PANOLOG: GEOMETRY AND GRADIENT-BASED PARTITIONING FOR PANORAMIC OUTDOOR RECONSTRUCTION

Anonymous authors
Paper under double-blind review

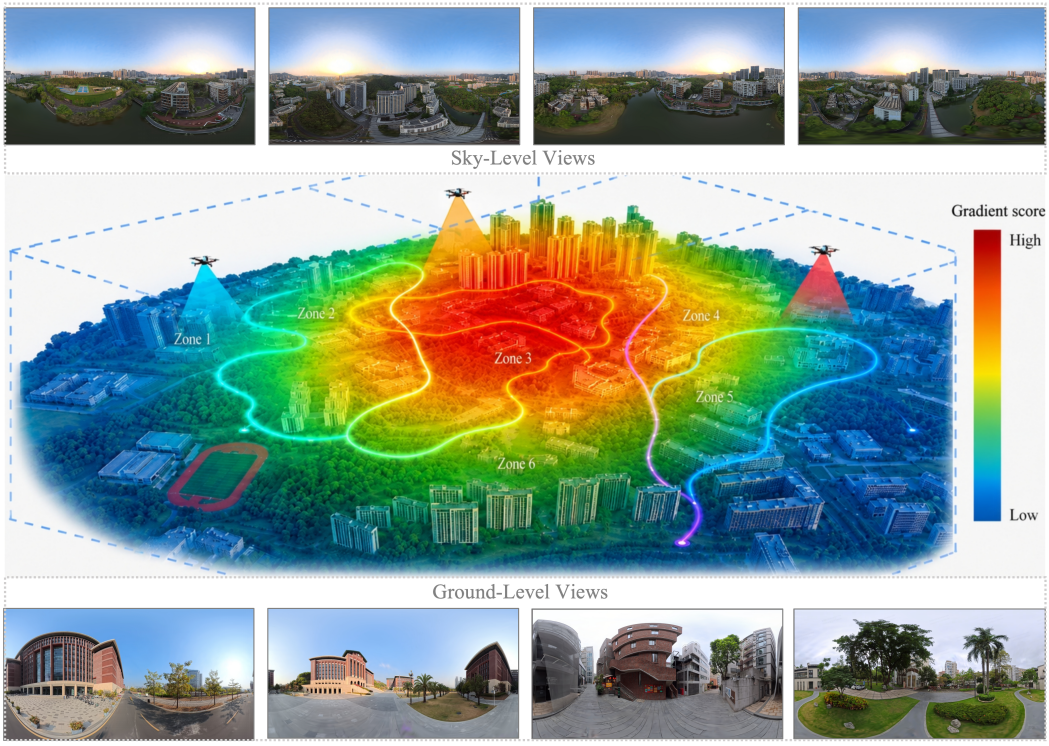

Figure 1: Overview of PanoLOG. Top and bottom: sky-level and ground-level 360° ERP panoramas from Pano360. Middle: conceptual illustration of G^{2}PS, a Geometry and Gradient-based Partitioning Strategy, where colors encode the gradient score (red: high; blue: low) used to allocate cameras to spatial blocks (Zones 1–6).

ABSTRACT

Scaling 3D Gaussian Splatting (3DGS) to large outdoor scenes is costly in both data acquisition and computation. Adopting panoramic images with equirectangular projection (ERP) can reduce capture effort via their full 360° field of view, yet the resulting omnipresent visibility invalidates existing partitioning strategies that rely on local camera frustums, causing block-wise optimization to degenerate into global training. Thus, we propose PanoLOG (**Pano**ramic **L**arge-scale **O**utdoor **G**aussian Splatting), a two-stage coarse-to-fine framework equipped with a Geometry and Gradient-based Partitioning Strategy (G^{2}PS) tailored for large-scale panoramic 3DGS reconstruction. In the global coarse stage, PanoLOG leverages sky-sphere modeling and panoramic monocular depth supervision for reliable geometry, while in the refinement stage, G^{2}PS builds adaptive bounding volumes via parallax-driven uncertainty and assigns cameras via gradient-based importance scoring. Furthermore, we construct Pano360, the first large-scale panoramic bench-

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.

mark for outdoor scene reconstruction. Extensive experiments demonstrate that PanoLOG achieves state-of-the-art rendering quality while maintaining scalable, block-parallel training. Our models, training code, and dataset will be publicly available.

1 INTRODUCTION

Novel View Synthesis (Chung et al., 2025; Wu et al., 2025; 2024) has emerged as a cornerstone of computer vision, underpinning applications from autonomous driving simulation to virtual reality. Recently, 3D Gaussian Splatting (3DGS) has become the mainstream paradigm for photorealistic reconstruction, leveraging anisotropic explicit Gaussian primitives and an efficient differentiable rasterization pipeline to achieve unprecedented rendering speed.

Despite its success, scaling 3DGS to expansive outdoor environments remains challenging due to the prohibitive costs of data acquisition and computational overhead. Conventional pipelines typically rely on capturing a massive volume of narrow field-of-view (FoV) images, subsequently adopting a divide-and-conquer strategy to partition the scene. However, such workflows are not only labor-intensive but also prone to structural inconsistencies across stitching boundaries, limiting their efficiency for large-scale deployment.

Inspired by omnidirectional vision, adopting a panoramic ERP representation offers a compelling alternative by capturing a full 360° field of view in a single shot, thereby streamlining data collection. Yet, this shift moves the complexity from acquisition to spatial partitioning. Traditional partitioning (Kerbl et al., 2024; Liu et al., 2024; Lin et al., 2024) relies on the local visibility of pinhole cameras, which is at odds with the 360° nature of panoramic vision. Without frustum constraints, the omnipresent visibility in panoramic scenes strips these strategies of their discriminative power, causing intended block-wise partitioning to degenerate back into inefficient global optimization. Thus, a question arises: can we design a partition strategy tailored to panoramic images?

To address this issue, we propose PanoLOG, a two-stage coarse-to-fine training pipeline alongside a Geometry and Gradient-based Partitioning Strategy (G^2PS; Figure 1). For geometry-based partitioning, we estimate a relative parallax distance using inter-camera baselines and perform depth uncertainty analysis, enabling adaptive expansion of spatial boundaries to convert unbounded scenes into stable reconstruction regions. For camera–block allocation, we leverage gradient signals from the coarse stage to evaluate each camera's actual contribution to different spatial blocks, enabling gradient-based importance assignment that effectively resolves the camera–block allocation problem in panoramic scenes. In this way, we obtain structurally sound spatial partitioning, enabling each sub-block to focus on its key observation regions, thereby enhancing the quality of block optimization and preventing global degeneracy.

To well evaluate our performance, we build a comprehensive benchmark that collects outdoor images across four panoramic scenes (campus and park via Antigravity A1; two street scenes via Insta360 X5). Covering over 2 million m^2 with 5,637 images (3840×1920), we construct a dataset named **Pano360**, which provides calibrated poses and sparse point clouds for reproducible evaluation.

Our contributions are threefold.

- We present PanoLOG, a two-stage coarse-to-fine 3DGS framework for large-scale outdoor panoramic reconstruction. As its name suggests, PanoLOG unifies four defining elements: **Pano**ramic image, **L**arge-scale scenes, **O**utdoor environments, and **G**aussian Splatting.
- We propose G^2PS, a Geometry and Gradient-based Partitioning Strategy that bounds unbounded scenes via parallax-driven depth uncertainty analysis and allocates cameras to blocks via gradient-based scoring, preventing block-wise optimization from degenerating into global optimization.
- We contribute Pano360, the first large-scale panoramic dataset and benchmark for outdoor scene reconstruction. Experiments on Pano360 demonstrate the effectiveness of PanoLOG.

