Seeing Across Skies and Streets: Feedforward 3D Reconstruction from Satellite, Drone, and Ground Images

Qiwei Wang*
ShanghaiTech University
Shanghai, China
wangqw2024@shanghaitech.edu.cn

Zhongyao Tuo*
ShanghaiTech University
Shanghai, China
tuozhy2025@shanghaitech.edu.cn

Xianghui Ze
Nanjing University of Science and Technology
Nanjing, China
zexh@njust.edu.cn

Yujiao Shi†
ShanghaiTech University
Shanghai, China
shiyj2@shanghaitech.edu.cn

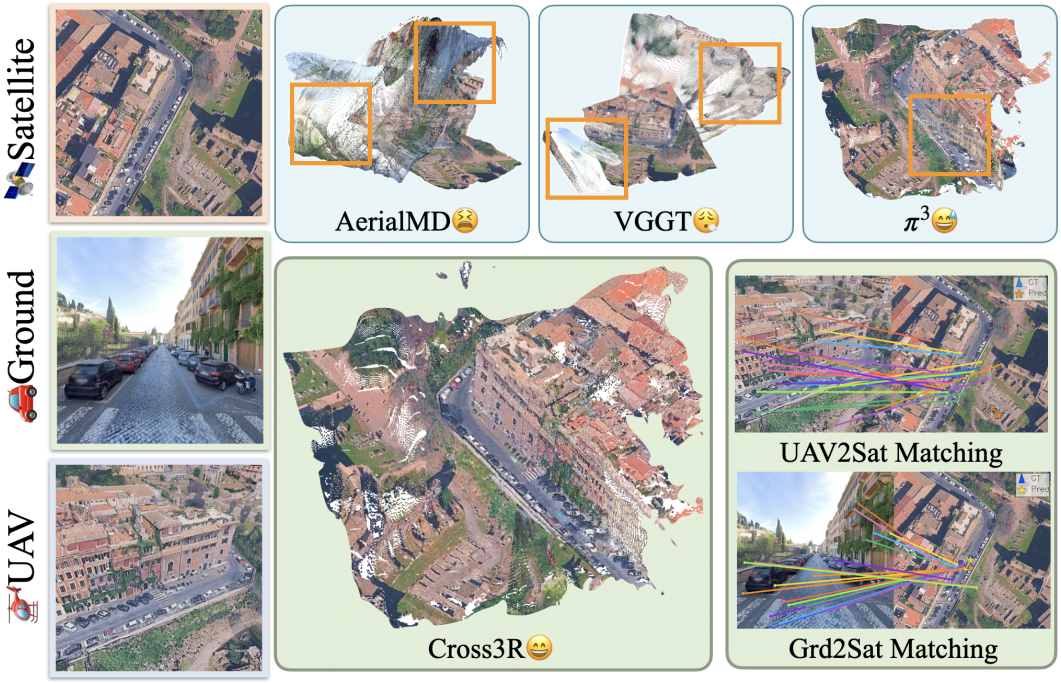


Figure 1: Cross3R ingests a satellite tile along with one or two perspective views (UAV, ground, or both) and jointly recovers a cross-view 3D point cloud, all camera poses, and the on-tile localization of each perspective camera.

Abstract

Cross-view localization classically asks: where does this ground image lie on the satellite tile? Existing methods are typically limited to 3-DoF estimates—an

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

(x, y) position and a yaw angle—because nadir satellite imagery provides no direct cues for roll, pitch, or altitude, forcing a reliance on planar-motion and zero-tilt assumptions. These assumptions break on real terrain with slopes, ramps, and tilted camera mounts. To overcome this, we introduce a single UAV image as an intermediate viewpoint: it reveals the 3D structure invisible from nadir, supplies the cues for roll, pitch, and altitude that the satellite alone cannot provide, and needs only spatial overlap with the ground camera—no known relative pose is required. Building on this insight, we propose **Cross3R**, a flexible feed-forward model that ingests a satellite tile together with a UAV image, a ground image, or both, and, in a single forward pass, recovers a cross-view 3D point cloud, the 6-DoF poses of every input camera, and the on-tile (x, y) position and yaw of each perspective camera. For training and evaluation, we also construct **CrossGeo**, a 278K-image tri-view dataset spanning 85 scenes across every continent except Antarctica. On CrossGeo, Cross3R consistently outperforms feed-forward 3D baselines in point-cloud reconstruction, 6-DoF camera-pose estimation, and cross-view localization. On KITTI, it outperforms dedicated cross-view methods trained on KITTI on most metrics, despite having no KITTI training itself.

1 Introduction

Ground-to-satellite camera localization aims to estimate the pose of a ground-view camera with respect to a satellite image. This paradigm is appealing because satellite imagery provides globally available, frequently updated, and cost-effective environmental coverage. In contrast, 3D maps built from LiDAR scans, structure-from-motion point clouds, meshes, or neural representations are expensive to construct and difficult to maintain, making satellite-based localization an especially scalable alternative for large-area deployments.

Despite its appeal, direct ground-to-satellite matching is challenging due to the drastic viewpoint gap, severe appearance changes, and ambiguous geometric correspondences between the two modalities. As a consequence, existing approaches [50, 35, 33, 17, 44, 41] restrict the problem to 3-DoF localization—recovering only the (x, y) position and yaw angle of the ground camera—because a single nadir tile provides no direct cues for roll, pitch, or altitude; these methods further assume the ground camera moves on a flat plane at a fixed height with zero roll and pitch. Real environments, however, contain slopes, ramps, uneven terrain, and tilted camera mounts—conditions common in urban scenes, construction zones, and robotic deployments—where these assumptions break and 3-DoF methods degrade sharply in both accuracy and robustness. Overcoming this limitation requires constraining the missing degrees of freedom with cues that satellite imagery alone cannot supply.

Our key insight is that a single UAV image captured at an intermediate altitude naturally bridges the large viewpoint gap between the satellite (top-down) and the ground (perspective) views: it simultaneously observes the top-down layout characteristic of satellite views and the oblique 3D structure visible from the ground. This connective viewpoint is what prior 3-DoF cross-view methods could not establish from a satellite tile alone. Crucially, our approach does not require known relative poses between the UAV and the ground or satellite cameras; only a shared field of view is necessary—a condition easily satisfied by cooperative UGV–UAV systems, on-demand drones, and patrol archives indexed with coarse GPS.

Building on this insight, we propose **Cross3R**, a feed-forward cross-view reconstruction model that breaks the 3-DoF ceiling of prior cross-view methods. Cross3R accepts satellite tiles paired with a ground image, a UAV image, or both, and incorporating the UAV image yields pronounced gains in ground-to-satellite localization (Fig. 7, Tab. 6). Beyond the UAV view, two other design choices are also important. First, a per-sample altitude redefinition (Sec. 3.2) stabilizes training by aligning the satellite, UAV, and ground depth ranges. Second, a single per-tile scale ρ replaces satellite intrinsics. Because ρ is a learned relative quantity (not true meters-per-pixel), any 3D point—including the ground- and UAV-camera centers—can be placed on the satellite tile simply by transforming to the satellite frame and dividing by ρ. The impact of this design is clearest on KITTI (Tab. 4), where Cross3R surpasses specialized cross-view methods trained directly on the KITTI split [50, 33, 17] on most metrics. Since no public benchmark provides all three views with full 6-DoF poses and metric depth (Tab. 1), we construct and release **CrossGeo**, an automatically curated dataset of 85 globally

Table 1: Datasets Comparison

Name	Data Source	Images	Grd	UAV	Map	Grd Pose	UAV Observation	Scenes
AnyVisLoc [55]	Real	18,000	✗	✓	✗	–	Multi-view	Multiple
University-1652 [58]	Real	50,218	✓	✓	✓	✗	no pose	Buildings
CVUSA [48]	Real	44,516	✓	✗	✓	3DoF	–	Streets
CVACT [23]	Real	44,416	✓	✗	✓	3DoF	–	Streets
KITTI [7]	Real	14,999	✓	✗	✓	3DoF	–	Streets
VIGOR [60]	Real	195,832	✓	✗	✓	3DoF	–	Multiple
ULTRRA Challenge [14]	Synthetic	1,207	✓	✓	✗	6DoF	Multi-view	Buildings
AerialMD [37]	Real+Synth	132,137	✓	✓	✗	6DoF	Multi-view	Multiple
Ours (CrossGeo)	**Real+Synth**	**277,812**	✓	✓	✓	**6DoF**	**Multi-view**	**Multiple**

distributed scenes sourced from Google Maps, Google Earth, and Google Street View. Evaluated on CrossGeo, KITTI, and AnyVisLoc [55], Cross3R substantially outperforms strong feed-forward 3D baselines [39, 45, 37] and dedicated 3-DoF cross-view methods [50, 33, 17].

2 Related Work

Cross-view Ground Image Localization. This task estimates the pose of a ground-level camera within a candidate satellite region [32, 50, 27, 28, 20, 16] and is challenging because of the large appearance and viewpoint gap between ground and aerial imagery. One line of work matches the two views in a learned feature space using contrastive objectives [36, 60, 41, 34], which is effective for coarse retrieval but struggles with fine-grained pose. A second line projects the ground view into a bird's-eye-view representation and establishes explicit geometric correspondences with the satellite image [35, 44, 49, 42], yielding more interpretable estimates. Both families share a fundamental limitation: a single nadir tile provides no cues for roll, pitch, or height, so they all adopt a planar-motion and zero-tilt assumption that reduces the problem to 3-DoF pose estimation and breaks on slopes, ramps, or tilted camera mounts. Cross3R lifts this restriction by adding a UAV image as an intermediate viewpoint that supplies the missing roll/pitch/altitude cues.

3D Scene Reconstruction. Recovering 3D geometry and cameras from images is a long-standing problem. Classical Structure-from-Motion [29, 30, 38, 5, 57, 19, 26, 22] and Multi-View Stereo [6, 9, 31, 54, 3, 2, 10, 47] pipelines, together with their modern learned variants, remain the workhorses for high-fidelity reconstruction, but require heavily overlapping views and degrade under sparse inputs or large viewpoint gaps. Recent feed-forward 3D models [43, 18, 52, 39, 45, 15, 40, 24, 8] drop per-scene optimization and instead predict geometry and cameras jointly in a single forward pass, which generalizes to novel scenes without per-instance tuning. Monocular depth foundation models [21] supply reusable priors, and AerialMD [37] specializes DUSt3R to the aerial–ground setting by fine-tuning on a dedicated UAV-ground corpus. All of these assume image collections from a roughly common altitude; none handles a satellite/UAV/ground triplet—the gap Cross3R fills.

