MapAnything: Universal Feed-Forward Metric 3D Reconstruction

map-anything.github.io

Nikhil Keetha [1,2] Norman Müller [1] Johannes Schönberger [1] Lorenzo Porzi [1] Yuchen Zhang [2]
Tobias Fischer [1] Arno Knapitsch [1] Duncan Zauss [1] Ethan Weber [1] Nelson Antunes [1]
Jonathon Luiten [1] Manuel Lopez-Antequera [1] Samuel Rota Bulò [1] Christian Richardt [1]
Deva Ramanan [2] Sebastian Scherer [2] Peter Kontschieder [1]

[1] Meta Reality Labs [2] Carnegie Mellon University

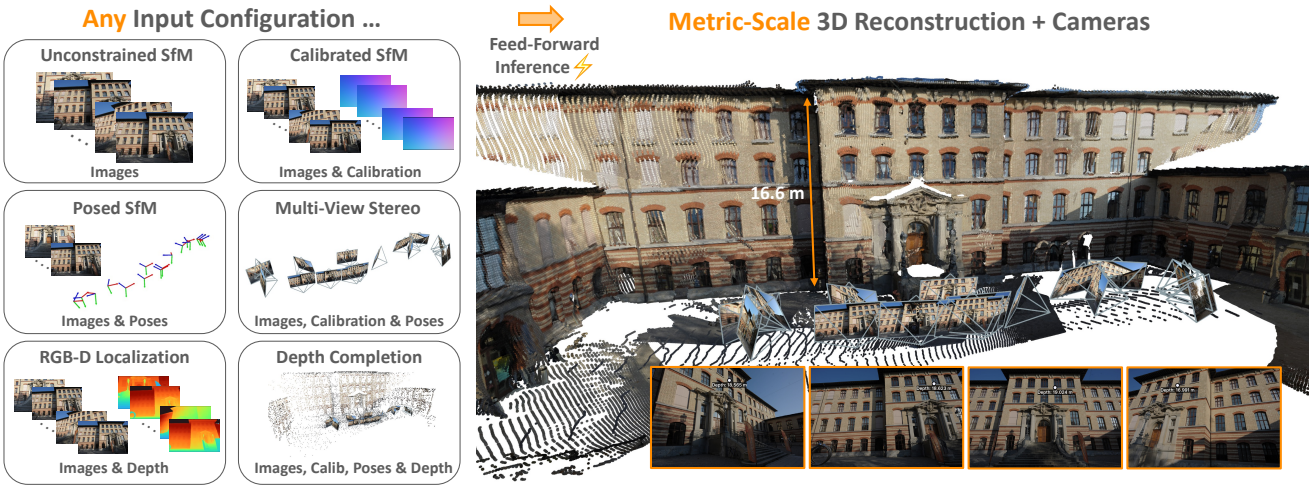

Figure 1. **MapAnything is a flexible, unified feed-forward 3D reconstruction model** that predicts metric 3D reconstructions with camera information from a set of N input images with optional camera poses, intrinsics, or depth maps. MapAnything supports over 12 different 3D reconstruction tasks, including camera localization, structure-from-motion (SfM), multi-view stereo, and metric depth completion, outperforming or matching the quality of specialist methods.

Abstract

We introduce MapAnything, a unified transformer-based feed-forward model that ingests one or more images along with optional geometric inputs such as camera intrinsics, poses, depth, or partial reconstructions, and directly regresses the metric 3D scene geometry and cameras. MapAnything leverages a factored representation of multi-view scene geometry, i.e., a collection of depth maps, local raymaps, camera poses, and a metric scale factor that effectively upgrades local reconstructions into a globally consistent metric frame. Standardizing the supervision and training across diverse datasets, along with flexible input augmentation, enables MapAnything to address a broad range of 3D vision tasks in a single feed-forward pass, including uncalibrated structure-from-motion, calibrated multi-view stereo, monocular depth estimation, camera localization, depth completion, and more. We provide extensive experimental analyses and model ablations demonstrating that MapAnything outperforms or matches specialist feed-forward models while offering more efficient joint training behavior, thus paving the way toward a universal 3D reconstruction backbone.

1. Introduction

The problem of image-based 3D reconstruction has traditionally been solved using structure-from-motion (SfM) [43, 52], photometric stereo [77], shape-from-shading [17], and so on. To make the problem tractable, classic approaches decompose it into distinct tasks, such as feature detection [33] and matching [49], two-view pose estimation [40], camera calibration [64] and resectioning [50], rotation [13] and translation averaging [43], bundle adjustment (BA) [60], multi-view stereo (MVS) [53], and/or monocular surface estimation [16]. Recent work has demonstrated tremendous potential in solving these problems in a unified way using feed-forward architectures [8, 22, 29, 68, 72, 84].

While prior feed-forward work has approached the different tasks separately or by not leveraging all the available input modalities, we present a unified end-to-end model for diverse 3D reconstruction tasks. Our method MapAnything can be used to solve the most general uncalibrated SfM problem as well as various combinations of sub-problems, such as calibrated SfM or multi-view stereo, monocular depth estimation, camera localization, and metric depth completion. To enable the training of such a unified model, we: (1) introduce a flexible input scheme that supports various geometric modalities when available, (2) propose a suitable output space that supports all of these diverse tasks, and (3) discuss flexible dataset aggregation and standardization.

MapAnything's key insight to address these challenges is the use of a *factored* representation of multi-view scene geometry. Instead of directly representing the scene as a collection of pointmaps, we represent the scene as a collection of depth maps, local raymaps, camera poses, and a metric scale factor that upgrade local reconstructions into a globally consistent metric frame. We use such a factored representation to represent both the outputs and (optional) inputs for MapAnything, allowing it to take advantage of auxiliary geometric inputs when available. For example, robotic applications [1, 15, 19, 27] may have knowledge of camera intrinsics (rays) and/or extrinsics (poses). Finally, a significant benefit of our factored representation is that it allows MapAnything to be effectively trained from diverse datasets with partial annotations, for example, datasets that may be annotated with only non-metric "up-to-scale" geometry. In summary, we make the following main contributions:

1. **Unified Feed-Forward Model** for multi-view metric 3D reconstruction that supports more than 12 different problem configurations. The end-to-end transformer is trained more efficiently than a naive set of bespoke models and leverages not only image inputs, but also optional geometric information such as camera intrinsics, extrinsics, depth, and/or metric scale factor, when available.

2. **Factored Scene Representation** that flexibly enables decoupled inputs and effective prediction of metric 3D reconstructions. Our model computes multi-view pixel-wise scene geometry and cameras directly, without redundancies or costly post-processing.

3. **State-of-the-Art Performance** compared to other feed-forward models, matching or surpassing expert models that are tailored for specific, isolated tasks.

4. **Open Source Release** of (a) code for data processing, inference, benchmarking, training & ablations, and (b) a pre-trained MapAnything model under the permissive Apache 2.0 license, thereby providing an extensible & modular framework plus model to facilitate future research on building 3D/4D foundation models.

2. Related Work

Towards Universal 3D Reconstruction. In contrast to the traditional approach of designing specialized methods for distinct reconstruction tasks, recent efforts have shown great promise in solving them jointly with a single feed-forward architecture. Early works like DeMoN [61], DeepTAM [87] or DeepV2D [57] explored this direction with CNNs but did not match the performance of classical expert models. Enabled by advances in deep learning, recent methods like PF-LRM [68], RayDiffusion [84], DUSt3R [72], VGGSfM [66], and VGGT [67] scale up transformers on large amounts of data. Despite this breakthrough, these methods are still limited to a subset of 3D reconstruction tasks with fixed inputs and output modalities, a small or fixed number of views, or they only work well in relatively constrained, typically object-centric, scenarios. With MapAnything, we overcome these limitations by designing a geometrically grounded and flexible architecture that supports heterogeneous input and output modalities for any number of input views.

Multi-View Feed-forward Reconstruction. DUSt3R and its metric follow-up MASt3R [29] predict a coupled scene representation (i.e., cameras, poses, and geometry are parameterized by a pointmap and need to be recovered post hoc) and require expensive post-processing & symmetric inference to perform multi-view unconstrained SfM. Follow-up work [10, 11, 39, 44] integrates MASt3R outputs into classical SfM and SLAM pipelines in a more principled manner. Recent works like Spann3R [65], CUT3R [69], and MUSt3R [6] remove the need for classical optimization and enable multi-view reconstruction via latent state memory in transformers. However, these works do not yet match the performance of traditional optimization applied to predicted two-view outputs from MASt3R [10, 39].

Recently, MV-DUSt3R+ [56] and VGGT [67] demonstrate multi-view inference by extending the DUSt3R architecture for multi-view reconstruction. Likewise, Reloc3r [8] focuses on camera re-localization and directly predicts multi-view camera poses. MV-DUSt3R+ achieves this by parallelizing the cross-attention transformer to support different reference views, leading to a significant increase in computation, while VGGT employs an alternating attention transformer to predict multi-view pointmaps, depth, pose, and features for tracking. FASt3R [78] uses positional encoding for long-sequence inference in LLMs for global attention trained on a few views to work on a larger number of views. More recently, π^3 [74] fine-tunes VGGT to remove the use of the first input frame as reference coordinate.