2 Related Work

2.1 Omnidirectional 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has established a new paradigm for real-time novel view synthesis, but its core formulation assumes a perspective camera model and is thus susceptible to the severe angular distortions of equirectangular projection (ERP). Recent research therefore adapts the 3DGS pipeline to omnidirectional images. Early efforts such as ODGS (Lee et al., 2024) project Gaussians onto local tangent planes of the unit sphere. Subsequent methods, including OmniGS (Li et al., 2025b) and SC-OmniGS (Huang et al., 2025a), derive closed-form derivatives for direct optimization in ERP space without intermediate rectification, with the latter further co-optimizing camera poses. More recently, 3DGEER (Huang et al., 2026) formulates an exact ray–Gaussian density integral applicable to arbitrary camera models.

Parallel efforts extend panoramic 3DGS to synthesis (Chen et al., 2025b; Zhang et al., 2025; Lee et al., 2025; Wang et al., 2026) and generation (Zhou et al., 2024; Ma et al., 2024; Huang et al., 2025b). These methods primarily target localized scenes and do not address the scalability required by expansive outdoor environments with unbounded skies and low-texture regions.

2.2 Large-Scale Scene Reconstruction

Scaling 3DGS to expansive environments introduces compounding hardware and optimization bottlenecks. To overcome GPU memory limits, the community has largely adopted divide-and-conquer strategies that leverage the inherent spatial constraints of pinhole cameras. VastGaussian (Lin et al., 2024) partitions the scene into cells using an airspace-aware visibility criterion, while CityGaussian (Liu et al., 2024) assigns cameras to spatial blocks using camera frustums and rendering similarity. DOGS (Chen & Lee, 2024) accelerates training via distributed ADMM-based optimization, achieving over 6× training speedup. Other works enhance scalability through hierarchical level-of-detail (LoD) structures (Kerbl et al., 2024; Kulhanek et al., 2025) or geometric constraints (Liu et al., 2025; Chen et al., 2025a) for smoother large-scale navigation, and Horizon-GS (Jiang et al., 2025) unifies aerial and street-view reconstruction via a coarse-to-fine training strategy. However, the omnipresent visibility of panoramic images renders such frustum-based partitioning non-discriminative.

3 Method

We present PanoLOG, a two-stage coarse-to-fine framework for large-scale panoramic 3DGS reconstruction (Figure 2). We first introduce the base structure and loss design for each block (Section 3.1), then detail the two-stage training pipeline (Section 3.2): global coarse training (Stage I) followed by block-wise refinement (Stage II).

3.1 Base Structure

Each spatial block shares a unified base structure: a panoramic 3DGS renderer with ERP projection, an explicit sky sphere, and panoramic depth supervision, all optimized under a common loss. We first describe the rendering formulation, then detail each auxiliary component.

Panoramic 3DGS. Inspired by Kerbl et al. (2023), we represent a scene with N anisotropic 3D Gaussians $\mathcal{G} = \{g_i\}_{i=1}^{N}$, each parameterized by a mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$, covariance $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$, color $\mathbf{c}_i \in \mathbb{R}^3$, and opacity $o_i \in (0, 1]$. Pixel color $\hat{C}(\mathbf{u})$ is computed via front-to-back alpha compositing:

$$\hat{C}(\mathbf{u}) = \sum_{i\in\mathcal{N}} \mathbf{c}_i \, \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \quad \alpha_i = o_i \cdot G_{2\mathrm{D},i}(\mathbf{u}), \tag{1}$$

where $G_{2\mathrm{D},i}(\mathbf{u})$ is the projected 2D Gaussian at pixel $\mathbf{u}$.

To handle the full 360° field of view, we adopt equirectangular projection (ERP). For a 3D point $\mathbf{X} = (x, y, z)^\top$ in camera coordinates, its spherical coordinates $(\theta, \phi) = (\mathrm{atan2}(x, z),\ \arcsin(y/\|\mathbf{X}\|))$ are mapped to pixel coordinates $\mathbf{u} = \boldsymbol{\pi}(\mathbf{X})$. The 2D splatting covariance is obtained by projecting

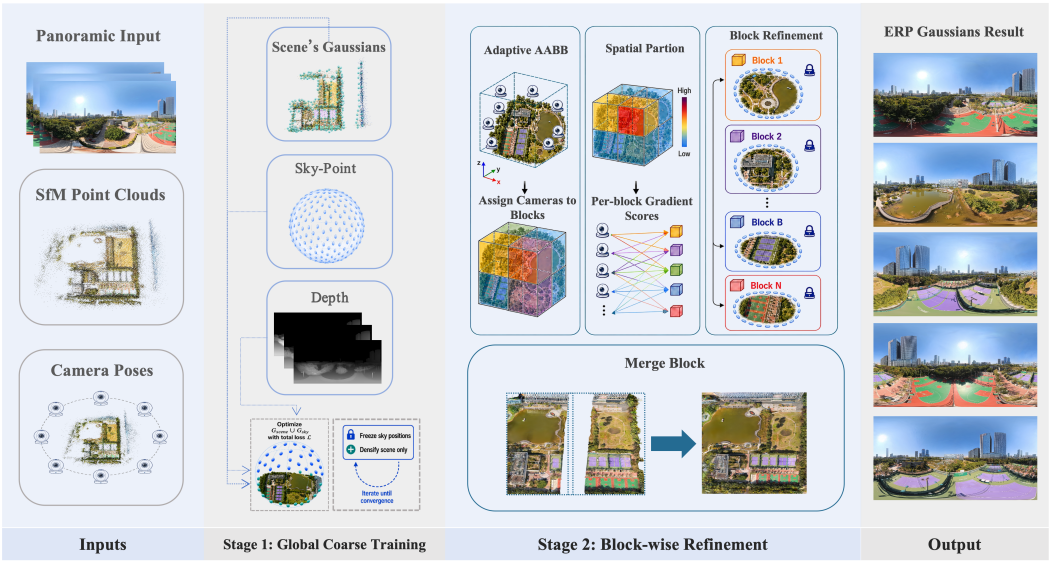

Figure 2: Overview of the PanoLOG training pipeline. **Stage I** jointly optimizes scene and explicit sky Gaussians under panoramic depth supervision to establish stable coarse geometry. **Stage II** applies G^{2}PS for geometry-based spatial partitioning and gradient-based camera–block allocation; each block is then refined independently with frozen sky Gaussians (lock icons) and merged into the final model.

the 3D covariance through the ERP Jacobian $\mathbf{J} = \partial\boldsymbol{\pi}/\partial\mathbf{X}$:

$$\boldsymbol{\Sigma}_{2\mathrm{D}} = \mathbf{J}\,\mathbf{W}\,\boldsymbol{\Sigma}\,\mathbf{W}^{\top}\mathbf{J}^{\top}, \tag{2}$$

where $\mathbf{W}$ is the world-to-camera rotation and $\boldsymbol{\Sigma} = \mathbf{R}\mathbf{S}\mathbf{S}^{\top}\mathbf{R}^{\top}$. Unlike the pinhole pipeline, $\boldsymbol{\Sigma}_{2\mathrm{D}}$ is computed via the nonlinear ERP Jacobian rather than a perspective projection matrix. The full derivation is provided in Section B.

Auxiliary Design. Our base representation comprises *explicit sky Gaussians* for modeling unbounded sky regions and *standard 3D Gaussians* for near-field scene geometry, complemented by panoramic monocular depth priors and a unified optimization objective. During block-wise refinement (Stage II), each block inherits the sky Gaussians trained in Stage I as a frozen global model, while the scene Gaussians are loaded as initialization and further optimized locally.