Multi-view Datasets. As summarized in Table 1, existing datasets cover only two modalities or omit poses. Ground-satellite benchmarks (CVUSA [48], CVACT [23], KITTI [7], VIGOR [60]) provide ground and satellite views but lack UAV images, restricting them to 3-DoF planar assumptions. UAV-focused datasets (SEUS-200 [59], GTA-UAV [12], UAV-VisLoc [51], AnyVisLoc [55]) include aerial imagery but omit ground-level views, missing the fine-grained street-level geometry. University-1652 [58] offers all three modalities but provides no camera poses, making it suitable only for image retrieval. Multi-view 3D datasets (ULTRRA [14], AerialMD [37]) offer 6-DoF poses but lack satellite imagery as a top-down reference. This absence has stalled joint satellite–UAV–ground reasoning, which CrossGeo closes with all three modalities, full 6-DoF poses, and dense metric depth.

3 Method

We present *Cross3R*, a feed-forward foundation model that jointly recovers tri-view 3D point clouds, all camera poses, and the 6-DoF cross-view localization of every input view in a single forward pass. Section 3.1 describes **CrossGeo**, the large-scale satellite / UAV / ground training corpus we construct from publicly available sources. Section 3.2 then details how Cross3R extends π^3 [45] with three modifications targeted at the satellite branch and how the model is trained.

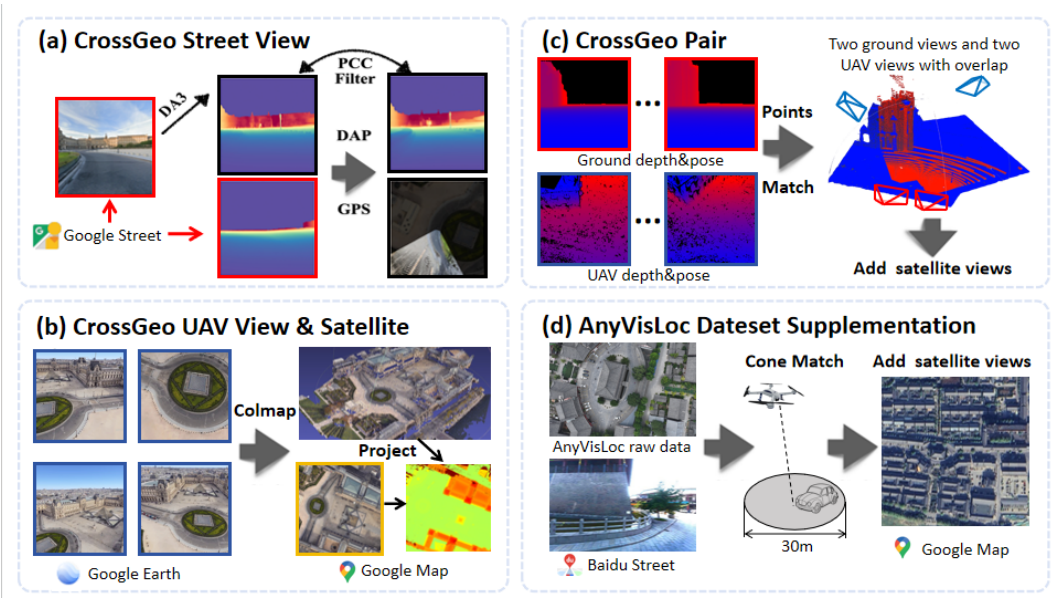


Figure 2: CrossGeo data sources. (a) Ground views and their coarse depth from *Google Street View*. (b) UAV captures rendered in *Google Earth* together with co-located satellite tiles from *Google Maps*. (c) Tri-view pairing across the three modalities. (d) AnyVisLoc test-set augmentation.

3.1 CrossGeo Dataset

Training a tri-view model requires paired (satellite, UAV, ground) data with full 6-DoF poses and dense metric depth across all three views—a combination that no public benchmark currently provides (Tab. 1). We therefore build **CrossGeo**, a tri-view corpus of 46,302 samples drawn from 85 globally distributed scenes spanning every continent except Antarctica and covering urban, suburban, rural, and natural terrain. Each sample bundles two satellite, two UAV, and two ground views of the same region with calibrated poses and dense metric depth, so CrossGeo contains $46{,}302 \times 6 = 277{,}812$ images in total. Figure 2 summarizes the four data sources; Figures 10–11 of the supplementary material visualize the resulting tri-view coverage and the construction pipeline.

Data collection. Acquisition is anchored on each scene's GPS center. We extract a $300\,\mathrm{m} \times 300\,\mathrm{m}$ Google Maps satellite tile, render UAV views in Google Earth at altitudes $H_{\text{drone}} \in [30, 120]\,\mathrm{m}$ and pitch angles $\in [0°, 90°]$, and collect ground images via the Google Street View API. The high-pitch range (60°–90°) is oversampled because high-altitude low-pitch renderings suffer from pixel stretching that degrades ground resolution and geometric stability; the resulting altitude/pitch distribution and the global geographic coverage are visualized in Fig. 10(b)–(c). We use Google Street View rather than Google Earth's street-level renderings, which suffer from geometric artifacts such as texture clipping through buildings and a larger domain gap to real-world deployment.

Pose recovery. We define a world frame whose z-axis points east, x-axis points south, and y-axis points vertically downward. For UAV and ground views, the Google Earth and Street View APIs already expose every quantity needed to assemble extrinsics—altitude, GPS, yaw, pitch—which we convert directly into a rotation–translation pair (R, T). Satellite poses are more involved because Google Maps exposes only the tile center's GPS. We therefore model every satellite image as a virtual camera looking straight down with a narrow $[2°, 5°]$ field of view, and recover its altitude by sweeping candidate values in Google Earth and stopping when the rendered top-down view at the same GPS matches the actual Google Maps tile. An altitude of $5{,}726\,\mathrm{m}$ yields near-perfect alignment and is adopted as the canonical satellite camera height.

Metric depth and tri-view pairing. Each modality acquires dense metric depth through a dedicated pipeline (Fig. 11). *Ground depth* (Fig. 2(a)) begins with Street View, which provides an absolute-scale but coarse depth map that captures the overall scene scale while missing fine structures such as distant buildings and street lamps. Because Google Street View imagery is sparsely distributed along

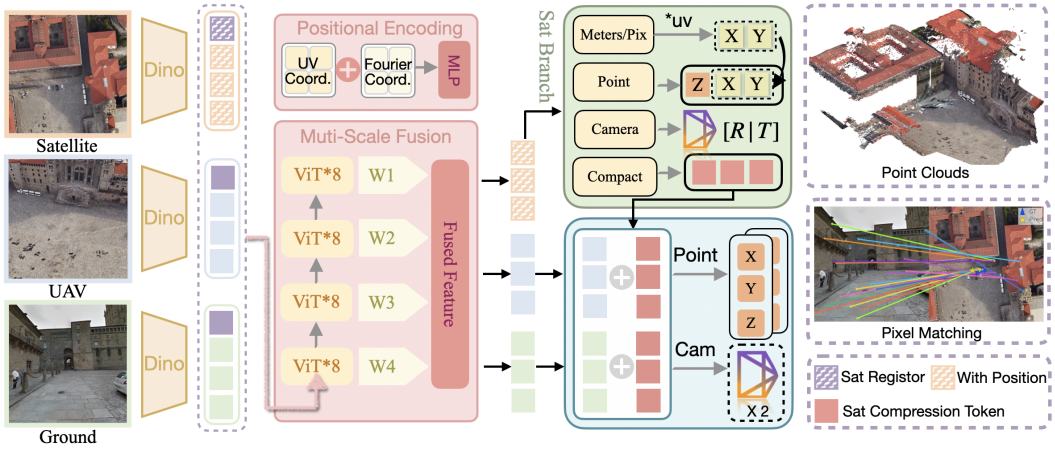


Figure 3: Overview of Cross3R. Satellite, UAV, and ground images are encoded and processed through an Alternating Attention module. Task-specific heads then predict: an orthographic representation (a per-tile scale ρ and per-pixel depth z) for the satellite view, local 3D point maps (x, y, z) for the UAV and ground views, and unified camera poses (R, T) for all views.

roadways, it is unsuitable for COLMAP [29] multi-view stereo. We therefore apply DEPTHANYTHING V3 [21] to each ground frame, obtaining a sharp but *relative* depth, and then fuse the two using *Prior Depth Anything* [46], which employs the Street View depth as a metric anchor to recover the global scale and shift of the relative prediction. As a safeguard, we discard samples where the rescaled depth exhibits low Pearson correlation [1] with the relative prediction. *UAV depth* (Fig. 2(b)) is recovered by running COLMAP [29] multi-view stereo on a denser auxiliary image set captured for each scene, and *satellite depth* is obtained for free by projecting the resulting UAV point cloud into each top-down virtual camera. With depth and pose for every image, we score cross-modality pairs by voxel overlap and select the highest-scoring six-image tuples (two views per modality) as our tri-view samples (Fig. 2(c)), requiring that all $\binom{6}{2}=15$ pairwise overlaps be non-empty.

Train / val / test split. The 85 scenes are partitioned at the *scene* level into three disjoint subsets, ensuring that no city appears in more than one split. The training split comprises 75 scenes (38,962 samples); the validation split contains 5 scenes (3,614 samples) drawn from North American cities; and the test split consists of 5 scenes (3,726 samples) sourced from cities with no overlap with the training set—probing generalization across both city and continent boundaries. This cross-continent test split serves as the in-distribution evaluation set used throughout Section 4.

Real-world OOD test set. CrossGeo's UAV imagery is rendered in Google Earth and is therefore synthetic. To stress-test Cross3R under real aerial photography, we repurpose AnyVisLoc [55], a benchmark of UAV photographs captured by physical drones, as an out-of-distribution test set (Fig. 2(d)). AnyVisLoc provides only the UAV view, so we augment every sample with the two missing modalities: for each AnyVisLoc UAV photograph we draw a Baidu Maps ground image within 30 m of the UAV's GPS, then center a 110 m $\times$ 110 m Google Maps tile on the ground camera with a random $\pm$20 m east–south shift to mimic deployment-time GPS uncertainty. The resulting 5,068 tri-view samples form a fully real-world test set used throughout the experiments.

3.2 The Cross3R Model

Cross3R takes a satellite tile I_s and up to two perspective views (a UAV image I_u and/or a ground image I_g) that share overlapping scene content. In a single forward pass it predicts, for every input view v, a per-pixel point map $P^v \in \mathbb{R}^{H\times W\times 3}$ and a 6-DoF camera pose (R^v, T^v), along with a per-image scalar ρ that ties the per-pixel scale of the satellite tile to the rest of the scene. Since ρ is a learned relative scale (not true meters-per-pixel), Cross3R recovers a self-consistent scene; any 3D point—including the ground- and UAV-camera centers—can then be located on the satellite tile simply by transforming to the satellite's local frame and dividing the (x, y) coordinate by ρ. In contrast, existing feed-forward 3D models [45, 39, 37] are designed for image collections captured from roughly similar altitudes. On satellite–UAV–ground triplets, two difficulties surface (Tab. 3,

Fig. 1): the predicted point clouds are individually plausible but drift apart and fail to align; moreover, these models output only point clouds and extrinsics, so projecting a ground or UAV camera onto the satellite tile requires inferring satellite intrinsics from an extremely narrow field of view—a fragile step that often returns wrong intrinsics and breaks cross-view localization.