In both MV-DUSt3R+ and FASt3R, the prediction is a coupled scene representation and cannot handle heterogeneous inputs. As shown in FASt3R, for the multi-view setup, the dense geometry prediction capabilities of the model are impacted by the pose estimation across non-visible views

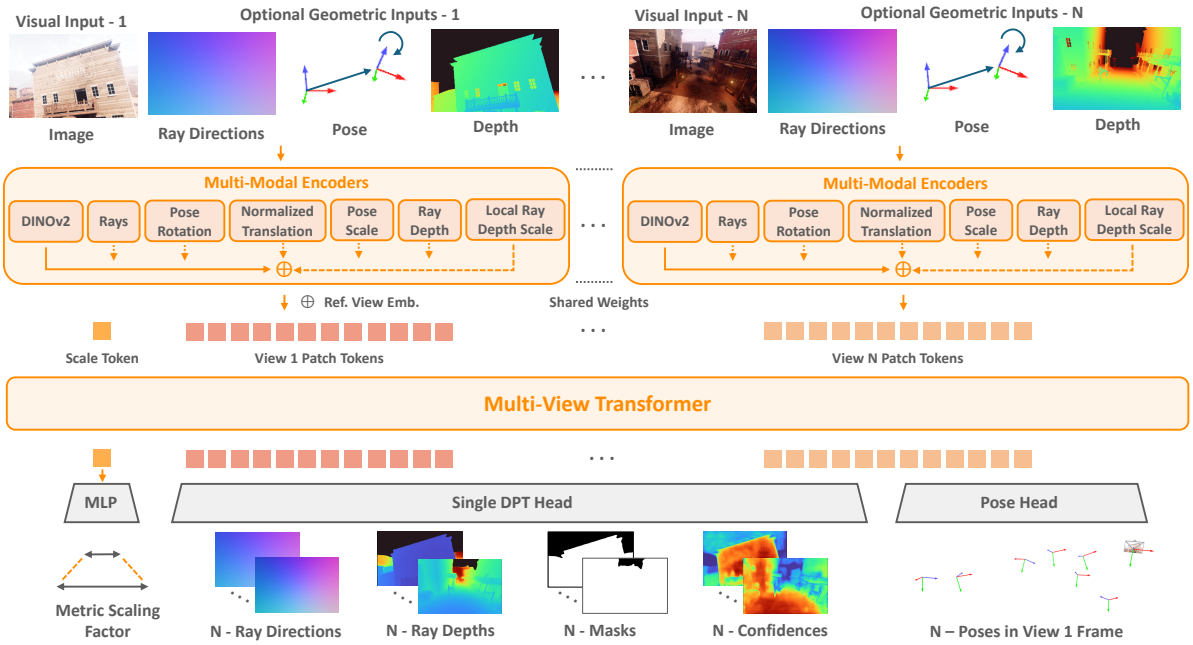

Figure 2. **Overview of the MapAnything Architecture.** Given N visual and optional geometric inputs, the model first encodes the images and the factored representation of the geometric inputs into a common latent space where the patch features (for images, rays & depth) and broadcasted global features (for translation, rotation, pose scale across all pose inputs & depth scale local to each frame) are summed together. Then, a fixed reference view embedding is added to the first view's features and a single learnable scale token is appended to the set of N view patch tokens. These tokens are then input into an alternating-attention transformer. We use a single DPT to decode the N view patch tokens into N dense outputs local to all the views. A single average pooling-based pose head also uses the N view patch tokens to predict N poses in the frame of view 1. Lastly, while these predictions exist in an up-to-scale space, the model passes the scale token through an MLP to predict the metric scaling factor, which when coupled with the other predictions, provides the dense metric 3D reconstruction.

(see their Table 5 & Section 5.1). To alleviate this issue, FASt3R predicts redundant pointmaps across all views with a dedicated DPT head for global and local pointmap prediction. Likewise, VGGT also predicts multiple redundant quantities through two separate branches, one for pointmaps and one for cameras and depth. While concurrent work, π^3 [74], fine-tunes VGGT to remove this redundancy by predicting up-to-scale decoupled local pointmaps and global pose, we find this design choice to be sub-optimal (see Table 5a). In contrast, MapAnything directly predicts a completely factored representation, i.e., local ray directions, depth along the ray, global camera pose for all views, and a single metric scaling factor for the scene. In this formulation, the task of predicting ray directions (akin to camera calibration) and depth-along-ray estimation are per-view and thus can be predicted from a single dense prediction head.

While prior work has paved the way for unconstrained multi-view inference and large-scale training, they are all limited to only image inputs and modeling a simple pinhole camera. In contrast, MapAnything supports various 3D reconstruction and calibration tasks from multiple views with heterogeneous inputs and a flexible camera model.

Geometry as Inputs or Conditioning. While not explicitly used for feed-forward 3D reconstruction like in MapAnything, quantities such as ray directions, origins, and depth maps have been explored as conditioning inputs for tasks like novel-view synthesis [24, 35, 75, 88], diffusion-based image generation [38, 85], dynamic video depth estimation [34], or 3D object shape completion [9]. Taskonomy [83] explored the benefits of multi-task learning for improved vision task performance. Later works like MultiMAE [3] build on these insights and devise auto-encoders to support flexible combination of heterogeneous inputs; however, this is not suitable for solving 3D reconstruction tasks. Pow3R [23] was the first work to incorporate known priors as inputs to feed-forward 3D reconstruction. In contrast to us, Pow3R only supports two pinhole camera images with a single focal length and centered principal point. Furthermore, Pow3R builds on top of DUSt3R and cannot condition on metric scale information. In contrast, MapAnything supports any number of input views and has a flexible input parameterization that supports metric scale and any camera with a central projection model.

3. MapAnything

MapAnything is an end-to-end model that takes as input N RGB images $\hat{\mathcal{I}} = (\hat{I}_i)_{i=1}^N$ and optional geometric inputs corresponding to all or a subset of the input views:

(**a**) generic central camera calibrations [12, 63, 84] as ray directions $\hat{\mathcal{R}} = (\hat{R}_i)_{i \in S_\mathrm{r}}$,

(**b**) poses in the frame of the first view $\hat{I}_1$ as quaternions $\hat{\mathcal{Q}} = (\hat{Q}_i)_{i \in S_\mathrm{q}}$ and translations $\hat{\mathcal{T}} = (\hat{T}_i)_{i \in S_\mathrm{t}}$, and

(**c**) ray depth for each pixel $\hat{\mathcal{D}} = (\hat{D}_i)_{i \in S_\mathrm{d}}$,

where $S_\mathrm{r}, S_\mathrm{q}, S_\mathrm{t}, S_\mathrm{d}$ are subsets of frame indices $[1, N]$.

MapAnything maps these inputs to an N-view factored metric 3D output (as shown in Figure 2):

$$f_{\text{MapAnything}}\big(\hat{\mathcal{I}}, [\hat{\mathcal{R}}, \hat{\mathcal{Q}}, \hat{\mathcal{T}}, \hat{\mathcal{D}}]\big) = \{m, (R_i, \tilde{D}_i, \tilde{P}_i)_{i=1}^{N})\}, \quad (1)$$

where $m \in \mathbb{R}$ is the predicted global metric scaling factor, and for each view i, $R_i \in \mathbb{R}^{3 \times H \times W}$ are the predicted local ray directions, $\tilde{D}_i \in \mathbb{R}^{1 \times H \times W}$ are the ray depths in a up-to-scale space (indicated by the tilde), and $\tilde{P}_i \in \mathbb{R}^{4 \times 4}$ is the pose of image $\hat{I}_i$ in the frame of image $\hat{I}_1$, represented as quaternion Q_i and up-to-scale translation $\tilde{T}_i \in \mathbb{R}^3$. We can further use this factored output to get the up-to-scale local pointmaps (3D points corresponding to each pixel) as $\tilde{L}_i = R_i \cdot \tilde{D}_i \in \mathbb{R}^{3 \times H \times W}$. Then, leveraging the rotation matrix O_i (obtained from Q_i) and up-to-scale translation, we can compute the N-view up-to-scale pointmaps in world frame as $\tilde{X}_i = O_i \cdot \tilde{L}_i + \tilde{T}_i$. The final metric 3D reconstruction for the N input views (in the frame of image I_1) is given by $X_i^{\text{metric}} = m \cdot \tilde{X}_i$ for $i \in [1, N]$.

3.1. Encoding Images & Geometric Inputs

Given N visual inputs and optional dense geometric inputs, we first encode them into a common latent space. For images, we use DINOv2 (Apache 2.0) [42]. Among a wide variety of pre-trained options, such as CroCov2 [76], DUSt3R's image encoder [72], RADIO [14, 48], and random-init linear patchification, we find DINOv2 to be optimal in terms of downstream performance, convergence speed, and generalization (especially when fine-tuned with a small learning rate). We use the 24th layer normalized patch features from DINOv2 ViT-G, $F_\mathrm{I} \in \mathbb{R}^{1536 \times H/14 \times W/14}$.

MapAnything can also encode other geometric quantities. Before feeding these geometric quantities to our network, we factorize them to enable training and inference across both metric and up-to-scale quantities. To support use cases where only rotation or translation might be individually present (for e.g., IMU & GPS priors) and to deal with the entanglement of translation with scale, we encode rotation and translation separately. Furthermore, since we don't assume depth & pose to always be provided together as input, we decouple their normalization (note that this is separate from the training objective where we normalize predicted depth & pose together since we want multi-view consistency).

In particular, when provided, the ray depths are first decoupled into average per-view depth $\hat{z}_{di} \in \mathbb{R}^+$ and normalized ray depths $\hat{D}_i / \hat{z}_{di}$. Furthermore, when translations $\hat{\mathcal{T}}$ are provided, MapAnything computes the pose scale as the average distance to the world frame, $\hat{z}_\mathrm{p} = \frac{1}{|S_t|} \sum_{i \in S_t} \|\hat{T}_i\|$. This pose scale is used as the same input for all frames with input translation and is also used to get the normalized translations $\hat{T}_i / \hat{z}_\mathrm{p}$. Since we are interested in effectively exploiting the metric scale information from geometric inputs, MapAnything only uses the pose scale and depth scales when the poses and depths provided for specific frames are metric. Furthermore, the metric scale values can be large and drastically vary across scene sizes, hence, we log-transform scales before encoding them.

We encode ray directions and normalized ray depths using a shallow convolutional encoder [38], where the spatial resizing only happens once with a pixel unshuffle of size 14. This projects the dense geometric inputs into the same spatial and latent dimension as the DINOv2 features, i.e., $F_\mathrm{R}, F_\mathrm{D} \in \mathbb{R}^{1536 \times H/14 \times W/14}$. For the global non-pixel quantities, i.e, rotations (represented as unit quaternions), translation directions, depth and pose scales, we use a 4-layer MLP with GeLU activations to project the quantities to features $F_\mathrm{Q}, F_\mathrm{T}, F_{\hat{z}_d}, F_{\hat{z}_\mathrm{p}} \in \mathbb{R}^{1536}$. Once all input quantities are encoded, they are passed through layer normalization, summed together, and followed by another layer normalization to obtain the final per-view encodings for each input view. These are then flattened into tokens $F_\mathrm{E} \in \mathbb{R}^{1536 \times (HW/256)}$.