Explicit Sky Sphere. Sky regions lack valid SfM geometry, causing near-field Gaussians to drift outward and produce floater artifacts. We initialize N_{sky} dedicated sky Gaussians on a distant sphere of radius $R_{\mathrm{sky}} = \kappa \cdot r_{\mathrm{scene}}$ ($\kappa = 10$), where $r_{\mathrm{scene}} = \max_i \|\mathbf{x}_i - \bar{\mathbf{C}}\|$ is the maximum extent of the SfM point cloud from the camera centroid $\bar{\mathbf{C}}$. Positions are uniformly sampled on the sphere:

$$\mathbf{p}_i^{\mathrm{sky}} = \bar{\mathbf{C}} + R_{\mathrm{sky}} \begin{pmatrix} \cos\theta_i \sin\phi_i \\ \sin\theta_i \sin\phi_i \\ \cos\phi_i \end{pmatrix}. \tag{3}$$

Panoramic Depth Supervision. ERP polar regions suffer from severe pixel stretching that degrades SfM triangulation, leaving the initial point cloud sparse and unreliable. We employ DAP (Lin et al., 2026), a depth estimator natively supporting ERP input, to generate monocular inverse depth maps aligned to SfM sparse depth via per-image affine parameters:

$$\tilde{D}_k^{-1} = s_k \cdot D_{\mathrm{mono},k}^{-1} + o_k, \tag{4}$$

where we operate in *radial* depth ($\|\mathbf{X}\|$) rather than z-axis depth, as it is geometrically natural for omnidirectional cameras. The depth loss is:

$$\mathcal{L}_{\mathrm{depth}} = w(t) \cdot \frac{1}{|\Omega|} \sum_{\mathbf{u}\in\Omega} \left| \hat{D}^{-1}(\mathbf{u}) - \tilde{D}_{\mathrm{mono}}^{-1}(\mathbf{u}) \right|, \tag{5}$$

where Ω is the set of valid pixels and $w(t)$ is an exponentially decaying weight that provides strong geometric guidance early while avoiding depth estimation errors from limiting final quality (schedule in Section C).

Unified Optimization Objective. The overall loss combines photometric and depth supervision:

$$\mathcal{L} = (1 - \lambda_{\text{ssim}})\,\mathcal{L}_1 + \lambda_{\text{ssim}}\,\mathcal{L}_{\text{D-SSIM}} + \mathcal{L}_{\text{depth}}, \tag{6}$$

where $\mathcal{L}_1$ and $\mathcal{L}_{\text{D-SSIM}}$ are the ℓ_1 photometric loss and structural dissimilarity loss. This objective is shared across both training stages.

3.2 Training Pipeline

With the base representation defined above, we describe the two-stage procedure. Stage I performs global coarse optimization to establish reliable geometry for partitioning; Stage II refines each spatial block independently to recover fine-grained detail.

3.2.1 Stage I: Global Coarse Training

Directly partitioning raw SfM point clouds leads to floaters in out-of-block regions and unreliable camera–block allocation due to low-fidelity renderings. We therefore first perform a global coarse optimization over all input panoramas to generate stable geometric priors. Unlike prior frameworks (Liu et al., 2024) that rely solely on photometric supervision, we incorporate panoramic monocular depth priors (Section 3.1) from the outset, providing robust geometric anchoring critical for the subsequent gradient-based partitioning (Section 3.2.2).

During this stage, sky Gaussian position gradients are zeroed to prevent drift into the near field, while their appearance parameters (color, rotation, opacity) are optimized normally; sky Gaussians are also excluded from all densification operations.

3.2.2 Stage II: Block-wise Refinement

Geometry and Gradient-based Partitioning Strategy (G^{2}PS). Equirectangular projection fundamentally violates the spatial visibility locality assumed by existing large-scale reconstruction frameworks: whereas conventional pinhole setups allow effective frustum-based partitioning because each block is observed by only a subset of cameras, the 360° observation range of panoramic scenes couples all spatial regions across viewpoints, rendering such strategies indiscriminate and degenerating into global optimization. G^{2}PS addresses this through two complementary mechanisms: *geometry-based spatial partitioning* that constructs a balanced block decomposition, and *gradient-based camera–block allocation* that assigns viewpoints by their actual observational contributions.

Geometry-based Spatial Partitioning. G^{2}PS constructs an adaptive axis-aligned bounding box (AABB) to define the effective reconstruction volume. Directly applying uniform grid partitioning in unbounded 3D space produces many nearly empty blocks with severe workload imbalance; we therefore contract the scene into a bounded AABB whose extent is determined by camera distribution and triangulation reliability.

We first compute the camera centroid $\bar{\mathbf{C}} = \frac{1}{K}\sum_k \mathbf{C}_k$ and the per-axis base radius $r_d = \max_k |\mathbf{C}_k^{(d)} - \bar{\mathbf{C}}^{(d)}|$ for $d \in \{x, y, z\}$, which encloses the camera trajectory. Since panoramic scene content extends well beyond the cameras, we expand the boundary according to the effective range of spherical feature triangulation. For ERP panoramas, matched features are lifted to unit bearing rays, and triangulation reliability is governed by the angular parallax γ. Under the small-parallax approximation, depth satisfies $z \sim b/\gamma$; by first-order error propagation, the relative depth uncertainty is $\sigma_z/z \approx (z/b)\,\sigma_\gamma$. Bounding this by a tolerance η and defining the triangulation range factor $\rho_{\text{tri}} = \eta/\sigma_\gamma$, we obtain $z_{\max} \approx b\,\rho_{\text{tri}}$.

The expansion margin is defined as:

$$\text{margin} = \hat{b} \cdot \rho_{\text{tri}}, \tag{7}$$

with the representative baseline $\hat{b}$ robustly estimated as the median nearest-neighbor camera distance:

$$\hat{b} = \text{median}_k \left(\min_{j \neq k} \|\mathbf{C}_k - \mathbf{C}_j\| \right). \tag{8}$$

The final AABB bounds are:

$$\mathbf{a}_{\min}^{(d)} = \bar{\mathbf{C}}^{(d)} - r_d - \text{margin}, \quad \mathbf{a}_{\max}^{(d)} = \bar{\mathbf{C}}^{(d)} + r_d + \text{margin}. \tag{9}$$

Gaussians extending beyond the AABB are contracted into the bounded region as prescribed by Liu et al. (2024). The contracted cubic space is then uniformly partitioned into spatial blocks, yielding a balanced distribution of Gaussians.

Gradient-based Camera–Block Allocation. In the refinement stage, each block must receive sufficient training supervision. We observe that the rendering loss gradient with respect to Gaussian positions naturally reflects their importance to the current viewpoint: regions with large contributions exhibit large gradients, while distant or occluded regions have near-zero gradients. After Stage I converges, we perform a single forward-backward pass over all training views to collect these gradients. For camera k, we compute the mean gradient magnitude over all Gaussians $\mathcal{G}_b$ within block b:

$$s_{k,b} = \frac{1}{|\mathcal{G}_b|} \sum_{g \in \mathcal{G}_b} \left\| \frac{\partial \mathcal{L}_k}{\partial \mathbf{x}_g} \right\|_1, \tag{10}$$

where $\mathcal{L}_k$ denotes the rendering loss for camera k and $\mathbf{x}_g$ is the position of Gaussian g. The camera–block assignment combines geometric membership with gradient-based importance:

$$\text{assign}(k,b) = \underbrace{\mathbb{1}[\mathbf{C}_k \in \text{Block}_b]}_{\text{geometric}} \vee \underbrace{\mathbb{1}\left[\frac{s_{k,b}}{\max_{b'} s_{k,b'}} > \tau_{\text{grad}}\right]}_{\text{gradient-based}}, \tag{11}$$

where $\tau_{\text{grad}} = 0.8$ is the gradient ratio threshold. A camera is assigned to a block if it is either geometrically located within that block or its normalized gradient score exceeds the threshold, ensuring that viewpoints with significant observational contributions are included even when physically located in adjacent blocks.