Cross3R realizes ρ and the joint registration described above by building on π^3 [45] and introducing three coordinated modifications to the encoder, the alternating-attention stack, and the decoding heads (Fig. 3). *Encoder.* We inject a Fourier positional embedding [25] of the satellite's normalized pixel grid directly into the satellite tokens, so the orthographic grid structure enters the network from the very first layer:

$$\mathrm{PE}(x) \;=\; W\big[\,x \,\|\, \sin(2\pi B^\top x) \,\|\, \cos(2\pi B^\top x)\,\big] + b, \tag{1}$$

where $x \in [0,1]^2$ is a normalized pixel coordinate, $B \in \mathbb{R}^{2\times K}$ is a fixed Gaussian random projection controlling the frequency bandwidth, and W, b are learnable parameters. We also double the size of π^3's register-token bank, giving the encoder extra capacity to disentangle satellite from perspective tokens. *Alternating attention.* Instead of attending only to the last layer, the heads fuse hidden states across layers with softmax-weighted gates, gaining simultaneous access to fine local geometry (early layers) and coarse global layout (late layers). *Decoding.* The satellite and perspective branches diverge. For the satellite branch we enforce an explicit orthographic geometric prior: a Google Maps tile is rendered by a virtual pinhole camera looking straight down with a very narrow $[2^\circ, 5^\circ]$ field of view from $5{,}726$ m altitude (Section 3.1). Although the tile is fundamentally a perspective rendering, the parallax across a single tile is sub-pixel, making an orthographic projection an excellent approximation. Under this model, every 3D point in the tile's local frame satisfies

$$(x_i, y_i) \;=\; (u_i, v_i)\cdot \rho, \tag{2}$$

where (u_i, v_i) is a pixel coordinate and ρ is the aforementioned learned relative scale. A tiny MLP regresses ρ per satellite image, and we replace the (x, y) output of the shared point head with $(u, v)\cdot\rho$, retaining only the per-pixel depth $\hat{z}$. The resulting satellite point cloud is internally scale-consistent by construction and shares the same relative scale as the rest of the scene. For the UAV and ground branches, satellite features from the same stage are compressed by a small MLP and broadcast-added into the perspective tokens before the shared point and camera heads, giving each perspective branch explicit guidance from the satellite coordinate frame. Without such an anchor, earlier methods must resort to fragile intrinsic estimation—precisely the failure mode we observe in the baselines.

Loss functions. Following π^3 [45], the training objective is a single weighted sum of four terms,

$$\mathcal{L} \;=\; \mathcal{L}_{\text{geo}} + \lambda_n \mathcal{L}_{\text{norm}} + \lambda_c \mathcal{L}_{\text{conf}} + \lambda_p \mathcal{L}_{\text{cam}}. \tag{3}$$

$\mathcal{L}_{\text{geo}}$ first solves for an optimal global scale $s^\star$ that aligns predicted to ground-truth point maps and then penalizes the depth-weighted L_1 residual at that scale; $\mathcal{L}_{\text{norm}}$ minimizes the angular error between predicted and ground-truth surface normals to encourage locally smooth geometry; $\mathcal{L}_{\text{conf}}$ supervises a per-pixel confidence map with binary cross-entropy, with target 1 when the scaled reconstruction error falls below a threshold ϵ; and $\mathcal{L}_{\text{cam}}$ supervises relative poses between all ordered view pairs with a geodesic rotation loss and a Huber translation loss on scale-corrected predictions. We set $\lambda_n = 1$, $\lambda_c = 0.05$, $\lambda_p = 0.1$, and enable $\mathcal{L}_{\text{norm}}$ only after a five-epoch warm-up.

Per-sample altitude redefinition. Raw satellite poses in CrossGeo are anchored at $5{,}726$ m above ground (Section 3.1), and back-propagating through translations that large destabilizes training. We therefore re-anchor altitudes locally at dataset-preparation time. CrossGeo is collected as image pairs, each containing two ground, two UAV, and two satellite views of overlapping scene content; within a pair we take the lower of the two ground cameras as the reference and place it at $y = 0$. The second ground camera then sits at its relative altitude H'_{grd} above this reference (a small offset on uneven terrain), the two UAV cameras at their relative altitudes $H_{\text{drone}} \in [30, 120]$ m, and the two satellite cameras at a constant $H_{\text{sat}} = 150$ m, chosen just above the maximum UAV flight altitude (Figure 4).

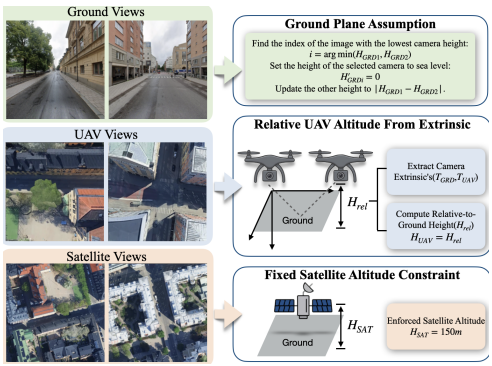


Figure 4: Per-sample altitude redefinition.

This keeps gradient magnitudes through the satellite translation bounded and gives the network a per-sample-constant satellite-to-ground depth prior that, in our experience, accelerates convergence.

4 Experiments

We benchmark Cross3R on two complementary task families. Section 4.1 evaluates *cross-altitude point cloud reconstruction and camera pose estimation*, where we recover dense geometry and the relative camera poses jointly across the satellite, UAV, and ground views. Section 4.2 then evaluates *cross-view ground-camera localization and pixel matching*, measuring how accurately the ground camera is placed on the satellite tile and how reliably pixel-level correspondences can be established across the three modalities. Section 4.3 additionally presents three ablation studies: one examining how the UAV pose—altitude, pitch, and horizontal distance to the ground camera—affects the UAV's contribution to ground-camera localization, another isolating the contribution of each architectural component, and a third measuring how Cross3R scales with the amount of CrossGeo training data.

Setup. We initialize Cross3R from the released π^3 [45] checkpoint, inheriting all weights of layers shared with π^3, and fine-tune the model on CrossGeo (Section 3.1). Despite the architectural changes described in Section 3.2, our additions are deliberately lightweight: Cross3R adds only $\sim 2.5\,\mathrm{M}$ parameters on top of π^3 ($961.20\,\mathrm{M}$ vs. $958.70\,\mathrm{M}$, a 0.26% increase), and per-triplet inference time is essentially unchanged ($137.8\,\mathrm{ms}$ vs. π^3's $138.5\,\mathrm{ms}$). Following π^3, fine-tuning runs in two stages on $8\times$ NVIDIA L40 GPUs: stage 1 uses 224×224 inputs for 6 hours with a per-GPU batch size of 64, and stage 2 uses variable resolutions in $[100{,}000, 255{,}000]$ pixels with aspect ratios drawn from $[0.5, 2.0]$ for 8 hours with a per-GPU batch size of 16. Each batch dynamically groups 2–3 frames; at inference, all images are resized to 504×504.

4.1 Cross-Altitude Point Cloud Reconstruction and Camera Pose Estimation

We first evaluate Cross3R on the CrossGeo test set (3,726 tri-view triplets, 504×504 inputs) for point-cloud quality and relative camera pose (Tab. 2). Following AerialMD [37], point-cloud quality is assessed with Acc-mean [11] ($\downarrow$, mean nearest-neighbor distance to ground truth) and $\delta@\tau$ [4] ($\uparrow$, fraction of points within $\tau \in \{0.5, 1, 2\}$ m); camera pose quality uses RRA and RTA [13] at $\theta \in \{5^\circ, 15^\circ, 25^\circ\}$ (relative rotation/translation-direction recall under θ) and AUC@30 [56] ($\uparrow$, normalized area under the cumulative pose-accuracy curve from 0° to 30°). Baselines include the recent feed-forward models VGGT [39] and π^3 [45], together with AerialMD [37]—a UAV/ground-specialized model obtained by fine-tuning Dust3R [43]. Because Cross3R builds on π^3, we additionally fine-tune π^3 on CrossGeo under the same schedule (denoted π^{3*}), such that $\pi^3 \to \pi^{3*}$ captures the contribution of our training data while $\pi^{3*} \to$ Cross3R isolates the effect of our architecture. Both π^{3*} and Cross3R are trained with our redefined satellite altitude instead of the raw map-tile altitude; as Tab. 6 shows, training with the raw altitude is unstable and led to divergence in our runs, so it is not a viable setting, and π^{3*} inherits the same redefinition. All numbers are obtained from a single forward pass per triplet, with no per-scene optimization.

Cross3R leads every column of Tab. 2. $\pi^3 \to \pi^{3*}$ alone delivers a large gain in both Acc-mean and AUC@30, attributing the bulk of the improvement to the CrossGeo data; $\pi^{3*} \to$ Cross3R then adds substantial further gains on RRA@5° and RTA@5°, attributing the remainder to the satellite-aware architecture. These same predicted point clouds and poses are reused, without any further processing, for the cross-view localization and pixel-matching evaluations in the next subsection.

4.2 Cross-View Camera Localization and Pixel Matching

We next evaluate cross-view localization—placing the ground camera on the satellite tile—and pixel-level matching across the three modalities. Cross3R produces these estimates directly from the predicted scale ρ and per-view poses, with no satellite intrinsics. We probe three datasets: in-distribution CrossGeo (3,726 triplets), out-of-distribution AnyVisLoc [55] (5,068 triplets, with the Google Maps and Baidu Maps modalities we add in Section 3.1), and zero-shot KITTI Test 2 [33] (n=7,542, no UAV view). For the KITTI dataset, following the cross-view benchmark of [33], we use the left-stereo image (90° horizontal FoV) paired with Google Maps tiles at $\sim 0.20\,\mathrm{m/px}$ that each cover roughly $100\,\mathrm{m} \times 100\,\mathrm{m}$, with the ground camera assumed to lie within a $40\,\mathrm{m} \times 40\,\mathrm{m}$ area at the tile center. Test 2 contains regions geographically disjoint from the training split, and we evaluate under

Table 2: Point cloud reconstruction and camera pose estimation on CrossGeo. π^{3*} denotes π^3 fine-tuned on CrossGeo. Best in **bold**, second-best underlined.

	Acc	δ (%)			RRA (%)			RTA (%)			AUC
Method	mean↓	0.5m	1m	2m	5°	15°	25°	5°	15°	25°	30°
AerialMD [37]	2.1334	28.48	48.56	70.12	3.00	47.00	62.50	7.50	54.50	70.00	35.41
VGGT [39]	2.3029	18.59	33.97	57.45	8.40	47.75	57.76	0.00	1.61	8.30	2.69
π^3 [45]	1.9818	24.86	42.31	65.47	26.45	64.51	72.97	1.34	16.43	32.44	15.98
π^{3*}	1.1771	39.18	61.40	83.55	86.99	95.40	96.15	29.93	93.28	98.98	71.87
Cross3R (Ours)	**1.0671**	**43.52**	**66.56**	**86.23**	**95.10**	**97.91**	**98.23**	**41.19**	**95.66**	**99.52**	**76.84**

Table 3: Cross-view ground- and UAV-camera localization on CrossGeo. Meter (m) is the on-tile translation error of the predicted camera position from ground truth; Yaw (°) is the rotation error around the up axis.

