On the Generalization of Metric Relative Pose Estimation Models to Unseen Environments

Master's Thesis (RO57035)

Bakul Jangley
Supervisors: **Prof. Julian Kooij, Mubariz Zaffar**
Faculty of Mechanical Engineering, TU Delft

On the Generalization of Metric Relative Pose Estimation Models to Unseen Environments

Bakul Jangley
Student Number: 6055826
Master's Robotics (RO57035: Thesis)
Supervisors: **Prof. Julian Kooij, Mubariz Zaffar**
Faculty of Mechanical Engineering, TU Delft

***Abstract*—Crowd-sourced imagery is increasingly important for urban mapping and visual localization. However, its reliability is limited by GPS inaccuracies and heterogeneous capture conditions, including device variability, viewpoint differences, illumination changes, and temporal shifts. In these settings, achieving metric-scale pose estimation remains a central challenge. Deep Learning-based pose estimation models address this problem by learning to estimate the 6-DoF pose using geometric cues between image views and metric supervision during training on large datasets. This encourages spatial consistency and supports generalization across diverse conditions. Recent learning-based architectures, often based on vision transformer encoders, approach the task through unified multi-task frameworks that jointly predict metric depthmaps and 2D–2D correspondences, with relative pose estimated downstream. This thesis evaluates whether such frameworks predict accurate metric depthmaps under domain shifts. Experiments show that, even with scale correction through data-driven fine-tuning with metric supervision, depth predictions from multi-task relative pose estimation models fail to generalize reliably to out-of-domain environments. In contrast, monocular models, trained on significantly larger and more varied datasets, demonstrate strong zero-shot reliability for metric depth prediction. A hybrid pipeline is proposed that combines the geometric consistency of relative pose models with the stable metric cues of monocular models, enabling robust pose estimation in crowd-sourced outdoor environments.**

I Introduction

Applications such as urban planning [1], infrastructure monitoring [2], and disaster resilience [3] increasingly rely on large-scale visual data. Crowd-sourced imagery provides a scalable and low-cost alternative for such applications, covering diverse locations and capture conditions that extend far beyond the scope of controlled datasets. For instance, crowd workers have used Google Street View panoramas to annotate urban objects such as street trees, generating accurate geo-tagged urban maps [4]. Similarly, platforms like **Bee Maps** [5] leverage driver-collected street-level imagery to support detailed mapping of urban infrastructure, while imagery from **Mapillary** [6] has been applied to identify road signs, map pedestrian infrastructure such as sidewalks and crosswalks, and enhance navigation services.

Despite these benefits, commodity GPS tags in crowd-sourced imagery often introduce errors of several meters, undermining their reliability as ground truth for mapping and localization. Such inaccuracies hinder the training and evaluation of deep learning methods that rely on accurate supervision, including models for visual place recognition, camera pose regression, and large-scale mapping. This problem is further compounded by the heterogeneity of the data: images originate from diverse devices, capture conditions, and viewpoints, which complicates consistent pose estimation and metric scale recovery. Structure-from-Motion (SfM) [7]–[9] remains the most reliable purely image-based approach for camera pose recovery, but is computationally expensive, sensitive to initialization, and requires auxiliary information for metric scale.

Early learning-based methods attempted to bypass these limitations by directly regressing camera poses from images, demonstrating the feasibility of end-to-end pose estimation but struggled in complex or dynamic environments [10]. *Relative* pose estimation is the task of estimating the relative pose between two images, whereas *absolute* pose estimation seeks to regress the pose of a new image directly with respect to a known reference frame. In deep learning–based methods, relative approaches often generalize better to unseen environments since they do not rely on scene-specific cues but instead learn features that transfer across domains [11], [12]. Later approaches shifted toward training specialized networks for individual components, such as depth estimation or feature matching, combining geometric principles with data-driven learning to improve scalability and accuracy [13].

More recently, unified architectures have emerged that jointly learn dense image-to-image correspondences and metric depth prediction within a multi-task framework. These models show strong promise for robust feature matching across diverse scenes and challenging conditions. Because metric scale recovery is inherently tied to coherent geometric understanding, such architectures implicitly enforce scale by embedding cross-view geometric constraints within their learned representations. Training on metric supervision encourages models to capture both structural and semantic priors relevant for scale estimation. Nevertheless, it remains unclear whether such representations generalize robustly to crowd-sourced outdoor imagery.

In particular, the depthmaps predicted by multi-task relative pose estimation models must be examined to determine whether they are *accurate* in metric scale, i.e., whether the predictions are aligned with ground-truth or absolute measurements. Beyond accuracy, it is also important to assess whether

the scale of these predictions remains *consistent*, meaning it does not change within a scene and is comparable across different scenes, or whether it varies depending on the input. The central issue is whether unified multi-task architectures, which couple correspondence learning with depth supervision to enforce metric scale, can generalize under domain shift. In contrast, monocular metric depth models, trained on vastly larger and more diverse datasets, may provide more robust and reliable metric cues even though they lack cross-view supervision.

To address these gaps, this work systematically compares the two paradigms: evaluating the capability of multi-task relative pose estimation approaches to enforce metric scale, benchmarking them against monocular metric depth models, and testing whether metric fine-tuning improves robustness or merely leads to scene-specific adaptation. The following research questions guide this study:

1) Can multi-task architectures supervised on metric depth generalize across diverse outdoor environments?
2) Are the depthmaps produced by multi-task architectures *accurate* in metric scale, and does the scale remain *consistent* within and across scenes?
3) Can heuristic or data-driven scale correction improve the accuracy of depthmap predictions from multi-task models and lead to more reliable downstream pose estimation?
4) Do monocular metric depth models, trained on larger datasets, achieve metric scale accuracy more reliably across domains?

II Related Work

Traditional Structure-from-Motion (SfM) [7]–[9] pipelines have long been regarded as the gold standard for visual localization (6-DoF pose estimation) and 3D scene reconstruction from images alone. These methods rely on detecting and matching local features across multiple images, followed by triangulation and bundle adjustment to jointly estimate camera poses and reconstruct scene geometry. While highly accurate under controlled scenarios, SfM demands dense image coverage, precise initialization, and substantial computational resources [7]. These requirements limit SfM's scalability in large-scale or dynamic outdoor environments, which often include challenges typical of crowd-sourced imagery: moving objects, illumination changes, temporal differences, and heterogeneous data quality. Additionally, SfM requires scale information from the scene such as known object sizes or ground-truth data collected via sensors such as LiDAR to recover metric scale [7].

Deep learning-based pose estimation reformulates camera localization as a learning problem, training neural networks to regress camera positions and orientations either directly from images or through intermediate representations such as depth, feature correspondences, or dense matching. These methods can be distinguished as *map-aware* or *map-free* depending on how they are applied. Map-aware methods, including *ACE* [14], *ACE0* [15], and *MAREPO* [11], achieve high accuracy by exploiting prior knowledge of a scene such as pre-computed maps or densely captured reference imagery with known poses. For instance, ACE [14] is a map-aware absolute pose estimation framework that leverages scene-specific re-training to improve depthmap predictions using only images and associated poses. While effective when this information is available, map-aware approaches are fundamentally constrained by their dependence on priors, limiting their applicability in scenarios with sparse coverage or unreliable ground truth. In contrast, map-free approaches, such as *Mickey* [16], avoid reliance on priors, generally trading some accuracy for greater generalization across heterogeneous data.

Recent map-free architectures for relative pose estimation, such as *DUSt3R* [17], advance learned pose estimation by jointly integrating correspondence search, depth regression, and geometric consistency into a unified model that processes image pairs. This multi-task framework encourages models to embed structural coherence across views alongside implicit scale cues, yielding robust performance under viewpoint and illumination changes. Its successor, *MASt3R* [18], enhances the framework with reciprocal matching strategies and descriptor-level supervision, and further introduces a "Fast Reciprocal Matching" algorithm with theoretical guarantees on computational efficiency. Building on this line of work, *MUSt3R* [19] extends pairwise inference to multi-view to impose richer geometric constraints and reduce memory when regressing 3D points. Meanwhile, *MASt3R-SfM* [20] integrates a frozen MASt3R backbone into a full SfM pipeline, combining efficient image retrieval, high-quality correspondence matching, and global optimization for scalable and accurate 3D reconstruction of unconstrained image collections. These developments highlight the versatility of such architectures: a single backbone can be adapted for correspondence estimation, relative pose prediction, and full 3D reconstruction.

Nevertheless, despite employing scale-aware losses during training, it remains unclear to what extent joint learning of scene geometry and correspondences enforces faithful metric scale under domain shifts and unseen environments. A persistent bottleneck in outdoor pose estimation research lies in the scarcity of suitable benchmarks [21]. Existing datasets often rely on SfM pipelines for ground truth generation and inherit their limitations. Examples include Aachen Day-Night [22] and the Map-Free Visual Localization Benchmark [23], both constructed from heterogeneous image sources. Automotive datasets such as RobotCar Seasons and CMU Seasons offer long-term variation in illumination and weather but exhibit relatively constrained viewpoint changes [21]. Ground truth in these datasets is typically generated through COLMAP reconstructions [8], [9], sometimes supplemented with manual annotation of 2D–3D correspondences [22] and curated pair selection. As a result, pose labels may still inherit errors from SfM reconstructions [23], or require multi-sensor fusion to mitigate inaccuracies [21].

Metric supervision encourages networks to capture typical object sizes and distance distributions in outdoor scenes, embedding metric scale within their feature representations.

By leveraging diverse geometric cues and semantic priors *between* images, relative pose estimation models internalize not only geometric consistency but also implicit metric scale information. However, their generalization ability depends critically on the size, accuracy, and diversity of available training data, which remains challenging to obtain in complex outdoor settings.

Unlike relative pose estimation models, monocular depth models are trained to predict dense depthmaps independently for each image, enabling access to far larger and more varied training datasets. While such models do not explicitly encode cross-view geometry or relative pose constraints, their depth predictions provide valuable metric cues that can complement multi-view pose estimation pipelines and strengthen their robustness. Typically, these models are first pre-trained on large-scale datasets to learn general depth cues, and then fine-tuned on smaller metric datasets to refine predictions and improve alignment with absolute scale [24]–[26]. For example, *Depth Anything V2* [26] is a metric monocular depth estimation model trained through large-scale teacher–student distillation. Leveraging over 62 million real-world images alongside synthetic data, it demonstrates strong zero-shot generalization across diverse environments.

Existing works have yet to fully evaluate how multi-task relative pose estimation models generalize metric scale under domain shifts, particularly in diverse outdoor and crowd-sourced imagery settings. Addressing this gap, this thesis makes three main contributions:

1) A systematic evaluation of metric depthmap predictions from scale-aware multi-task relative pose estimation models, compared against monocular metric depth models, assessing their accuracy and consistency under domain shifts.
2) An empirical analysis of scale variation in depthmaps from multi-task relative pose estimation models, demonstrating that predicted scale varies with input. The study further tests fine-tuning with metric depth supervision as a scale correction strategy and shows that improvements remain largely scene-specific rather than generalizable.
3) A scalable pipeline that integrates relative pose estimation models with robust monocular depth predictors, enabling generalizable relative pose estimation across heterogeneous outdoor scenes. The pipeline can be further extended by incorporating large-scale datasets such as ZOD [27] and Waymo [28] to broaden coverage by accurately localizing nearby images from crowd-sourced collections like Mapillary.

III Methodology

This section outlines the relative pose estimation framework studied in this thesis. Section III.A formally defines the problem and introduces a modular pipeline that combines dense 3D depthmaps with 2D image correspondences to estimate relative camera pose at metric scale. Sections III.B and III.C then present the models and baselines used to instantiate this pipeline. Finally, Sections III.D.1 and III.D.2 describe strategies to improve metric scale accuracy in depthmaps predicted by scale-aware multi-task models, covering explicit scale estimation techniques and implicit data-driven correction methods, respectively.

III.A Problem Formulation

Given a set $\mathcal{A}$ of anchor images defined as

$$\mathcal{A} = \{(I_i^a, \mathbf{P}_i^a, \mathbf{K}_i^a) \mid i = 1, \ldots, N_a\}, \tag{1}$$

where I_i^a is the i-th anchor image, $\mathbf{P}_i^a = (\mathbf{R}_i^a, \mathbf{t}_i^a)$ its known camera pose or assumed to be at the origin, and $\mathbf{K}_i^a$ the intrinsic parameters. Similarly, we define the set $\mathcal{Q}$ of query images

$$\mathcal{Q} = \left\{\left(I_j^q, \mathbf{P}_j^q, \mathbf{K}_j^q\right) \mid j = 1, \ldots, N_q\right\}, \tag{2}$$

where I_j^q is the j-th query image, $\mathbf{K}_j^q$ are the intrinsics, and $\mathbf{P}_j^q$ is the unknown camera pose to be estimated.

