SPLATANYWHERE: 3D GAUSSIAN SPLATTING FROM SPARSE AND INCONSISTENT STREET PANORAMAS

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) produces photorealistic scene models, yet reconstructing street scenes still requires dedicated capture for every target site, making reconstruction costs grow with every new location. Google Street View (GSV) already covers a large share of the world's roads with panoramas captured repeatedly over the years. Reconstruction from GSV would enable worldwide 3D digital coverage at no additional capture cost. However, GSV panoramas are sparse in camera centers, which leaves depth weakly constrained and yields models that look acceptable at training viewpoints but show floating artifacts under extrapolated views. They are also inconsistent in appearance across capture sessions, which adds photometric error that no shared color can remove. Standard photometric optimizers, ill-suited for this problem, compensate for the photometric error with wrong geometry. To reconstruct robustly from GSV, we present SPLATANYWHERE, an optimization-based pipeline that builds structure-aligned 3DGS scenes directly from GSV panoramas. SPLATANYWHERE adds structural supervision losses in the regions that Structure from Motion under-represents, suppresses splits where the views disagree, reducing the number of Gaussians while improving the geometry, and uses a session appearance adapter that takes up the photometric differences across capture sessions. Across street scenes sampled from five representative cities, SPLATANYWHERE reaches the best average PSNR, SSIM, and LPIPS compared with 3DGS baselines, lowers their 3D structural error by 26% to 89%, and holds these gains even for extrapolated views.

1 INTRODUCTION

High-fidelity 3D models of real-world street scenes are essential for autonomous driving, outdoor robotics, transportation, simulation, and digital twins (Dosovitskiy et al., 2017; Yang et al., 2023). As these emerging applications penetrate the market, there is a growing demand for methods that can build such models easily and at scale for arbitrary deployment sites. 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) offers photorealistic fidelity and real-time rendering. However, reconstructing street scenes with 3DGS remains expensive. Optimization-based (Kerbl et al., 2023; Liu et al., 2024) and feed-forward pipelines (Charatan et al., 2024; Wang et al., 2025a) require dedicated per-site capture with diverse and dense viewpoints under consistent illumination, so the cost grows with every new site. Generative methods (Schwarz et al., 2025; Gao et al., 2024) drop this requirement by synthesizing scenes from learned priors rather than observed data, but their output drifts from reality, producing plausible scenes rather than geometrically faithful reconstructions.

In this paper, we target reconstruction of street locations from publicly available data without dedicated capture, and we treat real-world fidelity and geometric accuracy as primary objectives. Leveraging the worldwide coverage of Google Street View (GSV) (Anguelov et al., 2010), we aim at the capability of turning any road in the world into a digital experimental field. To realize this goal, however, the inherent limitations of GSV present two challenges. First, GSV camera centers lie on a narrow strip of the road, with limited lateral spread and almost no height variation. By contrast, standard datasets (*e.g.,* Mip-NeRF 360 (Barron et al., 2022)) provide diverse viewpoints that can supply sufficient supervision through the photometric loss alone. Second, GSV panoramas captured in different years differ in exposure, white balance, sun position, and scene content, whereas standard datasets are captured in a single session with consistent illumination.

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.

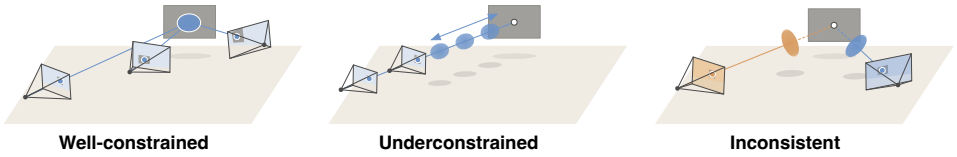


Figure 1: 3DGS under different capture conditions. Well-distributed captures constrain Gaussians to the surface and yield high-quality 3DGS (left), while underconstrained captures with limited viewpoint diversity may generate floaters that are weakly constrained along shared rays (middle). When images disagree on the appearance of the same surface, the optimizer may exploit this geometric freedom to reduce photometric error (right).

Recent work addresses each in isolation. Sparse-view and depth-supervised methods (Xiong et al., 2025; Li et al., 2026; Deng et al., 2022; Li et al., 2024; Xie et al., 2025) constrain geometry under sparse viewpoints but assume consistent appearance. In-the-wild methods (Kulhanek et al., 2024; Zhang et al., 2024) model appearance variation but rely on viewpoint diversity to keep geometry grounded. Under GSV, however, both conditions fail at once. Underconstrained depth gives the optimizer freedom to place Gaussians away from surfaces, and cross-session disagreement gives it a reason to use this freedom. When views from different years disagree on a surface's color, no single color fits all of them, and the optimizer can reduce the remaining error by changing geometry (Wang et al., 2024b; Deutsch et al., 2026), as illustrated in Figure 1. This suboptimal way of optimization therefore systematically preserves floaters and artifacts, which cannot be eliminated through the photometric loss alone. Worse, test views taken along the capture trajectory share the rays along which these artifacts move, so standard photometric evaluation cannot reveal the problem.

To address these challenges, we propose SPLATANYWHERE, an optimization-based 3DGS pipeline that robustly reconstructs from GSV panoramas. Our method focuses on recovering accurate geometry and handling view inconsistency at the same time. For geometry, supervising depth from SfM triangulation is a natural and low-cost choice (Deng et al., 2022), but under GSV the distribution of SfM points is severely biased toward textured building walls, while road surfaces, whose geometry should be equally important, receive disproportionately fewer samples due to weak texture. We strengthen road geometry with a *semantic road-height grid*. A low-cost *ray clearance loss* further removes the remaining floaters in front of the surfaces, using monocular depth only as a conservative bound. For inconsistency, we design a *session appearance adapter* that takes up the photometric differences across capture sessions, so that the optimizer does not have to pay for color differences with geometry. Additionally, we design a *gradient-based consensus gate* to adaptively limit oversplitting, reducing the ultimate reconstruction size while maintaining fidelity. On five sample cities across three continents, SPLATANYWHERE outperforms 3DGS baselines in average PSNR, SSIM, and LPIPS, and lowers their 3D structural error by 26% to 89%, while the consensus gate halves the number of Gaussians. The gains hold or even widen for extrapolated out-of-distribution perspectives. In summary, we make the following contributions:

- We present SPLATANYWHERE, a framework that robustly reconstructs 3DGS street scenes from publicly available sparse Google Street View panoramas without dedicated dense capture.
- We design an effective structural supervision mechanism that addresses the limitation of underconstrained GSV observations.
- We create a session appearance adapter to reduce geometric errors incurred during reconciliation of cross-year inconsistent observations.
- We propose a disagreement-aware splitting control mechanism for 3DGS densification, which halves the number of Gaussians and lowers the 3D structural error.
- Extensive evaluation establishes SPLATANYWHERE's state-of-the-art performance in both photometric and geometric metrics, with robust reconstructions on extrapolated views.

2 RELATED WORK

Street-level and large-scale scene reconstruction. Large-scale street scene models can provide essential testbeds for downstream tasks such as autonomous driving validation and traffic simulation. Existing systems differ fundamentally in the input they assume. Simulation platforms such as CARLA (Dosovitskiy et al., 2017) build environments from manually authored assets, which take substantial human effort to build. Generation-based systems synthesize street-level scenes from as

little as a single satellite image (Qian et al., 2026) or synthesize driving scenarios at scale (Wang et al., 2025b; NVIDIA et al., 2025), producing plausible approximations rather than faithful reconstructions. A fast-growing family of feed-forward geometry models, such as DUSt3R (Wang et al., 2024a) and VGGT (Wang et al., 2025a), recovers point maps and camera poses from image sequences (Deng et al., 2025; Chen et al., 2026; Wang et al., 2026b), and some attach 3DGS prediction heads (Lin et al., 2025), yet they rely on large-scale pretraining and coherently captured input sequences. Driving-scene reconstruction methods (Yan et al., 2024; Wang et al., 2026a) likewise build renderable street-scene models on an optimization-based backbone, and diffusion priors extend them beyond the driven lane (Yan et al., 2025; Ni et al., 2025). However, they build on well-annotated, high-quality sequential captures. Even under such captures, road surfaces remain a weak spot, since SfM yields few points there and street surface reconstruction reports incomplete roads as an open limitation (Cui et al., 2025). Different from these systems, we reconstruct photorealistic, structure-aligned 3DGS scenes from publicly available GSV panoramas with an optimization-based pipeline, requiring neither dedicated capture nor large-scale pretraining for this task.

Table 1: Characteristics of Common Data Sources

Data source	View diversity	Appearance consistency	Extra sensors	Capture coverage
Standard (Mip-NeRF 360)	✓	✓	✗	Scene
Driving (Waymo, PandaSet)	✗	✓	✓	Street
Tourism (Phototourism)	✓	✗	✗	Landmark
Sparse view (LLFF, DTU)	✗	✓	✗	Scene
Google Street View	✗	✗	✗	**World**

Reconstruction with imperfect data. Real-world captures rarely satisfy the ideal conditions that standard 3DGS assumes, and prior work has tackled two imperfections separately. To handle photometric inconsistency, in-the-wild methods attach learnable appearance embeddings to individual images and splats (Kulhanek et al., 2024; Zhang et al., 2024), ChronoGS (Wang et al., 2026c) further models temporal scene changes, and PPISP (Deutsch et al., 2026) explains camera-side variation through interpretable ISP transformations. To handle sparse or degenerate viewpoints, sparse-view methods distill geometric priors from monocular depth or diffusion models (Xiong et al., 2025; Li et al., 2024), depth-supervised methods constrain geometry with SfM or relative depth (Deng et al., 2022; Xie et al., 2025), and driving-scene works leverage diffusion priors to extend reconstruction beyond the captured trajectory (Yan et al., 2025; Ni et al., 2025). Each line, however, largely presumes the other imperfection away. Appearance modeling relies on abundant viewpoint diversity to disentangle appearance from geometry, while sparse-view priors and depth supervision assume consistent appearance across views. SparseGS-W (Li et al., 2026) touches both imperfections, yet it is designed for a small number of training images and relies heavily on diffusion priors, which makes it still more aligned with the line of work that handles sparsity. As Table 1 summarizes for common data sources (Barron et al., 2022; Sun et al., 2020; Xiao et al., 2021; Snavely et al., 2006; Mildenhall et al., 2019; Jensen et al., 2014), GSV is the only one that provides neither viewpoint diversity nor appearance consistency, while being the only one with worldwide street-level coverage. Our work targets this combination, routing structural supervision to the regions that SfM spatially under-represents and intercepting cross-session disagreement before it turns into geometry errors. SPLATANYWHERE establishes the capability to reconstruct high-fidelity 3D scenes at world scale.

3 METHODOLOGY

3.1 OVERVIEW

Problem formulation. To reconstruct a target scene, we take as input every GSV panorama within a radius of a query coordinate, together with its GPS position and capture date. Without loss of generality, each panorama is cut into $J = 12$ perspective crops (denoted as v) with a 90° field of view and shared intrinsics. All crops of one panorama share one camera center. Panoramas captured in the same year and month form a capture session, and $\tau(v)$ denotes the session of crop v.

The scene is represented by N_{GS} Gaussians with centers $\boldsymbol{\mu}_i$, scales $\mathbf{s}_i$, opacities α_i, and spherical harmonics colors $\mathbf{c}_i$ with zeroth-order band $\mathbf{c}_i^{(0)}$. Each pixel u of a crop defines a ray from the camera center through u. Following 3DGS (Kerbl et al., 2023), the Gaussians along this ray are sorted by depth d_k and composited with weights $w_k = \alpha_k \prod_{k'<k}(1-\alpha_{k'})$, which gives the rendered color $\hat{I}^{(u)} = \sum_k w_k \mathbf{c}_k$, accumulated opacity $A^{(u)} = \sum_k w_k$, and depth $\hat{d}^{(u)} = \sum_k w_k d_k / A^{(u)}$. The problem is to find the N_{GS} Gaussians $\{\boldsymbol{\mu}_i, \mathbf{s}_i, \alpha_i, \mathbf{c}_i\}_{i=1}^{N_{\mathrm{GS}}}$ that minimize the optimization loss $\mathcal{L}_{\mathrm{GS}}$ summed across the crops, $\sum_v \mathcal{L}_{\mathrm{GS}}(v)$. For standard 3DGS, $\mathcal{L}_{\mathrm{GS}}$ contains only the photometric loss

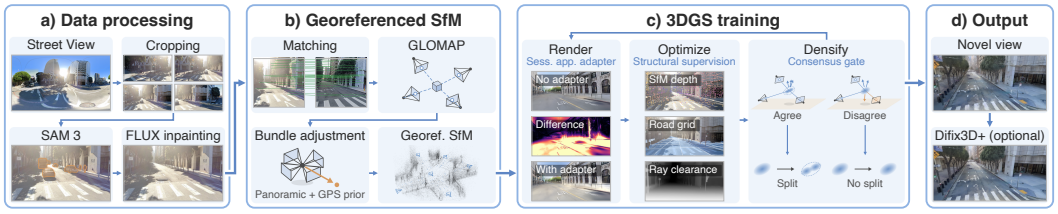


Figure 2: SPLATANYWHERE overview. (a) Each GSV panorama is cut into perspective crops, and transient objects are segmented with SAM 3 and inpainted with FLUX. (b) Crops of the same and neighboring panoramas are matched, GLOMAP recovers the initial poses, and a bundle adjustment with the panoramic constraint and a GPS prior georeferences them. (c) 3DGS training loops over three stages, each with one of our mechanisms. Rendering applies the session appearance adapter, shown for one crop without and with the adapter together with their difference. Optimization adds structural supervision from SfM depth, the road grid, and ray clearance. Densification splits a Gaussian only when the gradients from its crops agree. (d) The trained model renders the scene from novel viewpoints, and Difix3D+ can optionally refine the renderings.

$\mathcal{L}_{\text{rgb}}$ between the rendered colors $\hat{I}^{(u)}$ and the pixels of the crop, while we also focus on structural correctness of the generated results.

Photometric loss and geometry. Standard 3DGS places Gaussians by the photometric loss alone. The photometric loss constrains depth well when camera centers surround the scene, because a Gaussian at the wrong depth then projects to the wrong pixels in some crop, where the error is noticed and corrected. GSV camera centers, by contrast, lie on a narrow strip of road at a fixed height, so depth is weakly constrained along the rays these cameras share, and a Gaussian can move in depth there with little change in any training crop. Prior work observes that inconsistency among training images produces floaters (Martin-Brualla et al., 2021; Wang et al., 2024b), and the weak depth constraint of GSV can leave such Gaussians uncorrected.

Consider a surface point observed in a set of crops, where crop v shows it with color I_v, and session τ holds n_τ of these crops with mean color $\bar{I}_\tau$. The shared color that minimizes the photometric error is the mean $\bar{I}$ over all these crops, and the remaining error splits into two terms,

$$\sum_v \|I_v - \bar{I}\|^2 = \underbrace{\sum_\tau \sum_{v \in \tau} \|I_v - \bar{I}_\tau\|^2}_{E_{\text{crop}}} + \underbrace{\sum_\tau n_\tau \|\bar{I}_\tau - \bar{I}\|^2}_{E_{\text{sess}}}, \tag{1}$$

where E_{crop} collects the differences among the crops within each session and E_{sess} collects the differences between sessions. E_{sess} is nonzero whenever sessions disagree, and no shared color can reduce it. The optimizer can still lower it by sacrificing geometry, either by moving Gaussians off the surface, for example as semi-transparent Gaussians in front of it, or by adding Gaussians.