We append a single learnable scale token to the set of N view patch tokens and input the tokens into a multi-view transformer to allow information across multiple views to attend to each other and propagate. We use a 16-layer alternating-attention transformer [67] with 24 heads of multi-headed attention, a latent dimension of 1536 and an MLP ratio of 4, initialized using the last 16 layers of DINOv2 ViT-G [31, 41]. To distinguish the reference view (i.e., the first one), we add a constant reference view embedding to the set of patch tokens corresponding to view I_1. For simplicity, we do not use Rotary Positional Embedding (RoPE) [55]. We find that the patch-level positional encoding from DINOv2 suffices, and RoPE leads to unnecessary biases, given that it was originally applied in every attention layer.

3.2. Factored Scene Representation Prediction

Once the multi-view transformer fuses information across different views and outputs the N-view patch tokens and scale token, MapAnything further decodes these tokens into factored quantities representing the metric 3D geometry. In particular, we use a DPT head [46] to decode the N-view patch tokens into N dense per-view outputs, i.e., ray directions R_i (normalized to unit length), up-to-scale ray depths $\tilde{D}_i$, masks M_i representing non-ambiguous classes for depth, and world-frame pointmap confidence maps C_i. Furthermore, we input the N-view patch tokens into an average pooling-based convolutional pose head [7] to predict the unit quaternions Q_i and up-to-scale translations $\tilde{T}_i$. Finally, the scale token is passed through a 2-layer MLP with ReLU activations to predict the metric scaling factor. Since the metric scale of a scene can vary vastly, we exponentially scale the

prediction to obtain the metric scaling factor m. As shown in Table 5a, we find that this decoupling of scale prediction is critical to achieving universal metric feed-forward inference. Finally, as mentioned earlier, these factored predictions can be used together to obtain the metric 3D reconstruction.

3.3. Training Universal Metric 3D Reconstruction

We train MapAnything end-to-end using multiple losses depending on the available supervision. Since ray directions R_i and pose quaternions Q_i do not depend on scene scale, their losses are: $\mathcal{L}_{\text{rays}} = \sum_{i=1}^{N} \|\hat{R}_i - R_i\|$ and $\mathcal{L}_{\text{rot}} = \sum_{i=1}^{N} \min(\|\hat{Q}_i - Q_i\|, \|-\hat{Q}_i - Q_i\|)$. This accounts for the two-to-one mapping of unit quaternions, and the regression loss is similar to a geodesic angular distance.

For the predicted up-to-scale ray depths $\tilde{D}_i$, pose translations $\tilde{T}_i$, local pointmaps $\tilde{L}_i$ and world frame pointmaps $\tilde{X}_i$, we follow DUSt3R [72] and use the ground-truth validity masks V_i to compute the scaling factors for the ground truth $\hat{z} = \|(\hat{X}_i[V_i])_{i=1}^{N}\| / \sum_{i=1}^{N} V_i$ and the up-to-scale predictions $\tilde{z} = \|(\tilde{X}_i[V_i])_{i=1}^{N}\| / \sum_{i=1}^{N} V_i$. Likewise, to ensure that gradients from the scale loss do not influence the geometry, we use the predicted metric scaling factor m and detached up-to-scale norm scaling factor $\tilde{z}$ to compute the metric norm scaling factor $z^{\text{metric}} = m \cdot \text{sg}(\tilde{z})$, where sg indicates stop-grad.

Given these scaling factors, we compute the scale-invariant translation loss as $\mathcal{L}_{\text{translation}} = \sum_{i=1}^{N} \|\hat{T}_i/\hat{z} - \tilde{T}_i/\tilde{z}\|$. We find that it is critical to apply losses in log-space for ray depths, pointmaps and the metric scale factor. Specifically, we use $f_{\log} : \mathbf{x} \to (\mathbf{x}/\|\mathbf{x}\|) \cdot \log(1 + \|\mathbf{x}\|)$. Thus, the loss for the ray depths is $\mathcal{L}_{\text{depth}} = \sum_{i=1}^{N} \|f_{\log}(\hat{D}_i/\hat{z}) - f_{\log}(\tilde{D}_i/\tilde{z})\|$. Likewise, the loss for the local pointmaps is $\mathcal{L}_{\text{lpm}} = \sum_{i=1}^{N} \|f_{\log}(\hat{L}_i/\hat{z}) - f_{\log}(\tilde{L}_i/\tilde{z})\|$. We exclude the top 5% of per-pixel loss values to ignore imperfections and potential outliers in the training data. Similar to DUSt3R, we add $\mathcal{L}_{\text{pointmap}} = \sum_{i=1}^{N} (C_i \|f_{\log}(\hat{X}_i/\hat{z}) - f_{\log}(\tilde{X}_i/\tilde{z})\| - \alpha \log(C_i))$ as a confidence-weighted pointmap loss. Lastly, the factored metric scale loss is given by $\mathcal{L}_{\text{scale}} = \|f_{\log}(\hat{z}) - f_{\log}(z^{\text{metric}})\|$.

To capture fine details, we also employ a normal loss $\mathcal{L}_{\text{normal}}$ [70] on the local pointmaps, and a multi-scale gradient matching loss $\mathcal{L}_{\text{GM}}$ [47, 79] on the log of the z-depth in the local pointmaps. Since the geometry from real datasets can be coarse and noisy, we apply the $\mathcal{L}_{\text{normal}}$ and $\mathcal{L}_{\text{GM}}$ losses only to synthetic datasets. For the predicted non-ambiguous class masks, we use a binary cross entropy loss ($\mathcal{L}_{\text{mask}}$). Overall, we use the following total loss:

$$\begin{aligned}\mathcal{L} = &10 \cdot \mathcal{L}_{\text{pointmap}} + \mathcal{L}_{\text{rays}} + \mathcal{L}_{\text{rot}} + \mathcal{L}_{\text{translation}} + \mathcal{L}_{\text{depth}} \\ &+ \mathcal{L}_{\text{lpm}} + \mathcal{L}_{\text{scale}} + \mathcal{L}_{\text{normal}} + \mathcal{L}_{\text{GM}} + 0.1 \cdot \mathcal{L}_{\text{mask}}\end{aligned} \tag{2}$$

For the factored predictions, we find that up-weighting the global pointmap loss and down-weighting the mask loss is beneficial. For all the regression losses, we use an adaptive robust loss [4] (with parameters $c = 0.05$ and $\alpha = 0.5$) to help with robustness to outliers.

Training for Image & Geometric Inputs: To enable one-shot training of a universal model that supports various input configurations, we provide additional geometric inputs to the model with varying selection probabilities during training. Specifically, we use an overall geometric input probability of 0.9, where each individual factorization, i.e., ray directions, ray depth, and pose, has an input probability of 0.5 each. Whenever depth is selected as input, there is an equal probability of providing dense depth or 90% randomly sparsified depth. For robustness and flexibility in terms of which views have geometric information available as input, we use a per-view input probability of 0.95 and do not provide metric scale factors as input for metric-scale ground-truth datasets with a probability of 0.05. We provide further details regarding the training setup in the supplement.

Datasets: We train MapAnything on 13 high-quality datasets (see Table 1) with diversity across indoor, outdoor, and in-the-wild scenes. For ScanNet++ v2 and TartanAirV2-WB, we split the scenes into a training, validation, and a held-out test set, while other datasets are split into training and validation. While MPSD is originally a monocular metric depth dataset, we acquire the pose and camera information to enable a real-world multi-view metric scale dataset with ∼72K scenes. We open-sourced this MPSD metadata to enable future research. We release two pretrained models: one licensed under Apache 2.0 trained on six datasets, and one licensed under CC BY-NC 4.0 trained on an additional seven datasets (see Table 1). We provide comparisons between both variants in the supplementary.

Multi-View Sampling: For each dataset, we exhaustively precompute the pairwise covisibility of all images in a scene using a reprojection error check based on ground-truth depth and pose. During training, we use this precomputed covisibility with a selected covisibility threshold of 25% to perform

Table 1. **Datasets used for training and testing MapAnything.**

Dataset	License	# Scenes	Metric
BlendedMVS [80]	CC BY 4.0	493	✗
Mapillary Planet-Scale Depth [36]	CC BY-NC-SA[1]	71,428	✓
ScanNet++ v2 [81]	Non-commercial[1]	926	✓
Spring [37]	CC BY 4.0	37	✓
TartanAirV2-WB [73, 86]	CC BY 4.0	49	✓
UnrealStereo4K [59]	MIT	9	✓
Additionally used for our CC BY-NC model:			
Aria Synthetic Environments [2]	Non-commercial	103,890	✓
DL3DV-10K [32]	CC BY-NC 4.0	10,109	✗
Dynamic Replica [25]	Non-commercial	523	✓
MegaDepth [30]	CC BY 4.0[2]	269	✗
MVS-Synth [21]	Non-commercial	120	✓
ParallelDomain-4D [62]	Non-commercial	1,528	✓
SAIL-VOS 3D [20]	Non-commercial	171	✓
Unique held-out scenes for dense up-to-N-view benchmarking:			
ETH3D [54]	CC BY-NC-SA 4.0	13	✓
ScanNet++ v2 [81]	Non-commercial[1]	30	✓
TartanAirV2-WB [73, 86]	CC BY 4.0	5	✓

[1] We obtained approval from the dataset owners that allows training and model release under a permissive license. [2] Crowd-sourced images with varying licenses.