	Ground Camera						UAV Camera					
Method	Meter Mean↓	Meter Med↓	Yaw Mean↓	Yaw Med↓	PCK @2m↑	PCK @5m↑	Meter Mean↓	Meter Med↓	Yaw Mean↓	Yaw Med↓	PCK @2m↑	PCK @5m↑
AerialMD [37]	22.22	14.70	36.19	11.38	4.92	18.54	12.77	10.69	10.06	4.14	25.88	66.77
VGGT [39]	121.42	117.02	37.48	20.89	0.02	0.09	12.12	6.81	11.18	4.78	29.19	63.84
π^3 [45]	62.84	60.96	25.84	8.12	0.96	3.96	14.82	7.86	9.81	4.24	11.26	34.93
π^{3*}	7.16	5.19	6.51	2.05	11.04	42.62	12.06	8.77	2.99	1.42	8.66	33.33
Cross3R (Ours)	**3.68**	**2.14**	**3.66**	**1.30**	**42.60**	**76.34**	**2.38**	**1.64**	**1.92**	**0.87**	**67.75**	**91.23**

Table 4: Zero-shot cross-view ground-camera localization on KITTI Test 2 (n=7,542, unknown orientation; inputs: ground and satellite images only). Upper block: 2D-only baselines from CCVPE [50]; lower block: multi-view 3D reconstruction models. Best in **bold**, second-best underlined.

	Loc. (m)↓		Yaw (deg)↓		Lat. recall (%)↑			Lon. recall (%)↑			Ori. recall (%)↑	
Method	Mean	Median	Mean	Median	@1m	@3m	@5m	@1m	@3m	@5m	@1°	@3°
Traditional cross-view localization — 2D position only												
LM [33]	15.50	16.02	89.84	89.85	5.60	16.02	25.60	5.64	15.86	25.76	0.60	1.60
SliceMatch [17]	14.85	11.85	23.64	7.96	24.00	62.52	72.89	7.17	26.11	33.12	**31.69**	31.69
CCVPE [50]	13.94	10.98	77.84	63.84	23.42	49.15	60.46	**11.81**	**29.85**	**42.12**	4.39	14.09
Multi-view 3D reconstruction — camera pose + dense point cloud												
π^3 [45]	48.26	42.81	27.67	13.45	4.18	10.69	18.10	2.85	7.45	11.92	5.56	16.31
π^{3*}	16.28	14.85	**23.16**	4.09	9.44	27.61	44.18	4.34	12.86	21.36	15.27	40.27
Cross3R (Ours)	**11.69**	**10.52**	24.27	**3.51**	**39.11**	**76.32**	**85.95**	3.50	10.70	19.07	16.14	**44.90**

Table 5: Out-of-distribution cross-view ground- and UAV-camera localization on AnyVisLoc [55] (n=5,068; inputs: satellite, UAV, and ground images). Best in **bold**, second-best underlined.

	Ground Camera				UAV Camera			
Method	Meter-Mean↓	Meter-Med↓	Yaw-Mean↓	Yaw-Med↓	Meter-Mean↓	Meter-Med↓	Yaw-Mean↓	Yaw-Med↓
AerialMD [37]	48.90	43.23	59.81	42.56	40.17	23.25	71.62	69.40
VGGT [39]	100.51	97.57	30.61	17.33	71.77	56.39	44.25	26.54
π^3 [45]	57.36	51.80	15.76	7.00	58.31	51.82	59.20	28.56
π^{3*}	10.53	8.36	13.13	3.92	16.49	14.19	**38.11**	**11.24**
Cross3R (Ours)	**10.49**	**8.10**	**8.00**	**3.25**	**14.51**	**12.98**	39.39	12.27

the harder *no-orientation-prior* setting, where each aerial tile is rotated by a random angle drawn from the full 360° circular domain. On every dataset we report Meter-Mean/Med (m) and Yaw-Mean/Med (°); on CrossGeo we additionally report PCK@τ [53] for $\tau \in \{2, 5\}$ m, which doubles as a pixel-matching score; on KITTI Test 2 we follow CCVPE [50] with lateral/longitudinal recall at 1/3/5 m and orientation recall at 1°/3°. Beyond the multi-view 3D baselines from Sec. 4.1, we include three orientation-prior-free cross-view methods—LM [33], SliceMatch [17], and CCVPE [50]—trained directly on the KITTI cross-view split; their numbers are taken from the CCVPE paper.

On CrossGeo (Tab. 3) Cross3R leads all twelve columns: it halves π^{3*}'s ground translation error, nearly quadruples ground PCK@2m, and substantially shrinks the UAV translation error. On KITTI Test 2 (Tab. 4)—trained only on CrossGeo, never on KITTI—Cross3R still attains the lowest median translation error and the strongest lateral recall at every threshold, out-performing LM, SliceMatch, and CCVPE on most metrics despite their direct supervision on KITTI. On real-world AnyVisLoc (Tab. 5) Cross3R again has the best ground and UAV translation and the best ground yaw, with only UAV yaw trailing π^{3*} by a small margin. Fig. 6 shows geometrically consistent satellite–UAV–ground pixel correspondences in the wild and Fig. 5 the corresponding tri-view point clouds.

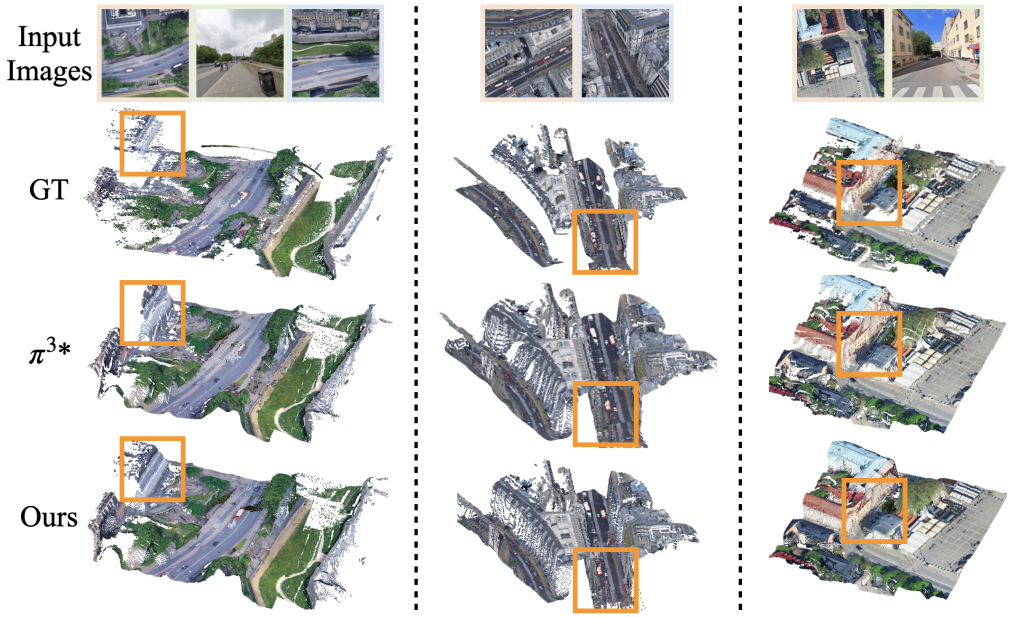


Figure 5: Predicted point clouds on three CrossGeo samples (left to right: satellite+ground+UAV, satellite+UAV, satellite+ground). Compared to π^3, Cross3R produces fewer holes and less misalignment in the regions highlighted by the orange boxes.

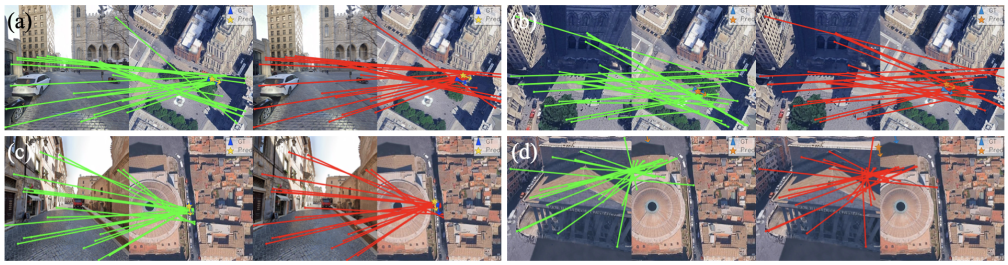


Figure 6: Cross-view localization and pixel matching on four CrossGeo samples (left two: ground–satellite; right two: UAV–satellite). Within each sample, the left column is Cross3R and the right is π^{3*}; green / red lines mark successful / failed matches. Cross3R produces more accurate results.

4.3 Ablation Studies

When does the UAV help ground-camera localization? For every test sample, we run Cross3R twice—once with the full satellite–UAV–ground triplet and once with only satellite and ground—and measure the reduction in ground-camera error when the UAV is present (Figure 7). On average, including the UAV lowers both the median translation error and the median yaw error; this is also corroborated by Tab. 6. The benefit depends on geometry: the UAV helps most when it flies at a moderate altitude, adopts a near-nadir viewpoint, and remains within a short horizontal distance of the ground camera. This contribution directly verifies our central insight—the UAV acts as a geometric bridge that supplies the vertical structure missing from satellite-only views, so cross-view ground-camera localization works reliably on a much wider range of scenes than before.

Architectural components. We isolate each design choice by training a leave-one-out variant of Cross3R on CrossGeo (Table 6). *Removing the orthographic head (ρ)* leaves Acc-mean nearly unchanged but causes cross-view localization to collapse—both UAV and ground translation errors increase sharply—confirming that ρ is the essential link that ties the three branches into a shared, scale-consistent coordinate frame. *Restoring the raw satellite altitude* ($\sim$5,715 m, i.e., undoing the altitude redefinition) is catastrophic on every metric: the satellite branch must regress depths above 5,000 m while the UAV and ground branches operate within a few hundred meters, and the satellite camera sits two orders of magnitude higher than the perspective cameras—a depth and translation span the shared backbone cannot learn to bridge. The remaining four modules—multi-scale decoder

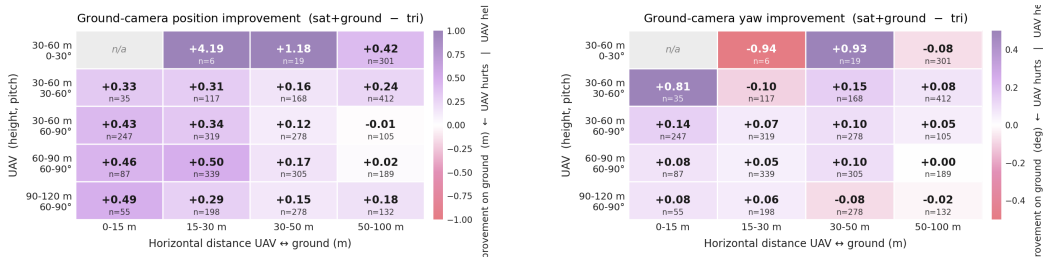
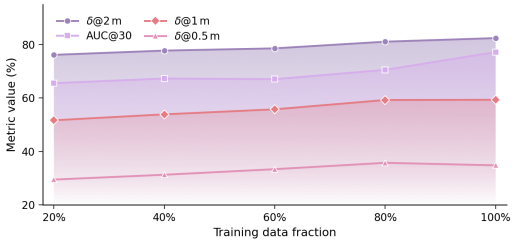


Figure 7: Per-cell value is $\text{metric}_{\text{sat+grd}} - \text{metric}_{\text{tri}}$ for ground position (*left*, m) and ground yaw (*right*, °); red = the UAV helps, blue = the UAV hurts.