Given a pair of images, I_i^a (anchor) and I_j^q (query), the *matching* neural network function

$$G : (I_i^a, I_j^q) \rightarrow \mathbf{M}_{ij} \tag{3}$$

takes the two images as input and predicts robust pixel correspondences $\mathcal{M}_{ij}$ between them by leveraging learned feature representations and enforcing mutual nearest-neighbor relationships for cyclic consistency.

Consider the *metric depth* neural network function

$$H : I_k \rightarrow \mathbf{D_k} \in \mathbb{R}^{H \times W \times 3} \tag{4}$$

that predicts a dense *depthmap* $\mathbf{D_k}$ defining a 3D point at every pixel in image I_k in the camera frame. Some models may directly predict metric 3D points, while others predict per-pixel metric depth values ($\mathbf{d_k} \in \mathbb{R}^{H \times W}$). Using the camera intrinsic parameters, each depth value at pixel (u, v) can be transformed into a corresponding 3D point expressed in the camera coordinate frame.

For I_i^a (anchor) and I_j^q (query), if $\mathbf{M}_{ij}$ contains P valid 2D–2D image matches, the projection model is then

$$\alpha\, \mathbf{u}_{j,p}^q = \mathbf{K}_j^q \big(\mathbf{R}\mathbf{D}_{i,p}^a + \mathbf{t}\big), \qquad p = 1, 2, \ldots, P. \tag{5}$$

$\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ define the relative transformation from the anchor to the query frame, $\mathbf{K}_j^q$ the query intrinsics, and $\alpha \in \mathbb{R}$ a scalar depth factor ($\alpha = 1$ if $\mathbf{D}$ is in metric scale).

Thus, the relative pose between anchor i and query j is estimated by finding the optimal $(\mathbf{R}, \mathbf{t})$ that minimizes the reprojection error over all correspondences in $\mathcal{M}_{ij}$ (as shown in Figure 1). Pose estimation is performed using OpenCV's `solvePnPRansac`, which robustly rejects outliers while solving this optimization.

III.B Choice of G (Image Matching)

A core component of the formulation in Section III.A is the matching function G, which establishes correspondences between image pairs. Dense matching methods typically exhibit quadratic complexity in the number of pixels, which becomes prohibitively slow for large-scale or crowd-sourced imagery if

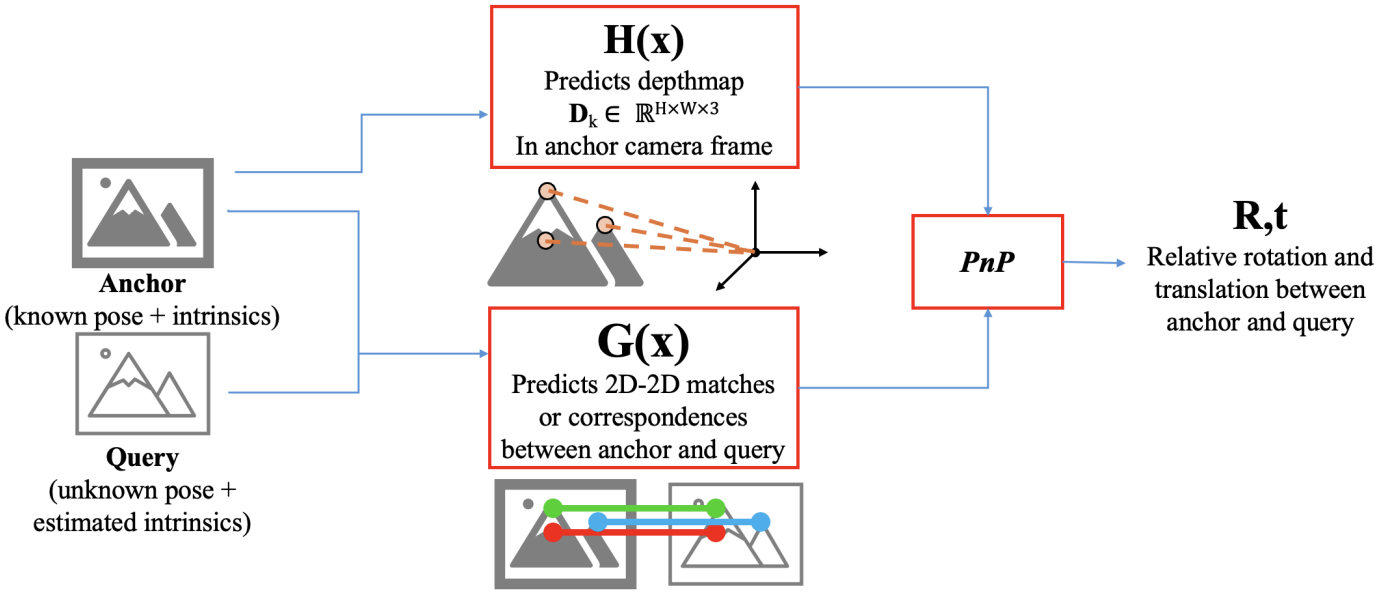


Fig. 1: Using DL based models to substitute for the functions G (image matching or correspondence search) and H (3D depthmap predictions), the relative pose (R, t) between anchor and query images can be estimated using PnP.

not carefully addressed. MASt3R introduces a *Fast Reciprocal Matching* scheme that reduces matching complexity by orders of magnitude while providing theoretical guarantees of correctness. This is used as G across all evaluations presented in this work. Experiments show that MASt3R outperforms prior approaches across a wide range of baselines and illumination changes, achieving a 30% absolute improvement in VCRE AUC[1] on the Map-Free localization benchmark [18].

III.C Choice of H (Depthmap Prediction)

The second component of the formulation is the function H, which provides dense geometric predictions used for pose estimation. The central question of this thesis is whether models trained with metric scale supervision generalize to unseen domains. To this end, we consider a diverse set of H functions:

- **MASt3R-DPT**: MASt3R uses the same learned descriptors to generate correspondences (Section III.B) and depthmaps via a distinct Dense Prediction Transformer (DPT) head. It also predicts confidence scores for depthmap predictions. The scale correction strategies described in Section III.D are applied to improve the metric scale for MASt3R-DPT. This configuration enables us to study whether multi-task supervision, which couples correspondence and geometry prediction, can produce metric depthmaps across different scenes.
- **Metric Monocular Depth Models**: We consider three monocular metric depth models: *ZoeDepth* [24], *DepthPro* [25], and *Depth Anything V2* [26]. ZoeDepth is first pretrained on multiple datasets for relative depth estimation to capture scene geometry and then fine-tuned on metric datasets to produce accurate metric-scale depthmaps. DepthPro is trained to predict metric depth directly from the start using diverse synthetic and real datasets, focusing on high-resolution, sharp depth outputs. Depth Anything V2 has the largest and most diverse training domain, including 62 million pseudo-labeled real images, subsequently fine-tuned on metric datasets for scale-consistent predictions. As detailed in Appendix A, these models leverage large and diverse training datasets with some overlap (Table V).
- **The Metric Depth Oracle** uses ground-truth LiDAR point clouds which serve as a reference for evaluating the metric accuracy of predicted depthmaps and facilitates scale correction. It also allows us to evaluate the MASt3R matching performance.

Together, these choices allow us to compare whether depthmaps produced by multi-task relative pose estimation models (MASt3R-DPT) can outperform specialized monocular depth models in terms of metric scale alignment with Oracle.

III.D Scale Correction Strategies for MASt3R-DPT

This section outlines the approaches used to quantify and analyze metric scale discrepancies between predicted scene geometry and ground-truth measurements. Section III.D.2 outlines explicit scale estimation methods that compare the ground truth LiDAR measurement to the MASt3R-DPT depthmaps. Section III.D.2 explains the data-driven fine-tuning strategy that implicitly tries to improve the model's predictions.

[1]VCRE AUC (Visual Correspondence Relative Error, Area Under Curve) measures the accuracy of predicted correspondences by quantifying the reprojection error between matched points and the true 3D geometry. It summarizes the proportion of correspondences with reprojection errors below varying thresholds as an area-under-curve score, with higher values indicating more accurate and robust matching.

III.D.1 Explicit Scale Correction

Explicit scale correction involves computing deterministic scale factors and subsequently *applying* these factors to MASt3R-DPT predictions before downstream pose estimation. This process does not update any model parameters; instead, the predicted 3D points are rescaled to align with LiDAR measurements in the anchor frame. Given two depthmaps defined in the anchor image frame $D^a(u,v) \in \mathbb{R}^{H\times W\times 3}$ composed of $\mathbf{x}_i^a \in \mathbb{R}^3$:

$$\mathbf{D}^a_{\text{gt}} = \{\mathbf{x}^a_{i,\text{gt}}\}_{i=1}^N \qquad \mathbf{D}^a_{\text{pred}} = \{\mathbf{x}^a_{i,\text{pred}}\}_{i=1}^N, \tag{6}$$

the objective is to compute scale factor(s) that best align $\mathbf{D}^a_{\text{pred}}$ to the ground-truth measurement $\mathbf{D}^a_{\text{gt}}$ using all (u,v) where we have valid 3D LiDAR points in the coordinate frame of the anchor image. We evaluate explicit scale correction using a the heuristic scale factors detailed below.

1) *Per-Axis Scale Estimation:* A per-axis scale vector $\mathbf{s} = [s_x, s_y, s_z]$ is computed as the ratio of norms along each coordinate axis

$$s_{\text{per-axis},k} = \frac{\|\mathbf{D}^a_{\text{gt},k}\|}{\|\mathbf{D}^a_{\text{pred},k}\|} \qquad k \in \{x,y,z\}, \tag{7}$$

where k represents the direction and $\|\cdot\|$ denotes a chosen norm (e.g., L_2 or L_1).

2) *Mean Per-Axis Scale:* Instead of estimating independent scale factors per coordinate axis, this method computes a single global scale s_{mean} by comparing the average L_2 norm across all axes

$$s_{\text{mean}} = \frac{\sum_{k\in\{x,y,z\}} \|\mathbf{D}^a_{\text{gt},k}\|_2}{\sum_{k\in\{x,y,z\}} \|\mathbf{D}^a_{\text{pred},k}\|_2}. \tag{8}$$

This approach assumes the same scale error across axes, unlike per-axis scaling which corrects each axis independently.

3) *Similarity Transform (Umeyama Method):* A full similarity transform estimating scale s, rotation $\mathbf{R}$, and translation $\mathbf{t}$ solves

$$s, \mathbf{R}, \mathbf{t} = \arg\min_{s,\,\mathbf{R},\,\mathbf{t}} \sum_{i=1}^{N} \left\| s\mathbf{R}\mathbf{x}^a_{i,\text{pred}} + \mathbf{t} - \mathbf{x}^a_{i,\text{gt}} \right\|_2^2, \tag{9}$$

with $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$. The Umeyama algorithm provides a closed-form solution including the optimal scale s; the translation and rotation from this solution are ignored.

III.D.2 Implicit Scale Correction

This section presents a strategy for implicitly encoding scale into the *depthmap* predictions generated by the neural network H. Rather than relying on post-hoc scaling corrections, this approach aims to improve the metric stability of the predictions by modifying the model's outputs directly, thereby producing accurate metric depthmaps without depending on LiDAR at test time. Supervision is provided via ground-truth LiDAR point clouds, with the goal of reducing scale error while preserving the geometric consistency learned by MASt3R.

In this approach, the MASt3R-DPT is fine-tuned using a regression loss that directly compares predicted and ground-truth 3D depthmaps. All weights are frozen, except those of the DPT head for the anchor image which is updated using

$$\mathcal{L}^{(1)}_{\text{reg}} = \frac{1}{|\mathcal{M}_1|} \sum_{(u,v)\in\mathcal{M}_1} \|\mathbf{X}_{1,\text{pred}}[u,v] - \mathbf{X}_{1,\text{gt}}[u,v]\|_1, \tag{10}$$

where $\mathcal{M}_1$ denotes the set of pixel locations with valid ground-truth depth from LiDAR. This training objective deviates from the original MASt3R and DUSt3R pipelines, which weigh the loss with confidence scores.

The central premise underlying this data-driven training is that the model learns a distribution specific to the training scenes. It is therefore of interest to determine whether targeted fine-tuning of the depth regression head can effectively correct scale-related errors and how well such adaptations transfer to novel scenes. This approach is conceptually similar to re-training strategies employed in map-aware frameworks such as ACE, which adapt depth predictions using scene-specific supervision, as well as to the final metric fine-tuning stages commonly used in monocular depth prediction models (e.g., ZoeDepth, Depth Anything V2). We aim to study the gains of such adaptations across scenes to assess generalization ability and to compare performance against other metric depth prediction networks.

IV Experiments

The experimental evaluation is divided into two parts. First, we assess performance on the Vision Benchmark Rome (VBR) dataset, comparing off-the-shelf models and scale correction strategies. Second, we analyze generalization on Mapillary imagery, providing a qualitative evaluation under real-world capture conditions.