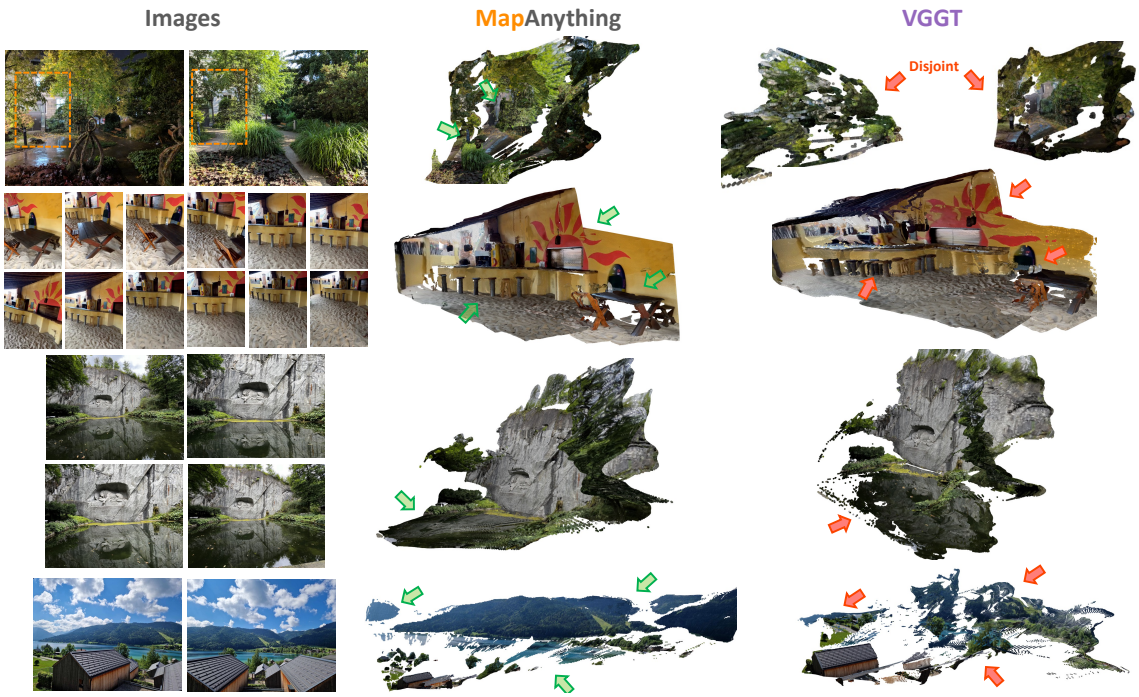

Figure 3. **Qualitative comparison of MapAnything to VGGT [67] using only in-the-wild images as input.** For a fair comparison, we apply the same normal-based edge mask post-processing and our sky mask to both methods. MapAnything more effectively deals with large disparity changes, seasonal shifts, textureless surfaces, water bodies and large scenes.

random walk sampling. This enables us to sample random single-connected component graphs of covisible views that have varying coverage and mutual information.

4. Benchmarking & Results

In this section, we benchmark MapAnything across a wide suite of 3D vision tasks. For each task, we compare against expert baselines specifically designed or trained for the task. We perform all experiments with a constant seed.

Multi-View Dense Reconstruction: We benchmark the performance of pointmaps, pose, depth & ray direction estimation on an undistorted version of ETH3D [54], ScanNet++ v2 [81], and TartanAirV2-WB [73, 86], where, for each test scene, we randomly sample up to N views that form a single connected component graph based on the pre-computed pairwise covisibility of all images in the scene (this prevents disjoint sets of images as input). Figure 4 shows that MapAnything provides state-of-the-art dense multi-view reconstruction performance over other baselines using only image input, including VGGT [67]. Beyond the performance using only images as input, we show that MapAnything can leverage additional auxiliary geometric inputs for feed-forward inference to further increase reconstruction performance by a significant factor. Furthermore, we find that MapAnything is better than the bundle adjustment (BA) variant of the two-view baseline, Pow3R [23], which is also designed to leverage scene priors. We also find that reconstruction outputs from MapAnything (using only images as input) display high fidelity, as shown in Figure 3.

Two-View Dense Reconstruction: We benchmark sparse-view reconstruction and image-matching performance against state-of-the-art feed-forward baselines in Table 2. MapAnything achieves state-of-the-art performance using only images as input. With additional input modalities, MapAnything significantly outperforms both image-only baselines and Pow3R [23], the only other two-view feed-forward method that uses scene or camera priors.

Single-View Calibration: We benchmark the single-view calibration performance of MapAnything and other expert calibration baselines on randomly sampled frames from the test scenes of undistorted ETH3D [54], ScanNet++ v2 [81], and TartanAirV2 [73]. To test non-centered principal points, we randomly crop frames with aspect ratios from 3:1 to 1:2. Despite not being trained specifically on single images, Table 3 shows that MapAnything achieves state-of-the-art performance for perspective calibration. This demonstrates MapAnything's effectiveness in modeling generic central camera systems and its potential to generalize to wide-angle models like fisheye with appropriate training.

Monocular & Multi-View Depth Estimation: In Table 4, we benchmark MapAnything against specialized models for single-view and multi-view depth estimation across various inputs. In the RMVD benchmark, note that we don't use

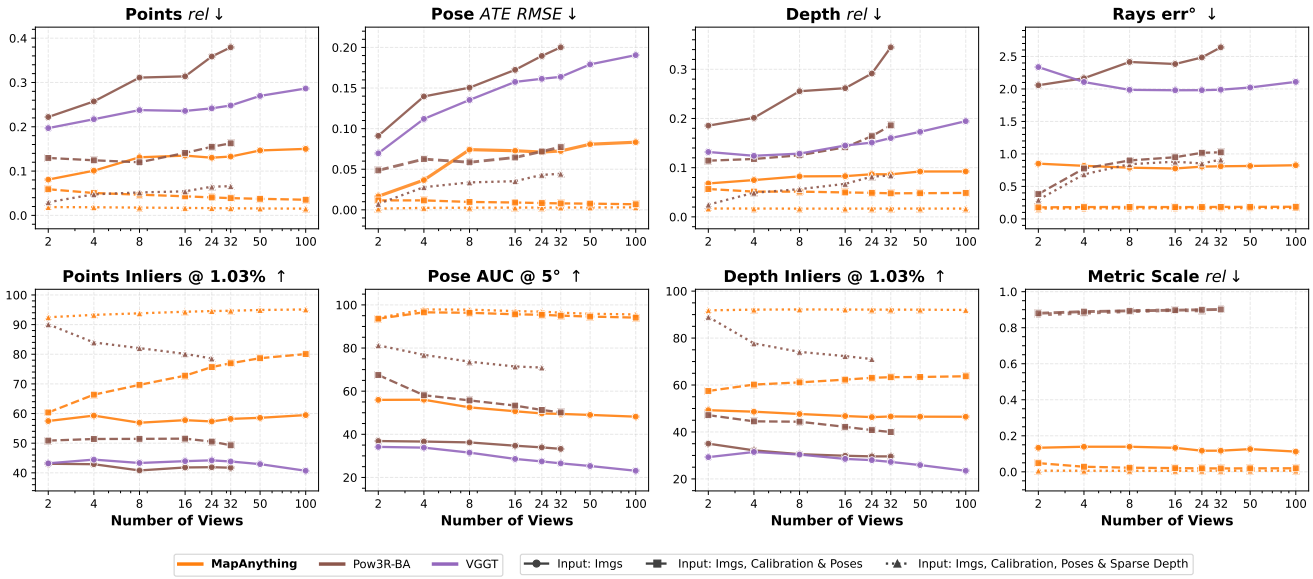

Figure 4. **MapAnything shows state-of-the-art dense multi-view reconstruction for input views ranging from 2 to 100 and under different input configurations.** We report the absolute relative error (rel), the inlier ratio at a relative threshold of 1.03% (τ), the average aligned trajectory error ($ATE\ RMSE$), the area under the curve at an error threshold of 5° ($AUC@5$), and the average angular error (err) in degrees (°), averaged over ETH3D, ScanNet++ v2 & TAv2. We do not report performance for baselines when the inference runs out of GPU memory. We provide results for individual datasets & the exhaustive input configurations of MapAnything in the supplement.

Table 2. **MapAnything showcases state-of-the-art two-view reconstruction under different input configurations.** We report the absolute relative error (rel), the inlier ratio at a relative threshold of 1.03% (τ), the average aligned trajectory error (ATE), the area under the curve at an error threshold of 5° (AUC), and the average angular error (err) in degrees (°). Best results are indicated in **bold**.

	Average across ETH3D, SN++v2 & TAV2							
	Scale	Points		Pose		Depth		Rays
Methods	rel ↓	rel ↓	τ ↑	ATE ↓	AUC ↑	rel ↓	τ ↑	err° ↓
a) Input: Images								
DUSt3R [72]	—	0.21	43.9	0.08	35.5	0.17	32.6	2.55
MASt3R [29]	0.38	0.25	30.2	0.07	37.3	0.19	24.8	7.03
Pow3R [23]	—	0.22	43.1	0.09	36.9	0.19	35.0	2.06
VGGT [67]	—	0.20	43.2	0.07	34.2	0.13	29.3	2.34
MapAnything	**0.13**	**0.08**	**57.5**	**0.02**	**56.0**	**0.07**	**49.3**	**0.85**
b) Input: Images & Intrinsics								
Pow3R [23]	—	0.20	46.0	0.08	51.3	0.15	43.2	0.40
MapAnything	**0.13**	**0.07**	**59.3**	**0.01**	**64.7**	**0.06**	**55.1**	**0.19**
c) Input: Images, Intrinsics & Poses								
Pow3R [23]	—	0.13	50.9	0.05	67.5	0.11	47.2	0.38
MapAnything	**0.05**	**0.06**	**60.4**	**0.01**	**93.6**	**0.06**	**57.5**	**0.18**
d) Input: Images, Intrinsics & Depth								
Pow3R [23]	—	0.13	**77.9**	0.04	66.5	0.07	**77.3**	0.29
MapAnything	**0.02**	**0.04**	77.8	**0.01**	**73.1**	**0.03**	76.6	**0.18**
e) Input: Images, Intrinsics, Poses & Depth								
Pow3R [23]	—	0.03	**90.1**	0.01	81.3	0.02	**89.0**	0.29
MapAnything	**0.01**	**0.02**	82.0	**0.00**	**94.8**	**0.02**	81.5	**0.16**

ETH3D due to the distortion issue mentioned in MVSA [22] and DTU & Tanks and Temples since they are not metric. Although not trained specifically for single-view metric depth, MapAnything achieves state-of-the-art or comparable

Table 3. **MapAnything shows state-of-the-art single-image calibration.** Note that MapAnything has not been trained specifically for single-image inputs. We report the average angular error (err) in degrees (°). Best results are indicated in **bold**.