Table 6: Leave-one-out architectural and view-mode ablations on CrossGeo full test (n=3,736). Lower is better for every column.

		Ground		UAV	
Method	Acc-mean↓	Met.Med↓	Yaw Med↓	Met.Med↓	Yaw Med↓
Cross3R	**1.07**	**2.14**	**1.30**	**1.64**	0.87
w/o ortho	1.09	4.04	1.38	7.33	0.77
w/o ms-fusion	1.16	2.24	1.35	1.79	0.82
w/o sat-pos-embed	1.16	2.20	1.35	1.76	**0.73**
w/o sat-inject	1.12	2.36	1.61	1.83	0.78
w/o double register	1.12	2.43	1.39	1.80	0.82
w/o altitude-redef	3.89	7.37	1.48	12.00	0.77
w/o UAV (sat+grd)	1.28	2.48	1.37	–	–
w/o ground (sat+UAV)	1.34	–	–	1.84	0.80

Figure 8: Cross3R data-scaling on CrossGeo (lowres 224×224, n=3,736).



fusion, Fourier positional embedding, satellite-to-perspective feature injection, and the doubled register-token bank—each bring a modest but consistent gain in point-cloud quality and ground/UAV translation; the injection module further improves ground-yaw accuracy. The bottom two rows of Table 6 evaluate the same checkpoint after dropping one perspective view at inference: excluding the UAV substantially degrades ground localization in both translation and yaw, whereas excluding the ground view only marginally affects UAV localization, mirroring the asymmetry seen in the heatmaps above (Fig. 7) and consistent with the wider satellite–ground viewpoint gap.

Data scaling. To validate the effectiveness of our dataset and assess the model's scaling potential, we train Cross3R at low resolution (224×224) on 20% - 100% of the training data (Figure 8). Almost every metric improves monotonically, and the curves show no sign of saturating even at the full training set. In particular, from 80% to 100% data, the AUC@30 metric still exhibits a substantial gain. These trends indicate that our model has considerable headroom to benefit from additional data and confirm that our data acquisition pipeline is effective.

5 Conclusion

We presented **Cross3R**, a feed-forward model that jointly recovers dense 3D point clouds, 6-DoF camera poses, and on-tile cross-view localization from a satellite tile paired with a UAV image, a ground image, or both. By introducing an intermediate UAV view as a geometric bridge, re-anchoring per-sample altitudes, and predicting a learned per-tile scale ρ in place of fragile satellite intrinsics, Cross3R lifts cross-view localization beyond planar 3-DoF assumptions. Supported by our 278K-image **CrossGeo** dataset across 85 global scenes, Cross3R consistently outperforms feed-forward 3D and dedicated cross-view baselines on CrossGeo, KITTI, and AnyVisLoc.

Limitations and future work. Several limitations point to clear directions for future work. First, limited training compute restricts each forward pass to at most three views; scaling to longer multi-view sequences will enable larger-area streaming reconstruction. Second, Cross3R currently assumes pinhole inputs, whereas extending the backbone and geometric heads to panoramic (360°) and fisheye cameras is important for autonomous driving and mobile robotics. Third, our data-scaling curves show substantial headroom for further scaling, and current map and street-view imagery is mostly captured in clear daytime conditions, leaving cross-view matching sensitive to harsh illumination or appearance shifts (Appendix C). We therefore plan to expand CrossGeo with broader geographic scenes and diverse weather (rain, fog, snow), lighting (dawn, dusk, nighttime, strong shadows), and seasonal environments for all-condition robustness.

Acknowledgments and Disclosure of Funding

The authors are grateful for the valuable comments and suggestions by the reviewers and ACs. We also thank Shaoxun Wu, Yijing Guo, and Mengjun Chao from ShanghaiTech University for their helpful discussions and support. This work was supported by NSFC (62406194), Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI), MoE Key Laboratory of Intelligent Perception and Human-Machine Collaboration (KLIP-HuMaCo), and HPC Platform of ShanghaiTech University. A part of the experiments of this work were supported by the core facility Platform of Computer Science and Communication, SIST, ShanghaiTech University.

References

[1] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[2] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024.

[3] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15324–15333, 2021.

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

[5] Sven Elflein, Qunjie Zhou, and Laura Leal-Taixé. Light3r-sfm: Towards feed-forward structure-from-motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16774–16784, 2025.

[6] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

[8] Yijing Guo, Mengjun Chao, Luo Wang, Tianyang Zhao, Haizhao Dai, Yingliang Zhang, Jingyi Yu, and Yujiao Shi. Panovggt: Feed-forward 3d reconstruction from panoramic imagery. *arXiv preprint arXiv:2603.17571*, 2026.

[9] Vu Hoang Hiep, Renaud Keriven, Patrick Labatut, and Jean-Philippe Pons. Towards high-resolution large-scale multi-view stereo. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1430–1437. IEEE, 2009.

[10] Sergio Izquierdo, Mohamed Sayed, Michael Firman, Guillermo Garcia-Hernando, Daniyar Turmukhambetov, Javier Civera, Oisin Mac Aodha, Gabriel Brostow, and Jamie Watson. Mvsanywhere: Zero-shot multi-view stereo. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11493–11504, 2025.

[11] 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*, pages 406–413, 2014.

[12] Yuxiang Ji, Boyong He, Zhuoyue Tan, and Liaoni Wu. Game4loc: A uav geo-localization benchmark from game data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3913–3921, 2025.

[13] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

[14] Neil Joshi, Joshua Carney, Nathanael Kuo, Homer Li, Cheng Peng, and Myron Brown. Ultrra challenge 2025, 2024. URL `https://dx.doi.org/10.21227/2zs6-ht63`.

[15] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

[16] Wooju Lee, Juhye Park, Dasol Hong, Changki Sung, Youngwoo Seo, Dongwan Kang, and Hyun Myung. Pidloc: Cross-view pose optimization network inspired by pid controllers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21981–21990, 2025.

[17] Ted Lentsch, Zimin Xia, Holger Caesar, and Julian FP Kooij. Slicematch: Geometry-guided aggregation for cross-view pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17225–17234, 2023.

[18] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[19] Yaxuan Li, Yewei Huang, Bijay Gaudel, Hamidreza Jafarnejadsani, and Brendan Englot. Cvd-sfm: A cross-view deep front-end structure-from-motion system for sparse localization in multi-altitude scenes. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10741–10748. IEEE, 2025.

[20] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):2020–2036, 2024.

[21] 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.

[22] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5987–5997, 2021.

[23] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5624–5633, 2019.

[24] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16651–16662, 2025.

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[26] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024.

[27] Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Bulo, Richard Newcombe, Peter Kontschieder, and Vasileios Balntas. OrienterNet: Visual Localization in 2D Public Maps with Neural Matching. In *CVPR*, 2023.

[28] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. Snap: Self-supervised neural maps for visual positioning and semantic understanding. *Advances in Neural Information Processing Systems*, 36:7697–7729, 2023.

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

[30] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[31] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017.

[32] Yujiao Shi and Hongdong Li. Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17010–17020, 2022.

[33] Yujiao Shi, Xin Yu, Liu Liu, Dylan Campbell, Piotr Koniusz, and Hongdong Li. Accurate 3-dof camera geo-localization via ground-to-satellite image matching. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):2682–2697, 2022.

[34] Yujiao Shi, Hongdong Li, Akhil Perincherry, and Ankit Vora. Weakly-supervised camera localization by ground-to-satellite image registration. In *European Conference on Computer Vision*, pages 39–57. Springer, 2024.

[35] Zhenbo Song, Jianfeng Lu, Yujiao Shi, et al. Learning dense flow field for highly-accurate cross-view camera localization. *Advances in Neural Information Processing Systems*, 36: 70612–70625, 2023.

[36] Shaowen Tong, Zimin Xia, Alexandre Alahi, Xuming He, and Yujiao Shi. Geodistill: Geometry-guided self-distillation for weakly supervised cross-view localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25357–25366, 2025.

[37] Khiem Vuong, Anurag Ghosh, Deva Ramanan, Srinivasa Narasimhan, and Shubham Tulsiani. Aerialmegadepth: Learning aerial-ground reconstruction and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21674–21684, 2025.

[38] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024.

[39] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[40] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[41] Qiwei Wang, Shaoxun Wu, and Yujiao Shi. Bevsplat: Resolving height ambiguity via feature-based gaussian primitives for weakly-supervised cross-view localization. *arXiv preprint arXiv:2502.09080*, 2025.

[42] Shan Wang, Chuong Nguyen, Jiawei Liu, Yanhao Zhang, Sundaram Muthu, Fahira Afzal Maken, Kaihao Zhang, and Hongdong Li. View from above: Orthogonal-view aware cross-view localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14843–14852, 2024.

[43] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024.

[44] Xiaolong Wang, Runsen Xu, Zhuofan Cui, Zeyu Wan, and Yu Zhang. Fine-grained cross-view geo-localization using a correlation-aware homography estimator. *Advances in Neural Information Processing Systems*, 36:5301–5319, 2023.

[45] 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. *arXiv preprint arXiv:2507.13347*, 2025.

[46] Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025.

[47] Zhaoying Wang, Xingxing Zuo, and Wei Dong. Flying co-stereo: Enabling long-range aerial dense mapping via collaborative stereo vision of dynamic-baseline. *IEEE Transactions on Robotics*, 2026.

[48] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–9, 2015. doi: 10.1109/ICCV.2015.451. Acceptance rate: 30.3%.

[49] Zimin Xia and Alexandre Alahi. Fgˆ 2: Fine-grained cross-view localization by fine-grained feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6362–6372, 2025.

[50] Zimin Xia, Olaf Booij, and Julian FP Kooij. Convolutional cross-view pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3813–3831, 2023.

[51] Wenjia Xu, Yaxuan Yao, Jiaqi Cao, Zhiwei Wei, Chunbo Liu, Jiuniu Wang, and Mugen Peng. Uav-visloc: A large-scale dataset for uav visual localization. *arXiv preprint arXiv:2405.11936*, 2024.

[52] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025.

[53] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392. IEEE, 2011.