Block-wise Optimization. Each block is initialized with the *full* set of Gaussians from Stage I and optimized independently using its assigned cameras. Sky Gaussians are fully frozen during this stage to maintain cross-block appearance consistency.

To adaptively determine each block's effective spatial extent, we leverage periodic opacity reset. After each reset, all scene Gaussian opacities are clamped to a low value o_{reset}; only Gaussians receiving sufficient photometric gradients from the block's assigned cameras recover high opacity, while the rest remain transparent and are pruned. The retained Gaussian set for block b is:

$$\tilde{\mathcal{G}}_b = \left\{ g \in \mathcal{G}_b \;\middle|\; \mathbf{x}_g^{\text{ctr}} \in [\mathbf{l}_b, \mathbf{u}_b) \;\wedge\; o_g^{(b)} > o_{\min} \right\}, \tag{12}$$

where $\mathcal{G}_b$ is the Gaussian set after block b's optimization, $\mathbf{x}_g^{\text{ctr}}$ is the center of Gaussian g in the contracted coordinate space, $[\mathbf{l}_b, \mathbf{u}_b)$ is the half-open axis-aligned cell of block b, $o_g^{(b)}$ is the converged opacity, and $o_{\min}$ is the pruning threshold. The half-open interval convention ensures each retained Gaussian is assigned to exactly one block, avoiding duplicate primitives in the final model. The merged model is:

$$\mathcal{G}_{\text{final}} = \mathcal{G}_{\text{sky}} \cup \bigcup_{b=1}^{B} \tilde{\mathcal{G}}_b. \tag{13}$$

Because adjacent blocks share overlapping camera sets (via Equation (11)), Gaussians near block boundaries receive consistent cross-boundary supervision, producing coherent geometry and appearance at block interfaces without explicit stitching.

4 Experiments

4.1 Experimental Setup

Datasets. We introduce **Pano360**, the first large-scale outdoor panoramic reconstruction dataset, comprising 5,637 high-resolution (3840×1920) images captured across four diverse urban environments. Pano360 consists of four subsets—NSC, NSK, BAX, and NSN—where NSC and NSK are

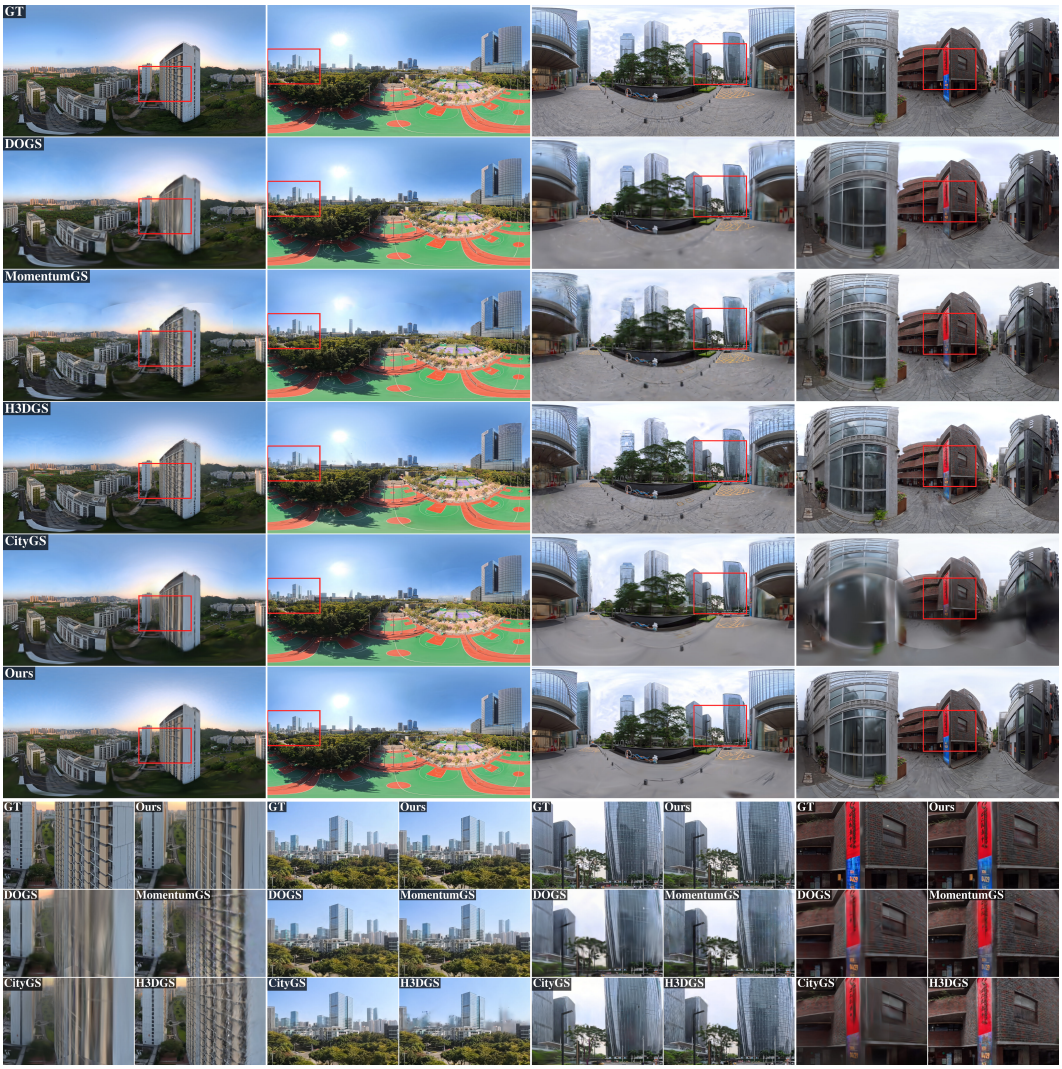

Figure 3: Qualitative comparison on Pano360. Columns: different Pano360 scenes; rows: GT, DOGS, Momentum-GS, H3DGS, CityGaussian (CityGS), and Ours; red boxes mark the regions magnified in the bottom panels. PanoLOG better recovers distant regions and complex glass facades, which are challenging in urban environments.

collected by the A1 drone, while BAX and NSN are acquired with the X5 handheld camera. The dataset provides calibrated poses and sparse point clouds with a 7:1 train/test split. To demonstrate generalizability, we further evaluate on the public Ricoh360 (Choi et al., 2023) and 360Roam (Huang et al., 2022) benchmarks. Additional dataset details (coverage, cropping protocol, and anonymization) are provided in Section E.

Baselines and Metrics. We compare against large-scale 3DGS methods—H3DGS (Kerbl et al., 2024), CityGaussian (Liu et al., 2024), DOGS (Chen & Lee, 2024), and Momentum-GS (Fan et al., 2025)—by converting ERP images into six cubemap views with pinhole camera models. On public datasets, we additionally compare with panoramic 3DGS methods: OmniGS (Li et al., 2025b), ODGS (Lee et al., 2024), SpaGS (Li et al., 2025a), and a cubemap-input 3DGS baseline. Metrics include PSNR, SSIM (Wang et al., 2004), LPIPS-VGG (Zhang et al., 2018), and model size.