IV.A Experimental Setup

This section details the datasets used, how training and evaluation splits were defined, and chosen evaluation metrics.

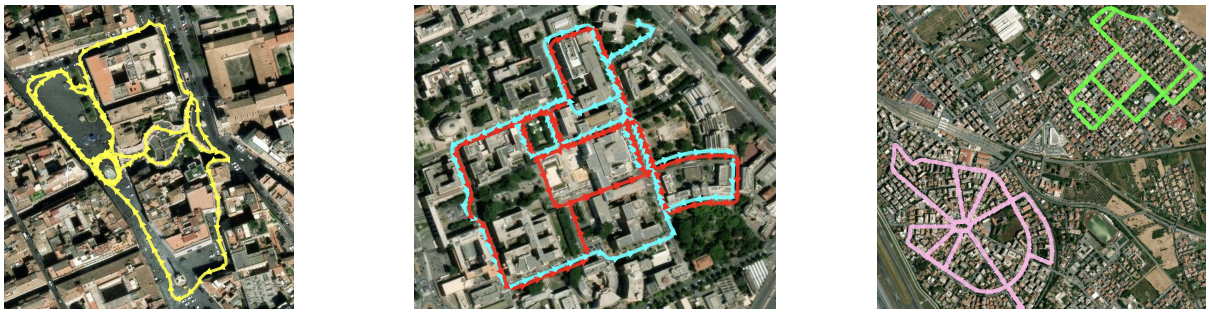
IV.A.1 Datasets

The **Vision Benchmark Rome** (VBR) [29] dataset contains trajectories collected across six locations in Rome: Spagna, Pincio, DIAG, Campus, Colosseum, and Ciampino. For this study, Pincio (dense vegetation), DIAG (indoor setting), and Colosseum (crowded with people) are excluded, focusing instead on **Spagna**, **Campus**, and **Ciampino**, which represent urban scenes featuring buildings and roads. The dataset also provides associated frame transformations between the LiDAR and camera sensors, as well as camera intrinsics and distortion parameters. Trajectories collected in the Campus scene originate from the same physical location at Sapienza University of Rome and thus treated as one unified scene (Figure 3b). In contrast, trajectories from the Ciampino scene are recorded in different physical locations with no overlap and these are split into two independent scenes, Ciampino1 and Ciampino2 (Figure 3c).

Each scene includes one or more trajectories consisting of images, local LiDAR point clouds, and corrected poses

(a) Spagna (b) Campus (c) Ciampino1 (d) Ciampino2

Fig. 2: Sample image pairs from four selected scenes in the VBR dataset.



(a) **Spagna:** contains only one trajectory (b) **Campus:** two trajectories in the same location (c) **Ciampino:** two trajectories at nearby but distinct locations

Fig. 3: Trajectory layouts for each scene in the VBR dataset.

expressed in a local coordinate frame defined at the start of the trajectory. The Spagna scene was recorded with a handheld Ouster OS0-128 sensor, which offers a shorter range but a wider vertical field-of-view. In contrast, the Campus and Ciampino trajectories were captured with a car-mounted Ouster OS0-64, providing slightly longer range. All captures used the same Manta G-125B/C camera at a resolution of 1388×700. Since absolute GPS ground truth is not available, Appendix B-4 details how the trajectories were globally aligned.

In addition to the VBR trajectories, we incorporate crowd-sourced **Mapillary** imagery collected around the **Spanish Steps** and **Campus** areas in Rome and mine anchors from the VBR dataset. Unlike the controlled VBR captures, these images (examples shown in Figure 4) exhibit substantial variation in capture conditions, including day–night shifts, shadows and illumination. They are recorded with heterogeneous consumer devices (smartphones, action cameras, etc.) leading to inconsistencies in focal length, resolution, and distortion. Further challenges arise from arbitrary viewpoints, moving pedestrians, and occasional motion blur. Mapillary additionally provides estimated camera intrinsics obtained via its `OpenSfM` processing pipeline. These intrinsics are used when performing PnP with Mapillary images as queries, ensuring that evaluation remains consistent with the metadata available in real-world scenarios where calibration information is not accessible.

IV.A.2 Evaluation Metrics

Translation error, rotation error and depthmap error were selected as evaluation metrics, as they provide an accurate measure of both pose accuracy and geometric consistency. We report the Median Translation Error (MTE), Median Rotation Error (MRE) and AbsRel for each scene.

Pose estimation accuracy is evaluated by comparing the estimated pose $\mathbf{p}_{\text{est}} = [\mathbf{t}_{\text{est}}, \mathbf{q}_{\text{est}}]$ to the ground-truth pose $\mathbf{p}_{\text{gt}} = [\mathbf{t}_{\text{gt}}, \mathbf{q}_{\text{gt}}]$, where $\mathbf{t}$ denotes translation and $\mathbf{q}$ denotes rotation quaternion.

1) *Translation error* (t_error) is defined as the Euclidean distance between estimated and ground-truth positions expressed as

$$t\_\text{error} = \|\mathbf{t}_{\text{est}} - \mathbf{t}_{\text{gt}}\|_2. \tag{11}$$

2) *Rotation error* is computed as the minimum angular distance between the estimated and ground-truth orientations represented by unit quaternions $\mathbf{q}_{\text{est}}, \mathbf{q}_{\text{gt}} \in \mathbb{S}^3$

$$\mathbf{q}_{\text{rel}} = \mathbf{q}_{\text{est}} \otimes \mathbf{q}_{\text{gt}}^{-1}, \tag{12}$$

$$\text{rot\_error} = 2\cos^{-1}(|q_{\text{rel},w}|)\,, \tag{13}$$

where $q_{\text{rel},w}$ is the scalar (real) part of the relative quaternion $\mathbf{q}_{\text{rel}}$, and $\otimes$ denotes quaternion multiplication.

Fig. 4: Examples of Mapillary imagery near the Spanish Steps. Images span day and night, different focal lengths, and are often affected by occlusions, shadows, and motion blur, reflecting the diversity of crowd-sourced data.

This formulation ensures the angular difference lies within $[0, \pi]$ radians.

3) *Absolute Relative Error (AbsRel)* evaluates the accuracy of predicted depthmaps ($\mathbf{D}^a_{pred}$) against LiDAR ground truth ($\mathbf{D}^a_{gt}$). It is computed using all (u, v) where we have valid 3D LiDAR point in the coordinate frame of the anchor image,

$$\text{AbsRel} = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\mathbf{D}^a_{gt,i} - \mathbf{D}^a_{pred,i}\|_2}{\|\mathbf{D}^a_{gt,i}\|_2}. \tag{14}$$

In addition, per-axis errors are reported by restricting the calculation to each coordinate dimension $k \in \{x, y, z\}$

$$\text{AbsRel}_k = \frac{1}{N_k} \sum_{i \in \{|\mathbf{D}^a_{gt,i_k}| > \epsilon\}} \frac{|\mathbf{D}^a_{gt,i_k} - \mathbf{D}^a_{pred,i_k}|}{|\mathbf{D}^a_{gt,i_k}|}, \tag{15}$$

where $q_{i,k}$ is the ground-truth coordinate along axis k, N_k is the number of valid samples for that axis, and ϵ is a small constant to avoid division by zero.

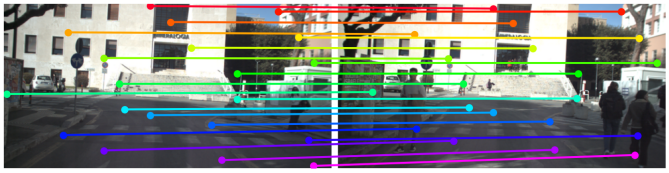
IV.A.3 Implementation Details

Table I summarizes the final dataset splits used for evaluation. For implicit scale correction, we follow the MASt3R training protocol with one modification: the regression loss is adapted to sparse LiDAR supervision (see Section III.D.2). Fine-tuning is performed with an initial learning rate of 3×10^{-5}, with a floor of 5×10^{-6} to reduce overfitting on the smaller training set. Details of image pair generation, data preprocessing, and additional hyperparameters are provided in Appendix B and Appendix C-1.

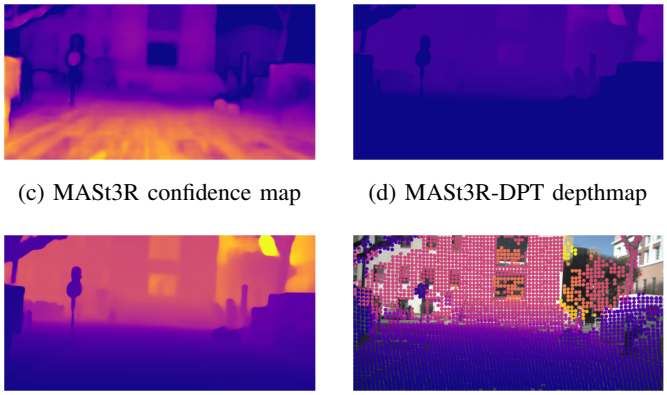
Scene	No. of Pairs Inliers > 200	No. of Pairs Inliers > 700	Train/Val /Test[1]/Test[2]
Spagna	2485	1168	817 / 175 / 175 / 1492
Campus	1929	1214	849 / 181 / 181 / 1027
Ciampino1	2030	1291	903 / 193 / 193 / 932
Ciampino2	1992	869	608 / 130 / 130 / 1253

TABLE I: Number of candidate pairs before/after filtering with number of inliers and final dataset splits.

(a) Anchor (left) and query (right) with 1987 matches, 1928 inliers



(b) MASt3R matches (random subset)



(c) MASt3R confidence map

(d) MASt3R-DPT depthmap

(e) MASt3R-DPT depthmap (scaled)

(f) Oracle

Fig. 5: MASt3R output on a Campus sample: Scaling (*per-axis L_1 norm*) aligns the MASt3R depthmap prediction to Oracle depth (ground truth). The corners and far away background matches have lower confidence.

IV.B Evaluation on VBR

The experimental evaluation is designed to assess the generalization ability of scale-aware relative pose estimation models in unseen outdoor environments. We begin by evaluating the off-the-shelf performance of MASt3R [18]. Next, we analyze how scale in depthmaps predicted by MASt3R-DPT varies within and across scenes using heuristics. We then investigate whether fine-tuning MASt3R-DPT on new environments improves the metric scale of predicted depthmaps, and whether such gains remain confined to the training scenes or transfer to novel settings. Finally, we assess hybrid pipelines pairing

MASt3R as the correspondence function G with monocular depth networks as H, including ZoeDepth [24], Depth Pro [25] and Depth Anything V2 [26].

IV.B.1 MASt3R: Off-the-shelf Generalization

Table II reports errors (MTE, MRE, AbsRel) for MASt3R as G combined with different variants of H on the test[1] splits. With Oracle depthmaps, MASt3R matches yield very low MTE (< 0.20 m across all scenes), confirming that correspondence quality is high. In contrast, using MASt3R-DPT depthmaps without scaling does not achieve sub-meter MTE. Confidence thresholding improves MTE but slightly worsens MRE. Figure 7 shows a consistent trend across scenes: baseline distance between anchor and query generally decreases as inliers increase, with wider baselines yielding fewer matches. At the same time, Oracle MTE remains low even at higher baseline distances, indicating that MASt3R correspondences can still support accurate pose estimation under wider viewpoint changes.

Figure 5 shows MASt3R depthmap, confidence map and 2D matches predictions. The network assigns high confidence to foreground structures such as buildings and signage, and lower confidence to edges and background regions.

IV.B.2 Scale Variation Across Scenes in MASt3R-DPT Depthmaps

We evaluate explicit scale correction in two ways: (i) per-pair scale factors computed by using LiDAR at test time and (ii) global per-scene scale factors derived from the training split (no LiDAR measurements used using inference). Among proposed scale estimation heuristics, per-axis L_1 scaling with LiDAR at test time yields the best pose accuracy and lowest AbsRel errors in predicted depthmaps, while isotropic scale factors (such mean per-axis L_2 scaling and similarity transforms) are less reliable. Train-set median scaling helps meaningfully only in Ciampino2.

Table III shows that computed per-axis L_1 scale factors differ substantially between scenes, even where LiDAR and cameras sensors are identical (Campus, Ciampino1, Ciampino2). Intra-scene variance is smaller for Campus and Ciampino, but Spagna, recorded with a short-range handheld LiDAR, exhibits higher scale and variance. Variability is largest along Z in all scenes.

Figures 5d and 5e compare MASt3R depthmaps before and after applying explicit per-axis L_1 scale correction, illustrating the improvement in scale alignment after scaling.

