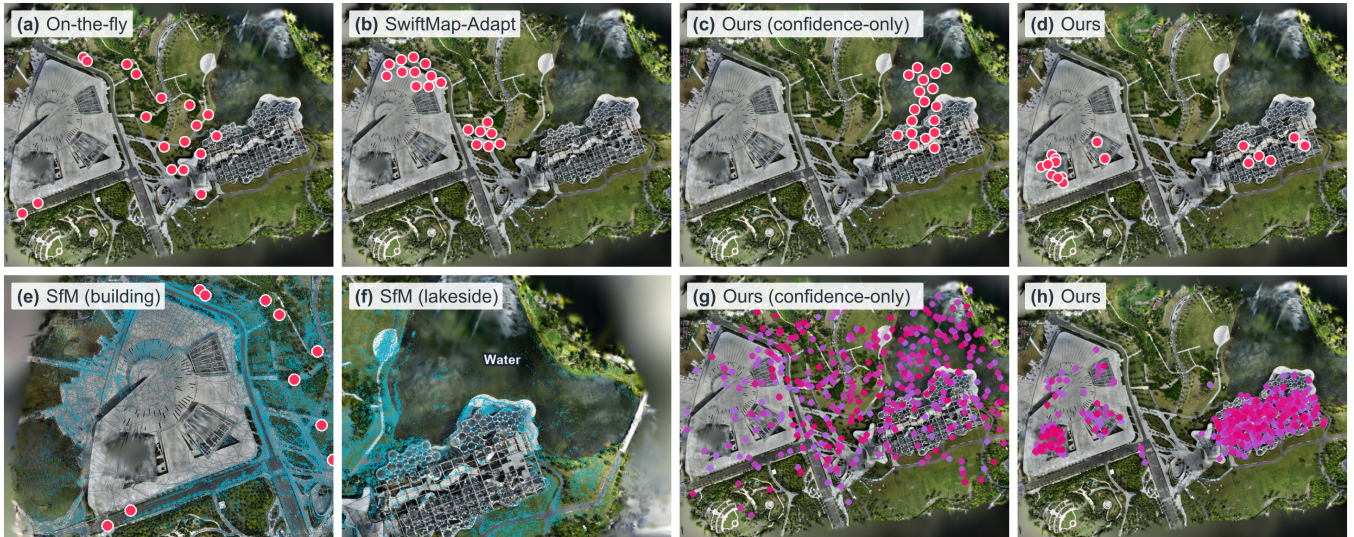


Fig. 4. Expo West: (a–d) 18 selected targets per method; (e, f) On-the-fly SfM support at the building and lakeside; (g, h) regional scores from Ours (confidence-only) and Ours (GS quality model), respectively. Pink: selected targets; cyan: SfM points; gray: mesh. Geometry overlays ignore occlusion.

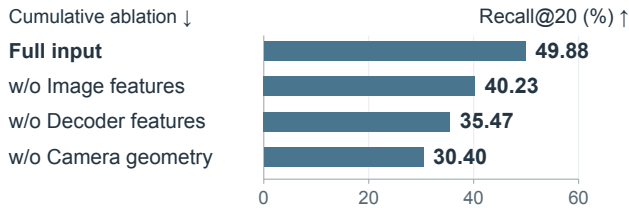


Fig. 5. Mean Recall@20 under cumulative feature ablation with fixed model parameters. Each w/o row additionally ablates the indicated group’s region-specific information.

TABLE V
TARGET-LOCAL QUALITY AND ACQUISITION COST

Scene	Method	Images	Path (km)	PSNR (dB)↑
NYC	Five-direction scan	162	1.564	25.82
	On-the-fly-Adapt	41	0.689	23.42
	SCONE-Adapt	76	1.372	22.20
	Ours	96	1.065	25.09
CitySample	Five-direction scan	414	4.007	24.82
	On-the-fly-Adapt	14	0.704	20.05
	SCONE-Adapt	112	2.676	22.29
	Ours	133	1.525	24.61

reacquisition. Each method is evaluated on initial-survey views registered in both its before and after reconstructions, together with its registered additional views. Within each paired initial-survey view set, the worst 20% are selected by uncorrected pre-reacquisition PSNR and kept fixed. Reported PSNR uses exposure and white-balance correction.

On these fixed weak-view subsets, PSNR changes by +0.57, -0.18, and -0.45 dB for Ours, SfM, and SwiftMap-Adapt, respectively. At the additional views used in joint reconstruction, all three methods improve over their pre-reacquisition models, with gains of 10.95, 4.90, and 6.35 dB, respectively. Our method improves both newly observed

areas and initially weak regions.

Fig. 6 compares the full-scene reconstructions and highlights a roof grating and a glass-roof region. Our method improves reconstruction quality in these regions, producing clearer grating edges and glass-panel boundaries.