Methods	Avg.	ETH3D	SN++v2	TAV2
VGGT [67]	4.00	2.83	5.21	3.95
MoGe-2 [71]	1.95	1.89	1.56	2.40
AnyCalib [58]	2.01	1.52	2.41	2.10
MapAnything	**1.06**	**1.33**	**0.39**	**1.47**

performance. For multi-view metric depth estimation using images only, MapAnything outperforms MASt3R-BA [29] & MUSt3R [6]. With auxiliary inputs like camera calibration and poses, MapAnything's performance improves and it delivers competitive results compared to task-specific specialized models. In comparison to baselines such as MoGe-2 [71] and MVSA [22], we find the metric scale estimation on ScanNet to be sub-optimal and believe this is likely due to lower benchmark dataset quality [22, 70]. As indicated in Table 4, we observe strong depth estimation performance on ScanNet when using median scale alignment.

Insights into enabling MapAnything: As shown in Table 5a, the factored representation of the scene as a multi-view set of rays, depth & pose (RDP) along with the metric scale is a key enabler for strong reconstruction performance while using images and optionally additional geometric inputs. In Table 5b, we find that our input probability-based training is efficient in training one universal model for various tasks and input configurations, where the performance

Table 4. **MapAnything shows versatile metric depth estimation under different input configurations on the Robust-MVD Benchmark [51]**. Note that MapAnything has not been trained for single-image inputs. We report the absolute relative error (rel) and the inlier ratio at a relative threshold of 1.03% (τ). The best result for each group is in **bold**; gray text indicates results where the evaluation dataset is in the training distribution [22].

Approach	K	Poses	KITTI rel ↓	KITTI τ ↑	ScanNet rel ↓	ScanNet τ ↑
a) Single-View Metric						
MoGe-2 [71]	✗	✗	14.21	6.8	**10.57**	**19.8**
MapAnything	✗	✗	**9.69**	**17.9**	27.77	2.9
Depth Pro [5]	✓	✗	13.60	14.3	9.20	19.7
UniDepthV2 [45]	✓	✗	13.70	4.8	3.20	61.3
Metric3DV2 [18]	✓	✗	8.70	13.2	6.20	19.3
MapAnything	✓	✗	**8.48**	**27.7**	**31.12**	**3.0**
b) Multi-View Metric						
MAST3R [29]	✗	✗	61.40	0.4	12.80	19.4
MUSt3R [6]	✗	✗	19.76	7.3	7.66	35.7
MapAnything	✗	✗	**5.45**	**45.7**	**22.23**	**10.6**
MapAnything	✓	✗	**8.45**	**27.5**	**24.94**	**8.2**
Fast-MVSNet [82]	✓	✓	12.10	37.4	287.10	9.4
Robust MVDB [51]	✓	✓	7.10	41.9	7.40	38.4
MAST3R Tri. [22]	✓	✓	3.40	66.6	4.50	63.0
MVSA [22]	✓	✓	**3.20**	**68.8**	**3.70**	**62.9**
MapAnything	✓	✓	4.63	51.6	5.58	48.1
c) Single-View w/ Alignment						
MoGe [70]	✗	✗	5.12	46.2	**3.59**	**65.3**
MoGe-2 [71]	✗	✗	**4.82**	**47.9**	3.77	63.1
VGGT [67]	✗	✗	7.50	33.0	3.33	70.8
π^3 [74]	✗	✗	6.00	40.1	2.90	73.9
MapAnything	✗	✗	6.12	42.2	4.95	55.6
Depth Pro [5]	✓	✗	6.10	39.6	4.30	58.4
DAV2 [79]	✓	✗	6.60	38.6	**4.00**	**58.6**
Metric3DV2 [18]	✓	✗	5.10	44.1	2.40	78.3
UniDepthV2 [45]	✓	✗	**4.00**	**55.3**	2.10	82.6
MapAnything	✓	✗	6.15	41.6	4.77	57.1
d) Multi-View w/ Alignment						
MAST3R [29]	✗	✗	3.30	67.7	4.30	64.0
MUSt3R [6]	✗	✗	4.47	56.7	3.22	69.2
VGGT [67]	✗	✗	4.60	53.0	2.34	80.6
π^3 [74]	✗	✗	**3.09**	**69.5**	1.98	83.6
MapAnything	✗	✗	4.04	60.3	**3.47**	**67.0**
DeMoN [61]	✓	✗	15.50	15.2	12.00	21.0
DeepV2D KITTI [57]	✓	✗	3.10	74.9	23.70	11.1
DeepV2D ScanNet [57]	✓	✗	10.00	36.2	4.40	54.8
MapAnything	✓	✗	**3.97**	**61.2**	**3.34**	**68.5**

of the universally trained model is equivalent to various bespoke models trained for specific input configurations.

5. Limitations

While MapAnything makes significant strides towards a universal multi-modal backbone for in-the-wild metric-scale 3D reconstruction, several limitations and future directions remain: (a) MapAnything does not explicitly account for noise or uncertainty in geometric inputs. (b) Although this is not currently supported, the architecture can be easily extended to handle tasks where images are not available for all input views. For example, in novel view synthesis, the target views for rendering will only have cameras available as input. (c) While the design of MapAnything supports iterative inference, it remains to be explored how effective scaling of test-time compute would be for 3D reconstruction (this ties into effectively handling noise in the inputs). (d) Multi-modal features are currently fused before being input; exploring more efficient ways to directly input different modalities to the transformer could be interesting.

Beyond multi-task capabilities, scalability is currently limited by the one-to-one mapping between input pixels and the output scene representation. We believe that significant work remains in effectively representing scenes in memory and decoding them as required, especially for large scenes. Our current scene parameterization does not capture dynamic motion or scene flow [26], which are promising areas.

Table 5. **Ablations providing insight into the key design choices.** We report the absolute relative error (rel) and the inlier ratio at a relative threshold of 1.03% (τ) at 50 views, averaged over ETH3D, ScanNet++ v2 & TAv2. Best results are **bold**. **Insights:** (a) The factored representation of rays, depth & pose (RDP) along with metric scale is key to achieving strong reconstruction performance under different input configurations. (b) MapAnything trained universally for 12+ tasks in one go with equivalent compute to two bespoke models is superior in terms of performance to three bespoke models trained for distinct input configurations. This indicates that the multi-task training of MapAnything is highly efficient.

(a) **Scene Representation**

Methods	Metric Scale rel ↓	Pointmaps rel ↓	Pointmaps τ ↑
Input: Images Only			
Local PM + Pose	**0.14**	0.32	33.2
RDP	0.17	0.33	32.6
LPMP & Scale	0.16	0.30	38.7
RDP & Scale (ours)	0.16	**0.28**	**40.7**
Input: Images, Intrinsics & Metric Poses			
Local PM + Pose	**0.04**	0.08	53.5
RDP	0.06	0.09	46.7
LPMP & Scale	0.06	**0.07**	55.9
RDP & Scale (ours)	0.05	**0.07**	**57.8**

(b) **Expert vs Universal Training**

Methods	Metric Scale rel ↓	Pointmaps rel ↓	Pointmaps τ ↑
Input: Images Only			
Expert Training	**0.16**	0.29	31.8
Universal Training	**0.16**	**0.28**	**40.7**
Input: Images, Intrinsics & Metric Poses			
Expert Training	**0.03**	**0.07**	56.2
Universal Training	0.05	**0.07**	**57.8**
Input: Images & Metric Depth			
Expert Training	**0.06**	**0.24**	53.0
Universal Training	**0.06**	0.25	**54.0**

6. Conclusion

MapAnything is the first universal transformer-based backbone that directly regresses metric 3D geometry and camera poses from flexible inputs – including images, camera intrinsics, poses, depth maps, or partial reconstructions – in a single pass. By using a factored representation of multi-view geometry (depth maps, ray maps, poses, and a global scale factor), MapAnything unifies local estimates into a global metric frame. MapAnything handles multiple tasks like uncalibrated structure-from-motion, calibrated multi-view stereo, monocular depth estimation, camera localization, depth completion, and more without task-specific tuning. Extensive experiments show that it surpasses or matches specialist models while enabling efficient joint training. Future extensions to dynamic scenes, uncertainty quantification, and scene understanding promise to further generalize MapAnything's capabilities and robustness, paving the way toward a truly universal 3D reconstruction backbone.

MapAnything: Universal Feed-Forward Metric 3D Reconstruction

Supplementary Material

A. Acknowledgments

We thank Michael Zollhöfer for his initial involvement in project discussions. We also thank Jeff Tan, Jianyuan Wang, Jay Karhade, Jinghao (Jensen) Zhou, Yifei Liu, Shubham Tulsiani, Khiem Vuong, Yuheng Qiu, Shibo Zhao, Omar Alama, Andrea Simonelli, Corinne Stucker, Denis Rozumny, Bardienus Duisterhof, and Wenshan Wang for their insightful discussions, feedback, and assistance with parts of the project. Lastly, we appreciate the support for compute infrastructure from Julio Gallegos, Tahaa Karim, and Ali Ganjei.

Funding at Carnegie Mellon University: Nikhil's time and parts of this work at CMU was supported by Defense Science and Technology Agency (DSTA) Contract `#DST000EC124000205`, DEVCOM Army Research Laboratory (ARL) SARA Degraded SLAM `CRA W911NF-20-S-0005`, and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number `140D0423C0074`. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government. The compute required for this work at CMU was supported by a hardware grant from Nvidia and used PSC Bridges-2 through allocation `cis220039p` from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program.