IV.B.3 Implicit Scale Correction via fine-tuning MASt3R-DPT head

MASt3R-DPT was fine-tuned on two different scenes: one trained on Ciampino1 (validated on Ciampino2), the other on Campus (validated on Ciampino2). In their respective domains, both models report lower MTE, MRE, and AbsRel errors than the off-the-shelf baseline. These gains are largely scene-specific: the Ciampino1-tuned model performs well on Ciampino but raises MTE on Spagna (Table II).

Scene	Median (X, Y, Z)	Variance (X, Y, Z)
All valid pairs (**Per Axis L_1 Norm**)		
Spagna	3.956, 4.025, 4.593	4.251, 4.832, 4.485
Campus	1.703, 1.572, 2.135	0.706, 0.694, 1.261
Ciampino1	1.943, 1.854, 2.544	0.474, 0.497, 1.099
Ciampino2	1.808, 1.737, 2.530	**0.339, 0.321, 0.680**
Train sets (**Per Axis L_1 Norm**)		
Spagna	3.690, 3.750, 4.016	3.598, 3.991, 4.112
Campus	1.539, 1.413, 1.894	0.648, 0.646, 1.171
Ciampino1	1.847, 1.765, 2.412	0.391, 0.392, 0.996
Ciampino2	2.067, 1.963, 2.871	**0.316, 0.310, 0.656**

TABLE III: Median and variance of heuristic scale factors computed per-pair.

Fine-tuning increases predicted depth values, aligning them more closely with the Oracle (Fig. 6). Cross-scene fine-tuning (*Campus→Ciampino2*) also improves over the pre-trained baseline but produces depthmaps with blurred boundaries for distant buildings, while the associated confidence maps lose detail after fine-tuning. Training curves are provided in Appendix C-2.

IV.B.4 Zero-shot Performance of Metric Monocular Depth Prediction Models

Among monocular depth networks, Depth Anything V2 as H achieves sub-meter MTE across all scenes and lower MRE than any MASt3R-DPT variant. It provides consistent AbsRel errors across scenes, demonstrating stable depth alignment without task-specific fine-tuning. DepthPro performs competitively, especially in rotation estimation, achieving the lowest MRE values overall (as low as $0.22°$ on Ciampino2). However, the depthmap predictions do not provide reliable metric scale across scenes and the AbsRel errors vary across scenes. ZoeDepth, despite metric fine-tuning on KITTI (same as Depth Anything V2), produces higher translation and AbsRel errors compared to pre-trained MASt3R-DPT. Nevertheless, it also achieves substantially lower MRE than MASt3R-DPT, highlighting that monocular models provide more reliable rotation cues. Overall, although fine-tuned MASt3R-DPT can sometimes match or surpass monocular networks in AbsRel, Depth Anything V2 and DepthPro depthmaps consistently provide better downstream pose accuracy.

Figure 8 visualizes depthmaps for different H functions and shows that Depth Anything V2 and Depth Pro provide off-the-shelf metric scale alignment with Oracle compared to ZoeDepth and MASt3R-DPT.

Inlier count can serve practical proxy for pose confidence, Figure 7 shows that MTE generally decreases as the number of inliers increases. Appendix D-1 further shows that across all choices of H, the translation component along z dominates overall translation error.

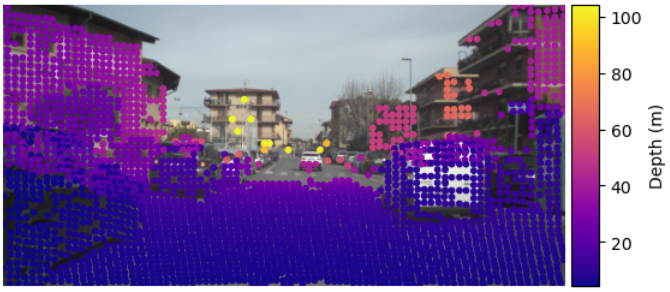
IV.C Evaluation on Mapillary Images

In this section, we extend the evaluation to crowd-sourced Mapillary imagery to assess robustness under real-world, uncontrolled capture conditions. Images from Mapillary are

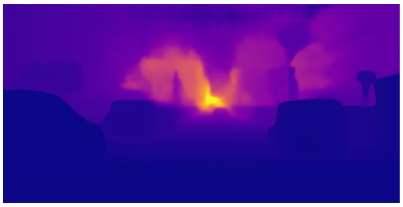
H	Spagna			Campus			Ciampino1			Ciampino2		
	MTE (m)	MRE (°)	AbsRel	MTE (m)	MRE (°)	AbsRel	MTE (m)	MRE (°)	AbsRel	MTE (m)	MRE (°)	AbsRel
Oracle	**0.17**	**0.90**	—	**0.11**	**0.19**	—	**0.10**	**0.20**	—	**0.08**	**0.17**	—
MASt3R-DPT (Pre-trained Model [18])	2.71	2.04	0.78	2.12	2.49	0.41	4.18	3.47	0.60	2.01	2.93	0.62
MASt3R-DPT (Confidence Thresholded)	2.50	2.06	0.77	1.73	2.51	0.40	2.55	3.72	0.61	1.69	4.21	0.61
MASt3R-DPT Scaled: *LiDAR used at test-time*												
Per Axis L_1 Norm (*)	**0.60**	2.08	0.15	**0.74**	2.08	0.17	**0.40**	2.08	0.14	**0.39**	1.95	0.18
Per Axis L_2 Norm (*)	0.81	2.24	0.16	1.01	2.10	0.21	0.50	2.05	0.16	0.50	2.00	0.21
Mean Per Axis L_2 Norm (*)	0.90	2.04	0.17	5.90	2.49	0.20	6.78	3.47	0.18	5.14	2.90	0.23
Similarity Transform (*)	1.01	2.04	0.22	6.06	2.49	0.20	6.39	3.44	0.20	4.70	2.97	0.23
MASt3R-DPT Scaled: *Train-set Median Scale*												
Per Axis L_1 Norm	0.91	2.18	2.14	2.20	2.11	4.92	2.32	2.07	4.58	0.89	2.01	2.00
Per Axis L_2 Norm	0.94	2.27	2.15	2.30	2.38	5.63	2.25	2.13	4.53	0.96	2.11	2.04
Mean Per Axis L_2 Norm	1.12	2.04	2.14	5.12	2.49	5.18	7.79	3.44	4.90	6.22	2.93	2.39
Similarity Transform	1.06	2.04	1.93	5.25	2.49	5.34	7.44	3.44	4.95	5.38	2.93	3.02
MASt3R-DPT (*Campus→Ciampino2*)	2.01	1.90	0.28	**0.83**	0.69	0.12	1.31	0.66	0.16	0.76	0.79	0.20
MASt3R-DPT (*Ciampino1→Ciampino2*)	2.95	2.10	0.29	1.15	**0.51**	0.13	**0.59**	**0.45**	0.11	**0.47**	**0.50**	0.15
ZoeDepth [24]	2.84	1.41	6.21	3.23	0.43	7.41	3.39	0.39	6.72	2.75	0.34	6.79
Depth Pro [25]	1.16	**0.96**	0.91	0.90	**0.25**	0.41	0.35	**0.27**	0.22	**0.28**	**0.22**	0.27
Depth Anything V2 [26]	**0.86**	1.26	**0.23**	**0.33**	0.29	**0.20**	0.38	0.32	**0.21**	0.59	0.31	**0.26**

TABLE II: Errors (median) on *test*[1] splits for each scene using MASt3R as G combined with different models for H. Lower values indicate better performance for all metrics (MTE, MRE, AbsRel). Rows marked with * use LiDAR-based per-pair scale at inference; unstarred rows use the train-set median scale. Fine-tuning MASt3R-DPT (*Train→Val*) shows mostly within-domain gains but can also lead higher MTE (shown in red).

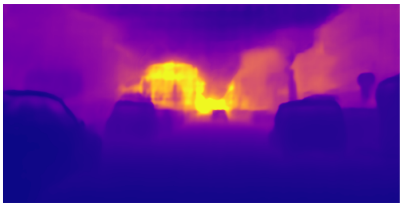
(g) Anchor image from Ciampino2



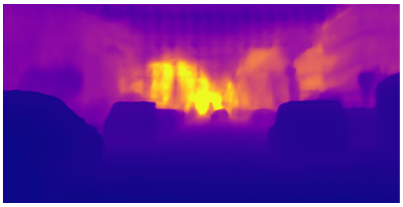
(h) Ground-truth depth from LiDAR



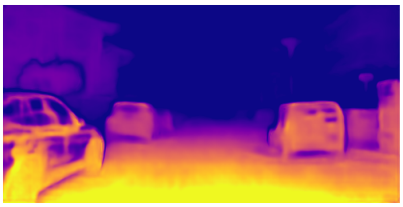
(i) Pre-trained



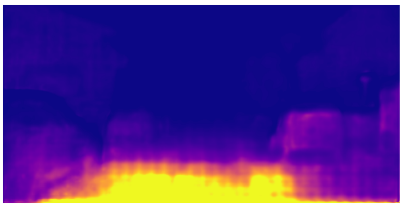
(j) Fine-tuned (*Ciampino1→Ciampino2*)



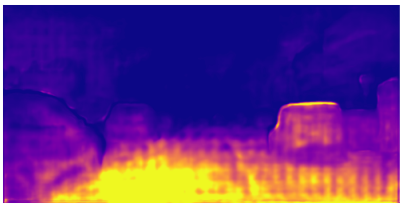
(k) Fine-tuned (*Campus→Ciampino2*)



(l) Pre-trained



(m) Fine-tuned (*Ciampino1→Ciampino2*)



(n) Fine-tuned (*Campus→Ciampino2*)

Fig. 6: Effect of fine-tuning on MASt3R-DPT. *Top*: Anchor image with LiDAR ground truth. *Middle*: depthmaps before/after fine-tuning (Train→Val), showing improved metric alignment but blurred distant boundaries. *Bottom*: confidence maps lose detail after fine-tuning.

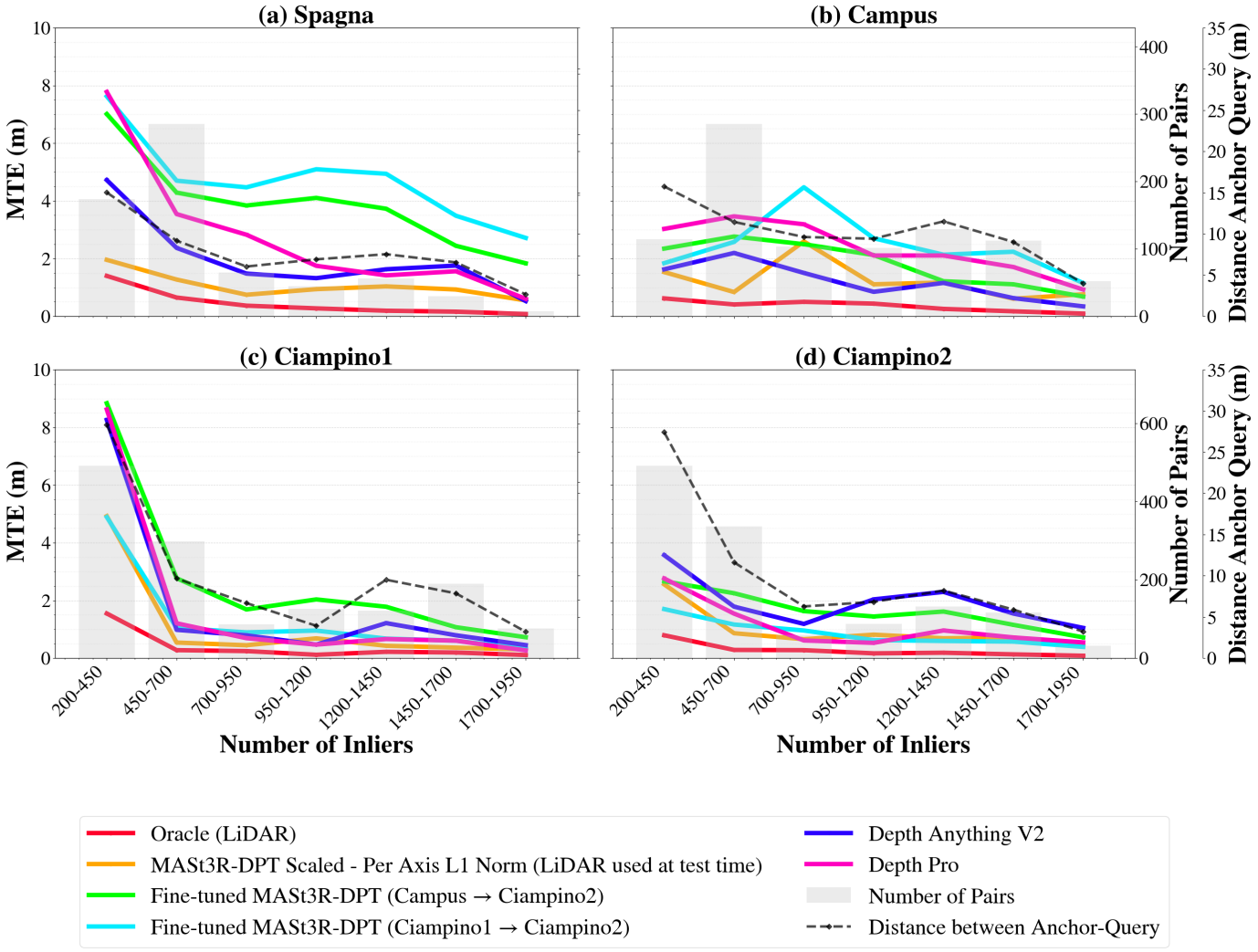
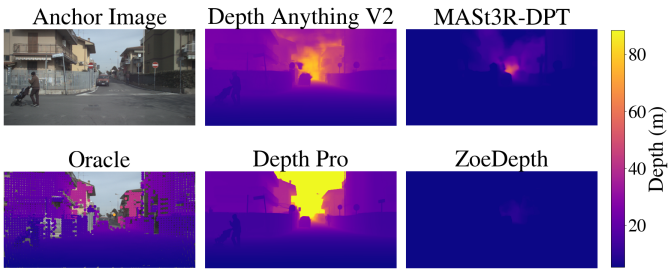


Fig. 7: Mean translation error (MTE) versus number of inliers on *test*[2] splits, shown per scene. Results compare scale corrected MASt3R-DPT against selected monocular depth prediction models as choices for H, with poor-performing models excluded for readability.