Interactive rendering for UAV simulation. We evaluate rendering on a large campus scene containing approximately 72.71 million Gaussians. Drawing on Gaussian rendering approaches in NanoGS [31] and Spark [32], we implement hierarchical LOD selection and on-demand loading in our AirSim integration. At 1920×1080 in UE 5.5 on an RTX 4090, it renders approximately 10 million active Gaussians with full SH3 colors at an average of 74.73 FPS. LumaAI 0.4.1 Niagara [33] averages 17.40 FPS with its default visible subset and static colors.

We plan to release the OpenFlyScan system, including the mobile apps, GS quality model and trained weights, the simulation framework, and GS reconstructions of all datasets used in this work.

V. CONCLUSION

OpenFlyScan connects capture with consumer drones, timely quality feedback, and targeted reacquisition in one on-site reconstruction workflow without additional onboard hardware. Experiments show that learned quality prediction improves average Recall@20 by 26.07 percentage points over On-the-fly. Targeted reacquisition improves PSNR at additional views by 10.95 dB and also improves initially weak regions. Compact color encoding reduces reconstruction memory use, while the rendering optimizations support using the scenes for UAV simulation. Future work will improve target selection under limited acquisition budgets to reduce missed opportunities for reacquisition after quality ranking. We will explore automated collection of UAV training and evaluation data in reconstructed simulation environments.

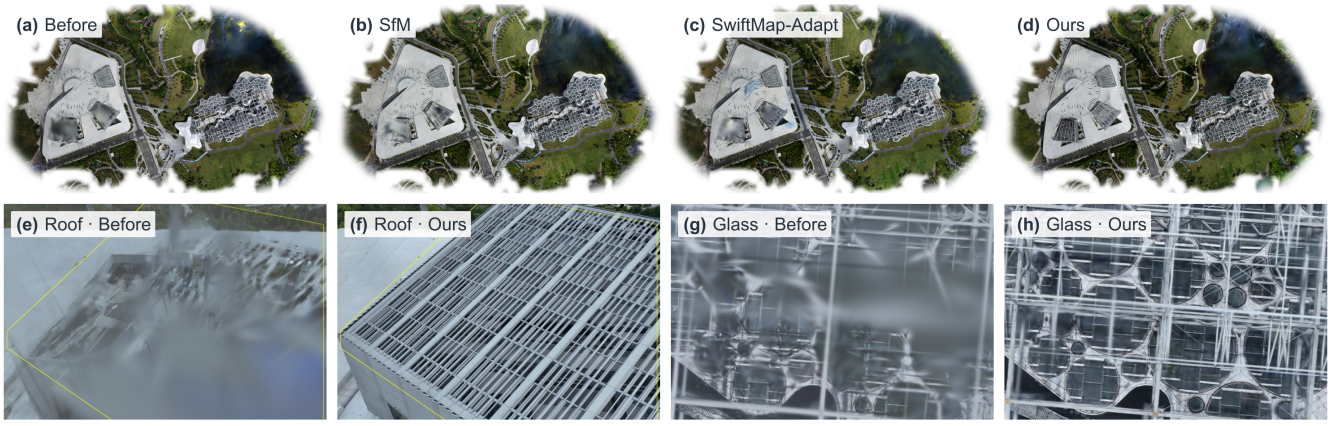


Fig. 6. Expo West before and after reacquisition: (a–d) scene overviews; (e–h) roof-grating and glass-roof details from our method.

ACKNOWLEDGMENTS

We thank Shanghai Artificial Intelligence Laboratory for providing computing resources, UAV platforms, experimental facilities, and technical support for model training and real-world deployment.