Implementation Details. All methods use official implementations with recommended hyperparameters, are trained for 30,000 iterations, and use two partitions by default. All experiments run

Table 1: Quantitative comparison on the A1 drone subsets (NSC, NSK) of Pano360 (Size in MB). **Bold red**: best result; **bold orange**: largest model. PanoLOG is best on five of the six quality metrics with the smallest model on both scenes.

	NSC				NSK			
Method	PSNR↑	SSIM↑	LPIPS↓	Size↓	PSNR↑	SSIM↑	LPIPS↓	Size↓
H3DGS (Kerbl et al., 2024)	27.7787	0.8564	0.2457	1002.1	24.1480	0.8154	0.1934	**1843.2**
CityGaussian (Liu et al., 2024)	27.7340	0.8453	0.2609	523.7	**24.8270**	0.8176	0.1983	1126.4
DOGS (Chen & Lee, 2024)	26.8486	0.8186	0.2769	**1024.0**	24.4606	0.7980	0.2174	1536.0
Momentum-GS (Fan et al., 2025)	26.4568	0.8311	0.2625	802.5	23.9123	0.7979	0.1986	1024.0
Ours	**28.1838**	**0.8594**	**0.2435**	463.5	24.65	**0.827**	**0.186**	667

Table 2: Quantitative comparison on the X5 handheld subsets (BAX, NSN) of Pano360 (Size in MB; highlighting as in Table 1). For all methods, the bottom $\sim 10\%$ of each ERP image is cropped before evaluation (Section E); H3DGS uses eight partitions due to GPU memory limitations. PanoLOG attains the best PSNR on both scenes; H3DGS obtains better LPIPS (and SSIM on NSN) with a 5.7–$7.5\times$ larger model.

	BAX				NSN			
Method	PSNR↑	SSIM↑	LPIPS↓	Size↓	PSNR↑	SSIM↑	LPIPS↓	Size↓
H3DGS (Kerbl et al., 2024)	20.7104	0.6626	**0.3936**	**7577.6**	23.4464	**0.7530**	**0.3018**	**5734.4**
CityGaussian (Liu et al., 2024)	19.8361	0.5995	0.4902	418.5	21.9492	0.6729	0.4221	526.6
DOGS (Chen & Lee, 2024)	19.6094	0.5745	0.5203	671.8	22.6743	0.6587	0.4454	774.9
Momentum-GS (Fan et al., 2025)	20.0993	0.5999	0.4896	1024.0	22.9746	0.7052	0.3703	1331.2
Ours	**21.3468**	**0.6656**	0.4169	1331.2	**24.6095**	0.7508	0.3347	766.8

Table 3: Quantitative comparison on the public Ricoh360 and 360Roam benchmarks, averaged over all scenes. 3DGS uses cubemap inputs; **bold red**: best. PanoLOG is best on all metrics.

	Ricoh360			360Roam		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
OmniGS (Li et al., 2025b)	26.00	0.828	0.210	24.93	0.800	0.253
3DGS (Kerbl et al., 2023)	26.26	0.825	0.225	24.98	0.800	0.271
ODGS (Lee et al., 2024)	22.71	0.748	0.326	19.74	0.627	0.476
SpaGS (Li et al., 2025a)	26.11	0.832	0.243	25.45	0.814	0.223
Ours	**26.48**	**0.845**	**0.183**	**25.83**	**0.821**	**0.222**

on a single NVIDIA RTX 4090 GPU (24 GB). Due to GPU memory limitations, H3DGS uses eight partitions on the BAX and NSN datasets. Full training configurations are provided in Section A.

4.2 Benchmarking Results

Results on Pano360. Table 1 and Table 2 report quantitative results on the A1 drone and X5 handheld subsets, respectively. PanoLOG achieves the best PSNR on three of four subsets (NSC, BAX, NSN) and the best SSIM and LPIPS on NSK. On the X5 scenes, PanoLOG outperforms the strongest baseline by 0.64 dB on BAX and 1.16 dB on NSN, while our models are 5.7–$7.5\times$ smaller than the strongest baseline. Figure 3 shows qualitative comparisons, with more results in Section G.1.

Results on Public Panoramic Datasets. As shown in Table 3, PanoLOG achieves the best PSNR, SSIM, and LPIPS on both Ricoh360 and 360Roam, outperforming all panoramic baselines (OmniGS, ODGS, SpaGS) and the cubemap 3DGS baseline across all metrics. This demonstrates that the panoramic-native design of PanoLOG generalizes beyond our own benchmark. Qualitative comparisons are provided in Section G.2.

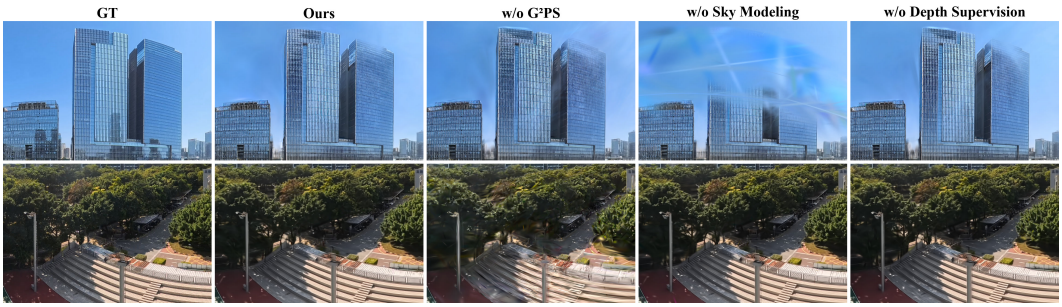

Figure 4: Qualitative ablation on two views from the NSK scene of Pano360. Left to right: GT, the full model (Ours), and variants replacing G^{2}PS with position-only partitioning, removing the explicit sky sphere (w/o Sky Modeling), or removing depth supervision. These components improve unbounded sky, glass facades, and dense vegetation.

Table 4: Component ablation on NSK. w/o G^{2}PS: position-only partitioning; Sup.: supervision.

Setting	PSNR↑	SSIM↑	LPIPS↓	Size↓
w/o G^{2}PS	24.14	0.816	0.194	**580M**
w/o Sky Sphere	24.15	0.822	0.195	683M
w/o Depth Sup.	24.61	0.824	0.191	588M
Full Model	**24.65**	**0.827**	**0.186**	667M

Table 5: Effect of τ_{grad} on NSK (Size in MB). The default $\tau_{\text{grad}} = 0.8$ gives the best quality at a moderate size.

Setting	PSNR↑	SSIM↑	LPIPS↓	Size↓
$\tau_{\text{grad}} = 0.7$	24.61	0.822	0.192	699M
$\tau_{\text{grad}} = 0.8$	**24.65**	**0.827**	**0.186**	667M
$\tau_{\text{grad}} = 0.9$	24.42	0.823	0.190	**654M**

4.3 Ablation Study

We ablate three key components: (1) replacing G^{2}PS with position-only partitioning, (2) removing the explicit sky sphere, and (3) removing the depth supervision. Results are in Table 4 and Figure 4. Replacing G^{2}PS reduces PSNR by 0.51 dB, confirming gradient-based allocation is essential. Without the sky sphere, floaters proliferate due to unconstrained Gaussian drift. Removing depth weakens geometric supervision in polar and distant areas.

4.4 Analysis of Partition Threshold

We vary $\tau_{\text{grad}} \in \{0.7, 0.8, 0.9\}$ in the camera–block allocation (Equation (11)). As shown in Table 5, the default $\tau_{\text{grad}} = 0.8$ achieves the best trade-off. Further analysis is provided in Section F.