B. Implementation Details

We use the AdamW [28] optimizer with a peak learning rate of $5 \cdot 10^{-6}$ for the pre-trained DINOv2 encoder [42], and 10^{-4} for everything else. For the learning rate schedule, we employ a 10% linear warmup to the peak and subsequently use a half-cycle cosine decay to a $100\times$ lower value. For the optimizer, we also use a weight decay of 0.05, $\beta_1 = 0.9$, and $\beta_2 = 0.95$. For every batch, the input images and dense geometric quantities are resized and cropped so that the maximum dimension is 518 pixels and aspect ratio is randomized from 3:1 to 1:2. We use color jitter, Gaussian blur, and grayscale conversion as augmentations. We further employ mixed precision training and gradient checkpointing for DINOv2 encoder to improve training efficiency and GPU memory utilization. We also use gradient norm clipping with a threshold of 1 for additional training stability. Lastly, we use a dynamic batching scheme where the batch size is changed based on the number of views in a batch. We find that it is effective to train the model with a two-stage curriculum (420K steps): (1) 6 days on 64 H200-140GB GPUs with an effective batch size varying from 768 to 1536 with the number of views varying from 4 to 2, respectively, and (2) 4 days on 64 H200-140GB GPUs with a $10\times$ lower peak LR and an effective batch size that varies from 128 to 1536 with views varying from 24 to 2, respectively. The training setup for the ablations presented in Section 4 is the same as above where the only difference is the effective batch size ($8\times$ lower but leading to sufficient convergence).

C. Additional Evaluation

Speed & Memory Profiling: We present profiling results for MapAnything and other concurrent models in Figure S.1.

Flexibility of Input Configurations: We present a representative set of input configurations for MapAnything in Table S.1, where the performance of MapAnything improves as more modalities are provided. The universal training of MapAnything with varying selection probabilities for geometric inputs as 6 factors (as described in Section 3.3) enables support for 64 exhaustive input combinations. While we primarily benchmark cases where the input modality is available for all views, MapAnything can also support optional geometric inputs for a subset of the input views.

Comparison of MapAnything Variants: We provide a comparison between different variants of MapAnything in Figure S.2, where the performance of VGGT [67] is provided as a baseline. First, we show that our two-stage training with covisibility-based view sampling is very effective, where the MapAnything model trained for up to 4 views as input already shows strong generalization to a significantly higher number of views. We also compare the performance of our different open-source models, where, as described in Table 1, the Apache model is trained on 6 datasets, while the CC-BY-NC one is trained on 13. While we observe a decrease in performance, the Apache variant is still competitive with the VGGT baseline and its performance further improves as additional geometric inputs are provided. We also provide a comparison between the various versions of the publicly released models in Figure S.3.

Design Choices: As shown in Table S.2, log scaling and alternating attention are key for strong performance when evaluating with 50 views, well beyond the 4 used in training.

Qualitative Examples: Figure S.4 and Figure S.5 showcase the versatility of MapAnything.

Comparison with Concurrent Models: Figure S.6 showcases the average results, while Figure S.7 and Figure S.8 show the performance for each individual dataset.

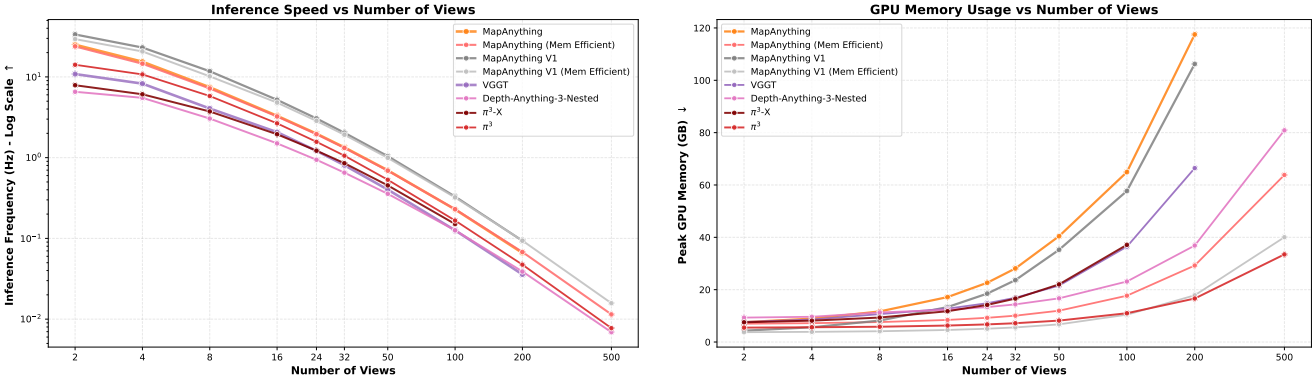

Figure S.1. **MapAnything shows the best speed and memory usage profile in comparison to latest concurrent state-of-the-art multi-view feed-forward reconstruction models.** We profile all the models on a H200-140GB GPU using appropriate floating point precision autocast. Upon profiling the naive inference call of MapAnything and comparing to concurrent methods, we notice a memory bottleneck in the dense per-pixel regression across views, since our implementation of the dense head is closer to the original DPT [46]. To alleviate this issue, i.e., in MapAnything (Mem Efficient), we run the per-view decodings in a mini-batched loop, where in the above profiling the mini-batch size is 1. We find that this looping leads to negligible trade-off in speed while significantly reducing the memory usage.

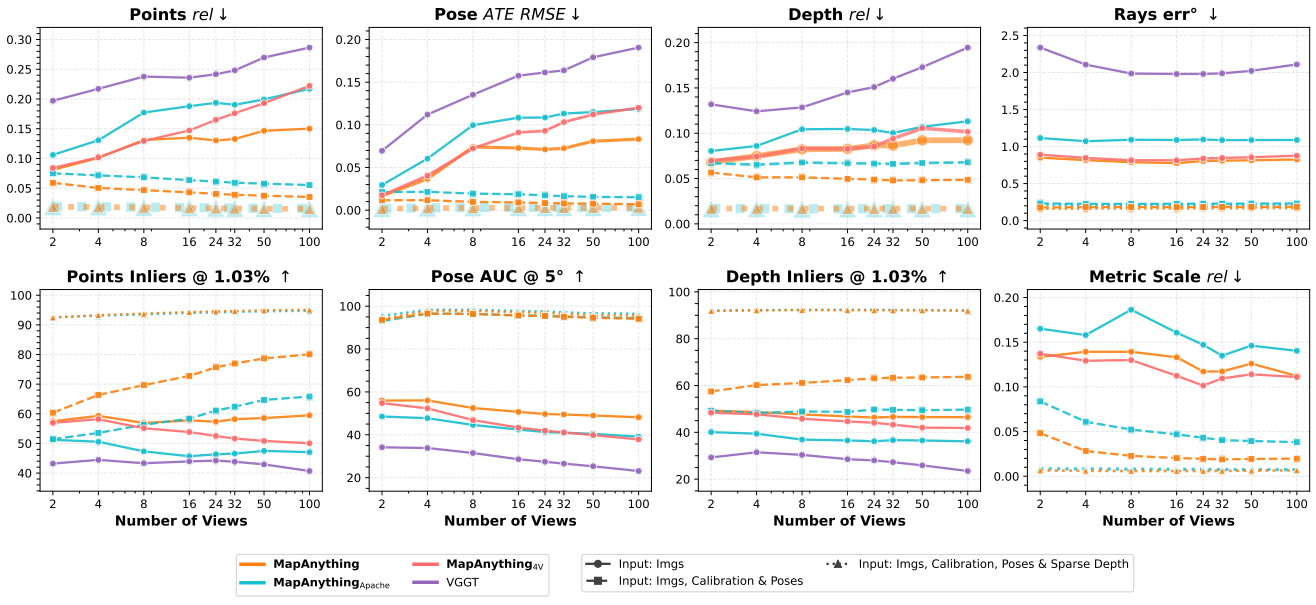

Figure S.2. **The Apache and first stage (up to 4 views) training variants of MapAnything show strong dense multi-view reconstruction for input views ranging from 2 to 100 and under different input configurations.** We report the absolute relative error (rel), the inlier ratio at a relative threshold of 1.03% (τ), the average aligned trajectory error ($ATE\ RMSE$), the area under the curve at an error threshold of 5° ($AUC@5$), and the average angular error (err) in degrees (°), averaged over ETH3D, ScanNet++ v2 & TAv2.