REFERENCES

- [1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3D Gaussian Splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–14, 2023, art. no. 139, doi: 10.1145/3592433.
- [2] A. Quach, M. Chahine, A. Amini, R. Hasani, and D. Rus, “Gaussian splatting to real world flight navigation transfer with liquid networks,” in *Proc. 8th Conf. Robot Learn.*, ser. Proceedings of Machine Learning Research, vol. 270, 2025, pp. 1069–1093.
- [3] Y. Gao *et al.*, “OpenFly: A comprehensive platform for aerial vision-language navigation,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026. [Online]. Available: <https://arxiv.org/abs/2502.18041>
- [4] Z. Li *et al.*, “SkyLume: A large-scale multi-illumination aerial benchmark for urban scene reconstruction and beyond,” *arXiv:2512.14200v3*, 2026.
- [5] C. Praschl, S. Schoibesberger, and D. C. Schedl, “Evaluation of novel view synthesis in the context of radiometric drone imagery,” in *Proc. 21st Int. Conf. Comput. Graph., Interact. Vis. Theory Appl. (GRIVAPP)*. SCITEPRESS, 2026, pp. 109–119, doi: 10.5220/0014460900004728.
- [6] C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof, “UAV-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2016, pp. 1–10, doi: 10.1109/CVPRW.2016.8.
- [7] L. Lou *et al.*, “On-the-fly feedback SfM: Online explore-and-exploit UAV photogrammetry with incremental mesh quality-aware indicator and predictive path planning,” *arXiv preprint arXiv:2512.02375*, 2025.
- [8] X. He *et al.*, “Detector-free structure from motion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 21 594–21 603. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024/html/He_Detector-Free_Structure_from_Motion_CVPR_2024_paper.html
- [9] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 4104–4113.
- [10] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, “VGGT: Visual geometry grounded transformer,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025, pp. 5294–5306.
- [11] Y. Wang *et al.*, “ π^3 : Permutation-equivariant visual geometry learning,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026.
- [12] B. Kerbl, A. Meuleman, G. Kopanas, M. Wimmer, A. Lanvin, and G. Drettakis, “A hierarchical 3D Gaussian representation for real-time rendering of very large datasets,” *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–15, 2024, doi: 10.1145/3658160.
- [13] Y. Liu, C. Luo, Z. Mao, J. Peng, and Z. Zhang, “CityGaussianV2: Efficient and geometrically accurate reconstruction for large-scale scenes,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.
- [14] DJI, “DJI Mobile SDK,” Official Android and iOS SDK documentation, accessed: Sep. 13, 2026. [Online]. Available: <https://github.com/dji-sdk>
- [15] A. Guédon, P. Monasse, and V. Lepetit, “SCONE: Surface coverage optimization in unknown environments by volumetric integration,” in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 20 731–20 743, doi: 10.52202/068431-1507.
- [16] A. Guédon, T. Monnier, P. Monasse, and V. Lepetit, “MACARONS: Mapping and coverage anticipation with RGB online self-supervision,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 940–951.
- [17] Y. Liu *et al.*, “Learning reconstructability for drone aerial path planning,” *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–17, 2022, doi: 10.1145/3550454.3555433.
- [18] L. Jin, X. Zhong, Y. Pan, J. Behley, C. Stachniss, and M. Popović, “ActiveGS: Active scene reconstruction using Gaussian splatting,” *IEEE Robot. Autom. Lett.*, vol. 10, no. 5, pp. 4866–4873, 2025, doi: 10.1109/LRA.2025.3555149.
- [19] W. Jiang, B. Lei, and K. Daniilidis, “FisherRF: Active view selection and mapping with radiance fields using Fisher information,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 422–440, doi: 10.1007/978-3-031-72624-8_24.
- [20] C. Shen *et al.*, “Auto3R: Automated 3D reconstruction and scanning via data-driven uncertainty quantification,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2026, accepted for publication.
- [21] X. Yang, Y. Wang, H. Li, and Y. Zhang, “UAVFF3D: A geometry-aware benchmark for feed-forward UAV 3D reconstruction,” *arXiv:2605.17942*, 2026.
- [22] X. Yang, Y. Wang, Y. Zhang, J. Li, H. Chen, and H. Li, “GeoFF3D: Coordinate-anchored feed-forward reconstruction for large-scale UAV mapping,” *arXiv:2608.28288*, 2026.
- [23] J. Xu *et al.*, “Towards fast and fully automatic drone mapping,” in *Proc. ACM Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2026, pp. 418–431, doi: 10.1145/3745756.3809213.
- [24] Pi3 Authors, “Pi3X: Engineering Update to Pi3,” Official software repository, accessed: Sep. 13, 2026. [Online]. Available: <https://github.com/yyfz/Pi3>
- [25] M. Oquab *et al.*, “DINOv2: Learning robust visual features without supervision,” *Transactions on Machine Learning Research*, 2024.
- [26] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981, doi: 10.1145/358669.358692.
- [27] D. Cernea, “OpenMVS: Open Multi-View Stereo Reconstruction Library,” Software repository, accessed: Sep. 13, 2026. [Online]. Available: <https://github.com/cdseacave/openMVS>
- [28] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics*, ser. Springer Proceedings in Advanced Robotics, vol. 5, 2018, pp. 621–635, doi: 10.1007/978-3-319-67361-5_40.

- [29] B. Xiong, N. Zheng, J. Liu, and Z. Li, “GauU-Scene V2: Assessing the reliability of image-based metrics with expansive LiDAR image dataset using 3DGS and NeRF,” arXiv:2404.04880, 2024.
- [30] L. Lin, Y. Liu, Y. Hu, X. Yan, K. Xie, and H. Huang, “Capturing, reconstructing, and simulating: The UrbanScene3D dataset,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 93–109, doi: 10.1007/978-3-031-20074-8_6.
- [31] TimChen1383, “NanoGaussianSplatting: Nanite-Style Gaussian Splatting Renderer for Unreal Engine,” Software repository (MIT), accessed: Sep. 13, 2026. [Online]. Available: <https://github.com/TimChen1383/NanoGaussianSplatting>
- [32] SparkJS Contributors, “Spark LoD Rendering / Level-of-Detail Deep Dive,” Software documentation, accessed: Sep. 13, 2026. [Online]. Available: <https://sparkjs.dev/docs/new-spark-renderer/>
- [33] Luma AI, “Luma Unreal Engine Plugin (0.41),” Official plugin documentation, accessed: Sep. 13, 2026. [Online]. Available: <https://lumaai.notion.site/Luma-Unreal-Engine-Plugin-0-4-8005919d93444c008982346185e933a1>