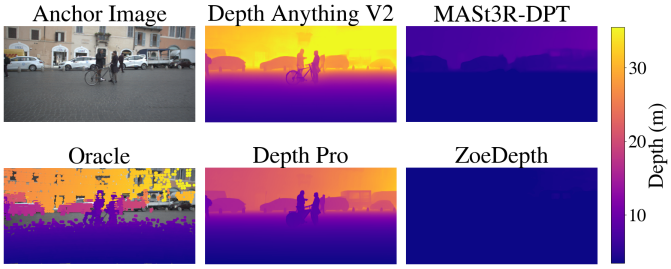
localized using VBR anchors, with MASt3R providing 2D correspondences (G) and the Oracle or Depth Anything V2 as H. Mapillary sequences with inaccurate GPS locations are shown in Figure 10, whereas inspection of the images indicates that the 'true' position should be different. Our anchor-based relative pose estimation corrects these errors reliably, pose estimates from Depth Anything V2 depthmaps closely track the Oracle. Figure 9 illustrates an especially challenging case where the query is captured at night with low visibility and limited overlap to the anchor, since the camera is tilted upward to avoid pedestrian areas. Such conditions are typical in crowd-sourced data and usually undermine feature matching. Despite this, our pipeline successfully refines the pose. The initial GPS-based estimate placed the query over the fountain, but after refinement the pose aligns correctly along the street. This demonstrates resilience to both appearance change and viewpoint bias.

IV.D Discussion

Experiments on both VBR and Mapillary confirm that correspondence quality is not the bottleneck for pose estimation. The main limitation lies in predicting metric depthmaps consistently across domains. In zero-shot settings, Depth Anything V2 consistently provides the lowest AbsRel errors across scenes and sub-meter MTE, demonstrating stable metric scale alignment without any scene-specific adaptation. Depth Pro also performs competitively, excelling in some scenes, but its errors vary more across environments, indicating less reliable zero-shot generalization. Notably, even in their training domains the fine-tuned MASt3R-DPT models did not achieve lower MTE (in relative pose estimation) compared to zero-shot monocular models, despite reporting lower AbsRel. These outcomes align with the observed variability in scale factors across scenes: adaptation to one local scale distribution fails to transfer elsewhere.



(a) Depthmaps generated on image from Ciampino2



(b) Depthmaps generated on image from Spagna

Fig. 8: Qualitative comparison of depthmaps from different H models.

(a) GPS locations on Satellite Map with Q marking the inaccurate Mapillary GPS for the query

(b) Anchor (left) from VBR and query (right) from Mapillary

(c) 2D image matches predicted using MASt3R as G

Fig. 9: MASt3R accurately predicts matches between pairs taken at different times of day leading to downstream pose correction when combined with Depth Anything V2

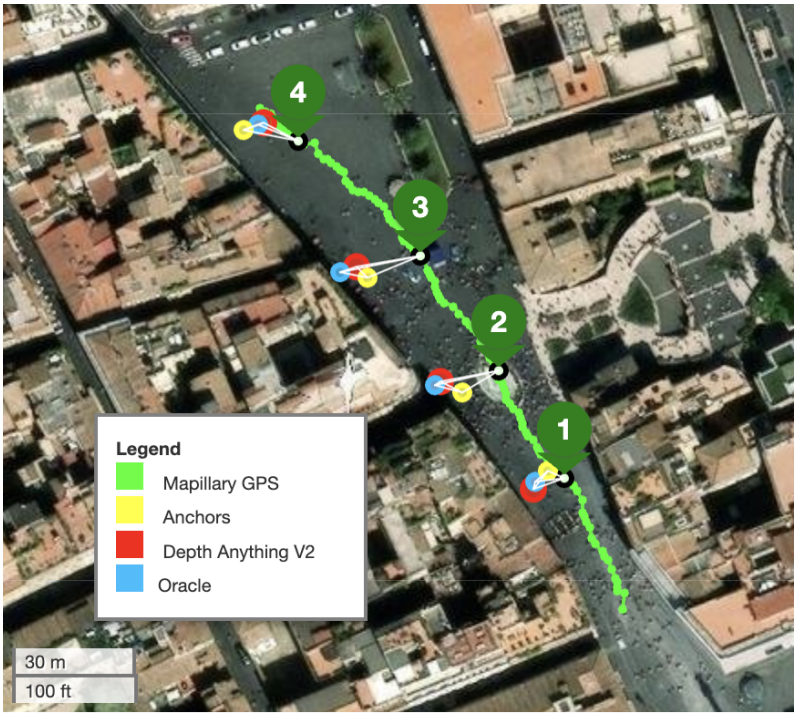
On queries from Mapillary (Section IV.C), the combination of MASt3R correspondences and metric depth from Depth Anything V2 is able to recover accurate relative pose under erroneous GPS, challenging illumination, and viewpoint shifts. These results are obtained on completely unseen data, without any scene-specific fine-tuning.

V Conclusion

This thesis examined whether multi-task relative pose estimation models can produce *metric* depthmaps in GPS-inaccurate, crowd-sourced outdoor imagery. While these models provide robust correspondences, their depth predictions lack consistent metric scale across scenes. Heuristic scale corrections improved alignment when auxiliary cues were available but remained tied to such measurements. Implicit strategies produced local gains within training scenes but failed to generalize. The results highlight that scale in depthmap predictions from these models remains dependent on input and scene. In contrast, monocular depth models trained on large and diverse datasets provided stable metric cues without scene-specific adaptation, with Depth Anything V2 standing out due to its *vastly* larger training domain.

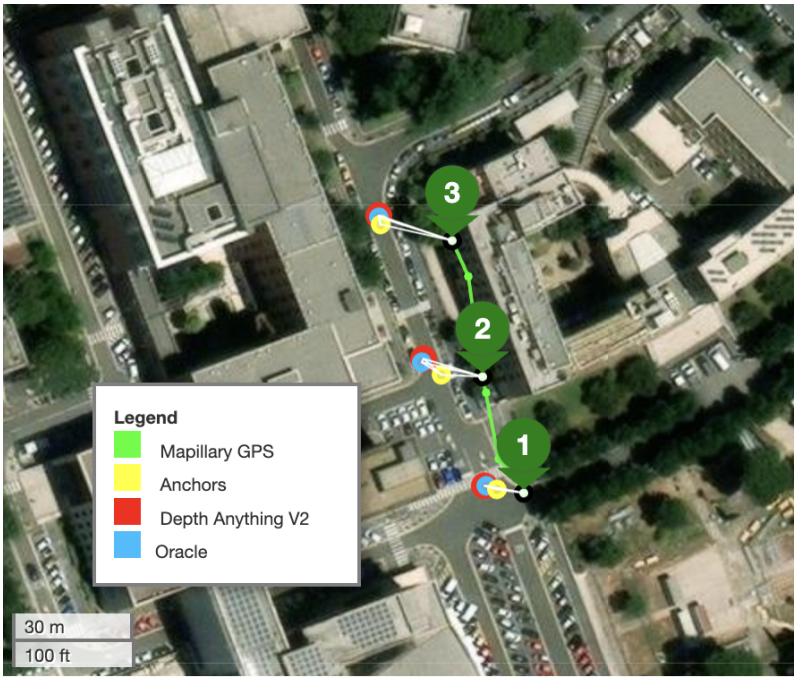
Two factors limited this work: the small amount of per-scene training data and the sparsity and range limits of LiDAR supervision, both of which weakened the available metric signal. Future work should explore enforcing scale through trajectory-level constraints, integrating an explicit scale-regression head in MASt3R-DPT, and evaluating on larger datasets with targeted fine-tuning for monocular models as well.

Overall, the results indicate that metric scale remains the central bottleneck in relative pose estimation. A practical next step is to couple multi-task relative pose architectures such as MASt3R with stable, zero-shot monocular depth predictors like Depth Anything V2, an approach that can improve robustness when applied to real-world, crowd-sourced data.



(a) Spagna: Query images are positioned closer to buildings, whereas the Mapillary GPS locations indicate positions near the center of the open square.

(b) Highlighted anchor–query pairs (Spagna).



(c) Campus: Queries from Mapillary (clearly captured over the road) are shown to have GPS locations over/inside buildings.

(d) Highlighted anchor–query pairs (Campus).

Fig. 10: Results from Spagna (top) and Campus (bottom). Anchors from VBR (yellow) are used to correct inaccurate GPS for Queries from Mapillary (green), and corrected positions using Oracle (blue) and Depth Anything V2 (red).

REFERENCES

[1] Raveena Marasinghe, Tan Yigitcanlar, Severine Mayere, Tracy Washington, and Mark Limb. Computer vision applications for urban planning: A systematic review of opportunities and constraints. *Sustainable Cities and Society*, 100:105047, 2024. 1

[2] Fan Zhang, Arianna Salazar-Miranda, Fábio Duarte, Lawrence Vale, Gary Hack, Min Chen, Yu Liu, Michael Batty, and Carlo Ratti. Urban visual intelligence: Studying cities with artificial intelligence and street-level imagery. *Annals of the American Association of Geographers*, 114(5):876–897, 2024. 1

[3] Vinodkumar Devarajan. Integrated AI-ML framework for disaster lifecycle management: From prediction to recovery. *World Journal of Advanced Research and Reviews*, 26:585–593, 05 2025. 1

[4] Sihang Qiu, Achilleas Psyllidis, Alessandro Bozzon, and Geert-Jan P. M. Houben. Crowd-mapping urban objects from street-level imagery. In *Proc. Web Conf. (WWW)*, pages 1521–1531, 2019. 1

[5] Bee maps. https://beemaps.com/blog/enhance-autonomous-vehicles-with-hivemappers-crowdsourced-maps. Accessed: 2025-08-16. 1

[6] Mapillary. https://www.mapillary.com/. Accessed: 2025-04-14. 1

[7] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017. 1, 2

[8] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1, 2

[9] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 501–518, 2016. 1, 2

[10] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2938–2946, 2015. 1

[11] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 20665–20674, 2024. 1, 2

[12] Hunter Blanton, Scott Workman, and Nathan Jacobs. A structure-aware method for direct pose estimation. In *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, pages 2019–2028, 2022. 1

[13] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 44(9):5847–5865, 2021. 1

[14] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using RGB and poses. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5044–5053, 2023. 2

[15] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Aron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 421–440, 2024. 2

[16] Axel Barroso-Laguna, Sowmya Munukutla, Victor Adrian Prisacariu, and Eric Brachmann. Matching 2D images in 3D: Metric relative pose from metric correspondences. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4852–4863, 2024. 2

[17] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709, 2024. 2

[18] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 71–91, 2024. 2, 4, 7, 9, 15, 16, 19

[19] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. MUSt3R: Multi-view network for stereo 3D reconstruction. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1050–1060, 2025. 2

[20] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MASt3R-SfM: A fully-integrated solution for unconstrained structure-from-motion. In *Proc. Int. Conf. on 3D Vision (3DV)*, pages 1–10, 2025. 2

[21] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6-DoF outdoor visual localization in changing conditions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8601–8610, 2018. 2

[22] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *International Journal of Computer Vision*, 129(4):821–844, 2021. 2