Method overview. To reconstruct robustly from GSV, we design our pipeline as shown in Figure 2. We first remove vehicles, pedestrians, and bicycles from every crop, segmenting them with SAM 3 (Carion et al., 2026) and inpainting them with FLUX (Black Forest Labs et al., 2025) (Section 3.2). We then recover camera poses with GLOMAP (Pan et al., 2024) with the help of the geographic information in the Google metadata (Section 3.3). After that, we train the Gaussians with three mechanisms added to 3DGS, each addressing a different way in which the optimizer may sacrifice geometry for the photometric loss. Structural supervision keeps the Gaussians on the surfaces (Section 3.4). Consensus-gated densification suppresses splits where the crops disagree (Section 3.5). The session appearance adapter lets the colors change with the capture date, so that differences between sessions need not be explained by geometry (Section 3.6).

3.2 PREPROCESSING: FROM RAW PANORAMAS TO INPAINTED CROPS

In GSV, vehicles, pedestrians, and bicycles appear at different places in different sessions and would create wrong correspondences if left in place. However, directly masking them out removes near-field features and weakens the pose constraints. In contrast, inpainting restores matchable texture without the moving content (Bescos et al., 2020). Hence, we segment these transient objects in every crop with SAM 3 (Carion et al., 2026) and fill them with FLUX (Black Forest Labs et al.,

2025). Note that diffusion-based inpainting can hallucinate content, but each crop is inpainted independently, so hallucinated content disagrees across crops and rarely survives triangulation in SfM.

3.3 Georeferenced Structure from Motion

We use SfM to recover camera poses and SfM points. However, given only the images, SfM often fails to converge to the correct geometry (see details in Appendix A.9). Therefore, we incorporate metadata to facilitate convergence and improve performance. Specifically, the information we use in the optimization includes the camera intrinsics, which are fixed by how we cut the crops, and the camera center locations provided by GSV.

The geoinformation is used for both efficiency and accuracy in the SfM process. For efficiency, since the total number of crops is large at our reconstruction scale, we construct a road graph to find the spatial connections between crops. With the road graph, we adopt SuperPoint (DeTone et al., 2018) and SuperGlue (Sarlin et al., 2020) to perform feature matching only between crops of the same or neighboring panoramas. The matching cost thus grows with the number of neighbors instead of with the square of the number of crops, which reduces the complexity from quadratic to linear. For accuracy, we incorporate the GSV positions, but these positions, especially the altitudes, carry errors (see details in Appendix A.11), so we use them only as a soft prior. Also, all J crops of one panorama share one camera center and differ only in heading. Following Vuong et al. (2024), we enforce this panoramic constraint and refine one rotation $\mathbf{R}_p$ and one center $\mathbf{o}_p$ per panorama p instead of one pose per crop. This process reduces the number of pose parameters by a factor of J. The refinement is a bundle adjustment of the GLOMAP (Pan et al., 2024) solution with a Huber penalty on the distance between each panorama center and its GPS position. The penalty weights the vertical direction less because the altitudes are noisier. After the initial optimization, we fix all poses and re-triangulate all matched points. In summary, the full objective is

$$\begin{aligned} \min_{\mathbf{R}_p,\mathbf{o}_p,\mathbf{X}_m} \; & \frac{1}{N_{\text{obs}}} \sum_{(v,m)} \rho_{\text{px}}\big(\big\|\pi\big(\mathbf{R}_v(\mathbf{X}_m-\mathbf{o}_v)\big)-\mathbf{x}_{vm}\big\|\big) + \frac{\lambda_{\text{gps}}}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} \rho_{\text{m}}\big(\big\|\mathbf{W}(\mathbf{o}_p-\mathbf{g}_p)\big\|\big) \\ \text{s.t.} \; & \mathbf{R}_v = \mathbf{R}_{\text{yaw}}(h_v)\,\mathbf{R}_{p(v)}, \quad \mathbf{o}_v = \mathbf{o}_{p(v)} \quad \text{for every crop } v. \end{aligned} \tag{2}$$

The first term is the standard reprojection error over all N_{obs} keypoint observations, where $\mathbf{x}_{vm}$ is the keypoint of the 3D point $\mathbf{X}_m$ in crop v, $\mathbf{R}_v$ and $\mathbf{o}_v$ are the rotation and center of that crop, and π is the pinhole projection with the crop intrinsics. Each crop takes the center $\mathbf{o}_p$ of its panorama $p(v)$ and the world-to-panorama rotation $\mathbf{R}_p$ turned by the known heading h_v about the vertical axis. The second term is the soft GPS prior over the set of panoramas $\mathcal{P}$. It pulls each panorama center toward its GPS position $\mathbf{g}_p$, and $\mathbf{W}$ down-weights its up component. The thresholds of the Huber penalties ρ_{px} and ρ_{m}, the weight λ_{gps}, and $\mathbf{W}$ are tunable parameters (Appendix A.8).

3.4 Structural Supervision as 3DGS Optimization Losses

Given the camera poses and the sparse points from SfM, 3DGS initializes the Gaussians and starts the optimization. Since GSV camera positions are concentrated along the road, the photometric loss leaves the depth dimension along the shared rays free, so the optimizer can move Gaussians off the surface without penalty from any training crop. Thus, keeping them on the surface needs references from outside the photometric loss. SfM points are good candidates for references (Deng et al., 2022), but they exist mainly in texture-rich areas. Under GSV, the road surface covers a large part of the scene that we focus on, yet its relatively uniform color yields few SfM points, so SfM points alone give the road little reference. We therefore design separate references for textured surfaces and for the road, and further clear the free space indicated by a monocular depth model, as follows.

Textured surfaces: SfM depth anchors. We project the 3D SfM points into each crop and get the reference depths at their projections. During optimization, we compare the rendered depth $\hat{d}^{(u_m)}$ at the projection u_m of point $\mathbf{X}_m$ with the depth d_m of $\mathbf{X}_m$ in the crop, and apply

$$\mathcal{L}_{\text{sfm}} = \frac{\sum_{m\in\mathcal{M}_v} \omega_m\, \rho_\delta\big((\hat{d}^{(u_m)}-d_m)/d_m\big)}{\sum_{m\in\mathcal{M}_v}\omega_m}, \tag{3}$$

where $\mathcal{M}_v$ collects the reliable SfM points that project into crop v at pixels with enough rendered opacity for a meaningful depth, ρ_δ is the Huber function, and ω_m gives more weight to points with longer tracks, wider triangulation angles, and smaller reprojection errors.

Ground surfaces: road geometry supervision. Because of their relatively uniform color, road surfaces have few SfM points to provide direct structural guidance. However, the road has a relatively simple shape, and current segmentation models can detect which pixels belong to it. We therefore prompt SAM 3 with terms for the ground surface on the inpainted crops to obtain road masks, divide the ground into square cells, and set the height H of each cell to the median height of the SfM points that fall inside the road masks. The supervision makes road rays end on this grid, as

$$\mathcal{L}_{\text{surf}} = \underset{u \in \mathcal{U}_{\text{surf}}}{\text{mean}} \, \rho_\delta\big(\hat{d}^{(u)}/d_{\text{g}}^{(u)} - 1\big), \tag{4}$$

where $\mathcal{L}_{\text{surf}}$ takes the road-mask pixels and regulates the corresponding rendered depth with the same relative Huber error as $\mathcal{L}_{\text{sfm}}$. Here $\hat{d}^{(u)}$ is the rendered depth of pixel u, and $d_{\text{g}}^{(u)}$ is the depth at which its ray meets the actual height grid. Two auxiliary terms with small weights, $\mathcal{L}_z$ and $\mathcal{L}_n$, help hold the road Gaussians at the grid height and align their normals with the rendered surface (see details in Appendix A.12). The total road loss is $\mathcal{L}_{\text{road}} = \mathcal{L}_{\text{surf}} + \lambda_z \mathcal{L}_z + \lambda_n \mathcal{L}_n$.

Free space: ray clearance. SfM depth and the road grid already hold most surfaces in place. As a low-cost addition, we further remove the remaining floaters in the free space in front of the surfaces. Depth Anything 3 (DA3) (Lin et al., 2025) indicates how far the space along each ray is free, but its depth is only relative and inaccurate near object boundaries, so it cannot serve as a target for fine structure. We therefore use it only as a conservative bound and make Gaussians transparent where they lie well inside this free space. With the DA3 depth aligned to the rendered depth as the reference $d_{\text{ref}}^{(u)}$, we penalize the opacity that accumulates in front of the reference,

$$\mathcal{L}_{\text{clear}} = \underset{u \in \mathcal{U}_{\text{clear}}}{\text{mean}} \, \text{relu}\Big(\sum_k \frac{w_k}{\max(d_k, 1)} - \kappa \, \frac{A^{(u)}}{d_{\text{ref}}^{(u)}}\Big), \tag{5}$$

where $\mathcal{U}_{\text{clear}}$ collects the pixels of all crops in the classes listed in Appendix A.8, the sum is the opacity-weighted inverse depth along the ray, and $\kappa > 1$ is a tolerance that keeps a margin in front of the reference. $\mathcal{L}_{\text{clear}}$ thus lowers the opacity of Gaussians only well inside the space that DA3 indicates as free, so the monocular depth acts as a bound rather than a target.

3.5 Consensus-Gated Densification

We observe that reconstructing from GSV tends to over-densify Gaussians, and this leads to unnecessarily large models. We propose a gradient-based mechanism that limits densification where crops disagree. In standard practice (Ye et al., 2024), the i-th Gaussian is split when its largest scale $\max(\mathbf{s}_i)$ exceeds θ_{s} and its image-space gradient averaged over crops $\bar{g}_i$ exceeds θ_{g}, where θ_{s} and θ_{g} are thresholds for scale and gradient, respectively. This gradient combines contributions from the pixels within one crop and from different crops. GDAGS (Zhou et al., 2026) examines the pixels within one crop and splits more where they pull a Gaussian in conflicting directions. On GSV, however, many conflicts arise from the inconsistency across crops. If these conflicts trigger splits, the new Gaussians fit the differences, which over-splits the model without structural gain. We therefore take the opposite action to GDAGS and weight the split criterion of a Gaussian by how much its crops agree on the direction of its gradient, measured by the consensus ratio C_i,

$$C_i = \frac{\|\sum_v \nabla_{\boldsymbol{\mu}_i} \mathcal{L}_{\text{GS}}(v)\|}{\sum_v \|\nabla_{\boldsymbol{\mu}_i} \mathcal{L}_{\text{GS}}(v)\|} \in [0,1], \qquad \text{split } i \;\Leftrightarrow\; C_i \, \bar{g}_i > \theta_{\text{g}} \text{ and } \max(\mathbf{s}_i) > \theta_{\text{s}}, \tag{6}$$

where v runs over the crops that have seen the i-th Gaussian since the last densification step, and $\nabla_{\boldsymbol{\mu}_i} \mathcal{L}_{\text{GS}}(v)$ is the gradient of the loss of crop v with respect to the center $\boldsymbol{\mu}_i$ of this Gaussian. Gradients that agree add up and keep C_i near 1. Gradients that conflict cancel and lower C_i, so a Gaussian whose crops disagree accumulates less densification gradient.

3.6 Session Appearance Adapter

Many differences between sessions come from exposure, white balance, and the seasonal color of vegetation. We design a session appearance adapter that absorbs the differences, so that the optimizer does not have to use geometry to explain them. We encode session τ as $\mathbf{e}_\tau$, and route it through two networks, $\text{MLP}_{\text{img}}(\mathbf{e}_\tau) \to (a_\tau, b_\tau, \boldsymbol{\beta}_\tau)$ for crops and $\text{MLP}_{\text{gs}}(\mathbf{e}_\tau) \to \mathbf{B}_\tau \in \mathbb{R}^{3\times q}$ for Gaussians,

$$\hat{I}_\tau = a_\tau \, \hat{I} \odot \boldsymbol{\beta}_\tau + b_\tau, \qquad \mathbf{c}_{i,\tau}^{(0)} = \mathbf{c}_i^{(0)} + \mathbf{B}_\tau \, \mathbf{z}_i, \tag{7}$$

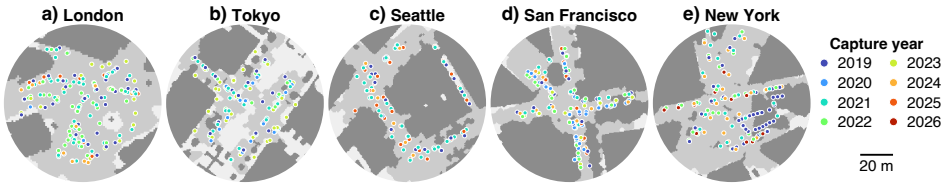


Figure 3: Captured panoramas of the five scenes seen from above. Each dot is one panorama at its reconstructed camera center, colored by capture year. Dark gray marks buildings and light gray marks roads in the airborne survey.

where a_τ, b_τ, and $\boldsymbol{\beta}_\tau$ are the luminance gain, offset, and white balance of the rendered crop $\hat{I}$, and $\mathbf{z}_i \in \mathbb{R}^q$ is the Gaussian color code of the i-th Gaussian. Previous per-image appearance methods (Kulhanek et al., 2024; Zhang et al., 2024) combine a per-image embedding with per-Gaussian features in one network and need pixels of each new view, such as half of the image, to obtain the embedding, whereas our adapter applies to any test view given only its session. During optimization, we limit the capacity of the adapter so that it does not absorb content changes, with a penalty $\mathcal{L}_{\text{reg}}$ that pulls $(a_\tau, b_\tau, \boldsymbol{\beta}_\tau)$ toward the identity and the color shift $\mathbf{B}_\tau \mathbf{z}_i$ toward zero.

Apart from this penalty, the adapter enters only $\mathcal{L}_{\text{rgb}}$, and the total loss is $\mathcal{L}_{\text{GS}} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{sfm}}\mathcal{L}_{\text{sfm}} + \lambda_r\mathcal{L}_{\text{road}} + \lambda_c\mathcal{L}_{\text{clear}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}$, with the weights in Appendix A.8.

4 Experiments

Datasets. We evaluate on five representative street scenes from London, Tokyo, Seattle, San Francisco, and New York to cover different conditions on different continents (Table 2 and Figure 3). The scenes cover commercial cores, an office district, a tourism landmark block, and a residential street. For each scene, we take all GSV panoramas within $40\,\text{m}$ of a target coordinate captured from 2019 onward, in total 691 panoramas in 74 sessions.

Table 2: Statistics of the Five Street Scenes

City	#Pano.	#Sess.	Years	Scene
London	192	22	2019∼2025	Commercial
Tokyo	122	8	2019∼2023	Commercial
Seattle	96	7	2019∼2025	Office
San Francisco	142	24	2019∼2025	Landmark
New York	139	13	2019∼2026	Residential
Total	691	74		

Test views. All panoramas are used for training, and we derive three groups of test views from the training panoramas. *1) Interpolation* keeps the panorama position and zero pitch, and only changes the heading, with three headings per panorama at least 5° away from the twelve training headings. It directly evaluates interpolation between the training crops, which resembles the common protocol of testing on unused frames of a capture sequence. *2) Pitch Shift* keeps the panorama position and looks 35° to 45° downward, which evaluates out-of-distribution viewing angles. *3) Lateral Shift* moves the camera $4\,\text{m}$ sideways from the capture trajectory at the capture height and turns it toward the scene interior, with one view per panorama. It evaluates out-of-distribution camera positions, where no ground-truth GSV photograph exists, so we use different metrics for it below.