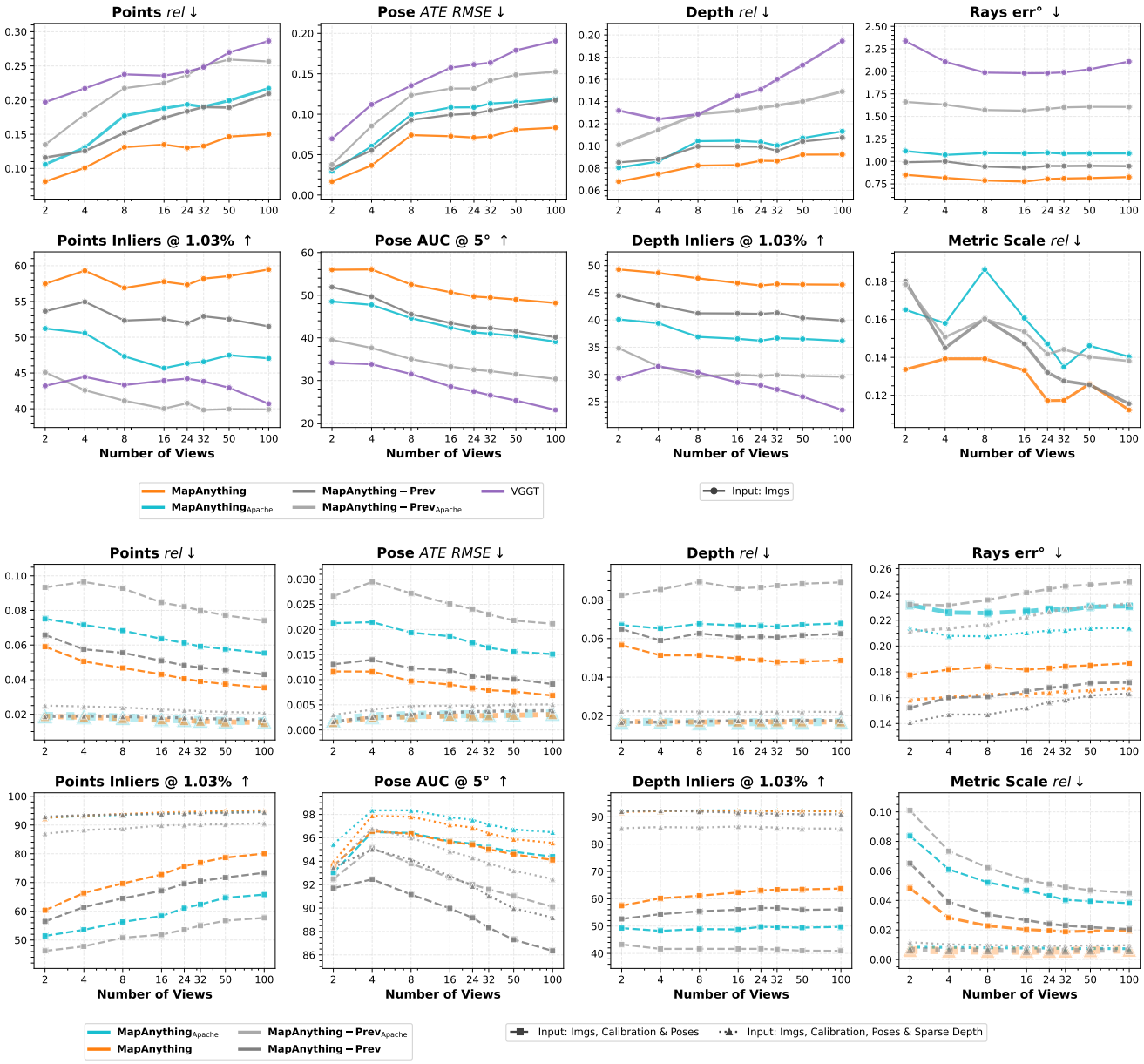

Figure S.3. **Dense multi-view reconstruction benchmarking comparing the performance of the latest checkpoint as of January 20th 2026 (MapAnything) to the one released in September 2025 (MapAnything-Prev).** The primary difference between the two variants (*latest* & *previous*) is: (a) Encoder: First 24 layers of 1536-dim DINOv2 ViT-G (*latest*) vs All 24 layers of 1024-dim DINOv2 ViT-L (*previous*), (b) Multi-View Transformer: Last 16 layers of 1536-dim DINOv2 ViT-G (*latest*) vs 24 layers of a randomly initialized 736-dim ViT-B size transformer. Both across large-scale trainings and ablation settings, we find that the DINOv2 initialization for the multi-view transformer helps significantly with convergence speed and final performance. In the above plots, we report the absolute relative error (rel), the inlier ratio at a relative threshold of 1.03% (τ), the average aligned trajectory error ($ATE\ RMSE$), the area under the curve at an error threshold of 5° ($AUC@5$), and the average angular error (err) in degrees (°), averaged over ETH3D, ScanNet++ v2 & TAv2.

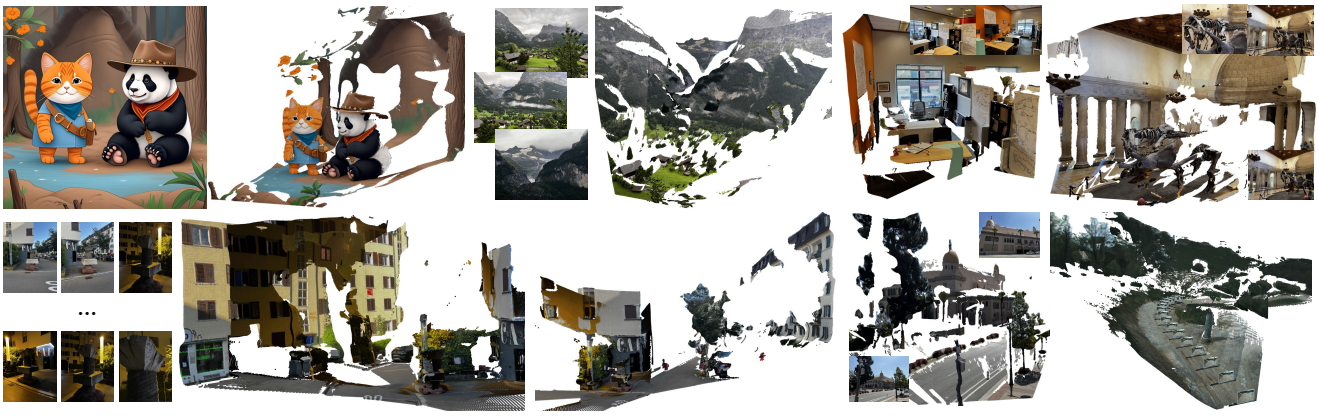

Figure S.4. **MapAnything provides high-fidelity dense geometric reconstructions** across varying domains and number of views. While MapAnything can support a flexible set of inputs, here we are showcasing its capabilities only using images as input. In particular, we show results across a varying number of captured views and from different domains such as indoor, landscape, art, object-centric, and off-road. It also works well on monocular and art images despite not being trained for it.

Table S.1. **MapAnything demonstrates remarkable flexibility in handling diverse input configurations, with performance improving as additional modalities are provided.** While our universal training supports 64 (i.e., 2^6) possible input combinations, we highlight 12 representative combinations in this table. We report the absolute relative error (rel) and the inlier ratio at a relative threshold of 1.03% (τ) for 50 views, averaged over ETH3D, ScanNet++ v2 & TAv2. 'K' denotes camera intrinsics and 'sparse' depth indicates that 90% of the valid depth is randomly masked out.

MapAnything Inputs							Avg. Performance		
			Depth		Metric Scale		Scale	Points	
Imgs	K	Poses	Dense	Sparse	Pose	Depth	rel ↓	rel ↓	τ ↑
a) Images Only									
✓	✗	✗	✗	✗	✗	✗	0.13	0.15	58.6
b) Images & Intrinsics									
✓	✓	✗	✗	✗	✗	✗	0.13	0.14	61.5
c) Images & Poses									
✓	✗	✓	✗	✗	✗	✗	0.11	0.04	76.8
✓	✗	✓	✗	✗	✓	✗	0.02	0.04	75.3
d) Images, Intrinsics & Poses									
✓	✓	✓	✗	✗	✗	✗	0.11	0.03	80.5
✓	✓	✓	✗	✗	✓	✗	0.02	0.04	78.7
e) Images & Depth									
✓	✓	✗	✗	✓	✗	✓	0.04	0.11	77.2
✓	✓	✗	✓	✗	✗	✓	0.05	0.11	67.8
f) Images, Intrinsics, Poses & Depth									
✓	✓	✓	✗	✓	✗	✗	0.12	0.02	92.0
✓	✓	✓	✓	✗	✗	✗	0.12	0.02	83.0
✓	✓	✓	✗	✓	✓	✓	0.01	0.02	94.9
✓	✓	✓	✓	✗	✓	✓	0.01	0.02	84.4

Table S.2. **Ablations showing loss and multi-view transformer attention design choices critical for strong reconstruction performance.** We report the absolute relative error (rel) and the inlier ratio at a relative threshold of 1.03% (τ) at 50 views, averaged over ETH3D, ScanNet++ v2 & TAv2. Best results are **bold**.

(a) **Loss Scheme**

	ETH3D, SN++v2 & TAV2		
	Metric Scale	Pointmaps	
Methods	rel ↓	rel ↓	τ ↑
Input: Images Only			
Overall Factored Loss	**0.16**	**0.29**	**31.8**
No Log Loss	0.17	0.39	27.3

(b) **Attention Scheme**

	ETH3D, SN++v2 & TAV2		
	Metric Scale	Pointmaps	
Methods	rel ↓	rel ↓	τ ↑
Input: Images Only			
Alternating [67]	**0.16**	**0.29**	**31.8**
Global w/ View PE [78]	0.20	0.53	19.7