[54] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[55] Yibin Ye, Xichao Teng, Shuo Chen, Zhang Li, Leqi Liu, Qifeng Yu, and Tao Tan. Exploring the best way for uav visual localization under low-altitude multi-view observation condition: a benchmark. *arXiv preprint arXiv:2503.10692*, 2025.

[56] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2666–2674, 2018.

[57] Qitao Zhao, Amy Lin, Jeff Tan, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Diffusionsfm: Predicting structure and motion via ray origin and endpoint diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6317–6326, 2025.

[58] Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In *Proceedings of the 28th ACM international conference on Multimedia*, pages 1395–1403, 2020.

[59] Runzhe Zhu, Ling Yin, Mingze Yang, Fei Wu, Yuncheng Yang, and Wenbo Hu. Sues-200: A multi-height multi-scene cross-view image benchmark across drone and satellite. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2023. doi: 10.1109/TCSVT.2023.3249204.

[60] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2021.

Supplementary Contents

A CrossGeo Dataset Details **16**

A.1 Dataset Overview 16

A.2 Dataset Coverage 16

A.3 Street-View Image Release Policy 16

A.4 Depth Acquisition Pipeline 17

B Satellite Altitude Ablation **19**

C Failure Cases on CrossGeo **19**

D Qualitative Results on KITTI **20**

E Qualitative Results on AnyVisLoc **20**

F Broader Impact **21**

A CrossGeo Dataset Details

A.1 Dataset Overview

To provide an intuitive overview of the CrossGeo dataset, Figure 9 visualizes its five core modalities across three representative scene types: **Urban**, **Hilly**, and **Rural**. The Urban scene is characterized by dense high-rise buildings and bustling streets; the Hilly scene features undulating terrain with a mix of low-rise structures and natural vegetation; and the Rural scene consists of flat open land with sparse, low-rise constructions. This diversity in urban density, topography, and architectural styles directly supports the "Multiple" scenes claim for CrossGeo in Table 1.

A.2 Dataset Coverage

Figure 10 visualizes CrossGeo's tri-view coverage at the sample level along with its global geographic distribution and UAV-pose statistics, complementing the data-source overview in Figure 2 of the main paper. Panel (a) shows a representative tri-view sample, in which the satellite (yellow), UAV (green), and ground (red) cameras are placed in a shared metric world frame, each carrying its own depth map and 6-DoF pose. Panel (b) shows the geographic coverage: the red markers and highlighted countries mark the 85 scenes, spanning every continent except Antarctica and a wide mix of urban, suburban, rural, and natural terrain. Panel (c) summarizes the UAV pose distribution per scenario—flight altitudes are dominated by the 30–60 m bin (about 1.8 k samples), with smaller bins at 60–90 m and 90–120 m (about 0.6 k each); pitch angles span 0°–90°, with the low-pitch range (0°–30°) oversampled to mitigate the pixel stretching that high-altitude high-pitch renderings produce.

A.3 Street-View Image Release Policy

The street-level ground images in CrossGeo and AnyVisLoc are pulled from Google Maps[3] and Baidu Maps[4] through their official APIs. Because of privacy protection and the providers' terms of service, we are not allowed to redistribute these street images ourselves. Instead, for every ground sample we release the image's `pano_id` (the unique ID that Google and Baidu assign to each street image), along with the camera pose and capture metadata we used. With this `pano_id`, anyone who

[3] `https://www.google.com.hk/earth/education/tools/street-view/`

[4] `https://lbsyun.baidu.com/products/map`

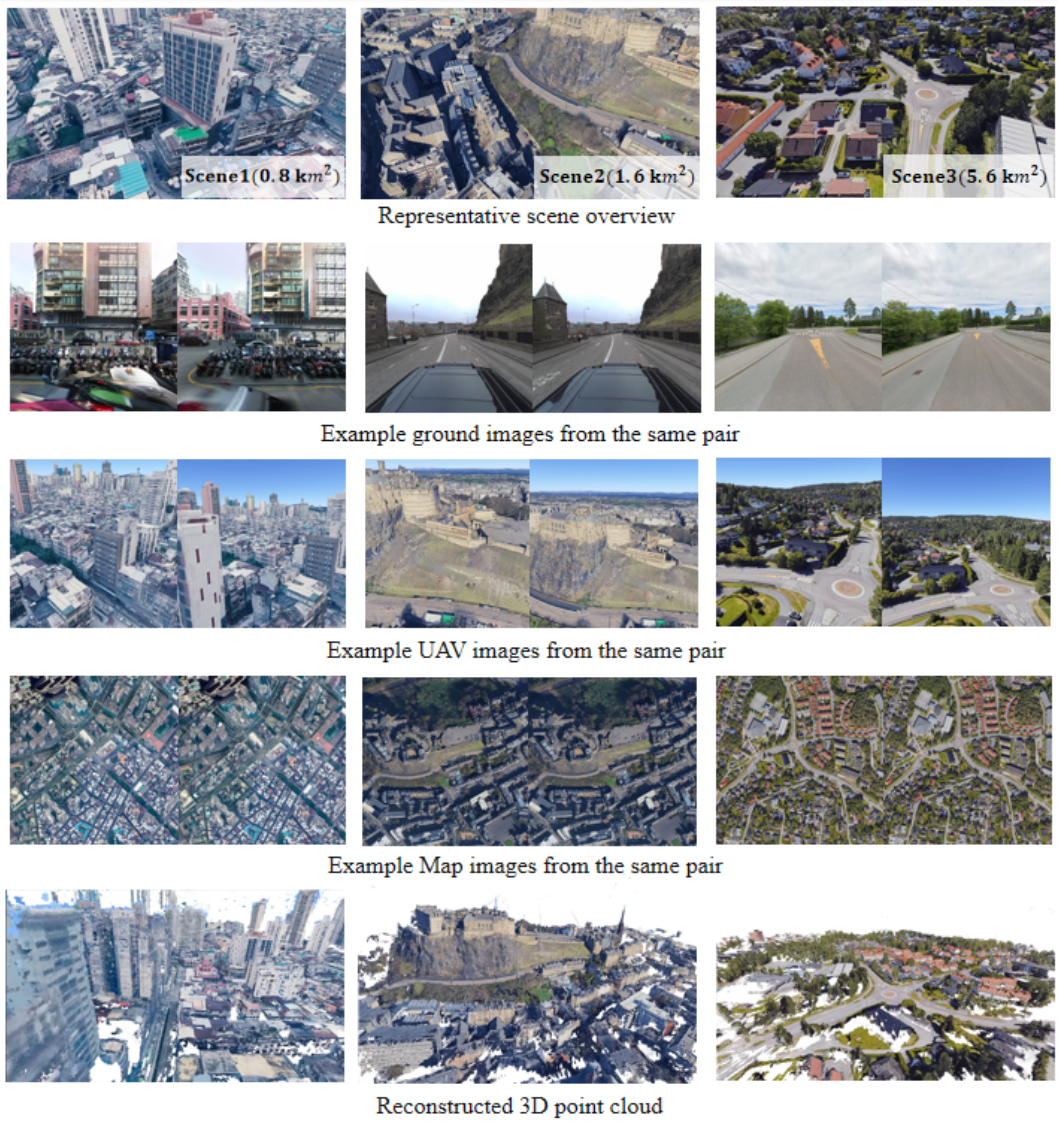


Figure 9: Overview of the five core modalities in the CrossGeo dataset. Row 1: representative scene overviews from three selected environments (Urban, Hilly, Rural), annotated with their coverage areas (0.8 km^2, 1.6 km^2, 5.6 km^2). Row 2: example ground images from the same triplet, showing dense buildings (Urban), undulating terrain and varied structures (Hilly), and flat open areas with low-rise constructions (Rural). Row 3: example UAV images from the same triplet, illustrating oblique aerial views across the three scene types. Row 4: example satellite map images (orthographic tiles) from the same triplet, providing top-down context. Row 5: reconstructed 3D point clouds for the corresponding scenes, visualizing the joint geometry recovered from satellite, UAV, and ground views.

has API access can call the same Google Maps or Baidu Maps endpoint, follow each provider's usage rules, and download exactly the same street image we trained and evaluated on. The UAV captures, satellite tiles, and all derived depth maps and poses are released directly as part of CrossGeo.

A.4 Depth Acquisition Pipeline

Figure 11 details the full processing pipeline we use to recover dense metric depth for every modality and to register the three views into a shared coordinate frame. *UAV branch.* Captures are processed by a calibrated, posed Structure-from-Motion stage [29] with known ray directions and camera

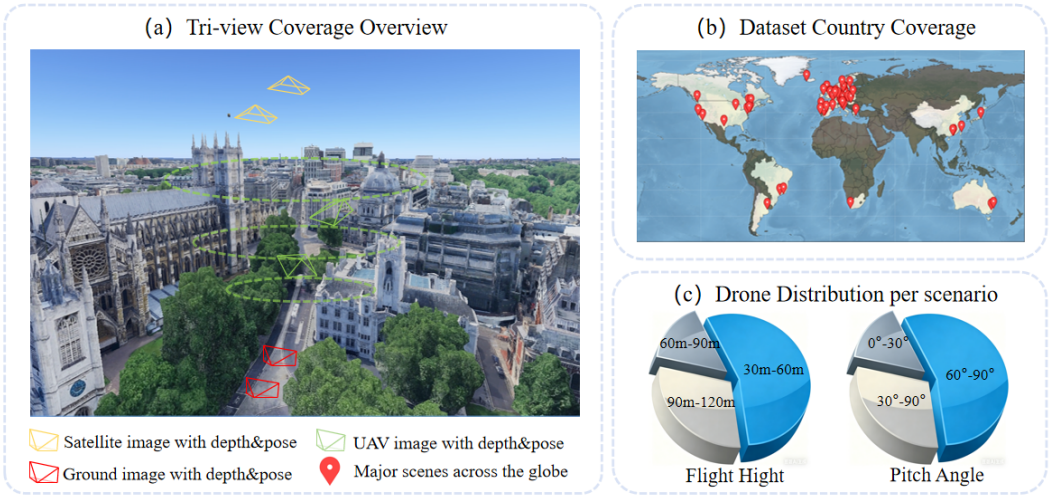


Figure 10: CrossGeo tri-view coverage and statistics: (a) a representative tri-view sample, (b) global geographic distribution of the 85 scenes, and (c) UAV altitude and pitch distributions.