Metrics. We evaluate appearance and structure with three groups of metrics. *1) PSNR, SSIM, and LPIPS*, the peak signal-to-noise ratio, the structural similarity (Wang et al., 2004), and the learned perceptual image patch similarity (Zhang et al., 2018), compare each Interpolation and Pitch Shift rendering with its ground-truth photograph on static pixels, with transient objects masked by SAM 3 (Carion et al., 2026). We average them over the test views, and they measure how closely the renderings match the photographs. *2) KID and FID*, the kernel (Bińkowski et al., 2018) and Fréchet (Heusel et al., 2017) inception distances, compare the Lateral Shift renderings as a set with the ground-truth photographs of the Interpolation views of the same scene. Since Lateral Shift has no ground truth, they measure how realistic the renderings look as a distribution rather than pixel by pixel. *3) 3D structural error (3DSE)* is a new metric that compares the rendered depth with the public airborne LiDAR survey of each city (Appendix A.20). For every LiDAR point in a view, we take the distance between the rendered surface and the point along the surface normal, and 3DSE is the median of this distance over the points in a view (at least 200), then the median over the views of a city. The median keeps outlier points from dominating the score. 3DSE measures whether the geometry is correct, which the photometric metrics cannot reveal.

Table 3: End-to-End Comparison with Baselines Averaged over Five Cities

Method	Interpolation				Pitch Shift				Lateral Shift		
	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	KID↓	FID↓	3DSE (m)↓
Vanilla 3DGS (Kerbl et al., 2023)	17.90	0.588	0.459	0.979	16.68	0.548	0.747	0.670	0.108	179.7	0.951
Scaffold-GS (Lu et al., 2024)	17.58	0.589	0.452	0.648	16.48	0.549	0.758	0.600	0.107	174.0	0.634
WildGaussians* (Kulhanek et al., 2024)	16.14	0.526	0.612	0.886	16.41	0.543	0.815	1.085	0.174	238.9	0.750
Vid2Sim (Xie et al., 2025)	17.42	0.602	0.393	0.305	16.67	0.545	0.716	0.229	0.085	161.8	0.297
ChronoGS (Wang et al., 2026c)	15.13	0.533	0.513	0.567	15.08	0.528	0.800	1.056	0.115	181.0	0.569
Google Earth	11.50	0.403	0.669	0.122	12.59	0.470	0.815	0.077	0.314	350.9	0.118
VGGT-X (Liu et al., 2025)	12.34	0.390	0.642	1.201	12.40	0.428	0.794	0.931	0.278	343.6	1.217
DA3-GS (Lin et al., 2025)	14.66	0.454	0.535	1.009	17.35	0.538	0.670	2.455	0.140	206.1	0.972
Ours (unconditioned)	17.52	0.626	0.386	0.209	17.19	0.568	0.695	0.123	0.080	159.0	0.220
Ours (conditioned)	18.16	0.631	0.382	0.209	17.81	0.572	0.691	0.123	0.088	164.6	0.220

* Stabilized variant of WildGaussians.

Baselines. We compare against *Vanilla 3DGS* (Kerbl et al., 2023), the anchor-based *Scaffold-GS* (Lu et al., 2024), the in-the-wild method *WildGaussians* (Kulhanek et al., 2024), the geometry-supervised *Vid2Sim* (Xie et al., 2025), and the time-conditioned *ChronoGS* (Wang et al., 2026c). Every baseline runs its released training code. All baselines use the same panoramas and camera intrinsics as our method, with camera poses from GLOMAP (Pan et al., 2024) and transient objects masked by SAM 3 in both feature extraction and training. The original WildGaussians is unstable on GSV, so we report a stabilized variant, *WildGaussians**, disabling its uncertainty mask and the periodic opacity reset (see details in Appendix A.22). We also include three reference systems that represent alternative approaches to worldwide 3D models, namely *Google Earth*, *VGGT-X* (Liu et al., 2025), and the *feed-forward 3DGS head of Depth Anything 3 (DA3-GS)* (Lin et al., 2025). We do not include baselines that handle sparsity because they rely on diffusion models and cannot scale.

SPLATANYWHERE variants. We report two versions of our method. *Ours (unconditioned)* renders without the session conditioning described in Section 3.6. *Ours (conditioned)* renders each test view in the session of its panorama, conditioned on the capture date.

Quantitative comparison. Table 3 reports end-to-end results averaged over the five cities (see Appendix A.24 for the results of each city). Our method reaches the best or second-best result on every metric, holding the advantages even under extrapolated views. When the session of the test view is known, as on Interpolation and Pitch Shift, Ours (conditioned) reaches the highest PSNR and SSIM, and on Lateral Shift, whose reference crops mix all sessions, Ours (unconditioned) reaches the lowest KID and FID. In 3D structural error, our method is second only to the Google Earth mesh, which is built from aerial capture. Google Earth reaches only $11.50\,\mathrm{dB}$ PSNR at street level. Among all methods that reconstruct from street-level images, ours has the lowest 3D structural error on every view group, 26 % to 89 % better than the 3DGS baselines. On Interpolation, the PSNR of the 3DGS baselines other than ChronoGS and WildGaussians* remains at a similar level to ours, yet their 3D structural errors are worse by up to a factor of 4.7, so photometric metrics in interpolated views alone cannot evaluate the geometry of these methods (Appendix A.3). Applying Difix3D+ (Wu et al., 2025) to every method keeps our lead, with the best or equal-best LPIPS in every city and view group, and on average 4 % to 29 % better than the baselines (Appendix A.7). This result indicates the importance of clean 3DGS backbones for diffusion-based refinement/fixing. Note that Vid2Sim is overall the strongest baseline, which provides depth supervision from monocular depth at a relative scale. Through a more comprehensive comparison (in Appendices A.5 to A.7), we show that our method still achieves overall the most robust results across different settings.

Qualitative comparison. Figure 4 shows two sample views for each test group. On Interpolation, Vanilla 3DGS, Scaffold-GS, and Vid2Sim blur the shop front in row 1 and the lettering on the glass in row 2, and Ours keeps both readable. On Pitch Shift, when the camera looks down in rows 3 and 4, the road of Vanilla 3DGS and Scaffold-GS turns into a foggy structure in which the lane markings can hardly be read, and Vid2Sim also loses half of the turn arrow, while ours draws both cleanly. Under the Lateral Shift setting, Vid2Sim shows poor structure of the curb and the street, while ours keeps the curbs and lane markings. DA3-GS is blurry and has dark holes, and the Google Earth mesh has the right layout but randomly warped building surfaces. The supplementary videos compare fly-through renderings of Ours and Vanilla 3DGS in all five cities.

Ablation study. To validate different components of our pipeline, we conduct an ablation study on London by adding its components one at a time (Table 4). *Base* trains 3DGS on camera poses and initial points using the bare-minimum setting (see Appendix A.8) without structural supervision, the consensus gate, or the session appearance adapter. Each row adds one component on top of the set-

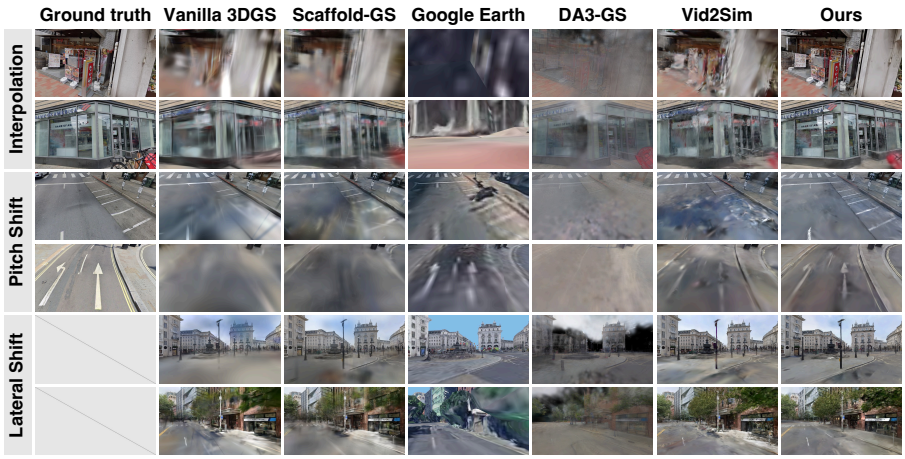


Figure 4: Qualitative comparison at novel views. Rows 1 and 2 are Interpolation views, rows 3 and 4 are Pitch Shift views, and rows 5 and 6 are Lateral Shift views, for which no photograph exists. The baselines blur storefronts and turn the road into foggy structures while ours keeps road markings, curbs, and storefronts sharp. Ours here is Ours (unconditioned).

Table 4: Ablation Study

	Interpolation				Pitch Shift				Lateral Shift			
Method	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	KID↓	FID↓	3DSE (m)↓	N_{GS} (M)↓
Base	18.83	0.687	0.355	1.425	17.39	0.620	0.703	1.104	0.158	189.1	1.406	3.42
+ consensus gate	18.48	0.678	0.372	1.348	17.60	0.622	0.707	1.055	0.141	181.0	1.328	1.39
+ appearance adapter	18.15	0.676	0.369	1.211	17.67	0.623	0.698	0.789	0.132	173.1	1.204	1.39
+ SfM depth	17.38	0.673	0.357	0.544	17.43	0.621	0.685	0.283	0.103	154.1	0.519	1.40
+ ray clearance	17.24	0.673	0.355	0.464	17.36	0.620	0.686	0.231	0.106	154.2	0.455	1.12
+ road grid	17.12	0.672	0.352	0.202	17.51	0.626	0.660	0.089	0.079	137.0	0.199	1.11

tings in the previous row. The last row is Ours (unconditioned). Besides the metrics used in Table 3, we also include the number of Gaussians N_{GS} to evaluate the efficiency of our method of reaching the same structure or quality with fewer Gaussians. From Table 4, as each of our components is added, the 3D structural error decreases and the Lateral Shift metrics improve overall, indicating that our method obtains stronger structure and more robust results in extrapolated views.

Densification control. We compare the consensus gate with the GDAGS rule (Zhou et al., 2026) (Appendix A.14). Removing the gate raises N_{GS} from 1.11 to 2.49 million and 3DSE by 44 % to 89 %, and the GDAGS rule still yields 2.20 million Gaussians and a 27 % to 52 % higher 3DSE.

SfM accuracy. Known camera intrinsics and the design of our SfM pipeline are essential for camera recovery. When GLOMAP estimates the focal length from the images alone, it scatters the twelve crops of one panorama over a huge area, with a diameter of 31 to 96 times the distance between neighboring panoramas. Using the known field of view forces them to stay within 0.2 of that distance (Appendix A.9). Inpainted crops also give the lowest median GPS residual of the camera positions, $0.15\,\mathrm{m}$ against $0.22\,\mathrm{m}$ for the raw and masked crops. The bundle adjustment with the panoramic constraint and the GPS prior lowers its 90th percentile from $0.89\,\mathrm{m}$ to $0.54\,\mathrm{m}$ (Appendix A.9).

Other experiments. We present additional results in the appendix, including but not limited to a GSV-like capture study on Mip-NeRF 360 (Appendix A.3), a detailed ablation of each component (Appendices A.13 to A.16), and the impact of multiple seeds (Appendix A.23).

5 Conclusion

We present SplatAnywhere, which reconstructs 3DGS models of street locations from publicly available Google Street View panoramas. GSV panoramas are sparse in camera center diversity and inconsistent across capture sessions, which lets the optimizer reduce photometric error by misplacing Gaussians and sacrificing geometry. SplatAnywhere constrains Gaussians with structural supervision, controls splitting with consensus-gated densification, and reconciles color changes through a session appearance adapter. On street scenes from five representative cities, SplatAnywhere lowers the 3D structural error by 26% to 89% relative to 3DGS baselines. SplatAnywhere will be open source.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing the methods and running experiments under the direction of the authors. We did not use generative AI tools to generate data for our experiments. We also used generative AI tools for polishing text for readability. The core ideas and innovations of the method and the structure of the paper are the authors' own. We have reviewed all AI-assisted work, the authors checked every reported result, and we take full responsibility for the final content.

ETHICS STATEMENT

We obtain the panoramas and their metadata from Google Street View through Google's Street View API endpoints and use them for research only. We do not release the panoramas or the crops derived from them. Google blurs faces and license plates in Street View, and our pipeline also removes vehicles, pedestrians, and bicycles from every crop before reconstruction.

REPRODUCIBILITY STATEMENT

We will release our code. Section 3 describes the pipeline, and Appendices A.2 and A.8 give the implementation details and the key hyperparameters. Appendices A.9 to A.11 describe structure from motion, the inpainting prompts, and the use of the Street View metadata. Section 4 and Appendix A.20 define the test views, the metrics, and the airborne LiDAR surveys. The baselines run their released code (Section 4), and Appendix A.22 describes the stabilized WildGaussians. Appendix A.23 repeats London with three training seeds.

REFERENCES

Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010. URL https://doi.org/10.1109/MC.2010.170.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469. IEEE, 2022. URL https://doi.org/10.1109/CVPR52688.2022.00539.

Berta Bescos, Cesar Cadena, and José Neira. Empty cities: A dynamic-object-invariant space for visual slam. *IEEE Transactions on Robotics*, 37(2):433–451, 2020. URL https://ieeexplore.ieee.org/abstract/document/9246713.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. URL https://arxiv.org/abs/2506.15742.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026. URL https://arxiv.org/abs/2511.16719.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467. IEEE, 2024. URL https://doi.org/10.1109/CVPR52733.2024.01840.

Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, et al. Geometric context transformer for streaming 3d reconstruction. *arXiv preprint arXiv:2604.14141*, 2026. URL https://arxiv.org/abs/2604.14141.

Xiao Cui, Weicai Ye, Yifan Wang, Guofeng Zhang, Wengang Zhou, Tong He, and Houqiang Li. Streetsurfgs: Scalable urban street surface reconstruction with planar-based gaussian splatting. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(9):8780–8793, 2025. URL https://ieeexplore.ieee.org/abstract/document/10929011.

Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it–pushing vggt's limits on kilometer-scale long rgb sequences. *arXiv preprint arXiv:2507.16443*, 2025. URL https://arxiv.org/abs/2507.16443.

Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12872–12881. IEEE, 2022. URL https://doi.org/10.1109/CVPR52688.2022.01254.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *2018 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 337–33712. IEEE, 2018. URL http://openaccess.thecvf.com/content_cvpr_2018_workshops/w9/html/DeTone_SuperPoint_Self-Supervised_Interest_CVPR_2018_paper.html.

Isaac Deutsch, Nicolas Moënne-Loccoz, Zan Gojcic, et al. Ppisp: Physically-plausible compensation and control of photometric variations in radiance field reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7289–7298,

2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Deutsch_PPISP_Physically-Plausible_Compensation_and_Control_of_Photometric_Variations_in_Radiance_CVPR_2026_paper.pdf.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017. URL http://proceedings.mlr.press/v78/dosovitskiy17a.html.

Environment Agency. National LIDAR Programme: LIDAR point cloud, survey P_12151 (tile TQ2580, december 2020). Defra Data Services Platform, https://environment.data.gov.uk/dataset/2e8d0733-4f43-48b4-9e51-631c25d1b0a9, 2020. Open Government Licence v3.0; accessed 2026-09-22.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/89e4433fec4b99f1d859db57af1e0a0f-Abstract-Conference.html.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 406–413, 2014. URL https://openaccess.thecvf.com/content_cvpr_2014/html/Jensen_Large_Scale_Multi-view_2014_CVPR_paper.html.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. URL https://doi.org/10.1145/3592433.

Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *arXiv preprint arXiv:2407.08447*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/25c0fe7b157821dd3140727dc07461da-Abstract-Conference.html.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20775–20785. IEEE, 2024. URL https://doi.org/10.1109/CVPR52733.2024.01963.

Yiqing Li, Xuan Wang, Jiawei Wu, Yikun Ma, and Zhi Jin. Sparsegs-w: Sparse-view gaussian splatting for unconstrained image collections with diffusion priors. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026. URL https://doi.org/10.1109/TCSVT.2026.3658313.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. URL https://arxiv.org/abs/2511.10647.

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. URL https://doi.org/10.1007/978-3-031-72640-8_15.