[23] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 690–708, 2022. 2, 15

[24] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 3, 4, 8, 9, 15, 16, 19

[25] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 3, 4, 8, 9, 15, 16, 19

[26] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 3, 4, 8, 9, 15, 16, 19

[27] Mina Alibeigi, William Ljungbergh, Adam Tonderski, Georg Hess, Adam Lilja, Carl Lindström, Daria Motórniuk, Junsheng Fu, Jenny Widahl, and Christoffer Petersson. Zenseact Open Dataset: A large-scale and diverse multimodal dataset for autonomous driving. In *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 20178–20188, 2023. 3

[28] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo Open Dataset. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 3, 15

[29] Leonardo Brizi, Emanuele Giacomini, Luca Di Giammarino, Simone Ferrari, Omar Salem, Lorenzo De Rebotti, and Giorgio Grisetti. VBR: A Vision Benchmark in Rome. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 15868–15874, 2024. 5

[30] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4340–4349, 2016. 15, 16

[31] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGB-D images. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 15

[32] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3D dataset (HM3d): 1000 large-scale 3D environments for embodied AI. In *Proc. NeurIPS Datasets and Benchmarks Track*, 2021. 15

[33] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 15, 16

[34] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes: A diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *Proc. NeurIPS Datasets and Benchmarks Track*, 2021. 15, 16

[35] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 15

[36] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A Large-Scale Dataset for Generalized Multi-View Stereo Networks. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1790–1799, 2020. 15, 16

[37] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3D indoor scenes. In *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 12–22, 2023. 15, 16

[38] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2021. 15

[39] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. RGBD objects in the Wild: Scaling real-world 3D object learning from RGB-D videos, 2024. 15

[40] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020. 15, 16

[41] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. SMD-Nets: Stereo mixture density networks. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8942–8952, 2021. 15, 16

[42] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual SLAM. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4909–4916, 2020. 15, 16

[43] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 15, 16

[44] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 311–320, 2018. 15

[45] Jaehoon Cho, Dongbo Min, Youngjung Kim, and Kwanghoon Sohn. DIML/CVL RGB-D Dataset: 2M RGB-D images of natural indoor and outdoor scenes. *arXiv preprint arXiv:2110.11590*, 2021. 15

[46] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2020. 15

[47] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes, 2019. 15

[48] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The ApolloScape Dataset for Autonomous Driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 954–960, 2018. 15

[49] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. IRS: A Large Naturalistic Indoor Robotics Stereo Dataset to Train Deep Models for Disparity and Surface Normal Estimation. *arXiv preprint arXiv:1912.09678*, 2019. 15, 16

[50] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 15

[51] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 10912–10922, 2021. 15, 16

[52] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint arXiv:1810.08705*, 2018. 15

[53] Jose L. Gómez, Manuel Silva, Antonio Seoane, Agnès Borràs, Mario Noriega, German Ros, Jose A. Iglesias-Guitian, and Antonio M. López. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes. *Neurocomputing*, 637:130038, 2025. 15

[54] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 15

[55] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8737, 2023. 15

[56] Youngjung Kim, Bumsub Ham, Changjae Oh, and Kwanghoon Sohn. Structure selective depth superresolution for RGB-D cameras. *IEEE Trans. on Image Processing*, 25(11):5227–5238, 2016. 15

[57] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 15, 16

[58] Anastasiia Kornilova, Marsel Faizullin, Konstantin Pakulev, Andrey Sadkov, Denis Kukushkin, Azat Akhmetyanov, Timur Akhtyamov, Hekmat Taherinejad, and Gonzalo Ferrer. Smartportraits: Depth powered handheld smartphone dataset of human portraits for state estimation, reconstruction and synthesis. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 21318–21329, 2022. 15

[59] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3D ken burns effect from a single image. *ACM Trans. on Graphics*, 38(6):184:1–184:15, 2019. 15

[60] Hoang-An Le, Partha Das, Thomas Mensink, Sezer Karaoglu, and Theo Gevers. EDEN: Multimodal synthetic dataset of enclosed garden scenes. In *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, 2021. 15

[61] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 15

[62] Y.-T. Hu, J. Wang, R. A. Yeh, and A. G. Schwing. SAIL-VOS 3D: A synthetic dataset and baselines for object detection and 3D mesh reconstruction from video data. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 15

[63] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2636–2645, 2020. 15

[64] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2: A large-scale benchmark for instance-level recognition and retrieval. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2575–2584, 2020. 15

[65] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 15

[66] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 15

[67] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 8429–8438, 2019. 15

[68] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The Open Images Dataset v4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *Int. Journal of Computer Vision*, 128(7):1956–1981, 2020. 15

[69] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2017. 15

[70] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 4015–4026, 2023. 15

[71] Daan Zwaneveld. Tu delft report template. https://dzwaneveld.github.io, 2024. Licensed under CC BY-NC 4.0. 18

[72] IEEE. Ieee manuscript templates for conference proceedings. https://www.ieee.org/conferences/publishing/templates.html, 2023. 18

[73] Pixabay. World map abstract blue. https://pixabay.com/photos/map-world-dotted-blue-abstract-1025845/, 2015. CC0 License, no attribution required. 18

[74] Esri, Maxar, Earthstar Geographics, and the GIS User Community. World imagery basemap. https://www.arcgis.com/home/item.html?id=10df2279f9684e4a9f6a7f08febac2a9, 2024. Accessed: 20-09-2025. 18

APPENDIX A
TRAINING DATASETS FOR METRIC DEPTH PREDICTION MODELS

Table IV presents the datasets used to train the various models proposed as depth prediction functions H. There is some overlap in the training sets as reported in Table V. Depth Anything V2 and ZoeDepth models are finetuned on KITTI [30] (outdoor) and NYUDepthV2 [31] (indoor) to produce metric checkpoints.

The training curriculum of these metric depth models differ primarily in how they balance learning relative depth and achieving accurate absolute scale.

- **ZoeDepth** [24]: Follows a two-stage pipeline. Stage 1 pretrains on diverse datasets to optimize relative depth estimation. Stage 2 fine-tunes on metric datasets, with specialized prediction heads enabling scale-consistent outputs.
- **DepthPro** [25]: Employs a two-stage training regime oriented towards absolute metric predictions from the outset. Stage 1 trains on a mix of synthetic and real datasets, using metric supervision where available and scale-invariant normalization otherwise. Stage 2 fine-tunes on high-quality synthetic metric datasets with boundary-focused losses, yielding sharp and accurate scale-consistent depthmaps.
- **Depth Anything V2** [26]: Uses a three-stage training curriculum. Stage 1 trains a teacher network on accurate synthetic datasets. Stage 2 distills knowledge into student networks via over 62 million pseudo-labeled real-world images, boosting robustness and domain generalization. An optional Stage 3 fine-tunes on metric datasets to align absolute scale.

Model	Datasets
MASt3R [18]	Habitat [32], MegaDepth [33], ARKitScenes [34], StaticScenes3D [35], BlendedMVS [36], ScanNet++ [37], CO3D-v2 [38], Waymo [28], Map-free [23], WildRgb [39], VirtualKitti [40], Unreal4K [41], TartanAir [42], Internal Dataset (undisclosed)
ZoeDepth [24]	HRWSI [43], BlendedMVS [36], ReDWeb [44], DIML-Indoor [45], 3DMovies [46], MegaDepth [33], WSVD [47], TartanAir [42], ApolloScape [48], IRS [49], KITTI [50], NYUDepth v2 [31]
Depth Pro [25]	**Stage 1:** Hypersim [51], TartanAir [42], Synscapes [52], Urbansyn [53], Dynamic Replica [54], Bedlam [55], IRS [49], Virtual Kitti2 [30], Sailvos3d, ARKitScenes [34], Diml Indoor [56], Scannet [57], Smart Portraits [58], UnrealStereo4k [41], 3D Ken Burns [59], EDEN [60], MVS Synth [61], HRWSI [43], BlendedMVS [36] **Stage 2:** Hypersim [51], TartanAir [42], Synscapes [52], Urbansyn [53], Dynamic Replica [54], Bedlam [55], IRS [49], Virtual Kitti2 [30], Sailvos3d [62]
Depth Anything V2 [26]	**Precise Synthetic (595K):** BlendedMVS [36], Hypersim [51], IRS [49], TartanAir [42], VKITTI2 [30] **Pseudo-labeled Real (62M):** BDD100K [63], GoogleLandmarks [64], ImageNet-21K [65], LSUN [66], Objects365 [67], OpenImagesV7 [68], Places365 [69], SA-1B [70]

TABLE IV: Training datasets used by models (H)

APPENDIX B
DATASET PREPARATION

This subsection details the preprocessing steps applied to the Vision Benchmark Rome (VBR) dataset. Dataset creation included generation of candidate image pairs, scaling input images, filtering based on inlier thresholds, and global trajectory alignment.

B-1 Pair Construction

Overlapping sequences within each trajectory were grouped, designating one as the anchor and the others as queries. Anchor and query frames were sub-sampled at different step sizes to balance coverage and computational overhead. For each anchor–query pair, features were extracted with MASt3R, and correspondences were evaluated via fundamental matrix estimation. To improve robustness, forward and reverse matches were computed. Only the top-n anchors with the highest inlier support were retained per query, yielding a final set of reliable pairs with inliers > 200.

B-2 Pair Filtering and Splits

Table VI summarizes the mining parameters and resulting pair counts per scene. Filtering was performed via OpenCV's fundamental matrix estimation with RANSAC (reprojection threshold $1\,\text{px}$, confidence 0.99). Pairs with fewer than 700 inliers were discarded to ensure geometric consistency for training and validation. The remaining pairs were randomly shuffled and partitioned into train/val/test[1] splits using a $0.7/0.15/0.15$ ratio. An additional test[2] set was defined by reintroducing rejected pairs with $200 < \text{inliers} < 700$, allowing evaluation on more challenging cases. Table I reports the resulting counts.

B-3 Matching Resolution

Only coarse features are used during matching. Inputs are re-scaled such that the longest image side is reduced to 512 pixels while maintaining the original aspect ratio, followed by cropping to the model patch size (16).

B-4 Global Localization from Image–Map Correspondences

Since the VBR dataset does not provide absolute GPS locations, global localization of image trajectories was achieved by manually selecting correspondences between landmarks visible in the images and their locations on a satellite map. The process proceeds in three main stages:

1) **Georeferencing correspondences:**
Distinct visual markers on the ground (e.g., poles, corners) were identified in the camera images and localized on the satellite map. Their geographic coordinates were converted from latitude/longitude to UTM coordinates using PyProj, ensuring consistency with the local metric

Dataset	MASt3R [18]	ZoeDepth [24]	Depth Pro [25]	Depth Anything V2 [26]
BlendedMVS [36]	✓	✓	✓	✓
MegaDepth [33]	✓	✓	–	–
TartanAir [42]	✓	✓	✓	✓
IRS [49]	–	✓	✓	✓
VirtualKitti [40] / VKITTI2 [30]	✓	–	✓	✓
Hypersim [51]	–	–	✓	✓
HRWSI [43]	–	✓	✓	–
ARKitScenes [34]	✓	–	✓	–
Unreal4K [41]	✓	–	✓	–
ScanNet++ [37] / ScanNet [57]	✓	–	✓	–

TABLE V: Dataset overlaps between models

Scene	Sub-sampling (Query/Anchor)	Top n Anchors	Total Pairs	Valid Pairs (>200 Inliers)
Spagna	50 / 50	10	2580	2485
Campus	20 / 10	10	2060	1929
Ciampino1	20 / 10	5	2060	2030
Ciampino2	20 / 10	7	2212	1992

TABLE VI: Summary of mining parameters per scene. Sub-sampling values indicate the frame step size for query and anchor sequences. Top n anchors shows the number of highest-scoring anchors retained per query. Valid pairs indicate the number of anchor–query pairs with more than 200 inliers.

frame. Figure 11 shows an image with selected markers and the corresponding points on a satellite map.

2) **Camera pose estimation:**
For images with pixel-map correspondences, a Perspective-n-Point (PnP) problem was solved using the 2D pixel coordinates of the selected markers and their corresponding 3D UTM coordinates projected onto the ground plane. This provided extrinsic estimates (rotation and translation) for each camera relative to the global UTM frame.