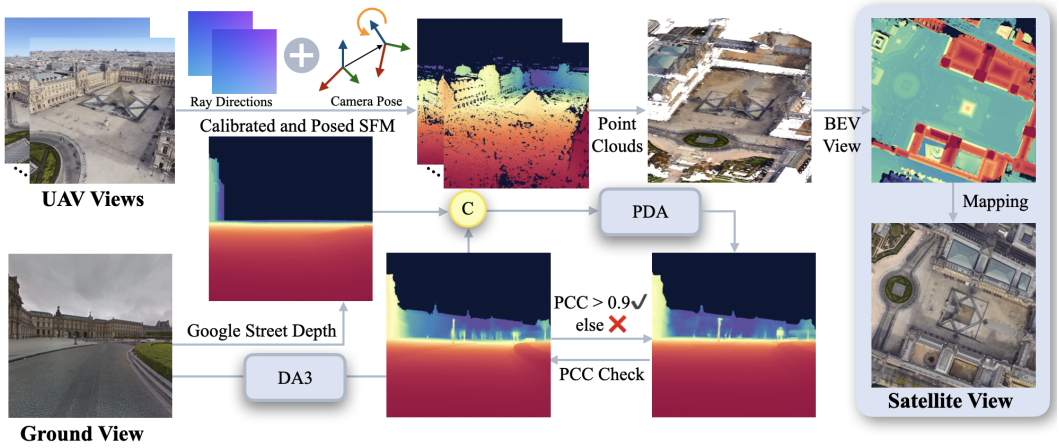


Figure 11: CrossGeo depth-acquisition and tri-view alignment pipeline: each modality produces a dense metric depth map and is registered into a shared coordinate frame.

poses, producing a dense metric UAV point cloud whose BEV projection yields per-pixel UAV depth. *Ground branch.* Each view is paired with the coarse, absolute-scale depth from *Google Street View* and refined by a monocular depth backbone (DA3) [21]; the two depth signals are concatenated (**C**) and fused into a metric ground depth by a pose–depth alignment module (**PDA**) [46] that uses the Street View depth as a metric anchor for the high-resolution but relative DA3 prediction. *Satellite branch.* Tiles are sourced from *Google Maps*; we associate them with the UAV point cloud's BEV depth to obtain a metric depth for each satellite pixel. The output is paired (satellite, UAV, ground) views with consistent poses and dense metric depth across all three modalities.

Figure 12 shows representative qualitative results of the ground-depth fusion together with our PCC-based quality filter. The top two rows are successful samples whose rescaled depth has a Pearson correlation [1] greater than 0.9 with the relative DA3 [21] prediction; the geometry of buildings, road surface, and street furniture is faithfully recovered and the sample is kept. The bottom row is a failure case with PCC below 0.9 and is therefore discarded: the Google Street View depth on which we anchor the metric scale lacks any depth values for the buildings, so the fused ground depth collapses to an incorrect plane that disagrees with the relative DA3 prediction in regions where buildings dominate the view.

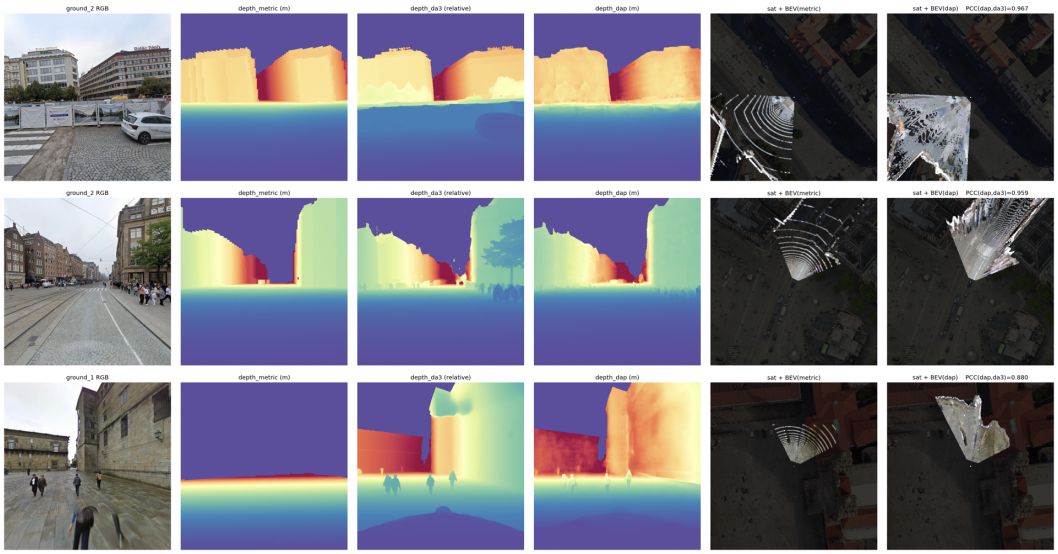


Figure 12: Qualitative ground-depth samples and the PCC quality filter. *Top two rows*: successful fusions (PCC$>$0.9) that are kept. *Bottom row*: a failure case (PCC$<$0.9) that is discarded, caused by missing building depth in the Google Street View anchor.

Table 7: Satellite-altitude ablation on CrossGeo (lowres 224×224, n=3,736). All rows share the Cross3R architecture and the same compute budget; only the redefined satellite altitude changes. Best in **bold**.

	Point cloud				Camera pose (%)				
Altitude	Acc-mean↓	δ@0.5m↑	δ@1m↑	δ@2m↑	RRA@5°	RRA@15°	RTA@5°	RTA@15°	AUC@30
100 m	**1.27**	35.06	59.29	82.35	**84.85**	**93.92**	1.39	52.81	44.10
150 m **(Cross3R)**	**1.27**	35.25	**59.93**	82.21	84.64	92.37	**70.85**	**97.11**	**79.85**
200 m	**1.27**	34.79	59.32	**82.41**	83.89	92.16	60.25	95.48	77.18

B Satellite Altitude Ablation

Training is unstable at the raw 5,726 m tile altitude (Tab. 6), so before training we have to pick a redefined satellite altitude. We try three values—100, 150, and 200 m—and retrain Cross3R end-to-end at each setting, keeping everything else the same and evaluating on the same CrossGeo test set (Table 7). Point-cloud quality stays the same across all three runs, so this choice does not really affect the 3D geometry. Camera pose, on the other hand, is very sensitive to the altitude. At 100 m the satellite virtual camera ends up inside the UAV altitude range (30–120 m), so the satellite no longer offers a clear top-down view that is different from the UAV; the model loses its top-down anchor and translation accuracy drops sharply. At 200 m the satellite sits far above the UAV and ground views, and the gap between the satellite depth range and the perspective depth range becomes large enough that the model struggles to match them in scale. 150 m sits between these two cases: it is high enough to stay above every UAV altitude in our data, and low enough that the satellite, UAV, and ground branches still share a similar depth range—the condition Tab. 6 already showed to be necessary. We therefore use 150 m as the redefined satellite altitude in every main-paper experiment.

C Failure Cases on CrossGeo

Figure 13 contrasts a failure case (top row) with a successful case (bottom row) drawn from the CrossGeo test set. *Top row.* When the ground image lacks clear localization cues—for example, a long, straight street with little distinctive structure and poor lighting that washes out the road surface—Cross3R can be drawn to a visually similar but geometrically wrong location on the satellite tile. This is a well-known and largely unavoidable failure mode of any cross-view localization method, since under such conditions the ground image is genuinely ambiguous with respect to the satellite tile.

Bottom row. When the ground view is well-lit and contains discriminative structures (curbs, building facades, road markings), Cross3R produces both an accurate ground-camera pose and a high-quality reconstruction: the ground point cloud aligns almost seamlessly with the satellite tile, illustrating that the model can fully exploit reliable visual cues when they are available.

D Qualitative Results on KITTI

Figure 14 visualizes four KITTI [7] samples (a)–(d): each sample's left half shows ground-to-satellite localization and pixel matching, the right half the stacked point cloud. The ground reconstruction lands cleanly on top of the satellite branch, and is nearly seamless in (a) and (c). All predictions are zero-shot—Cross3R is never trained on KITTI—corroborating Tab. 4 of the main paper.

E Qualitative Results on AnyVisLoc

Figure 15 visualizes Cross3R on four AnyVisLoc [55] samples; for each sample the top-left panel shows ground-to-satellite localization and pixel matching, the bottom-left panel the corresponding UAV-to-satellite result, and the right half the predicted tri-view point cloud. Although Cross3R wins most quantitative columns on AnyVisLoc (Tab. 5), the qualitative point clouds are noticeably noisier than on CrossGeo and KITTI for two reasons specific to this benchmark: the Google Maps tiles in these regions are served at a lower resolution and the buildings are visually repetitive, so the satellite branch carries less localization signal; and most AnyVisLoc UAV photographs are captured above 120 m, outside CrossGeo's 30–120 m training range. Cross3R nonetheless produces a coherent tri-view layout, which we consider acceptable for a feed-forward model evaluated entirely out of distribution.

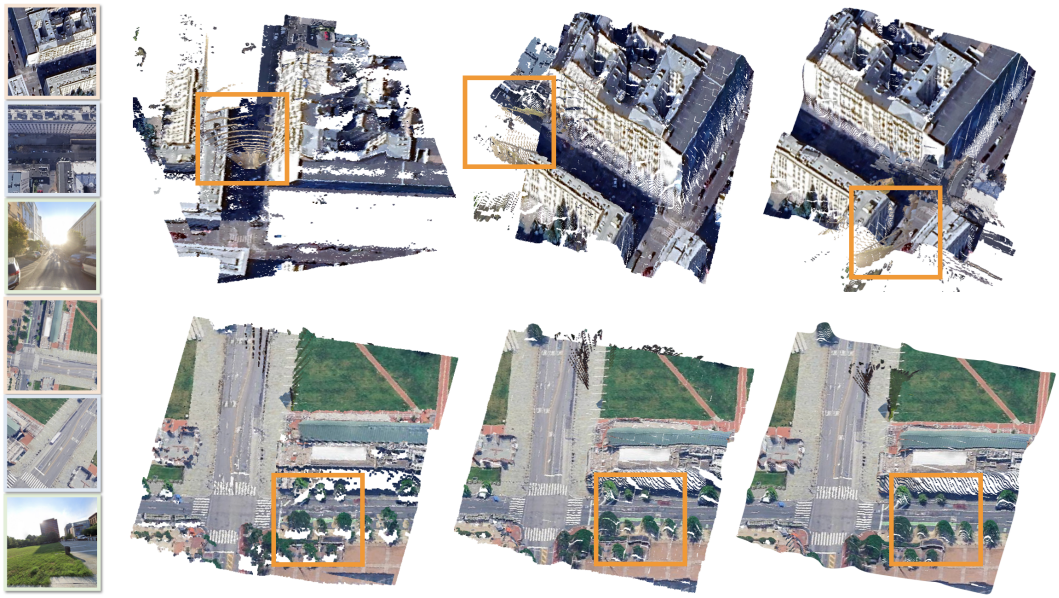


Figure 13: Failure (*top row*) and success (*bottom row*) cases of Cross3R on CrossGeo.