Yang Liu, Chuanchen Luo, Zimo Tang, Junran Peng, and Zhaoxiang Zhang. VGGT-X: When VGGT meets dense novel view synthesis. *arXiv preprint arXiv:2509.25191*, 2025. URL https://arxiv.org/abs/2509.25191.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D Gaussians for view-adaptive rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20654–20664. IEEE, 2024. URL https://ieeexplore.ieee.org/abstract/document/10658518.

Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 7206–7215. IEEE, 2021. URL https://ieeexplore.ieee.org/abstract/document/9578784.

Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019. URL https://doi.org/10.1145/3306346.3322980.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e6d58fc68c0f3c36ae6e0e64478a69c0-Abstract-Conference.html.

Chaojun Ni, Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Wenkang Qin, Guan Huang, Chen Liu, Yuyin Chen, Yida Wang, Xueyang Zhang, et al. Recondreamer: Crafting world models for driving scene reconstruction via online restoration. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1560–1569. IEEE, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ni_ReconDreamer_Crafting_World_Models_for_Driving_Scene_Reconstruction_via_Online_CVPR_2025_paper.html.

NVIDIA, Yulong Cao, Riccardo de Lutio, Sanja Fidler, Guillermo Garcia Cobo, Zan Gojcic, Maximilian Igl, Boris Ivanovic, Peter Karkus, Janick Martinez Esturo, Marco Pavone, Aaron Smith, Ellie Tanimura, Michal Tyszkiewicz, Michael Watson, Qi Wu, and Le Zhang. AlpaSim: A modular, lightweight, and data-driven research simulator for autonomous driving. Software, https://github.com/NVlabs/alpasim, October 2025.

OCM Partners. 2017 NYC topobathy lidar: New York City (digital coast dataset 9306). NOAA Office for Coastal Management, Digital Coast, https://www.fisheries.noaa.gov/inport/item/64728, 2017. Public domain (U.S. Government work); accessed 2026-09-22.

Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pp. 58–77. Springer, 2024. URL https://doi.org/10.1007/978-3-031-73661-2_4.

Ming Qian, Zimin Xia, Changkun Liu, Shuailei Ma, Wen Wang, Zeran Ke, Bin Tan, Hang Zhang, and Gui-Song Xia. Sat3dgen: Comprehensive street-level 3d scene generation from single satellite image. *arXiv preprint arXiv:2605.14984*, 2026. URL https://arxiv.org/abs/2605.14984.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4937–4946. IEEE, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Sarlin_SuperGlue_Learning_Feature_Matching_With_Graph_Neural_Networks_CVPR_2020_paper.html.

Katja Schwarz, Norman Müller, and Peter Kontschieder. Generative gaussian splatting: Generating 3d scenes with video diffusion priors. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27510–27520. IEEE, 2025. URL https://arxiv.org/abs/2503.13272.

Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pp. 835–846. 2006. URL https://doi.org/10.1145/1179352.1141964.

Larry J Sugarbaker, Eric W Constance, Hans Karl Heidemann, Allyson L Jason, Vicki Lukas, David L Saghy, and Jason M Stoker. The 3D Elevation Program initiative: a call for action. Circular 1399, US Geological Survey, 2014. URL https://pubs.usgs.gov/circ/1399/.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451. IEEE, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Sun_Scalability_in_Perception_for_Autonomous_Driving_Waymo_Open_Dataset_CVPR_2020_paper.html.

Tokyo Metropolitan Government. Tokyo Digital Twin Realization Project: 3d point cloud data for the 23 wards (airborne laser survey, 2023). G-Spatial Information Center, https://www.geospatial.jp/ckan/dataset/tokyopc-23ku-2024, 2024. CC BY 4.0; accessed 2026-09-22.

Khiem Vuong, Robert Tamburo, and Srinivasa G Narasimhan. Toward planet-wide traffic camera calibration. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 8538–8547. IEEE, 2024. URL https://ieeexplore.ieee.org/abstract/document/10484499.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

Jiawei Wang, Haowei Sun, Xintao Yan, Shuo Feng, Jun Gao, and Henry X Liu. Terasim-world: Worldwide safety-critical data synthesis for end-to-end autonomous driving. *arXiv preprint arXiv:2509.13164*, 2025b. URL https://arxiv.org/abs/2509.13164.

Jingkang Wang, Henry Che, Yun Chen, Ze Yang, Lily Goli, Sivabalan Manivasagam, and Raquel Urtasun. Flux4d: Flow-based unsupervised 4d reconstruction. *Advances in Neural Information Processing Systems*, 38:105673–105693, 2026a. URL https://arxiv.org/abs/2512.03210.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024a. URL https://doi.org/10.1109/CVPR52733.2024.01956.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2507.13347.

Yuehao Wang, Chaoyi Wang, Bingchen Gong, and Tianfan Xue. Bilateral guided radiance field processing. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024b. URL https://doi.org/10.1145/3658148.

Zhongtao Wang, Jiaqi Dai, Qingtian Zhu, Yilong Li, Mai Su, Fei Zhu, Meng Gai, Shaorong Wang, Chengwei Pan, Yisong Chen, et al. Chronogs: Disentangling invariants and changes in multi-period scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8298–8307, 2026c. URL https://arxiv.org/abs/2511.18794.

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*, 13(4):600–612, 2004. URL https://doi.org/10.1109/TIP.2003.819861.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*

(CVPR), pp. 26024–26035. IEEE, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_DIFIX3D_Improving_3D_Reconstructions_with_Single-Step_Diffusion_Models_CVPR_2025_paper.html.

Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE international intelligent transportation systems conference (ITSC)*, pp. 3095–3101. IEEE, 2021. URL https://arxiv.org/abs/2112.12610.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/64f1f27bf1b4ec22924fd0acb550c235-Abstract.html.

Ziyang Xie, Zhizheng Liu, Zhenghao Peng, Wayne Wu, and Bolei Zhou. Vid2sim: Realistic and interactive simulation from video for urban navigation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1581–1591. IEEE, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xie_Vid2Sim_Realistic_and_Interactive_Simulation_from_Video_for_Urban_Navigation_CVPR_2025_paper.html.

Haolin Xiong, Sairisheek Muttukuru, Hanyuan Xiao, Rishi Upadhyay, Pradyumna Chari, Yajie Zhao, and Achuta Kadambi. Sparsegs: Sparse view synthesis using 3d gaussian splatting. In *2025 International Conference on 3D Vision (3DV)*, pp. 1032–1041, 2025. URL https://doi.org/10.1109/3DV66043.2025.00100.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pp. 156–173. Springer, 2024. URL https://arxiv.org/abs/2401.01339.

Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, et al. Streetcrafter: Street view synthesis with controllable video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 822–832. IEEE, 2025. URL https://arxiv.org/abs/2412.13188.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1389–1399. IEEE, 2023. URL https://doi.org/10.1109/CVPR52729.2023.00140.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 1053–1061, 2024. URL https://doi.org/10.1145/3664647.3681361.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19447–19456. IEEE, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yu_Mip-Splatting_Alias-free_3D_Gaussian_Splatting_CVPR_2024_paper.html.

Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. In *European Conference on Computer Vision*, pp. 341–359. Springer, 2024. URL https://doi.org/10.1007/978-3-031-73116-7_20.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Zheng Zhou, Yu-Jie Xiong, Jia-Chen Zhang, Chun-Ming Xia, Xihe Qiu, and Hongjian Zhan. Gradient-direction-aware density control for 3d gaussian splatting. In *International Conference on Learning Representations*, volume 2026, pp. 21415–21437, 2026. URL https://openreview.net/forum?id=6qDxK4Gz7F.

A Appendix

A.1 Limitations

Our pipeline has several limitations. First, our transient-object removal cannot guarantee complete removal or perfect background restoration. SAM 3 occasionally misses partially occluded objects, and FLUX fills masked regions with plausible rather than observed content. The generated hallucinations or unclean final crops may limit the final performance. Second, the reconstruction inherits the limits of its data source. GSV covers public roads only, its panoramas cap the achievable fidelity below that of dedicated capture, and access is subject to the provider's terms. Finally, we reconstruct street-level scenes around a target coordinate, and scaling to district-level extents increases computation that our neighbor-restricted matching reduces but does not eliminate.

A.2 Experimental Setup Details

- **Images.** All training crops and test views are 912×512 pixels with a 90° horizontal field of view.
- **Structure from motion.** GLOMAP (Pan et al., 2024) with SuperPoint (DeTone et al., 2018) features and SuperGlue (Sarlin et al., 2020) matching (Section 3.3 and Appendix A.9).
- **Training.** Every model trains for 30 k iterations, and ours use seed 42. Our method and its ablations were trained on an NVIDIA RTX 5090 or an RTX 6000 Ada GPU.
- **Baseline frames.** Each baseline reconstruction lies in the frame of its own GLOMAP solve, and we align it to our frame with one similarity transform per city, fitted on the camera centers that both solves share (median residual 0.022 to $0.046\,\mathrm{m}$).

Metrics.

- **PSNR, SSIM, and LPIPS.** Computed on the static pixels of the Interpolation and Pitch Shift views, with transient objects masked by SAM 3. Pitch Shift uses only the views that look down.
- **KID and FID.** Lateral Shift renders one view per panorama, and KID and FID compare these renderings with the ground-truth photographs of the Interpolation views of the same scene. For methods whose colors depend on the capture, the scores are averaged over the captured sessions.
- **3D structural error (3DSE).** For every survey point in a view, the depth difference along the ray is projected onto the survey normal. A view scores the median over at least 200 such points, a city the median over its views, and the five-city value the mean over the cities. The surveys are clipped to a disc of $80\,\mathrm{m}$ radius and shifted by the frame offset of Appendix A.11.
- **Five-city values.** Every five-city number in the tables is the mean of the five city values.

A.3 Controlled Study of One-Sided Capture

To test whether test views taken along the capture trajectory can hide geometric error, we run a controlled experiment on the bicycle scene of Mip-NeRF 360 (Barron et al., 2022), which has test photographs all around the scene. We train two models with standard 3DGS and 40 views each. The *wedge* model uses 40 views taken from one side of the scene, within a 79° wedge of azimuth, and the *all-round* model uses 40 views spread around the scene. We score both models on the 25 standard test photographs and on the 98 other photographs that neither model used for training. Only 8 of them lie within the wedge, since the remaining photographs there are training views of the wedge model. To exclude content that the wedge model never observed, Figure 5(d) scores only image patches around 3D points that both training sets observed and keeps views with at least 50 such points.

Within the wedge, the wedge model looks better than the all-round model, with $2.8\,\mathrm{dB}$ higher PSNR on the full images (22.32 against $19.52\,\mathrm{dB}$) and $4.7\,\mathrm{dB}$ higher PSNR on the shared content. Beyond the wedge, the same model renders the observed content $6.2\,\mathrm{dB}$ worse than within it, while the all-round model stays level. The wedge model still leads 10° beyond its wedge but falls behind from about 20°. Over three seeds, the difference between the two drops ranges from 5.6 to $6.6\,\mathrm{dB}$, and it ranges from 7.4 to $8.2\,\mathrm{dB}$ on the 25 standard photographs alone. Figure 5(e) shows why. Within the wedge, both models reproduce the bicycle and the bench. At 56° beyond the wedge, the bench and the grass of the wedge model are smeared, and at 71° the boundary between the path and the grass is lost. The all-round model renders both views cleanly. Test views taken along the capture trajectory therefore reward a model that fits the trajectory, even when its geometry fails from other viewpoints. GSV capture resembles the wedge case (Figure 5a to c). Each surface point is seen from one side of

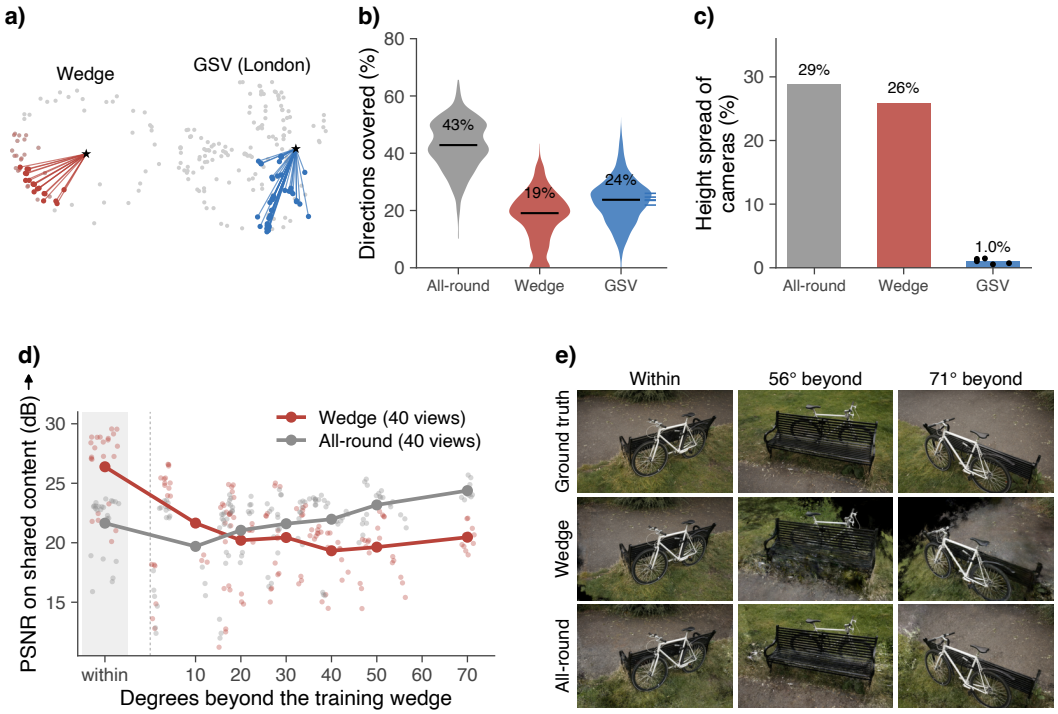


Figure 5: Controlled study on the bicycle scene with 40 training views per model and its similarity to GSV capture. (a) One surface point near the scene center and the training cameras whose SfM track contains it, seen from above, for the wedge and for London, where 42 panoramas observed the point. Both sets of cameras lie on one side of the point. (b) Fraction of the viewing directions around each SfM point that a training camera covers within $15°$. The ticks mark the five GSV scenes. (c) Spread of the camera centers along their third principal axis relative to their first. (d) PSNR on content that both training sets observed, against the angle beyond the training wedge. Dots show single views of three seeds, and points show bin means. (e) Test views within the wedge and $56°$ and $71°$ beyond it. The wedge model is better within its wedge and worse beyond it, while the all-round model changes little. Black regions are content that the wedge model never observed.

the street, the training cameras cover 24 % of the viewing directions around a point against 19 % for the wedge and 43 % for the all-round set, and the GSV cameras lie at one height, so their vertical spread is about 1 % of their horizontal spread against 26 % for the wedge. For this reason, Table 3 includes Pitch Shift and Lateral Shift and reports 3D structural error against LiDAR.

A.4 Held-Out Panoramas

Interpolation and Pitch Shift keep the camera centers of training panoramas. To test camera positions that no method was trained on, we hold out every eighth London panorama, 24 in total, and train every method without their crops (Table 5). Their camera poses still come from structure from motion over all panoramas, for our method and for the baselines. On the 287 views of these panoramas, our 3D structural error is $0.208\,\mathrm{m}$, 2.6 to 6 times better than that of every baseline, our LPIPS is the lowest, 0.412 against at least 0.433, and Ours (conditioned) has the highest PSNR, 15.56 dB. Figure 6 shows four of these views, drawn at random. The baselines blur the road and cover it with haze, while ours keeps the road markings and curbs.