5 Conclusion

We presented PanoLOG, a two-stage coarse-to-fine framework for large-scale outdoor panoramic 3DGS reconstruction. The central contribution is G^{2}PS, which resolves the omnipresent visibility problem inherent to panoramic scenes through geometry-based spatial partitioning and gradient-based camera–block allocation. Complementary components—explicit sky sphere modeling and panoramic monocular depth supervision—further stabilize reconstruction in unbounded and geometrically challenging regions. We also introduced Pano360, the first large-scale panoramic outdoor benchmark, comprising 5,637 high-resolution images across four diverse urban environments. Experiments on Pano360 and public benchmarks show that PanoLOG achieves state-of-the-art rendering quality while producing models 2.2–$7.5\times$ smaller than H3DGS. These results confirm that panoramic-aware partitioning is essential for scalable, high-fidelity reconstruction from omnidirectional images.

AI Use Statement

We used generative AI tools only to aid and polish the writing and to check and proofread the paper formatting. Generative AI tools were not used for any other part of this research. All AI-assisted content was reviewed by the authors, who take full responsibility for the final content of this work.

Ethics Statement

Our ethics statement, including privacy protection for Pano360, is provided in Section H.

References

Junyi Chen, Weicai Ye, Yifan Wang, Danpeng Chen, Di Huang, Wanli Ouyang, Guofeng Zhang, Yu Qiao, and Tong He. Gigags: Scaling up planar-based 3d gaussians for large scene surface reconstruction. In *AAAI*, 2025a. 3

Yu Chen and Gim Hee Lee. Dogs: Distributed-oriented gaussian splatting for large-scale 3d reconstruction via gaussian consensus. In *NeurIPS*, 2024. 3, 7, 8

Zheng Chen, Chenming Wu, Zhelun Shen, Chen Zhao, Weicai Ye, Haocheng Feng, Errui Ding, and Song-Hai Zhang. Splatter-360: Generalizable 360 gaussian splatting for wide-baseline panoramic images. In *CVPR*, 2025b. 3

Changwoon Choi, Sang Min Kim, and Young Min Kim. Balanced spherical grid for egocentric view synthesis. In *CVPR*, 2023. 7

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 2

Jixuan Fan, Wanhua Li, Yifei Han, Tianru Dai, and Yansong Tang. Momentum-gs: Momentum gaussian self-distillation for high-quality large scene reconstruction. In *ICCV*, 2025. 7, 8

Huajian Huang, Yingshu Chen, Tianjian Zhang, and Sai-Kit Yeung. 360roam: Real-time indoor roaming using geometry-aware 360◦ radiance fields. In *arXiv preprint arXiv:2208.02705*, 2022. 7

Huajian Huang, Yingshu Chen, Longwei Li, Hui Cheng, Tristan Braud, Yajie Zhao, and Sai-Kit Yeung. Sc-omnigs: Self-calibrating omnidirectional gaussian splatting. In *ICLR*, 2025a. 3

Zilong Huang, Jun He, Junyan Ye, Lihan Jiang, Weijia Li, Yiping Chen, and Ting Han. Scene4u: Hierarchical layered 3d scene reconstruction from single panoramic image for your immerse exploration. In *CVPR*, 2025b. 3

Zixun Huang, Cho-Ying Wu, Yuliang Guo, Xinyu Huang, and Liu Ren. 3dgeer: 3d gaussian rendering made exact and efficient for generic cameras. In *ICLR*, 2026. 3

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 *CVPR*, 2025. 3

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023. 3, 8

Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 2024. 2, 3, 7, 8

Jonas Kulhanek, Marie-Julie Rakotosaona, Fabian Manhardt, Christina Tsalicoglou, Michael Niemeyer, Torsten Sattler, Songyou Peng, and Federico Tombari. Lodge: Level-of-detail large-scale gaussian splatting with efficient rendering. In *NeurIPS*, 2025. 3

Suyoung Lee, Jaeyoung Chung, Jaeyoo Huh, and Kyoung Mu Lee. Odgs: 3d scene reconstruction from omnidirectional images with 3d gaussian splattings. In *NeurIPS*, 2024. 3, 7, 8

Suyoung Lee, Jaeyoung Chung, Kihoon Kim, Jaeyoo Huh, Gunhee Lee, Minsoo Lee, and Kyoung Mu Lee. Omnisplat: Taming feed-forward 3d gaussian splatting for omnidirectional images with editable capabilities. In *CVPR*, 2025. 3

Junbo Li, Florian Hahlbohm, Timon Scholz, Martin Eisemann, Jan-Philipp Tauscher, and Marcus Magnor. Spags: Fast and accurate 3d gaussian splatting for spherical panoramas. In *Computer Graphics Forum*, 2025a. 7, 8

Longwei Li, Huajian Huang, Sai-Kit Yeung, and Hui Cheng. Omnigs: Fast radiance field reconstruction using omnidirectional gaussian splatting. In *WACV*, 2025b. 3, 7, 8

Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, et al. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *CVPR*, 2024. 2, 3

Xin Lin, Meixi Song, Dizhe Zhang, Wenxuan Lu, Haodong Li, Bo Du, Ming-Hsuan Yang, Truong Nguyen, and Lu Qi. Depth any panoramas: A foundation model for panoramic depth estimation. In *CVPR*, 2026. 4, 12

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 *ECCV*, 2024. 2, 3, 5, 6, 7, 8

Yang Liu, Chuanchen Luo, Zhongkai Mao, Junran Peng, and Zhaoxiang Zhang. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes. In *ICLR*, 2025. 3

Yikun Ma, Dandan Zhan, and Zhi Jin. Fastscene: Text-driven fast 3d indoor scene generation via panoramic gaussian splatting. In *IJCAI*, 2024. 3

Qiwei Wang, Xianghui Ze, Jingyi Yu, and Yujiao Shi. Cylindersplat: 3d gaussian splatting with cylindrical triplanes for panoramic novel view synthesis. In *ICLR*, 2026. 3

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 7

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 2024. 2

Jiamin Wu, Kenkun Liu, Xiaoke Jiang, Yuan Yao, and Lei Zhang. Unigs: Modeling unitary 3d gaussians for novel view synthesis from sparse-view images. In *ICCV*, 2025. 2

Cheng Zhang, Haofei Xu, Qianyi Wu, Camilo Cruz Gambardella, Dinh Phung, and Jianfei Cai. Pansplat: 4k panorama synthesis with feed-forward gaussian splatting. In *CVPR*, 2025. 3

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In *ECCV*, 2024. 3

APPENDIX

A ADDITIONAL IMPLEMENTATION DETAILS

- **Training Configuration:** Both Stage I (global coarse) and Stage II (block-wise refinement) are trained for 30,000 iterations following the standard 3DGS densification schedule. All methods use identical learning rates: 1.6×10^{-4} for Gaussian positions and 2.5×10^{-3} for spherical harmonics coefficients. The densification interval is 100 iterations with a gradient threshold of 2×10^{-4}. Opacity is reset every 3,000 iterations in Stage II.
- **Partitioning Parameters:** The triangulation range factor is set to $\rho_{\text{tri}} = 5$ across all scenes. The gradient ratio threshold is $\tau_{\text{grad}} = 0.8$. The sky sphere multiplier is $\kappa = 10$ with $N_{\text{sky}} = 100{,}000$ sky Gaussians.
- **Depth Supervision:** We use DAP (Lin et al., 2026) to generate monocular inverse depth maps. The depth weight schedule follows Equation (17) with $w_0 = 0.5$ and $w_T = 0.01$.
- **Hardware and Software:** All experiments run on a single NVIDIA RTX 4090 GPU (24 GB VRAM). The codebase is built on PyTorch 2.1 with custom CUDA kernels for ERP-based Gaussian rasterization.