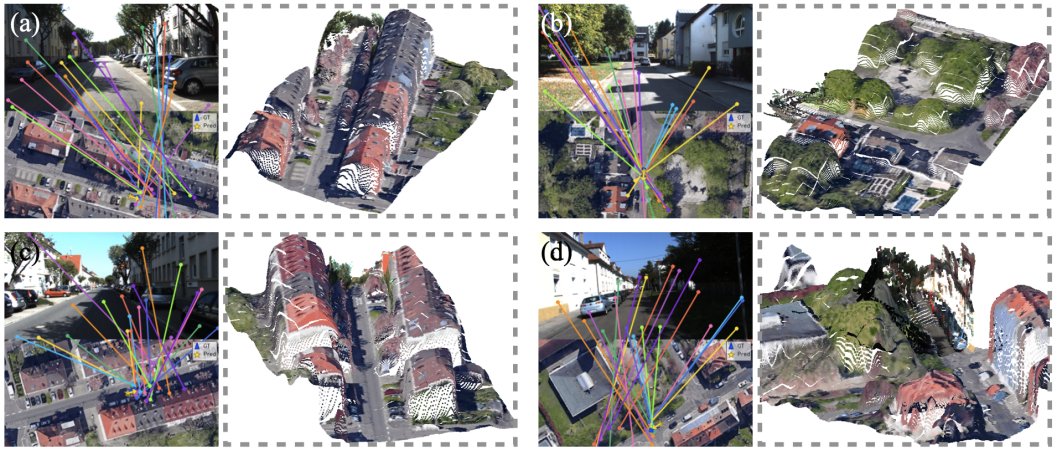


Figure 14: Zero-shot cross-view localization, pixel matching, and tri-view reconstruction on KITTI, samples (a)–(d). *Left of each panel*: ground-to-satellite localization and pixel-matching. *Right of each panel*: corresponding stacked point cloud.

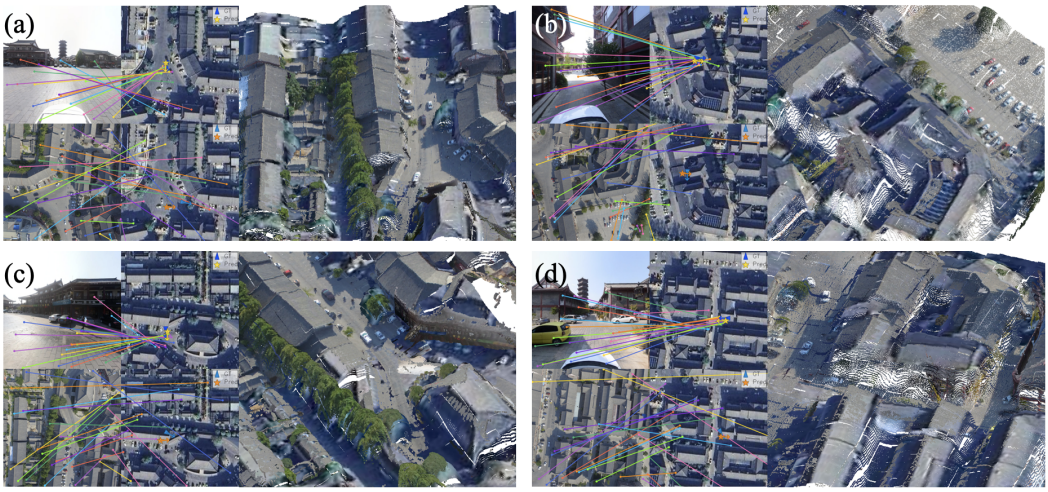


Figure 15: Out-of-distribution cross-view localization, pixel matching, and tri-view reconstruction on AnyVisLoc. For each sample: *top-left*: ground-to-satellite, *bottom-left*: UAV-to-satellite, *right*: stacked tri-view point cloud.

F Broader Impact

Cross3R aims to push cross-view ground-camera localization beyond the 3-DoF limit of prior methods, enabling reliable operation in real-world settings where the ground camera may be tilted or placed on uneven terrain. We envision applications such as autonomous driving on rough roads, robot navigation in unfamiliar environments, and search-and-rescue missions where a ground team collaborates with a drone. At the same time, accurate cross-view localization can be misused: a single ground photo could be localized without consent, enabling unwanted tracking or surveillance. To mitigate this risk, we built CrossGeo without any personally identifiable annotations. We also do not redistribute the original Google Street View images; we release only the `pano_id` for each image, so that anyone using the data must fetch it through Google's official API and comply with Google's terms of service. We urge those building upon Cross3R to respect the privacy and consent policies of the data providers and the legal requirements of the regions where the model is deployed.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sec. 1 state three contributions: (i) a feed-forward cross-view reconstruction model that outputs cross-view point clouds, 6-DoF camera poses, and cross-view localization in a single forward pass; (ii) flexible inputs (sat+ground, sat+UAV, or sat+UAV+ground) and the empirical finding that the UAV image substantially improves ground-to-satellite localization; (iii) the new CrossGeo dataset. All three are supported by the experiments in Sec. 4 (Tabs. 2–5, Tab. 6, Fig. 7, Fig. 8).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Sec. 5 discusses the main limitations: each forward pass currently handles at most three pinhole images due to compute, the network supports only pinhole inputs (not panoramic or fisheye), and the model is data-hungry with clear headroom as the dataset scales. Sec. 3.2 additionally clarifies that the predicted scale ρ is a learned relative quantity rather than a true meters-per-pixel, so Cross3R recovers a self-consistent scene rather than an absolute metric reconstruction.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results; the contributions are an empirical model architecture (Cross3R) and a new dataset (CrossGeo), evaluated experimentally.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 3.1 fully documents how CrossGeo is constructed (data sources, pose recovery, depth pipeline, train/val/test split). Sec. 3.2 describes the Cross3R architecture (encoder modifications, alternating-attention fusion, decoding heads, loss functions, ρ regression, altitude redefinition). Sec. 4 reports the training schedule (hardware, batch sizes, two-stage curriculum, per-stage durations and resolutions). Together with the released CrossGeo `pano_id` pointers and depth-pipeline description in the supplementary (Sec. A.3, A.4), this provides sufficient information for a third party to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and pretrained weights will be released upon acceptance. The CrossGeo dataset will be released alongside, with the modality split described in Sec. A.3: the rendered UAV captures, Google Maps satellite tiles, derived dense metric depth maps, and 6-DoF camera poses are released directly, while the Google Street View ground images are not redistributed—we release the per-sample `pano_id` together with the camera pose and capture metadata, so that downstream users can fetch identical images through the official Google API under the providers' terms of service.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4 (Setup) reports the initialization (released π^3 checkpoint), parameter count, per-triplet inference time, two-stage fine-tuning curriculum (resolution, duration, per-GPU batch size, dynamic frame grouping), and inference-time resizing. Sec. 3.2 specifies the loss formulation, the four loss weights (λ_n, λ_c, λ_p), and the warm-up schedule for $\mathcal{L}_{\text{norm}}$. Sec. 3.1 describes the scene-level train/val/test split. Evaluation metrics and the test-set

definitions (CrossGeo n=3,726, KITTI Test 2 n=7,542, AnyVisLoc n=5,068) are detailed in Sec. 4.1 and Sec. 4.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We do not report error bars on the main experiments. Each Cross3R / Pi3* training run requires substantial compute on $8\times$ NVIDIA L40 GPUs (Sec. 4, Setup), making multi-seed retraining for every entry impractical, which is consistent with the convention in recent feed-forward 3D reconstruction work (DUSt3R [43], VGGT [39], π^3 [45]). All reported numbers are deterministic single-run evaluations on the same fixed test sets. The within-sample paired analysis in Sec. 4.3 (Fig. 7) and the data-scaling sweep (Fig. 8) further mitigate single-run noise by varying one factor at a time on the same seed.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Sec. 4 (Setup) reports compute details: fine-tuning runs on $8\times$ NVIDIA L40 GPUs, with stage-1 (224×224, per-GPU batch 64) taking 6 hours and stage-2 (variable resolution in $[100{,}000, 255{,}000]$ pixels, per-GPU batch 16) taking 8 hours; per-triplet inference is 137.8 ms versus π^3's 138.5 ms. The data-scaling experiments in Sec. 4.3 hold the gradient-step budget ($\sim$5,100 steps) fixed, as detailed in Sec. B of the supplementary.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and our work conforms in all relevant respects. CrossGeo is built from publicly accessible map and street-imagery APIs (Google Maps, Google Earth, Google Street View) and contains no personally identifiable annotations; we do not redistribute the original Street View ground images and instead release only `pano_id` pointers (Sec. A.3). Privacy and dual-use considerations are discussed in Sec. F.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Sec. F discusses both sides: positive uses (autonomous driving on uneven roads, robot navigation in unfamiliar areas, ground-and-drone search-and-rescue) and the misuse risk that accurate cross-view localization could enable tracking or surveillance from a single ground photo. We outline the mitigations we adopt for the dataset (no personally identifiable annotations; `pano_id`-only release for Street View ground images; downstream users must obtain images through the official Google API and follow its terms of service) and recommend that downstream users follow the privacy and consent rules of the data providers and relevant jurisdictions.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Cross3R is a feed-forward 3D reconstruction model rather than a generative image model or a large language model, so it does not pose generative misuse risks (e.g., deepfakes). The dataset-side risk—that accurate cross-view localization could be misused for tracking—is mitigated by the release policy in Sec. A.3: CrossGeo carries no personally identifiable annotations, the Street View ground images are not redistributed, and only `pano_id` pointers are released so that downstream users must obtain the imagery through Google's official API under its terms of service. Sec. F additionally discusses these privacy and dual-use considerations.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets used in the paper are properly cited. Cross3R is built on the released π^3 [45] checkpoint and compared against VGGT [39], AerialMD [37], LM [33], SliceMatch [17], and CCVPE [50], all cited at first use. Depth predictions use DEPTHANYTHING V3 [21] and *Prior Depth Anything* [46]; UAV depth uses COLMAP [29]. Datasets are cited where used: KITTI [7] and AnyVisLoc [55] for evaluation. The CrossGeo modalities sourced from Google Maps, Google Earth, and Google Street View are obtained through the providers' official APIs and used under their respective terms of service; we do not redistribute the Street View ground images and instead release only `pano_id` pointers (Sec. A.3).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The two new assets are the Cross3R model and the CrossGeo dataset, both planned for release upon acceptance (Sec. A.3). Cross3R is fully documented architecturally (Sec. 3.2), including the encoder modifications, alternating-attention fusion, decoding heads, loss functions, and the per-sample altitude redefinition. CrossGeo is documented in Sec. 3.1 and the supplementary (Sec. A–A.3), covering data collection, pose recovery, depth pipeline, train/val/test split, scene composition, UAV-pose distribution, and the per-modality release policy. The dataset is built entirely from publicly accessible APIs without any human-subject involvement, so no informed-consent procedure was applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our work does not involve any crowdsourcing or human-subject research. CrossGeo is constructed entirely from publicly accessible map and street-imagery APIs (Google Maps, Google Earth, Google Street View) and from the existing AnyVisLoc benchmark; no human annotators were employed.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not involve human-subject research, so IRB review is not applicable. CrossGeo is constructed entirely from publicly accessible map and street-imagery APIs (Google Maps, Google Earth, Google Street View), without any participant recruitment or interaction.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not a component of our core methodology. Cross3R's training and inference pipeline relies on standard computer vision components (π^3 backbone, COLMAP, DepthAnything v3, Prior Depth Anything); LLMs are not used for any modelling, data filtering, or evaluation step.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.