A.5 Baselines with Our SfM and Inpainted Crops

In Table 3, every baseline trains on crops with transient objects masked by SAM 3 and on camera poses from GLOMAP. To test whether our lead in geometry comes from our SfM and inpainting, we retrain every 3DGS baseline with its settings in Table 3 on our inpainted crops, with the camera poses and initial points of our SfM, and score it as in Table 3 (Table 6). With our SfM and inpainted crops, every baseline keeps a 3D structural error 1.6 to 9 times ours on the five-city means. The geometric gain of our method therefore comes from its training rather than from its SfM or inpainting.

Table 5: Held-Out Panoramas on London

Method	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓
Vanilla 3DGS (Kerbl et al., 2023)	14.65	0.627	0.483	1.259
Scaffold-GS (Lu et al., 2024)	15.23	0.645	0.436	0.694
WildGaussians* (Kulhanek et al., 2024)	15.38	0.608	0.525	0.774
Vid2Sim (Xie et al., 2025)	15.25	0.633	0.433	0.620
ChronoGS (Wang et al., 2026c)	15.13	0.623	0.484	0.543
Ours (unconditioned)	15.33	0.611	0.412	0.208
Ours (conditioned)	15.56	0.611	0.412	0.208



Figure 6: Held-out panoramas on London. Four views drawn at random, each from a different panorama that no method was trained on. Ours is rendered in the session of each view.

Table 6: 3D Structural Error of the Baselines with GLOMAP on Masked Crops and with Our SfM on Inpainted Crops, Averaged over Five Cities

			3DSE (m)↓		
Method	SfM	Crops	Interpolation	Pitch Shift	Lateral Shift
Vanilla 3DGS (Kerbl et al., 2023)	GLOMAP	Masked	0.979	0.670	0.951
	Ours	Inpainted	0.924	0.564	0.874
Scaffold-GS (Lu et al., 2024)	GLOMAP	Masked	0.648	0.600	0.634
	Ours	Inpainted	0.696	0.460	0.673
WildGaussians* (Kulhanek et al., 2024)	GLOMAP	Masked	0.886	1.085	0.750
	Ours	Inpainted	0.778	1.105	0.737
Vid2Sim (Xie et al., 2025)	GLOMAP	Masked	0.305	0.229	0.297
	Ours	Inpainted	0.350	0.207	0.345
ChronoGS (Wang et al., 2026c)	GLOMAP	Masked	0.567	1.056	0.569
	Ours	Inpainted	0.631	0.642	0.628
Ours	Ours	Inpainted	0.209	0.123	0.220

*Stabilized variant of WildGaussians.

A.6 3D Structural Error Breakdown

Table 7 splits the 3D structural error of Table 3 by the class of each survey point. Ground is ground other than the road, and Poles also covers trees. Pitch Shift reports only the two ground classes, because downward views contain almost no walls, poles, or trees. Road and other ground make up 86 % of the points on Interpolation and 99 % on Pitch Shift, and among the trained methods ours has the lowest error on every reported class in every view group, including building walls, where it reaches 0.573 against $0.692\,\mathrm{m}$ for Vid2Sim on Interpolation. Table 8 shows that ours also has the lowest 90th percentile of the per-view error among the trained methods, 0.419, 0.302, and $0.436\,\mathrm{m}$ on the three view groups, and that ours and the five 3DGS baselines each score at least 98.5 % of the survey points visible from their views.

Table 7: 3D Structural Error by Surface Class Averaged over Five Cities

	Interpolation				Pitch Shift		Lateral Shift			
Method	Road↓	Ground↓	Walls↓	Poles↓	Road↓	Ground↓	Road↓	Ground↓	Walls↓	Poles↓
Vanilla 3DGS (Kerbl et al., 2023)	0.948	1.043	3.309	2.251	0.705	0.668	0.896	0.997	2.832	2.138
Scaffold-GS (Lu et al., 2024)	0.598	0.631	1.493	1.739	0.594	0.504	0.573	0.630	1.367	1.761
WildGaussians* (Kulhanek et al., 2024)	0.672	4.188	0.865	2.576	0.859	6.544	0.650	2.585	0.826	2.219
Vid2Sim (Xie et al., 2025)	0.277	0.327	0.692	1.619	0.224	0.264	0.275	0.318	0.662	1.530
ChronoGS (Wang et al., 2026c)	0.479	0.448	1.347	1.829	0.903	0.758	0.490	0.481	1.343	1.778
Google Earth	0.100	0.166	0.493	1.417	0.072	0.103	0.095	0.157	0.465	1.363
VGGT-X (Liu et al., 2025)	1.100	1.145	4.330	3.010	0.927	0.892	1.064	1.090	4.390	2.897
DA3-GS (Lin et al., 2025)	0.427	0.978	19.081	8.517	1.756	1.608	0.416	0.948	19.215	7.882
Ours	0.187	0.312	0.573	1.294	0.137	0.259	0.193	0.304	0.575	1.251
Share of points (%)	56	30	5	4	73	26	58	27	5	4

Table 8: Tail, Coverage, and Vertical Offset of the 3D Structural Error

	Interpolation				Pitch Shift				Lateral Shift			
Method	P90↓	Cov.↑	Before↓	After↓	P90↓	Cov.↑	Before↓	After↓	P90↓	Cov.↑	Before↓	After↓
Vanilla 3DGS (Kerbl et al., 2023)	1.679	99.6	0.765	0.979	1.077	99.4	0.511	0.670	1.605	99.8	0.748	0.951
Scaffold-GS (Lu et al., 2024)	1.197	99.3	0.570	0.648	1.741	99.5	0.692	0.600	1.111	99.5	0.561	0.634
WildGaussians* (Kulhanek et al., 2024)	13.966	100.0	0.745	0.886	48.132	100.0	1.066	1.085	11.258	100.0	0.615	0.750
Vid2Sim (Xie et al., 2025)	0.591	99.6	0.289	0.305	0.454	99.1	0.234	0.229	0.539	99.8	0.282	0.297
ChronoGS (Wang et al., 2026c)	1.251	98.7	0.601	0.567	2.403	98.6	1.302	1.056	1.137	99.0	0.589	0.569
Google Earth	0.474	99.0	0.236	0.122	0.198	99.4	0.243	0.077	0.611	99.1	0.243	0.118
VGGT-X (Liu et al., 2025)	2.297	79.6	-	1.201	3.646	81.3	-	0.931	2.437	80.1	-	1.217
DA3-GS (Lin et al., 2025)	9.693	93.0	1.308	1.009	10.437	95.7	2.263	2.455	6.165	93.8	1.150	0.972
Ours	0.419	99.9	0.215	0.209	0.302	99.8	0.224	0.123	0.436	99.9	0.211	0.220

A.7 Refinement with Difix3D+

We refine the renderings of every trained method with Difix3D+ (Wu et al., 2025), applied in the same way to all methods, and report LPIPS in Table 9. After refinement, our method has the best photometric results in most cities and settings. Averaged over the five cities, refinement lowers our LPIPS from 0.382 to 0.316 on Interpolation and from 0.691 to 0.553 on Pitch Shift, against 0.330 and 0.578 for Vid2Sim after refinement.

A.8 Key Hyperparameters

All scenes share the settings below.

- **SfM depth.** Weight $\lambda_{\text{sfm}} = 0.5$. A point is kept with a track of at least 5 crops, a reprojection error of at most 1.5 px, and a triangulation angle of at least 3°. Its weight is $\omega_m = 0.3 + 0.7\,\hat{\omega}_m / \max_{m'} \hat{\omega}_{m'}$ with $\hat{\omega}_m = \min(\ell_m/10, 1)\,\min(\alpha_m/10°, 1)\,\min(1.5/(e_m + 0.5), 1)$ for track length ℓ_m, triangulation angle α_m, and reprojection error e_m in pixels. The denominator of the relative error in Eq. (3) is clamped at 1 m. A point enters $\mathcal{L}_{\text{sfm}}$ only where the accumulated opacity $A^{(u_m)}$ exceeds 0.7, and the Huber threshold is $\delta = 0.05$.
- **Road grid.** Road surface weight $\lambda_r = 0.5$ on $\mathcal{L}_{\text{road}}$, which gives effective weights of $\lambda_r\lambda_z = 0.05$ for the height term and $\lambda_r\lambda_n = 0.01$ for the normal term. The height term acts on Gaussians within 0.75 m of a cell height with a Huber transition of 0.10 m, and the normal term gives lower weight to pixels on strong image edges. The cells are 4 m wide. An SfM point counts as road when at least 60 % of its at least three observations fall inside the road masks, and a cell takes the median height of at least six such points. Cells without enough road points take a fallback height from the SfM points below the cameras.
- **Ray clearance.** Weight $\lambda_c = 0.2$ with tolerance $\kappa = 1.5$. Pixels whose reference lies within 2 m of the camera are skipped. The clearance loss and the discarding of new Gaussians apply to the pixels that SegFormer (Xie et al., 2021) labels with the ADE20K tree, palm, plant, building, or sky class. The reference $d_{\text{ref}}^{(u)}$ comes from the DA3 depth of the crop, brought to the scale of the rendered depth by an affine fit in inverse depth over the pixels with accumulated opacity above 0.9 and refit on the three quarters of them with the smallest residuals.
- **Session appearance adapter.** Gaussian color code of rank $q = 4$ and penalty weight $\lambda_{\text{reg}} = 10$ on both of its parts. The penalty $\mathcal{L}_{\text{reg}}$ averages the squared deviations $(a_\tau - 1, b_\tau, \boldsymbol{\beta}_\tau - \mathbf{1})$ over all sessions and the squared color shifts $\mathbf{B}_\tau \mathbf{z}_i$ over all Gaussians for the session of the current view.

Table 9: LPIPS↓ after Refining Every Trained Method with Difix3D+

	Interpolation						Pitch Shift					
Method	London	Tokyo	Seattle	San Francisco	New York	Mean	London	Tokyo	Seattle	San Francisco	New York	Mean
Vanilla 3DGS	0.333	0.368	0.396	0.343	0.371	0.362	0.614	0.625	0.600	0.585	0.634	0.611
Scaffold-GS	0.316	0.377	0.410	0.351	0.382	0.367	0.605	0.636	0.602	0.594	0.646	0.617
WildGaussians*	0.361	0.465	0.490	0.442	0.461	0.444	0.630	0.714	0.704	0.694	0.724	0.693
Vid2Sim	0.306	0.337	0.357	0.326	0.324	0.330	0.589	0.586	0.576	0.560	0.579	0.578
ChronoGS	0.374	0.411	0.434	0.439	0.429	0.418	0.647	0.659	0.654	0.672	0.664	0.659
Ours (unconditioned)	0.295	0.338	0.321	0.334	0.323	0.322	0.547	0.584	0.529	0.554	0.569	0.557
Ours (conditioned)	0.290	0.333	0.316	0.326	0.316	0.316	0.545	0.582	0.525	0.549	0.566	0.553

- **Densification.** A new Gaussian is discarded when at least 60 % of its projections into 12 sampled training crops, with at least three in frame, fall on vegetation, building, or sky and no initial point lies within 1 m. The ratio C_i of Section 3.5 uses the gradient of the total loss, accumulated over the 100 iterations between densification steps, and a Gaussian that receives no gradient in this window gets $C_i = 0$.
- **Training.** 30 k iterations with the 3D smoothing filter of Mip-Splatting (Yu et al., 2024) and without a scale regularizer. The photometric loss is $\mathcal{L}_{\text{rgb}} = (1 - \lambda_{\text{ssim}})\,\mathcal{L}_1 + \lambda_{\text{ssim}}\,(1 - \text{SSIM})$ as in 3DGS.
- **Bundle adjustment.** GPS prior weight $\lambda_{\text{gps}} = 1$ with a Huber transition of 3 m and a vertical weight $w_z = 0.3$, and a Huber threshold of 2 px on the reprojection error.

A.9 Structure from Motion

Crop versions. Tables 11 and 12 test how inpainting and masking the crops affect camera recovery. We run the same matcher, pairing graph, and bundle adjustment on three versions of the crops.

- **Inpainted crops** have transient objects filled in by FLUX.
- **Raw crops** are the original perspective crops.
- **Masked crops** are raw crops whose keypoints inside the dilated SAM 3 vehicle masks are discarded.

Table 11 reports the initial GLOMAP solve, in which every crop is still an independent camera, and Table 12 reports the result after the bundle adjustment with the panoramic constraint and the GPS prior and after retriangulation. The reprojection error is computed on the same points in every version, namely the points whose observations all lie outside the vehicle masks. The GPS residual is the median over panoramas of the distance between the recovered camera center and the GPS position after a similarity alignment, with the 90th percentile in parentheses. San Francisco and New York count only the panoramas shared by all three solves, 137 of 142 and 136 of 139.

Table 10 summarizes the GPS residual of the camera positions of both stages for the three crop versions.

Table 10: GPS Residual of the Camera Positions after Each Structure-from-Motion Stage

Residual (m)↓	Inpainted crops		Raw crops		Masked crops	
City	GLOMAP	+ Pano., GPS	GLOMAP	+ Pano., GPS	GLOMAP	+ Pano., GPS
London	0.16 (0.89)	0.14 (0.65)	0.32 (0.59)	0.28 (0.52)	0.24 (1.26)	0.30 (0.62)
Seattle	0.13 (1.40)	0.17 (0.69)	0.32 (0.68)	0.12 (0.32)	0.10 (0.37)	0.10 (0.28)
San Francisco	0.19 (0.82)	0.18 (0.48)	0.85 (6.49)	0.40 (2.89)	0.33 (1.04)	0.21 (0.42)
New York	0.20 (1.21)	0.20 (0.69)	0.22 (1.10)	0.24 (0.48)	0.17 (1.02)	0.21 (0.42)
Tokyo	0.09 (0.19)	0.08 (0.16)	0.07 (0.13)	0.06 (0.12)	0.36 (0.63)	0.19 (0.33)
Overall	0.15 (0.89)	0.15 (0.54)	0.29 (1.13)	0.22 (0.58)	0.24 (0.89)	0.22 (0.47)

Nearly every crop is registered with each crop version (8,190 of 8,196 with inpainted crops), and the reprojection errors differ by less than 0.06 px, so the GPS residual separates the three versions. Inpainted crops give the lowest median residual, 0.15 m against 0.22 m for raw and masked crops after the bundle adjustment with the panoramic constraint and the GPS prior, about a third lower. Masking keypoints removes 21 % to 33 % of the sparse points of the raw initial solve, which suggests that masking mainly loses useful features rather than removing wrong matches. The bundle

adjustment also corrects badly placed cameras and lowers the 90th percentile of the inpainted crops from 0.89 to 0.54 m.