B ERP PROJECTION AND GAUSSIAN RENDERING DETAILS

This section derives the full equirectangular projection pipeline summarized in Section 3.1.

ERP Pixel Mapping. For a 3D point $\mathbf{X} = (x, y, z)^\top$ in camera coordinates with spherical coordinates (θ, ϕ), the ERP pixel coordinates $\mathbf{u} = (u, v)^\top$ are:

$$\mathbf{u} = \boldsymbol{\pi}(\mathbf{X}) = \begin{pmatrix} \frac{W}{2\pi} \operatorname{atan2}(x, z) + \frac{W}{2} \\ \frac{H}{\pi} \arcsin\left(\frac{y}{\|\mathbf{X}\|}\right) + \frac{H}{2} \end{pmatrix}, \tag{14}$$

where W and H denote the panoramic image width and height.

ERP Jacobian. The Jacobian $\mathbf{J} \in \mathbb{R}^{2\times 3}$ of the projection $\boldsymbol{\pi}$ at point $\mathbf{X}$ is:

$$\mathbf{J} = \frac{\partial \boldsymbol{\pi}}{\partial \mathbf{X}} = \begin{pmatrix} \frac{W}{2\pi} \cdot \frac{z}{x^2+z^2} & 0 & -\frac{W}{2\pi} \cdot \frac{x}{x^2+z^2} \\ -\frac{H}{\pi} \cdot \frac{xy}{\|\mathbf{X}\|^2 \sqrt{x^2+z^2}} & \frac{H}{\pi} \cdot \frac{\sqrt{x^2+z^2}}{\|\mathbf{X}\|^2} & -\frac{H}{\pi} \cdot \frac{zy}{\|\mathbf{X}\|^2 \sqrt{x^2+z^2}} \end{pmatrix}. \tag{15}$$

2D Gaussian Evaluation. Given the projected center $\mathbf{u}_s$ obtained via Equation (14) and a sampling pixel $\mathbf{u}$, the 2D Gaussian is evaluated as:

$$G_{\text{2D}}(\mathbf{u}) = \exp\left\{ -\frac{1}{2} (\mathbf{u} - \mathbf{u}_s)^\top \boldsymbol{\Sigma}_{\text{2D}}^{-1} (\mathbf{u} - \mathbf{u}_s) \right\}, \tag{16}$$

where $\boldsymbol{\Sigma}_{\text{2D}}$ is the projected 2D covariance defined in Equation (2). The panoramic rendering then follows Equation (1) with this ERP-specific G_{2D}.

C DEPTH WEIGHT DECAY SCHEDULE

The time-decaying weight $w(t)$ in the depth loss (Equation (5)) follows an exponential schedule:

$$w(t) = w_0 \cdot \left(\frac{w_T}{w_0}\right)^{t/T}, \tag{17}$$

where w_0 and w_T are the initial and final depth weights, and T is the total number of training iterations. This yields $w(0) = w_0$ and $w(T) = w_T$ exactly. The exponential form ensures rapid early guidance, steering Gaussians from sparse SfM initialization toward geometrically plausible positions, while smoothly transitioning to photometric-dominated optimization without abrupt weight changes.

D BLOCK-WISE OPTIMIZATION DETAILS

This section supplements Section 3.2.2 with additional design rationale.

Full Initialization. Each block is initialized by loading the *full* set of Gaussians from Stage I rather than only those geometrically inside the block. This prevents newly densified Gaussians near block boundaries from replacing pruned structures that were well-optimized during coarse training, which would otherwise introduce artifacts.

Soft-Boundary via Opacity Reset. To adaptively determine each block's effective spatial extent, we leverage periodic opacity reset. The reset operation periodically clamps all scene Gaussian opacities to a low value o_{reset}, after which only those receiving sufficient photometric gradients from the block's assigned cameras recover high opacity through gradient-driven updates. Insufficiently observed Gaussians remain transparent and are subsequently pruned. The half-open interval $[\mathbf{l}_b, \mathbf{u}_b)$ ensures each retained Gaussian is uniquely assigned to one block, avoiding duplicate primitives in the final merged model. Because adjacent blocks share overlapping camera sets (via Equation (11)), Gaussians near block boundaries receive consistent cross-boundary supervision, yielding coherent geometry and appearance at block interfaces without explicit stitching.

Sky Gaussian Freezing. All sky Gaussian parameter gradients are zeroed during block-wise refinement. Since each block observes only a partial set of viewpoints, allowing sky Gaussians to update independently across blocks would cause inconsistent appearance drift. Full freezing maintains cross-block sky consistency.

Depth Supervision in Block Refinement. The depth supervision introduced in Section 3.1 remains active during block refinement, where its role becomes even more critical: each block utilizes only a subset of camera viewpoints, further weakening the geometric constraints available from SfM. The monocular depth prior compensates for this reduced multi-view coverage, stabilizing optimization in geometrically under-constrained regions.

E DATASET DETAILS

Pano360 spans over 2 million m^2 across four diverse urban environments. All subset names (NSC, NSK, BAX, NSN) are anonymized identifiers to avoid disclosing specific geographic locations. Throughout, A1 refers to the Antigravity A1 panoramic UAV, and X5 denotes the Insta360 X5 panoramic camera.

Capture Protocol. NSC and NSK are captured by the A1 drone at an elevated viewpoint, which inherently avoids ground-level pedestrians. BAX and NSN are acquired with the X5 handheld camera at ground level, where the camera operator and other pedestrians are persistently visible near the image nadir.

Evaluation Cropping. Since the X5 camera is handheld, dynamic figures appear near the bottom of the ERP image. To ensure fair comparison, when reporting quantitative metrics on the X5 sub-scenes, we uniformly crop approximately 10% of the ERP image height from the bottom for all evaluated methods (including baselines), thereby excluding dynamic figures from the evaluation region. No such cropping is applied to the A1 sequences.

F ANALYSIS OF PARTITION THRESHOLD

We vary the gradient ratio threshold $\tau_{\text{grad}} \in \{0.7, 0.8, 0.9\}$ in the camera–block allocation (Equation (11)). Lower thresholds assign more cameras per block: at $\tau_{\text{grad}} = 0.7$, each block receives broader supervision, which slightly improves rendering quality but increases model size due to more overlapping Gaussians across blocks. Higher thresholds produce more compact models by restricting camera assignments, but risk leaving boundary regions under-supervised. At $\tau_{\text{grad}} = 0.9$, the restriction becomes excessive, reducing quality on certain blocks. The default $\tau_{\text{grad}} = 0.8$ achieves the best balance between rendering quality and model compactness across all tested scenes.