3) **Trajectory alignment:**
To align the VBR local trajectory with the GPS-aligned estimates, we solve for a similarity transform

$$\min_{s,R,t} \sum_{i=1}^{N} \| sRx_i + t - y_i \|^2 ,$$

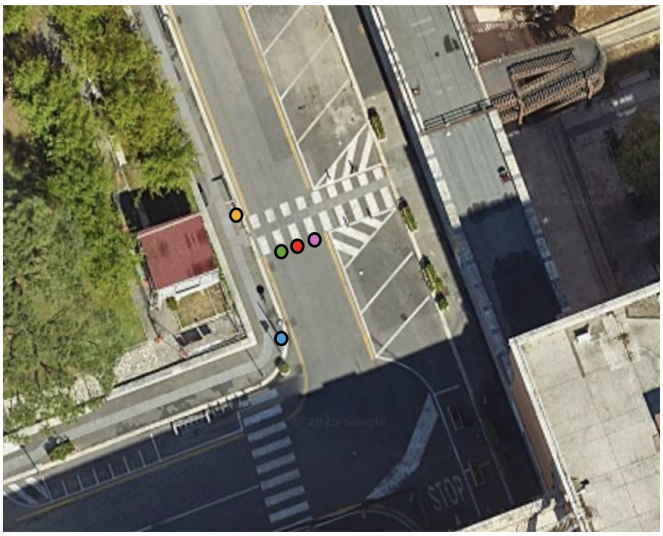
where x_i are local trajectory points and y_i their GPS-aligned counterparts. The correspondence set $\{(x_i, y_i)\}_{i=1}^{N}$ is obtained from 3–4 images that were manually matched with the map (via Step 1).
The closed-form solution (Umeyama, 1991) provides:

$$R = VDU^\top, \qquad s = \frac{\mathrm{tr}(D\Sigma)}{\sum_i \|x_i'\|^2}, \qquad t = \mu_y - sR\mu_x,$$

where (μ_x, μ_y) are centroids, Σ is the cross-covariance, and D enforces a proper rotation.

The resulting similarity transform was applied to the full local trajectory, yielding an aligned trajectory in the UTM/global frame (latitude, longitude and heading). The final resulting trajectories shown in Figure 3). This procedure yields a globally aligned camera trajectory even when absolute GPS is unavailable, enabling evaluation on Mapillary images.

(a) Selected markers on image from the Campus scene

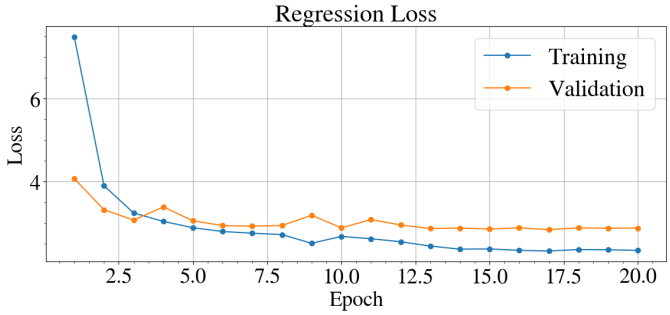


(b) Corresponding markers on Satellite Map

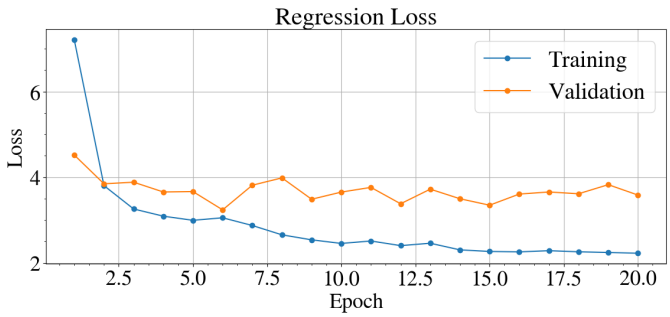
Fig. 11: By finding distinguishable markers on the image and then localizing them on the satellite map, images can be globally localized in the UTM frame

APPENDIX C
ADDITIONAL DETAILS: IMPLICIT SCALE CORRECTION

This section provides additional details on the fine-tuning of MASt3R-DPT for implicit scale correction. We compare the

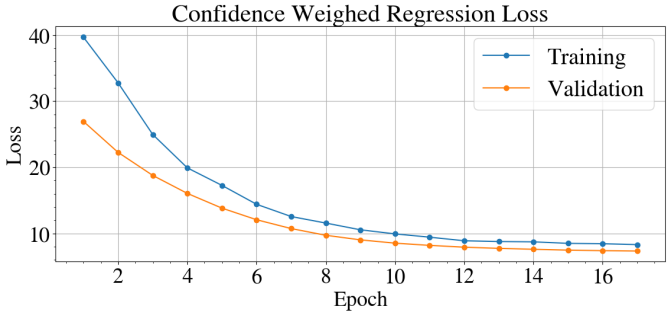


(a) *Ciampino1→Ciampino2* .

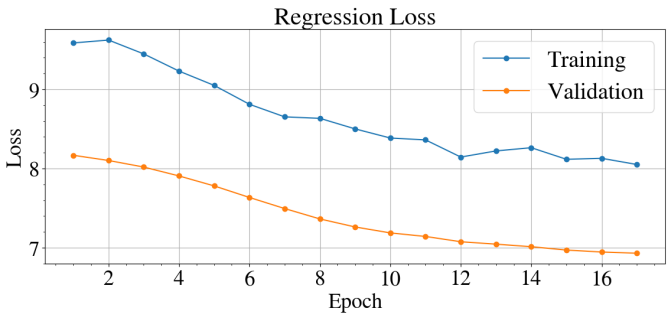


(b) *Campus→Ciampino2*

Fig. 12: Training curves for MASt3R model finetuned on *Training Set→Validation Set* with regression loss



(a) Confidence weighed regression loss



(b) Compared to fine-tuning with regression loss, confidence weighing does not show the same reduction in regression loss

Fig. 13: Training curves for MASt3R model finetuned on *Campus→Ciampino2* with confidence weighed regression loss with lower learning rate: 1×10^{-6} (minimum 1×10^{-7})

modified regression loss used in this work against the original confidence-weighted objective, and analyze how these choices affect the stability of MASt3R-DPT predictions during fine-tuning. Training curves are included to illustrate convergence behaviour, and the exact hyperparameters and implementation settings are reported for reproducibility.

C-1 Hyperparameters Used

All the necessary hyperparameters required to replicate the training the pointmap prediction for the anchor image using regression loss are listed in Table VII.

C-2 Training Curves

The standard training procedure in MASt3R and DUSt3R employs a confidence weighted regression loss, where per-point confidence scores are used to weigh the regression error. For the VBR dataset, however, we made two adjustments when fine-tuning the depth prediction head:

- A lower learning rate was adopted to prevent overfitting, given the limited dataset size.
- Confidence weighed loss was replaced with simple regression loss, since the LiDAR supervision in VBR is sparse.

When confidence weighting is used with the same learning rate as presented in Table VII, we observe that the predicted confidence values collapse within the first epoch, saturating near 1 for all points. This renders the confidence estimates meaningless. We also tried reducing the learning rate to 1×10^{-6} (minimum 1×10^{-7}), however this configuration does not reduce the regression loss (Figure 13), which is our main objective, as effectively as unweighted regression (Figure 12). Further analyzing how the confidence prediction changes shows that using confidence weighing in the loss function always leads to the confidence values all collapsing (as shown by the overwhelmingly blue confidence maps in Figure 14c).

Appendix D
Analyzing Per-Axis Errors

This section examines the per axis errors in translation error and also per-axis AbsRel errors for different models as H and MASt3R as G. Finally Section D-2 presents a more in-depth analysis of how per-axis scaling affects the MASt3R-DPT depthmaps.

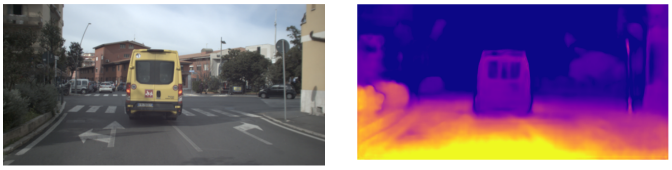
D-1 Per-Axis Translation and AbsRel Accuracy

Tables VIII and IX highlight two consistent patterns. First, the translation error (MTE) is always dominated by the z component, irrespective of the method. Even for the monocular depth estimation baselines (Depth Anything V2 and ZoeDepth), the z-axis translation error is consistently larger than the x and y components, confirming that depth uncertainty directly drives pose recovery through PnP.

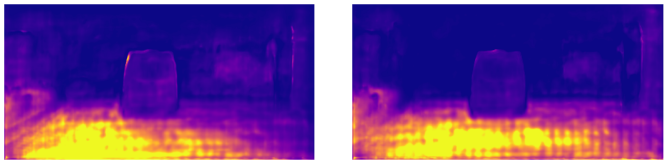
Second, the per-axis AbsRel error exhibits different behaviors depending on the model family and scaling strategy. For MASt3R-DPT and it's variants that use LiDAR-based scaling at test time, the largest AbsRel is usually observed

Hyperparameter	Value
Training Set Resolutions	(512, 384), (512, 336), (512, 288), (512, 256), (512, 160)
Validation Set Resolutions	(512, 384)
Model	AsymmetricMASt3R (ViT-L / DPT head)
Pretrained init	`MASt3R_ViTLarge_BaseDecoder_512_catmlpdpt_metric`
Batch size	8 (gradient accumulation $\times 4$)
Optimizer	AdamW
Learning rate	5×10^{-5}
Min. learning rate	3×10^{-6}
Weight decay	0.05
Warmup epochs	0
Total epochs	20
Save frequency	every 1 epoch (keep every 2)
Eval frequency	every 1 epoch
Loss	`Regr3D(L21, norm_mode=avg_dis, gt_scale=True)`

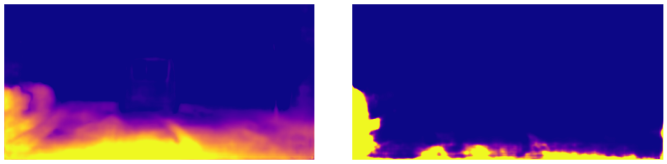
TABLE VII: Fine-tuning hyperparameters for MASt3R-DPT on VBR sequences.



(a) Anchor image (left) and predicted confidence map from the pre-trained model (right).



(b) Confidence maps for MASt3R-DPT trained with *regression loss* at epoch 4 (left) and 12 (right).



(c) Confidence maps for MASt3R-DPT trained with *confidence-weighted regression loss* at epoch 4 (left) and 12 (right), learning rate 1×10^{-6} (minimum 1×10^{-7}).

Fig. 14: Effect of fine-tuning (*Campus→Ciampino2*) with different loss functions on predicted confidence maps. Hyperparameters used are detailed in C-1 unless indicated otherwise.

along the x axis. In contrast, for train-set–scaled variants, the dominant error shifts to the z axis and for fine-tuned variants it generally shifts to the y axis. Meanwhile, the monocular depth estimation models (Depth Anything V2 and ZoeDepth) produce nearly isotropic AbsRel errors, with x, y, and z of similar magnitude. This is expected since the models only predict per pixel depth which are transformed into 3D points using camera intrinsics.

Although fine-tuned MASt3R-DPT reports reduced z-axis AbsRel compared to Depth Anything V2, downstream relative pose estimation shows higher MTE. Overall, while AbsRel distributions differ across methods, the z axis remains the dominant source of error in translation. The rotation errors reported for different H are also significantly lower for monocular depth estimation models. Thus improving depth reliability is the most effective avenue for reducing overall localization error. These results show that lower AbsRel along z does not necessarily imply more accurate pose estimation.

D-2 Per-Axis Explicit Scale Estimation

To investigate the difference between per-axis scaling and a single global scale factor, we examine the distributions of MASt3R-DPT depthmaps (scaled using these explicit scale estimation methods) along the x, y, and z spatial axes and comparing them against the ground-truth Oracle distributions.

Figure 15 shows two representative image pairs, alongside scaled and unscaled MASt3R-DPT depthmaps. The histograms show plots corresponding to each spatial axis, illustrating the spread of points along that axis. The *mean per-axis* L_2 *norm* computes a single global scalar, implicitly assuming isotropic scale error in the MASt3R pointmaps and applying an identical correction across all axes. In contrast, the *per-axis* L_1 *norm* estimates independent scale factors for each axis direction.

The histograms reveal that scale errors are indeed axis-dependent: under global scaling, the coordinate distributions along the x, y, and z axes remain unevenly offset from the LiDAR reference. By contrast, per-axis scaling brings the distributions into closer alignment, matching both their shapes and central statistics (means and medians) to the LiDAR ground truth. This demonstrates that MASt3R predictions require distinct correction factors along individual axes to recover accurate metric geometry.