Table 11: Camera Recovery after the Initial GLOMAP Solve for Three Crop Versions

		Registered crops↑			Reprojection error (px)↓			GPS residual (m)↓		
City	Crops	Inpainted	Raw	Masked	Inpainted	Raw	Masked	Inpainted	Raw	Masked
London	2,304	2,301	2,304	2,301	1.43±0.64	1.45±0.66	1.46±0.66	0.16 (0.89)	0.32 (0.59)	0.24 (1.26)
Seattle	1,152	1,151	1,152	1,151	1.39±0.61	1.40±0.63	1.38±0.61	0.13 (1.40)	0.32 (0.68)	0.10 (0.37)
San Francisco	1,644	1,644	1,644	1,644	1.48±0.65	1.48±0.65	1.47±0.62	0.19 (0.82)	0.85 (6.49)	0.33 (1.04)
New York	1,632	1,632	1,632	1,631	1.41±0.65	1.41±0.65	1.41±0.65	0.20 (1.21)	0.22 (1.10)	0.17 (1.02)
Tokyo	1,464	1,462	1,463	1,463	1.40±0.61	1.43±0.61	1.42±0.61	0.09 (0.19)	0.07 (0.13)	0.36 (0.63)
Overall	8,196	8,190	8,195	8,190	1.42±0.64	1.44±0.64	1.43±0.63	0.15 (0.89)	0.29 (1.13)	0.24 (0.89)

Table 12: Camera Recovery after Bundle Adjustment with the Panoramic Constraint and Retriangulation for Three Crop Versions

		Registered crops↑			Reprojection error (px)↓			GPS residual (m)↓		
City	Crops	Inpainted	Raw	Masked	Inpainted	Raw	Masked	Inpainted	Raw	Masked
London	2,304	2,301	2,304	2,301	1.53±0.65	1.55±0.65	1.54±0.64	0.14 (0.65)	0.28 (0.52)	0.30 (0.62)
Seattle	1,152	1,151	1,152	1,151	1.41±0.66	1.42±0.66	1.41±0.66	0.17 (0.69)	0.12 (0.32)	0.10 (0.28)
San Francisco	1,644	1,644	1,644	1,644	1.49±0.67	1.49±0.67	1.49±0.66	0.18 (0.48)	0.40 (2.89)	0.21 (0.42)
New York	1,632	1,632	1,632	1,631	1.44±0.66	1.44±0.67	1.44±0.66	0.20 (0.69)	0.24 (0.48)	0.21 (0.42)
Tokyo	1,464	1,462	1,463	1,463	1.42±0.64	1.43±0.63	1.42±0.63	0.08 (0.16)	0.06 (0.12)	0.19 (0.33)
Overall	8,196	8,190	8,195	8,190	1.46±0.66	1.47±0.66	1.47±0.66	0.15 (0.54)	0.22 (0.58)	0.22 (0.47)

Without metadata. We also run GLOMAP as released on the image content alone, with SIFT features, exhaustive matching, and self-calibrated intrinsics, so that no GPS position, neighbor list, or known field of view enters (Table 13). GLOMAP registers 82 to 100 % of the crops, so the registration counts would pass, but the geometry is wrong in every city. The twelve crops of one panorama, which share one camera center, end up 31 to 96 panorama spacings apart, the self-calibrated focal length is off by 21 to 484 % in four of the five cities, and the reprojection error stays small at 0.5 to 2.1 px. Given only the known field of view of the crops, GLOMAP is sound in all five cities, with the crops of each panorama within 0.2 spacings of each other. The failure therefore comes from self-calibration on crops that share one center and differ only in heading, and every baseline receives the crop intrinsics.

Pose constraints. The bundle adjustment minimizes Eq. (2) of Section 3.3. Table 14 adds the pose constraints one at a time on London. A solve is hard to judge on its own, so we train plain 3DGS with the photometric loss only on each set of poses and score it on the interpolation and Pitch Shift views of Table 3. Indented rows replace or extend the row above them. The initial GLOMAP solve reaches 14.41 dB. Removing two badly placed cameras recovers part of the loss (17.93 dB), and the panoramic constraint, which ties the twelve crops of a panorama to one camera center with known headings, lifts it to 19.73 dB. The GPS prior, the elevation prior, and retriangulation each change the Interpolation PSNR by less than 0.1 dB. Their main purpose is to place the scene in metric, georeferenced coordinates. This plain model fits the training crops with misplaced geometry, with a 3D structural error of 1.513 m against 0.202 m for our full model.

A.10 Inpainting Prompts

We fill the transient-object masks with FLUX.1 Fill using a prompt that describes the road surface, “weathered asphalt road with white lane markings, concrete sidewalk with curb, manhole covers, drain grates” (D). To check how often objects survive the fill, we regenerate 300 sampled crops, 60 per city, with the same masks and seed under three simpler prompts, an empty prompt (A), “empty street” (B), and a prompt that also asks for no vehicles and no people (C). We then detect vehicles, pedestrians, and bicycles inside the filled region with SAM 3 and with OWLv2 (Minderer et al., 2023), an independent open-vocabulary detector (Table 15 and Figure 7). The table gives the share of filled crops with a detection inside the fill (Res), the share of fill pixels that detections cover (Area), the vehicles, pedestrians, and bicycles per 100 filled crops, and the share of crops with a detection outside the mask (Out). Our prompt leaves objects in 9.7% of the crops by SAM 3

Table 13: GLOMAP without Metadata on the Raw Crops

City	Crops	Registered crops↑		Self-calibrated	Panorama spread (p90)↓	
		Image only	+ Intrinsics	focal f_x / f_y (true 456)	Image only	+ Intrinsics
London	2,304	2,298	2,302	553 / 529	53.5	0.185
Tokyo	1,464	1,419	1,464	1493 / 1776	95.7	0.118
Seattle	1,152	947	972	715 / 2352	48.1	0.076
San Francisco	1,644	1,643	1,644	597 / 2663	70.6	0.095
New York	1,632	1,600	1,620	468 / 475	31.0	0.168

Table 14: Effect of Each Pose Constraint on London Measured with Plain 3DGS

Pose constraints	Interpolation			Pitch Shift	
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	LPIPS↓
Initial GLOMAP solve	14.41	0.522	0.707	16.65	0.715
outlier cameras removed	17.93	0.633	0.435	16.98	0.770
+ panoramic constraint	19.73	0.695	0.364	17.72	0.716
+ soft GPS prior	19.77	0.697	0.361	17.77	0.712
hard GPS constraint instead	19.70	0.696	0.359	17.80	0.714
one-step optimization instead	19.77	0.696	0.363	17.74	0.712
followed by retriangulation	19.75	0.695	0.363	17.56	0.713
+ elevation prior	19.81	0.707	0.350	17.76	0.709
+ retriangulation (ours)	19.82	0.707	0.350	17.71	0.710

and 12.3% by OWLv2, against 53.3% and 46.3% for the empty prompt and 41.0% and 35.3% for "empty street" ($p < 0.001$ for both detectors, McNemar test with Holm correction). The prompt with explicit negatives leaves objects in 12.7 % and 15.0 %. Over all 8292 training crops, our prompt leaves objects in 8.6% (SAM 3) and 10.5% (OWLv2) of the 7860 crops with a fill.

A.11 Street View Metadata

Street View supplies a position, an altitude, and a capture date for every panorama. We use the positions and altitudes only as a soft prior in structure from motion (Section 3.3), because they carry errors at three scales (Table 16).

Individual panoramas scatter little. After a similarity alignment, the reconstructed camera centers lie within a median of 0.05 to $0.12\,\mathrm{m}$ of the metadata positions, and the altitudes scatter around the reconstructed camera heights by a median of 0.016 to $0.030\,\mathrm{m}$ in four cities and $0.076\,\mathrm{m}$ in Tokyo.

Whole sessions can be offset. In New York, the altitudes of the session from May 2019 lie $1.64\,\mathrm{m}$ below those of the other sessions at the same places, while structure from motion keeps this session within $0.10\,\mathrm{m}$ of the others. In London and San Francisco, one recent session lies about $0.5\,\mathrm{m}$ lower in both the metadata and the reconstruction, which points to a lower camera rather than an altitude error. Fifteen Tokyo panoramas captured in a road tunnel carry altitudes about $6\,\mathrm{m}$ below their neighbors, and we exclude them.

The altitudes of a whole city can be biased against the national surveys. Because the reconstructions are georeferenced through these altitudes, the reconstructed ground lies 0.07 to $0.55\,\mathrm{m}$ below the airborne survey (New York after its frame shift in Appendix A.20), most in Tokyo. We measure this offset without any trained model, as the peak of the height difference between the SfM points and the survey ground. The survey ground is the median height of the returns that the survey classifies as ground, per $0.5\,\mathrm{m}$ cell. We take the SfM points above cells that contain ground returns, and the offset is the peak of the histogram of their height above the survey ground within $\pm 1.2\,\mathrm{m}$. Once the offset is applied, the panorama camera centers sit a median 2.46 to $2.53\,\mathrm{m}$ above the survey ground in London, San Francisco, Seattle, and New York, within $0.04\,\mathrm{m}$ of the nominal Street View mounting height of about $2.5\,\mathrm{m}$, and $1.92\,\mathrm{m}$ in Tokyo, consistent with the lower camera mount reported for Japan. The offsets are therefore consistent with the camera geometry to within the uncertainty of the nominal height, about $0.1\,\mathrm{m}$.In Tokyo, the altitudes place the cameras only $1.35\,\mathrm{m}$ above the survey ground, which is not the height of a Street View camera, while the survey agrees with an independent national elevation model. The offset therefore lies in the altitudes and not in the survey.

Table 15: Inpainting Prompt Residual Detection Rates

	SAM 3						OWLv2					
	Res↓	Area↓	Veh↓	Ped↓	Bic↓	Out	Res↓	Area↓	Veh↓	Ped	Bic↓	Out
(R) Raw crop	99.0	63.17	475.0	384.7	43.7	0.7	73.7	39.64	203.3	0.0	49.3	5.3
(A) Empty string	53.3	20.21	136.3	52.7	5.3	4.3	46.3	19.59	155.0	4.0	14.0	18.7
(B) "empty street"	41.0	12.99	84.0	32.7	3.0	6.3	35.3	13.17	93.0	0.0	10.7	19.0
(C) Explicit negative	12.7	2.49	17.3	1.7	0.7	4.0	15.0	2.31	24.0	0.0	4.0	16.7
(D) Ours	9.7	0.96	9.7	2.3	1.3	5.7	12.3	0.80	17.0	0.0	3.3	15.3
(D) Ours, all crops	8.6	0.76	7.7	3.7	0.7	4.1	10.5	0.76	10.9	0.0	5.5	13.8

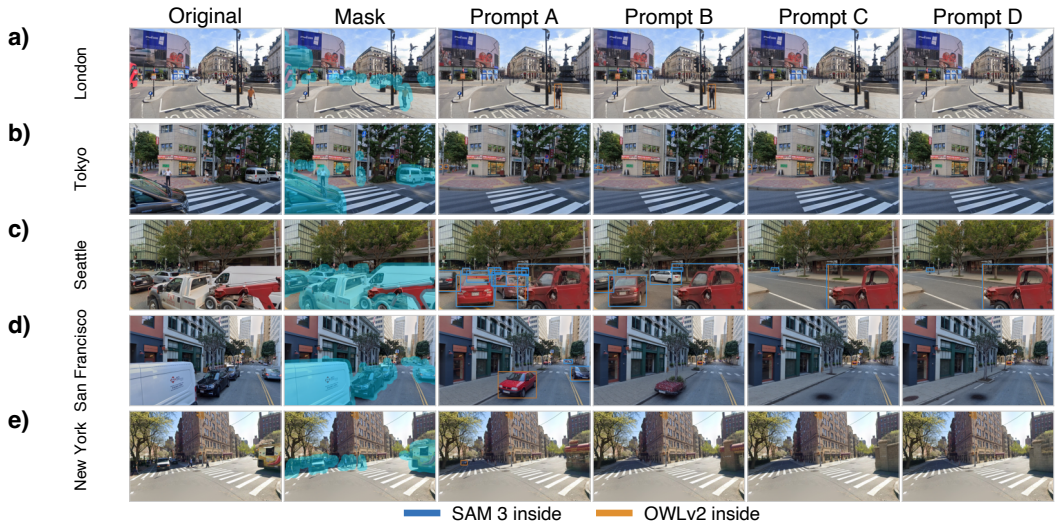


Figure 7: Inpainting prompt residuals. One crop per city, drawn at random from the sample crops in which a detector finds an object inside the fill under at least one of prompts A to C. The columns show the original crop, the mask, and the fills under prompts A to D of Table 15, and boxes mark the detections inside the fill.

The 3D structural error in Section 4 is computed after shifting each survey by this offset. The offset is the same for all methods, because the baselines are aligned to our frame.

A.12 Auxiliary Road Terms

Besides $\mathcal{L}_{\text{surf}}$ in Eq. (4), the road loss has a height term and a normal term,

$$\mathcal{L}_z = \underset{i \in \mathcal{G}_{\text{road}}}{\text{mean}} \, \rho_{\delta_z}\big(z_i - H(\boldsymbol{\mu}_i)\big), \qquad \mathcal{L}_n = \underset{u \in \mathcal{U}_{\text{road}}}{\text{mean}} \, \gamma^{(u)} \big\| \hat{\mathbf{n}}^{(u)} - \mathbf{n}_{\text{d}}^{(u)} \big\|_1. \tag{8}$$

The height term $\mathcal{L}_z$ pulls the Gaussians $\mathcal{G}_{\text{road}}$ whose center lies above a road cell and within $0.75\,\text{m}$ of its height toward the road surface, where z_i is the height of the center $\boldsymbol{\mu}_i$, $H(\boldsymbol{\mu}_i)$ is the height of the cell below it, and ρ_{δ_z} is the Huber function with $\delta_z = 0.1\,\text{m}$. Because it acts on the Gaussians directly, it also reaches road Gaussians that no training ray covers. The normal term $\mathcal{L}_n$ runs over the road-mask pixels $\mathcal{U}_{\text{road}}$ with an accumulated opacity above 0.7. It compares the rendered normal $\hat{\mathbf{n}}^{(u)}$, composited like the color from the shortest axis of each Gaussian, with the normal $\mathbf{n}_{\text{d}}^{(u)}$ of the rendered depth, which is held fixed, and $\gamma^{(u)}$ lowers the weight of pixels on strong image edges. With $\lambda_r = 0.5$, the effective weights are 0.05 for $\mathcal{L}_z$ and 0.01 for $\mathcal{L}_n$, against 0.5 for $\mathcal{L}_{\text{surf}}$.

A.13 Structural Supervision Breakdown

In the ladder of Table 4, the structural terms together lower Interpolation 3D structural error the most, from 1.211 to $0.202\,\text{m}$. SfM depth lowers it to $0.544\,\text{m}$. Ray clearance lowers it to $0.464\,\text{m}$ and reduces N_{GS} from 1.40 to 1.12 million. The road grid lowers it to $0.202\,\text{m}$, and Pitch Shift 3D

Table 16: Errors of the Street View Metadata in Each City

City	Position residual (m)	Altitude scatter (m)	Largest session offset (m)		Ground below survey (m)
			Metadata	SfM	
London	0.11	0.016	−0.54	−0.57	0.07
San Francisco	0.11	0.023	−0.53	−0.61	0.45
New York	0.12	0.028	−1.64	+0.10	0.31
Seattle	0.11	0.030	−0.07	−0.05	0.31
Tokyo	0.05	0.076	+0.02	+0.02	0.55

structural error, which mostly measures the road, falls from 0.231 to $0.089\,\mathrm{m}$. The last row also has the lowest KID and FID, 0.079 and 137.0.

A.14 Densification Variants

In the ladder of Table 4, the consensus gate first controls model capacity. Before any structural supervision, it reduces N_{GS} from 3.42 to 1.39 million, lowers the 3D structural error slightly in every view group, and lowers Lateral Shift KID and FID from 0.158 and 189.1 to 0.141 and 181.0. Its geometric effect is larger once the structural terms are present, as the variants below show.

Table 17 changes the densification of our model on London. Turning off the semantic filter that discards new Gaussians (Appendix A.8) changes the 3D structural error by at most $0.004\,\mathrm{m}$ and N_{GS} by less than 0.02 million, so the result does not depend on the filter. Removing the consensus gate more than doubles N_{GS} to 2.49 million and raises the 3D structural error from 0.202 to $0.292\,\mathrm{m}$ on Interpolation and from 0.089 to $0.168\,\mathrm{m}$ on Pitch Shift, while the Interpolation PSNR stays at 17.13 against 17.12 dB. Replacing the consensus gate with the densification rule of GDAGS (Zhou et al., 2026) doubles N_{GS} to 2.20 million and raises the 3D structural error from 0.202 to $0.256\,\mathrm{m}$ on Interpolation and from 0.089 to $0.135\,\mathrm{m}$ on Pitch Shift, with an Interpolation PSNR of 17.19 against 17.12 dB.