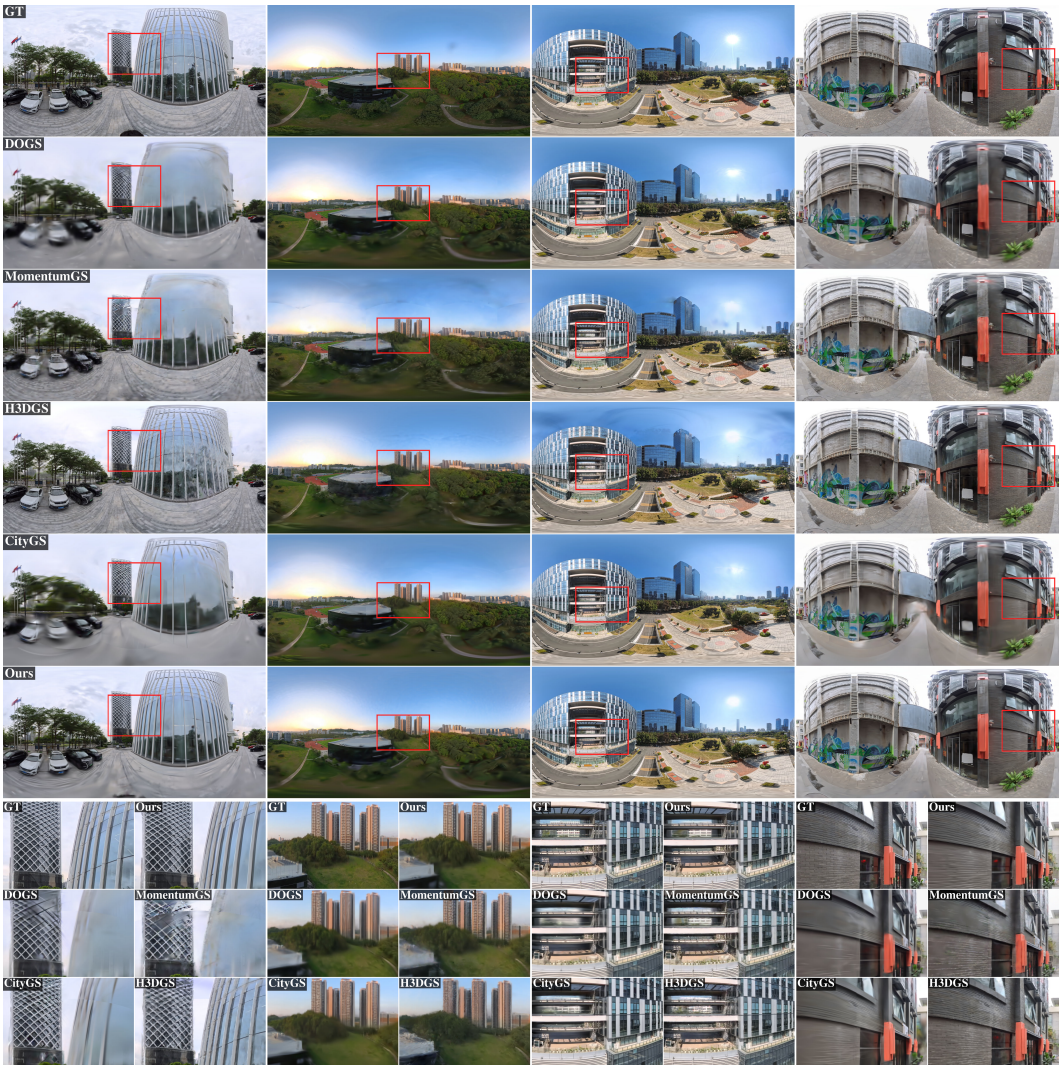

Figure G.1: Additional qualitative comparisons on Pano360, complementing Figure 3 in the main text. The layout follows Figure 3: rows show GT, DOGS, Momentum-GS (MomentumGS), H3DGS, CityGaussian (CityGS), and Ours; red boxes mark the regions magnified in the bottom panels.

G Additional Experimental Results

G.1 Additional Qualitative Results on Pano360

G.2 Qualitative Results on Public Panoramic Datasets

H Ethics Statement

Pano360 was captured in outdoor campus, park, and street environments. To protect privacy, human faces and vehicle license plates in all panoramas are blurred, and the subsets are named with anonymized identifiers that do not disclose specific geographic locations (Section E). All authors have read and adhere to the ICLR Code of Ethics.

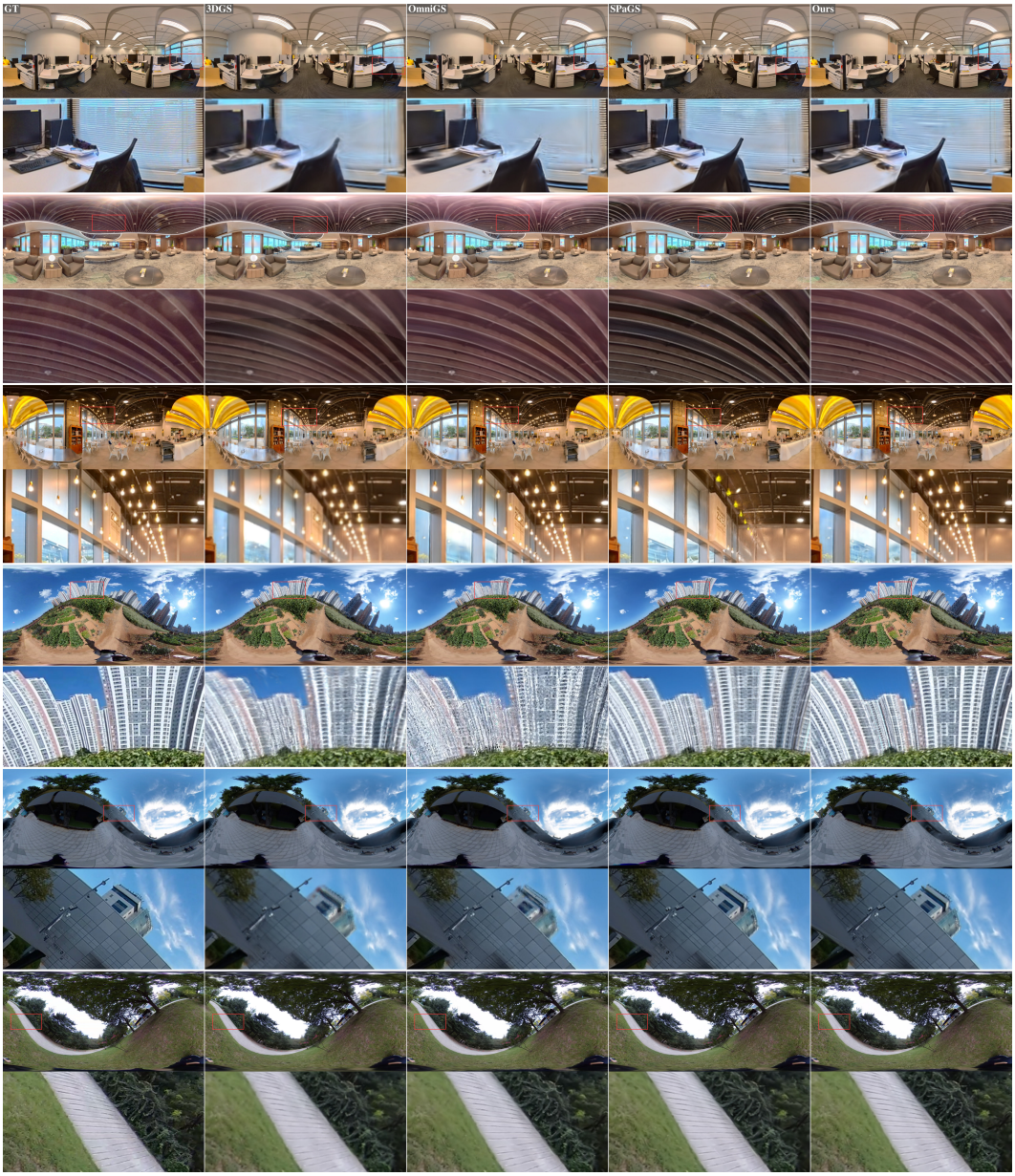

Figure G.2: Additional qualitative comparison on the public 360Roam and Ricoh360 benchmarks. Columns: GT, cubemap-input 3DGS, OmniGS, SpaGS (SPaGS), and Ours; each full ERP rendering is followed by a magnified view of its red-boxed region. PanoLOG produces sharper textures and fewer artifacts in both indoor and outdoor scenes. ODGS is not visualized here; its quantitative results are reported in Table 3.