Attribution of External Resources

This thesis makes use of external resources for formatting and figures:

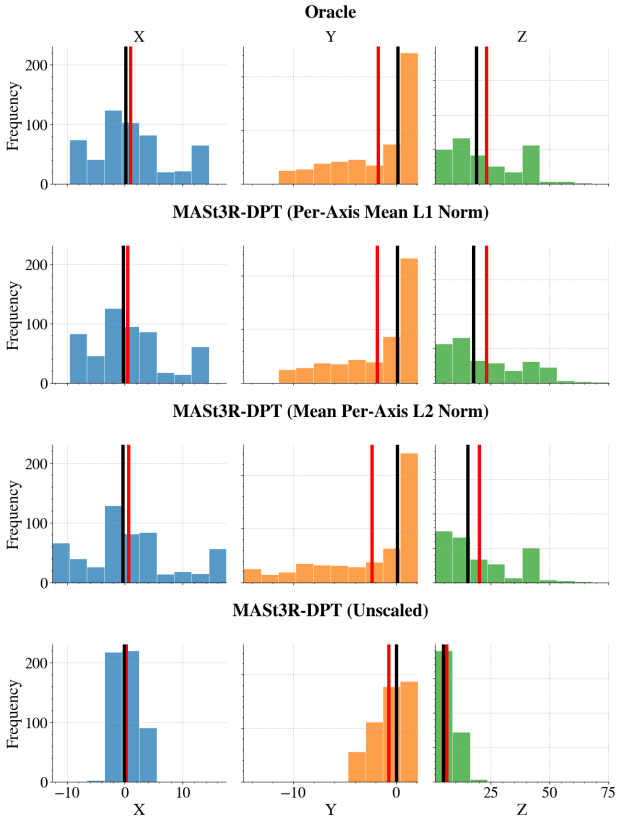
- TU Delft report class template [71]
- IEEE LaTeX template [72]
- Pixabay cover image [73]
- Esri World Imagery (for all satellite imagery) [74]

Method	Spagna				Campus				Ciampino1				Ciampino2			
	x	y	z	MTE	x	y	z	MTE	x	y	z	MTE	x	y	z	MTE
Oracle	0.06	0.04	**0.10**	0.17	0.02	0.05	**0.08**	0.11	0.03	0.05	**0.08**	0.10	0.02	0.03	**0.07**	0.08
MASt3R-DPT (Pre-trained Model [18])	0.83	0.14	**1.42**	2.71	0.28	0.09	**2.04**	2.12	0.35	0.14	**4.15**	4.18	0.25	0.12	**1.89**	2.01
MASt3R-DPT (Confidence Thresholded)	0.81	0.13	**1.39**	2.50	0.27	0.09	**1.65**	1.73	0.39	0.11	**2.52**	2.55	0.30	0.08	**1.67**	1.69
MASt3R-DPT Scaled: *LiDAR used during inference*																
Per Axis L_1 Norm (*)	0.14	0.11	**0.39**	0.60	0.09	0.22	**0.58**	0.74	0.06	0.12	**0.35**	0.40	0.07	0.06	**0.34**	0.39
Per Axis L_2 Norm (*)	0.19	0.15	**0.57**	0.81	0.10	0.24	**0.87**	1.01	0.07	0.15	**0.43**	0.50	0.09	0.09	**0.43**	0.50
Mean Per Axis L_2 Norm (*)	0.22	0.09	**0.76**	0.90	0.44	0.21	**5.86**	5.90	0.71	0.28	**6.66**	6.78	0.47	0.24	**5.13**	5.14
Similarity Transform (*)	0.27	0.09	**0.68**	1.01	0.44	0.21	**5.79**	6.06	0.59	0.27	**6.25**	6.39	0.43	0.22	**4.69**	4.70
MASt3R-DPT Scaled: *Train-set Median Scale*																
Per Axis L_1 Norm	0.27	0.07	**0.67**	0.91	0.18	0.24	**2.14**	2.20	0.17	0.12	**2.31**	2.32	0.08	0.09	**0.84**	0.89
Per Axis L_2 Norm	0.28	0.11	**0.67**	0.94	0.20	0.36	**2.26**	2.30	0.16	0.16	**2.18**	2.25	0.10	0.16	**0.87**	0.96
Mean Per Axis L_2 Norm	0.31	0.10	**0.89**	1.12	0.46	0.17	**5.09**	5.12	0.71	0.30	**7.78**	7.79	0.56	0.27	**6.21**	6.22
Similarity Transform	0.32	0.10	**0.82**	1.06	0.47	0.17	**5.23**	5.25	0.64	0.27	**7.36**	7.44	0.50	0.24	**5.33**	5.38
MASt3R-DPT (*Campus→Ciampino2*)	0.45	0.13	1.74	2.01	0.08	0.08	**0.82**	0.83	0.11	0.10	**1.29**	1.31	0.08	0.08	**0.72**	0.76
MASt3R-DPT (*Ciampino1→Ciampino2*)	0.46	0.17	2.70	2.95	0.11	0.09	**1.12**	1.15	0.07	0.08	**0.57**	0.59	0.06	0.05	**0.44**	**0.47**
ZoeDepth [24]	0.91	0.14	**1.59**	2.84	0.20	0.02	**3.22**	3.23	0.28	0.03	**3.37**	3.39	0.21	0.02	**2.73**	2.75
Depth Pro [25]	0.25	0.06	**0.63**	1.16	0.06	0.05	**0.89**	0.90	0.05	0.05	**0.32**	0.35	0.04	0.04	**0.24**	0.28
Depth Anything V2 [26]	0.39	0.09	**0.60**	0.86	0.07	0.04	**0.32**	0.33	0.08	0.04	**0.33**	0.38	0.09	0.03	**0.57**	0.59

TABLE VIII: Median translation error per direction (X, Y, Z) and overall median (MTE) on *test*[1] splits (*meter*). * indicates LiDAR-based per-pair scale estimation at inference; unstarred rows use train-set median scale. **Bold** values denote the highest error per axes for each scene for MASt3R-DPT variants. Fine-tuning MASt3R-DPT (*Train→Val*) may result in higher translation errors (shown in red).

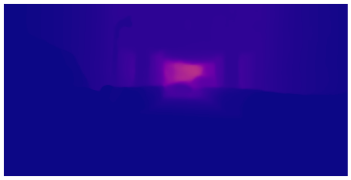
Method	Spagna				Campus				Ciampino1				Ciampino2			
	x	y	z	Med	x	y	z	Med	x	y	z	Med	x	y	z	Med
MASt3R-DPT (Pre-trained Model [18])	**0.84**	0.76	0.78	0.78	**0.72**	0.27	0.42	0.41	**0.67**	0.50	0.61	0.60	**0.68**	0.50	0.64	0.62
MASt3R-DPT (Confidence Thresholded)	**0.85**	0.76	0.77	0.77	**0.72**	0.25	0.41	0.40	**0.66**	0.50	0.61	0.61	**0.68**	0.49	0.63	0.61
MASt3R-DPT Scaled: *LiDAR used during inference*																
Per Axis L_1 Norm (*)	**0.70**	0.19	0.14	0.15	**0.82**	0.16	0.16	0.17	**0.59**	0.15	0.13	0.14	**0.72**	0.19	0.17	0.18
Per Axis L_2 Norm (*)	**0.69**	0.20	0.15	0.16	**0.89**	0.19	0.20	0.21	**0.60**	0.16	0.15	0.16	**0.73**	0.20	0.20	0.21
Mean Per Axis L_2 Norm (*)	**0.75**	0.22	0.16	0.17	**1.09**	0.42	0.16	0.20	**0.80**	0.36	0.14	0.18	**1.00**	0.47	0.17	0.23
Similarity Transform (*)	**0.71**	0.26	0.21	0.22	**1.06**	0.39	0.17	0.20	**0.73**	0.27	0.17	0.20	**0.81**	0.36	0.19	0.23
MASt3R-DPT Scaled: *Train-set Median Scale*																
Per Axis L_1 Norm	0.54	0.24	**2.00**	2.14	1.18	0.58	**4.64**	4.92	1.09	0.45	**4.28**	4.58	0.56	0.26	**1.85**	2.00
Per Axis L_2 Norm	0.56	0.26	**2.00**	2.15	1.35	0.65	**5.34**	5.63	1.09	0.45	**4.28**	4.53	0.56	0.27	**1.85**	2.04
Mean Per Axis L_2 Norm	0.61	0.29	**1.93**	2.14	1.45	0.79	**4.62**	5.18	0.96	0.29	**4.54**	4.90	0.95	0.50	**2.08**	2.39
Similarity Transform	0.54	0.25	**1.76**	1.93	1.52	0.82	**4.72**	5.34	1.03	0.38	**4.49**	4.95	0.66	0.32	**2.72**	3.02
MASt3R-DPT (*Campus→Ciampino2*)	0.45	**0.53**	0.26	0.28	0.22	**0.27**	0.11	0.12	0.25	**0.28**	0.16	0.16	0.34	**0.44**	0.20	0.20
MASt3R-DPT(*Ciampino1→Ciampino2*)	**0.49**	0.45	0.26	0.29	0.19	**0.23**	0.12	0.13	0.16	**0.20**	0.11	0.11	0.20	**0.31**	0.15	0.15
ZoeDepth [24]	6.21	6.22	6.21	6.21	7.42	7.44	7.41	7.41	6.72	6.72	6.72	6.72	6.80	6.81	6.79	6.79
Depth Pro [25]	0.91	0.92	0.91	0.91	0.41	0.41	0.41	0.41	0.22	0.22	0.22	0.22	0.28	0.28	0.28	0.28
Depth Anything V2 [26]	0.23	0.23	0.23	0.23	0.21	0.21	0.20	0.20	0.21	0.21	0.21	0.21	0.27	0.27	0.26	0.26

TABLE IX: Median AbsRel error per direction (X, Y, Z) and overall median on *test*[1] splits. * indicates LiDAR-based per-pair scale; unstarred rows use train-set median scale. **Bold** values denote the highest error among per axes for each scene for MASt3R-DPT variants. Depth Anything V2 has the most consistent performance across scenes (shown in green).

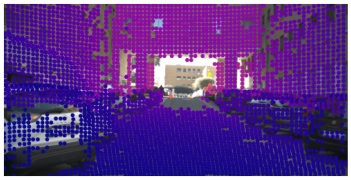


(a) Spatial Distributions

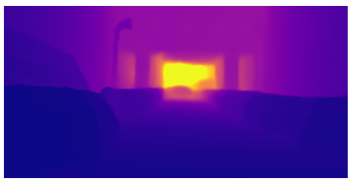
(b) Anchor (left) and query (right) with 1532 matches, 1506 inliers



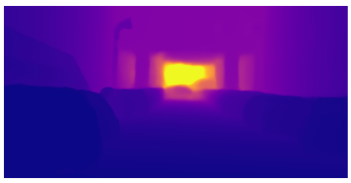
(c) MASt3R-DPT Unscaled



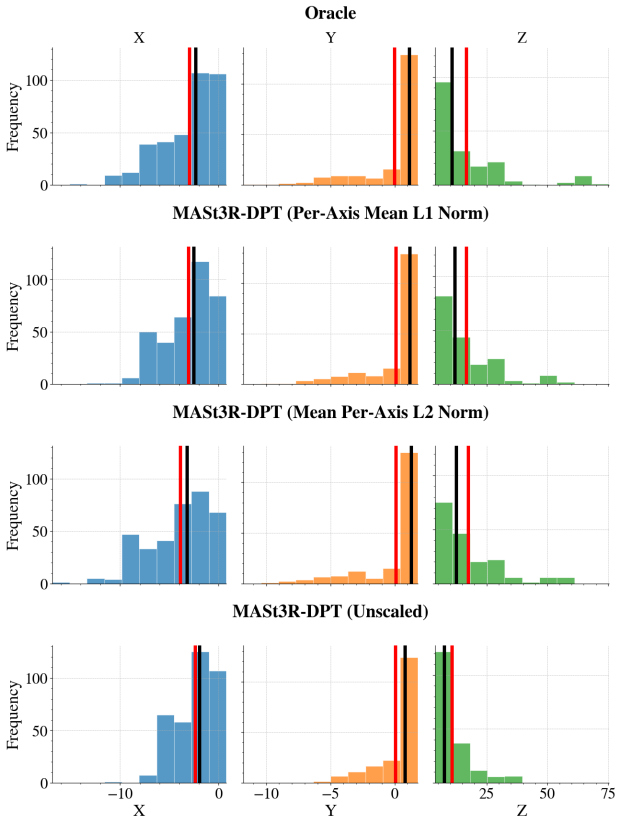
(d) Oracle



(e) MASt3R-DPT (Per-Axis L_1 Norm)



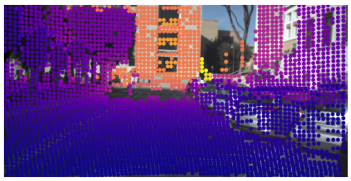
(f) MASt3R-DPT (Mean Per-Axis L_2 Norm)



(g) Spatial Distributions

(h) Anchor (left) and query (right) with 1064 matches, 894 inliers

(i) MASt3R-DPT Unscaled



(j) Oracle

(k) MASt3R-DPT (Per-Axis L_1 Norm)

(l) MASt3R-DPT (Mean Per-Axis L_2 Norm)

Fig. 15: *Left:* Distribution of matched 3D points along each spatial axis (X: blue, Y: orange, Z: green). *Right:* Anchor-query pairs with ground truth and MASt3R-DPT depthmaps. Spatial distributions show that per-axis scaling aligns **mean** and **median** (vertical lines) better.