Table 17: Densification Variants on London

Method	Interpolation		3DSE (m)↓			N_{GS} (M)↓
	PSNR↑	LPIPS↓	Interp.	Pitch	Lateral	
Ours (unconditioned)	17.12	0.352	0.202	0.089	0.199	1.11
Ours (w/o semantic filter)	17.16	0.350	0.198	0.088	0.201	1.12
Ours (w/o consensus gate)	17.13	0.337	0.292	0.168	0.286	2.49
Ours (GDAGS instead of gate)	17.19	0.339	0.256	0.135	0.257	2.20

A.15 Session Appearance Adapter Breakdown

In the ladder of Table 4, the session appearance adapter lowers 3D structural error again, most on Pitch Shift, from 1.055 to $0.789\,\mathrm{m}$, and lowers KID and FID to 0.132 and 173.1. It takes up color differences between sessions that the model otherwise fits partly with geometry. Its main effect is on appearance, where session conditioning raises Interpolation PSNR by $0.64\,\mathrm{dB}$ over Ours (unconditioned) in Table 3.

Table 18 examines the session appearance adapter on London on the interpolation views at zero pitch. The unconditioned PSNR renders the shared model without any session, and the conditioned PSNR applies the session of each test view. With both parts, conditioning raises PSNR from 17.12 dB to 17.72 dB. The image-level part alone reaches 17.32 dB, so the Gaussian color code adds most of the gain. The unconditioned PSNR drops slightly as more variation moves into the adapter, from 17.21 dB without it to 17.12 dB with both parts.

The lower block replaces the conditioning on the session with finer per-panorama and per-image codes. These reach higher conditioned PSNR, from 17.84 dB to 18.15 dB, but only because every test view here is another heading of a training panorama, whose learned codes can be looked up at test time. Such codes do not exist for a camera at a new position, which is why we condition on the session.

Table 18: Variants of the Session Appearance Adapter on London

Method	Unconditioned PSNR↑	Conditioned PSNR↑
Ours (w/o adapter)	17.21	-
Ours (w/o Gaussian color code)	17.18	17.32
Ours	17.12	17.72
Ours (per-panorama codes)	17.03	18.15
Ours (per-image codes)	17.10	17.90
Ours (per-image codes, MLP)	17.10	17.84

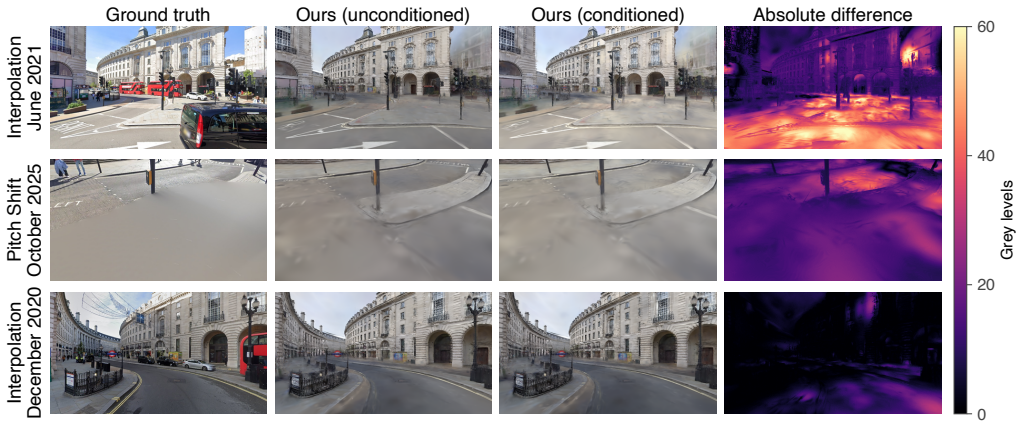


Figure 8: The session appearance adapter changes colors and leaves the geometry in place. Each row shows a London test view as ground truth, rendered without session conditioning, and rendered in the session of its panorama, followed by the absolute difference of the two renderings. The change is large for sessions whose look departs from the shared colors, such as June 2021, and small for sessions close to them, such as December 2020. Vehicles appear only in the ground truth because every model removes transient objects.

Table 19 repeats the adapter step of Table 4 in four cities, adding the adapter to the gated model without structural supervision. It lowers the 3D structural error in every city and view group.

Table 19: Adding the Session Appearance Adapter before Structural Supervision in Four Cities

	Without adapter, 3DSE (m)↓			With adapter, 3DSE (m)↓		
City	Interp.	Pitch	Lateral	Interp.	Pitch	Lateral
London	1.348	1.055	1.328	1.211	0.789	1.204
Tokyo	0.889	0.475	0.681	0.810	0.432	0.667
San Francisco	1.346	0.776	1.029	1.228	0.737	0.950
New York	1.176	0.751	1.085	1.038	0.619	1.047

A.16 Session Appearance Adapter Visualization

Figures 8 and 9 visualize the session appearance adapter on London. Conditioning a test view on its session changes colors without moving any surface, and the size of the change follows how far the session departs from the shared colors. June 2021 brightens the view of Figure 9 by about 12 grey levels through the image-level correction and 10 through the Gaussian color code, while July 2024 darkens it by about 5 through each. The per-session gains stay within 3 % of the identity, so the offset carries most of the image-level change. The Gaussian color code changes the road most, by 9.5 grey levels on average over 72 test views, followed by the sky at 7.0 and the sidewalk at 5.9, while the image-level correction changes the road, sky, sidewalk, and buildings by about 3 grey levels.

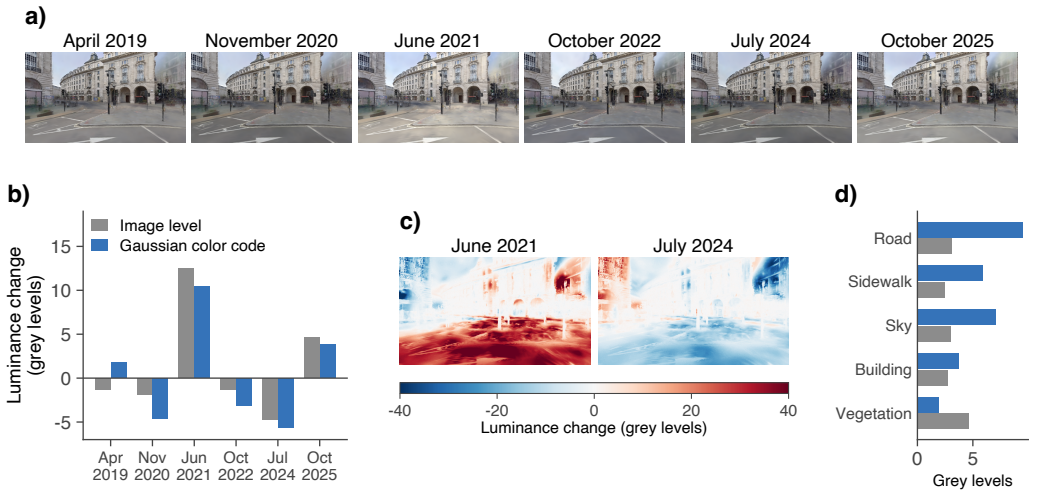


Figure 9: Each session brightens or darkens the scene in its own way, mostly on the road and the sky. (a) One test view rendered in six captured sessions. (b) Mean luminance change of this view, split into the image-level correction and the Gaussian color code. (c) Luminance change caused by the Gaussian color code alone, with red for brighter and blue for darker. (d) Mean absolute change per class over 72 test views, with the Gaussian color code in blue and the image-level correction in gray.

Table 20: Sensitivity to the Rank of the Gaussian Color Code on London

Rank q	Unconditioned PSNR↑	Conditioned PSNR↑	3DSE (m)↓
4 (ours)	17.12	17.72	0.202
8	17.08	17.88	0.210
16	17.10	17.99	0.210
32	17.09	18.05	0.204
64	17.08	18.00	0.209

A.17 Sensitivity Analysis

Adapter capacity. Table 20 varies the rank q of the Gaussian color code. The conditioned PSNR rises from 17.72 dB at rank 4 to about 18.0 dB at ranks 16 to 64, while the unconditioned PSNR changes by at most 0.04 dB. 3D structural error is lowest at rank 4 ($0.202\,\mathrm{m}$ against 0.204 to $0.210\,\mathrm{m}$), so we use rank 4.

GPS prior weight. Table 21 varies the weight of the GPS prior in the bundle adjustment and scores a plain 3DGS model trained on each solve, as in Table 14. Each row is the solve after the bundle adjustment, before the elevation prior and retriangulation, and its residual is measured in the frame of that adjustment without a new alignment. A larger weight pulls the camera centers closer to the GPS positions and lowers the median residual from $0.28\,\mathrm{m}$ to $0.09\,\mathrm{m}$, while PSNR changes by at most 0.12 dB and every setting registers 2,301 of 2,304 crops. Because the GPS readings themselves carry errors (Appendix A.11), we use a weight of 1.

Structural loss weights. Table 22 lowers and raises each structural weight around our setting. λ_{sfm} and λ_c are absolute weights, and λ_r multiplies the three road terms together. Raising any of the three weights lowers 3D structural error, to $0.139\,\mathrm{m}$ for λ_{sfm}, $0.120\,\mathrm{m}$ for λ_r, and $0.192\,\mathrm{m}$ for λ_c against $0.202\,\mathrm{m}$ for ours, at a cost of 0.21, 0.09, and 0.07 dB in PSNR. Lowering them raises 3D structural error, most for the road weight ($0.357\,\mathrm{m}$). We keep the middle setting.

Test viewpoint. Figure 10 shows how quality changes on London as the test view tilts down, and how the methods rank under Lateral Shift.

A.18 Number of Sessions

To measure how much the older sessions of the archive contribute, we train Ours (unconditioned) only on the sessions captured from a given year onward. Structure from motion, the initial points, the

Table 21: Sensitivity to the GPS Prior Weight on London

λ_{gps}	GPS residual (m)↓	Interpolation PSNR↑	Pitch Shift PSNR↑
0.3	0.28	19.73	17.65
1 (default)	0.24	19.77	17.77
3	0.17	19.70	17.73
10	0.09	19.78	17.77

Table 22: Sensitivity to the Structural Loss Weights on London

Weight	Value	Interpolation PSNR↑	3DSE (m)↓
λ_{sfm} (SfM depth)	0.1	17.20	0.263
	0.5 (ours)	17.12	0.202
	2.0	16.91	0.139
λ_r (road)	0.1	17.21	0.357
	0.5 (ours)	17.12	0.202
	2.5	17.03	0.120
λ_c (ray clearance)	0.05	17.19	0.213
	0.2 (ours)	17.12	0.202
	1.0	17.05	0.192

depth anchors, and the road grids come from the whole archive in every model, so only the images of the photometric loss change. The smallest pools hold 11 to 33 panoramas, and the whole archives hold 96 to 192 (Table 2). We render every model without session conditioning and score it on the test views of Section 4. The whole-archive points equal Ours (unconditioned) in Appendix A.24.

Figure 11 shows the results. From the smallest pool to the whole archive, Interpolation PSNR rises by 2.1 to 4.3 dB, and Lateral Shift KID and FID fall in every city. Interpolation 3D structural error falls in every city and reaches 0.16 to 0.27 m with the whole archive.

A.19 Reference Systems

Per-session reference systems. VGGT-X and DA3-GS expect a single consistent session, so we build one model per session and score every test view with the model of its own session. DA3-GS is scored on the 62 of 74 sessions whose DA3 poses align to our frame, and VGGT-X on the 47 sessions where it succeeds and aligns to our frame.

A.20 Airborne LiDAR Surveys

The 3D structural error of Table 3 is measured against a public airborne LiDAR survey of each scene, clipped to a disc of 80 m radius.

- **Seattle.** USGS 3DEP survey `WA_KingCounty_2021_B21`, April 2021, 10 points per m^2 (Sugarbaker et al., 2014).
- **San Francisco.** USGS 3DEP survey `CA_SanFrancisco_B23`, April 2023, 114 points per m^2 (Sugarbaker et al., 2014).
- **London.** Survey `P_12151` of the Environment Agency National LIDAR Programme, December 2020, 1.8 points per m^2 (Environment Agency, 2020).
- **Tokyo.** Ward-area point cloud of the Tokyo Metropolitan Government Digital Twin project, March and April 2023, 54 points per m^2 (Tokyo Metropolitan Government, 2024).
- **New York.** NOAA Digital Coast topobathymetric lidar, May 2017, 21 points per m^2 (OCM Partners, 2017).

Each survey is placed in the SfM frame of its scene by a similarity transform fitted between the GPS positions of the panoramas and their SfM camera centers, with the vertical datum taken from the survey metadata (GEOID18, OSGM15, or GSIGEO2011). The London survey is the sparsest, so its evaluation cell is 1.2 m instead of 0.5 m. The vertical offsets are handled as follows.

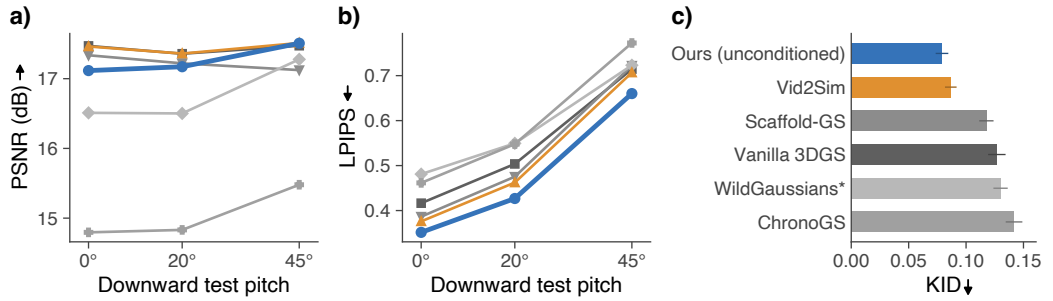


Figure 10: Sensitivity to the test viewpoint on London. Panels (a) to (c) follow the protocol of Table 3, and the method names in (c) give the colors of (a) and (b). As the view tilts down, PSNR changes little for most methods because downward views mostly show the smooth road surface, while LPIPS separates the methods. Ours (unconditioned) has the lowest LPIPS at every pitch and the lowest KID under Lateral Shift.

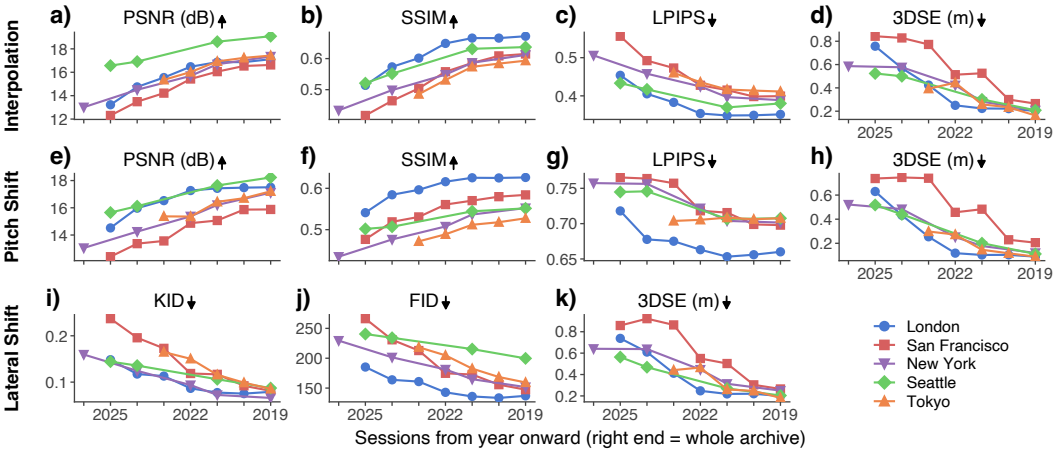


Figure 11: Quality of Ours (unconditioned) when training uses only the sessions from a given year onward. Moving right adds older sessions and their panoramas, and the rightmost points use the whole archive. Structure from motion, initial points, depth anchors, and road grids come from the whole archive in every model. From the smallest pool to the whole archive, PSNR rises and KID, FID, and 3D structural error fall in every city.

- **New York frame.** The New York frame is first shifted vertically by $-1.25\,\mathrm{m}$. This constant is shared by all methods.
- **Frame offset.** Every survey is then shifted by the frame offset of Appendix A.11. This offset is the same for all methods, because the baselines are aligned to our frame.
- **Google Earth.** Google Earth is a separate source with its own vertical datum. We remove its per-city vertical offset against the survey, estimated as the median ground height difference on open ground (Seattle $-0.46\,\mathrm{m}$, San Francisco $-0.50\,\mathrm{m}$, London $-0.09\,\mathrm{m}$, Tokyo $-0.65\,\mathrm{m}$, New York $+0.03\,\mathrm{m}$), and score it against the unshifted survey, as we remove the offset of our frame.

Attributions. The London survey is used under the Open Government Licence v3.0 and contains Environment Agency information © Environment Agency and/or database right. The Tokyo survey is the 3D point cloud data for the 23 wards of the Tokyo Digital Twin Realization Project (Tokyo Metropolitan Government), licensed under CC BY 4.0. Elevation data for Seattle and San Francisco are courtesy of the U.S. Geological Survey 3D Elevation Program (3DEP). The New York survey is the 2017 NYC Topobathy Lidar of OCM Partners, NOAA Office for Coastal Management Digital Coast.

A.21 System Overhead

Onboarding a scene is dominated by transient removal, which takes about 1.7 to 2 min per panorama on an RTX 5090. FLUX inpainting takes about 1.4 min of this and the SAM 3 masks 12 to 32 s.

Table 23: Running Time of Each Stage of Our Pipeline for Each Scene

Stage	London	Tokyo†	Seattle	San Francisco	New York	Mean
Perspective crops	2304	1644	1152	1704	1668	1694
Crop pairs matched (reduction)	287k (9×)	210k (6×)	88k (7×)	203k (7×)	213k (7×)	-
SuperPoint	34 s	20 s	14 s	24 s	21 s	23 s
SuperGlue matching	4h04m	2h13m	52m	1h58m	2h28m	2h19m
SfM optimization	1h09m	35m	22m	44m	40m	42m
Training (30k)	28m	30m	20m	28m	32m	28m
Total without transient removal	5h49m	3h22m	1h37m	3h14m	3h44m	3h33m
Gaussians, ours (M)	1.11	1.77	1.18	1.28	1.62	1.39
Gaussians, Vid2Sim (M)	1.62	2.92	2.16	2.24	2.25	2.24

Table 23 reports the running time of each stage of our pipeline. SfM stages ran on one NVIDIA RTX 6000 Ada GPU with 8 CPU cores. Its total excludes transient removal, which adds about 5.4 to $6.4\,\mathrm{h}$ on London, and its last two rows give the number of Gaussians of our model and of Vid2Sim. Feature matching takes most of the time. The pairing graph built from the panorama metadata matches 6 to 9 times fewer crop pairs than exhaustive matching, for example 287 k pairs on London, yet SuperGlue matching still takes 2 h 19 min on average, against 42 min for the remaining SfM optimization and 28 min for training. The Tokyo column (†) was timed on 137 panoramas. The total also includes short stages that are not listed, such as the elevation bundle adjustment and the road grid. On London, our densest scene, matching takes 78 % of the five-hour structure from motion stage, and the converged model holds 1.1 M Gaussians against 2.6 M for a plain 3DGS model trained on the same camera poses and renders 912×512 views at 472 fps against 245 fps on an RTX 5090.

A.22 Stabilized WildGaussians

The released configuration of WildGaussians fails on every scene. Its uncertainty mask, designed to ignore moving objects, treats the appearance differences between capture sessions as transient content. On London, the fraction of pixels that stay supervised drops from 1.0 to 0.038 by 11.8 k iterations and reaches zero at 16 k, and the same collapse occurs at the same point of training in every city. Once the mask closes, the photometric loss no longer supervises the model, and the released configuration ends at 5.7 to 6.9 dB interpolation PSNR (Table 24). The stabilized variant WildGaussians* used in all our tables disables the mask and the periodic opacity reset and keeps every other setting of the release.

Table 24: Interpolation PSNR of the Released and the Stabilized WildGaussians

	London	New York	Seattle	San Francisco	Tokyo	Mean
Released	5.68	6.11	6.86	6.12	6.50	6.25
WildGaussians*	16.51	15.82	16.62	15.64	16.13	16.14

A.23 Training Seeds

Table 25 reports the mean and standard deviation of every metric of Table 3 on London when each method is trained with three seeds, 42, 43, and 44 for our method and 0, 1, and 2 for the baselines. Seed 42 is the run behind the main table. Across the four rows, the standard deviation over seeds is at most $0.09\,\mathrm{dB}$ for PSNR, below 0.001 for SSIM and LPIPS, at most $0.034\,\mathrm{m}$ for 3D structural error, at most 0.002 for KID, and at most 1.3 for FID. Every claim that the main text makes about our method holds on the mean over seeds and on each seed.

A.24 Per-Scene Results

Tables 26 to 30 report each scene in the format of Table 3. As expected from its calibrated aerial capture, the Google Earth mesh has the best 3D structural error in nearly every city and view group.

Table 25: Three Training Seeds in London

Method	Interpolation				Pitch Shift				Lateral Shift		
	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	KID↓	FID↓	3DSE (m)↓
Vanilla 3DGS (Kerbl et al., 2023)	$17.50_{\pm 0.02}$	$0.656_{\pm 0.000}$	$0.416_{\pm 0.000}$	$1.178_{\pm 0.025}$	$17.52_{\pm 0.07}$	$0.613_{\pm 0.000}$	$0.715_{\pm 0.001}$	$0.937_{\pm 0.034}$	$0.127_{\pm 0.000}$	$166.8_{\pm 0.6}$	$1.157_{\pm 0.027}$
Vid2Sim (Xie et al., 2025)	$17.47_{\pm 0.01}$	$0.666_{\pm 0.000}$	$0.376_{\pm 0.001}$	$0.535_{\pm 0.008}$	$17.43_{\pm 0.08}$	$0.618_{\pm 0.001}$	$0.707_{\pm 0.001}$	$0.341_{\pm 0.017}$	$0.086_{\pm 0.002}$	$141.0_{\pm 1.3}$	$0.526_{\pm 0.004}$
Ours (unconditioned)	$17.11_{\pm 0.00}$	$0.672_{\pm 0.001}$	$0.352_{\pm 0.001}$	$0.199_{\pm 0.004}$	$17.57_{\pm 0.06}$	$0.627_{\pm 0.000}$	$0.660_{\pm 0.000}$	$0.091_{\pm 0.002}$	$0.081_{\pm 0.002}$	$138.0_{\pm 1.1}$	$0.197_{\pm 0.002}$
Ours (conditioned)	$17.73_{\pm 0.01}$	$0.675_{\pm 0.001}$	$0.348_{\pm 0.001}$	$0.199_{\pm 0.004}$	$18.50_{\pm 0.09}$	$0.631_{\pm 0.000}$	$0.656_{\pm 0.001}$	$0.091_{\pm 0.002}$	$0.086_{\pm 0.002}$	$141.5_{\pm 0.6}$	$0.197_{\pm 0.002}$

Table 26: Comparison with Baselines on London

Method	Interpolation				Pitch Shift				Lateral Shift		
	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	KID↓	FID↓	3DSE (m)↓
Vanilla 3DGS (Kerbl et al., 2023)	17.47	0.655	0.416	1.154	17.47	0.613	0.714	0.929	0.127	166.7	1.128
Scaffold-GS (Lu et al., 2024)	17.34	0.664	0.387	0.631	17.12	0.613	0.722	0.432	0.118	160.1	0.639
WildGaussians* (Kulhanek et al., 2024)	16.51	0.622	0.481	0.690	17.28	0.613	0.723	0.379	0.130	172.9	0.664
Vid2Sim (Xie et al., 2025)	17.46	0.666	0.376	0.541	17.51	0.619	0.707	0.346	0.087	141.6	0.530
ChronoGS (Wang et al., 2026c)	14.80	0.610	0.462	0.564	15.48	0.592	0.773	0.618	0.142	177.0	0.542
Google Earth	12.84	0.488	0.540	0.140	16.09	0.585	0.752	0.071	0.325	331.4	0.138
VGGT-X (Liu et al., 2025)	14.06	0.449	0.464	0.987	14.60	0.482	0.722	0.687	0.191	248.2	1.043
DA3-GS (Lin et al., 2025)	14.76	0.524	0.473	0.442	18.25	0.606	0.641	1.145	0.114	165.7	0.340
Ours (unconditioned)	17.12	0.672	0.352	0.202	17.51	0.626	0.660	0.089	0.079	137.0	0.199
Ours (conditioned)	17.72	0.676	0.348	0.202	18.39	0.631	0.657	0.089	0.084	140.8	0.199

*Stabilized variant of WildGaussians.

Table 27: Comparison with Baselines on Tokyo

Method	Interpolation				Pitch Shift				Lateral Shift		
	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	KID↓	FID↓	3DSE (m)↓
Vanilla 3DGS (Kerbl et al., 2023)	18.00	0.564	0.463	0.746	16.67	0.511	0.745	0.434	0.074	153.2	0.734
Scaffold-GS (Lu et al., 2024)	17.71	0.564	0.462	0.656	16.28	0.507	0.761	0.730	0.076	151.1	0.710
WildGaussians* (Kulhanek et al., 2024)	16.13	0.487	0.641	0.654	16.23	0.505	0.830	0.603	0.178	247.9	0.664
Vid2Sim (Xie et al., 2025)	17.60	0.581	0.404	0.243	16.84	0.515	0.708	0.202	0.086	160.9	0.250
ChronoGS (Wang et al., 2026c)	15.87	0.519	0.507	0.684	16.01	0.502	0.781	1.123	0.070	148.0	0.763
Google Earth	10.63	0.362	0.712	0.159	10.93	0.414	0.824	0.098	0.304	344.1	0.161
VGGT-X (Liu et al., 2025)	12.39	0.358	0.686	1.144	12.34	0.414	0.787	0.780	0.205	331.6	1.244
DA3-GS (Lin et al., 2025)	14.82	0.415	0.553	1.623	16.82	0.507	0.667	2.872	0.142	203.7	2.175
Ours (unconditioned)	17.43	0.593	0.411	0.163	17.21	0.527	0.708	0.089	0.085	159.4	0.181
Ours (conditioned)	18.08	0.599	0.408	0.163	17.54	0.529	0.706	0.089	0.096	167.1	0.181

*Stabilized variant of WildGaussians.

Table 28: Comparison with Baselines on Seattle

Method	Interpolation				Pitch Shift				Lateral Shift		
	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	KID↓	FID↓	3DSE (m)↓
Vanilla 3DGS (Kerbl et al., 2023)	18.53	0.560	0.491	0.874	16.68	0.511	0.777	0.614	0.116	213.8	0.909
Scaffold-GS (Lu et al., 2024)	18.29	0.565	0.484	0.609	16.88	0.521	0.775	0.686	0.108	200.5	0.606
WildGaussians* (Kulhanek et al., 2024)	16.62	0.493	0.679	1.629	17.11	0.515	0.861	2.895	0.195	281.6	0.960
Vid2Sim (Xie et al., 2025)	18.04	0.580	0.413	0.260	16.65	0.481	0.740	0.187	0.097	204.0	0.240
ChronoGS (Wang et al., 2026c)	16.69	0.520	0.525	0.473	15.52	0.502	0.812	1.056	0.120	212.5	0.471
Google Earth	12.48	0.393	0.718	0.135	13.65	0.445	0.850	0.099	0.319	378.8	0.124
VGGT-X (Liu et al., 2025)	10.69	0.395	0.783	1.292	10.30	0.444	0.861	1.037	0.440	434.2	1.311
DA3-GS (Lin et al., 2025)	15.21	0.442	0.555	0.465	18.14	0.509	0.668	0.926	0.143	234.9	0.388
Ours (unconditioned)	19.08	0.637	0.380	0.208	18.23	0.552	0.708	0.112	0.087	199.7	0.204
Ours (conditioned)	19.61	0.641	0.376	0.208	18.92	0.555	0.703	0.112	0.096	204.7	0.204

*Stabilized variant of WildGaussians.

Table 29: Comparison with Baselines on San Francisco

Method	Interpolation				Pitch Shift				Lateral Shift		
	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	KID↓	FID↓	3DSE (m)↓
Vanilla 3DGS (Kerbl et al., 2023)	17.73	0.598	0.440	1.146	15.77	0.574	0.725	0.769	0.130	190.5	1.029
Scaffold-GS (Lu et al., 2024)	17.19	0.596	0.438	0.721	15.50	0.571	0.736	0.633	0.131	183.2	0.628
WildGaussians* (Kulhanek et al., 2024)	15.64	0.523	0.628	0.703	15.44	0.563	0.816	0.821	0.223	274.6	0.672
Vid2Sim (Xie et al., 2025)	16.69	0.603	0.381	0.234	15.70	0.575	0.707	0.206	0.083	149.6	0.223
ChronoGS (Wang et al., 2026c)	13.84	0.504	0.550	0.571	13.49	0.535	0.830	1.085	0.156	203.6	0.549
Google Earth	10.25	0.377	0.683	0.085	9.73	0.422	0.829	0.067	0.327	354.5	0.085
VGGT-X (Liu et al., 2025)	12.53	0.364	0.610	1.049	12.52	0.422	0.790	0.941	0.289	344.0	1.025
DA3-GS (Lin et al., 2025)	13.74	0.446	0.551	1.653	16.51	0.555	0.701	5.720	0.163	221.7	1.384
Ours (unconditioned)	16.63	0.615	0.399	0.265	15.88	0.584	0.698	0.205	0.081	147.4	0.265
Ours (conditioned)	17.31	0.620	0.393	0.265	16.54	0.588	0.693	0.205	0.091	153.7	0.265

*Stabilized variant of WildGaussians.

Table 30: Comparison with Baselines on New York

Method	Interpolation				Pitch Shift				Lateral Shift		
	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	PSNR↑	SSIM↑	LPIPS↓	3DSE (m)↓	KID↓	FID↓	3DSE (m)↓
Vanilla 3DGS (Kerbl et al., 2023)	17.77	0.561	0.482	0.977	16.83	0.531	0.771	0.604	0.092	174.3	0.956
Scaffold-GS (Lu et al., 2024)	17.37	0.558	0.490	0.622	16.64	0.531	0.794	0.516	0.100	175.3	0.587
WildGaussians* (Kulhanek et al., 2024)	15.82	0.503	0.632	0.756	16.01	0.519	0.842	0.725	0.144	217.5	0.789
Vid2Sim (Xie et al., 2025)	17.31	0.583	0.392	0.248	16.65	0.534	0.716	0.205	0.071	153.1	0.239
ChronoGS (Wang et al., 2026c)	14.45	0.509	0.523	0.541	14.90	0.512	0.805	1.396	0.085	163.8	0.518
Google Earth	11.28	0.395	0.692	0.089	12.52	0.484	0.819	0.049	0.296	345.5	0.084
VGGT-X (Liu et al., 2025)	12.02	0.386	0.668	1.532	12.24	0.380	0.807	1.207	0.262	359.9	1.461
DA3-GS (Lin et al., 2025)	14.76	0.444	0.541	0.862	17.02	0.515	0.673	1.611	0.137	204.7	0.574
Ours (unconditioned)	17.35	0.612	0.389	0.207	17.11	0.552	0.701	0.120	0.066	151.7	0.250
Ours (conditioned)	18.09	0.617	0.383	0.207	17.67	0.556	0.697	0.120	0.073	156.6	0.250

*Stabilized variant of WildGaussians.