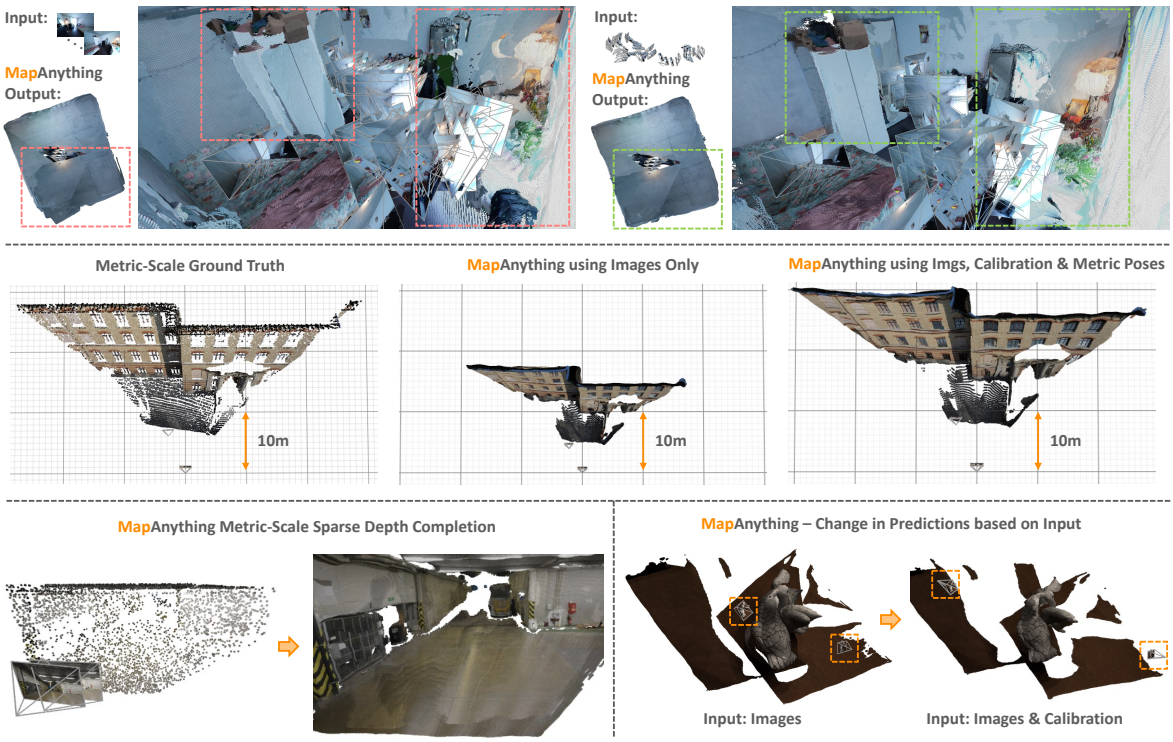

Figure S.5. **Auxiliary geometric inputs improve feed-forward performance of MapAnything.** `(Top)` While MapAnything & other baselines using 100 input images show duplication of 3D structure, when provided with the camera calibration and poses, the 3D reconstruction significantly improves, showing aligned geometry. `(Middle)` MapAnything using images only as input shows non-precise metric scale estimation on ETH3D (a zero-shot dataset). However, when the calibration and metric poses are provided as additional input, the estimated metric scale significantly improves and approximately matches the ground truth. `(Bottom-Left)` We show that MapAnything can leverage a sparse metric point cloud as input to perform dense metric depth completion. `(Bottom-Right)` Despite not being trained for object-centric data, we show how the scene geometry and cameras change based on the input provided.

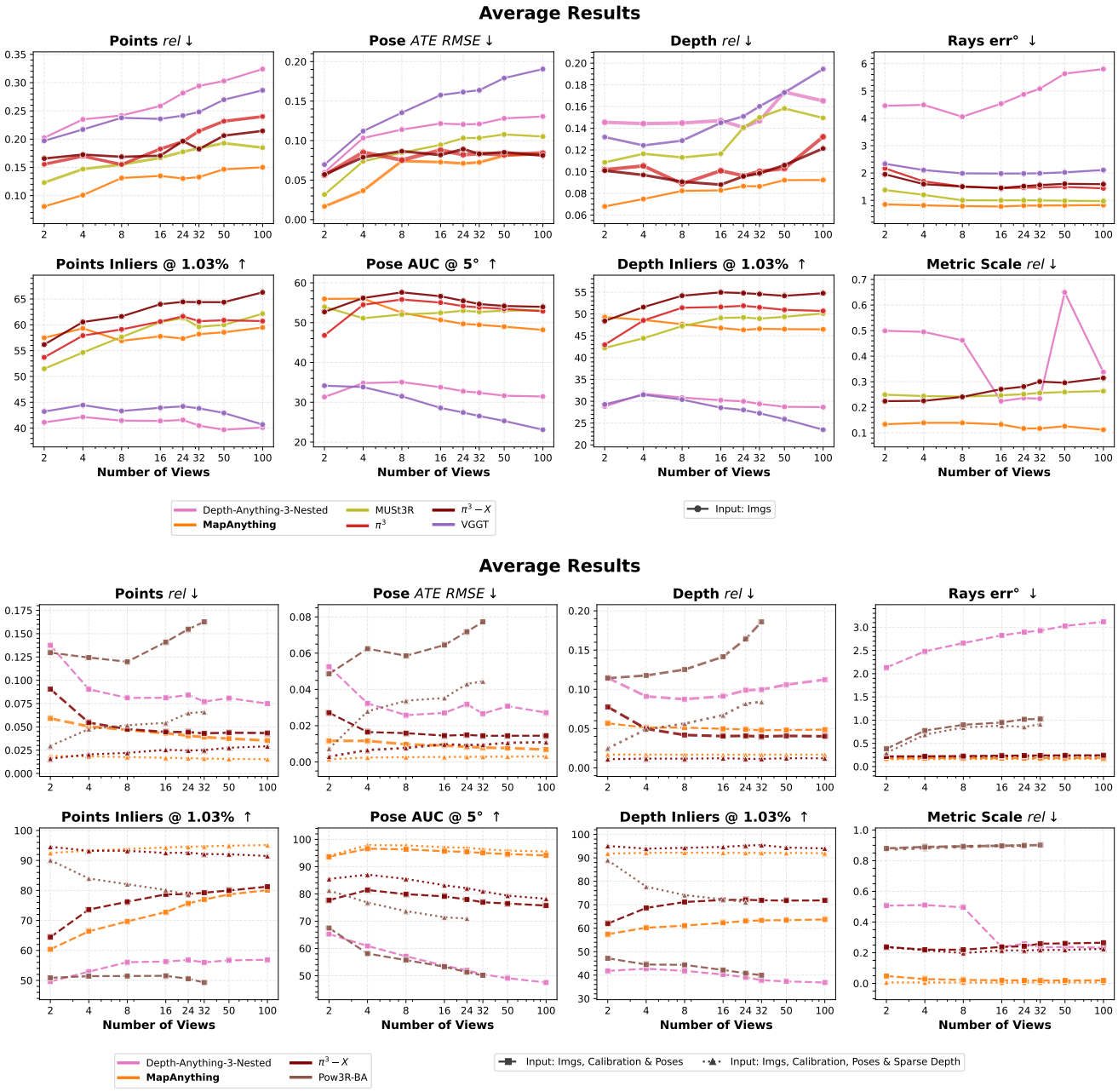

Figure S.6. **MapAnything showcases state-of-the-art and competitive dense multi-view reconstruction performance in comparison to latest public concurrent state-of-the-art models as of January 20th 2026, i.e., DA3-Nested & π^3-X, despite being trained on significantly less amount of data.** We report the absolute relative error (rel), the inlier ratio at a relative threshold of 1.03% (τ), the average aligned trajectory error ($ATE\ RMSE$), the area under the curve at an error threshold of 5° ($AUC@5$), and the average angular error (err) in degrees (°), averaged over ETH3D, ScanNet++ v2 & TAv2.

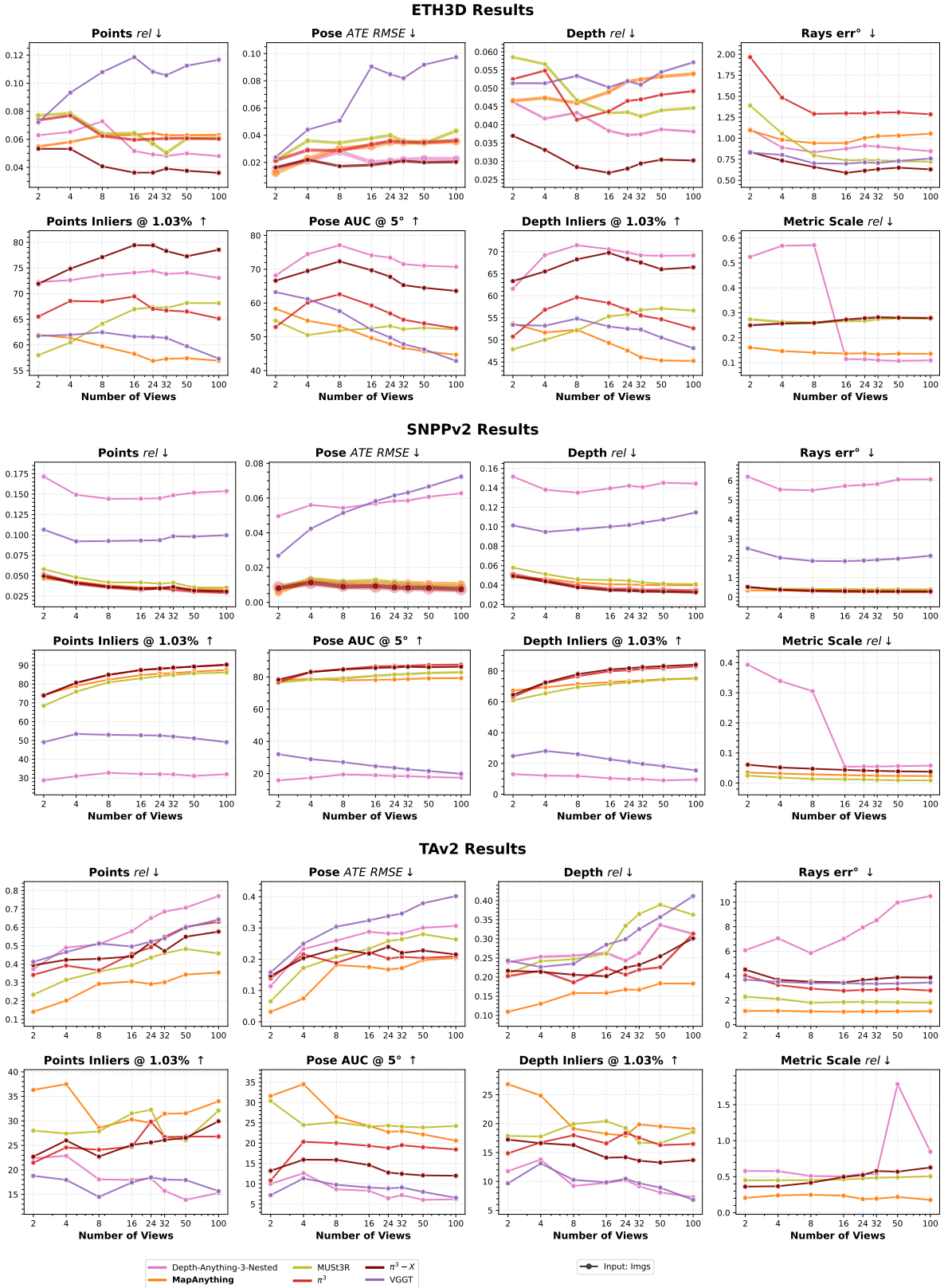

Figure S.7. **Dense multi-view reconstruction benchmarking across individual datasets using only images as input.** We also include results for latest public concurrent state-of-the-art models as of January 20th 2026. Please see Figure S.6 for details & the averaged version.

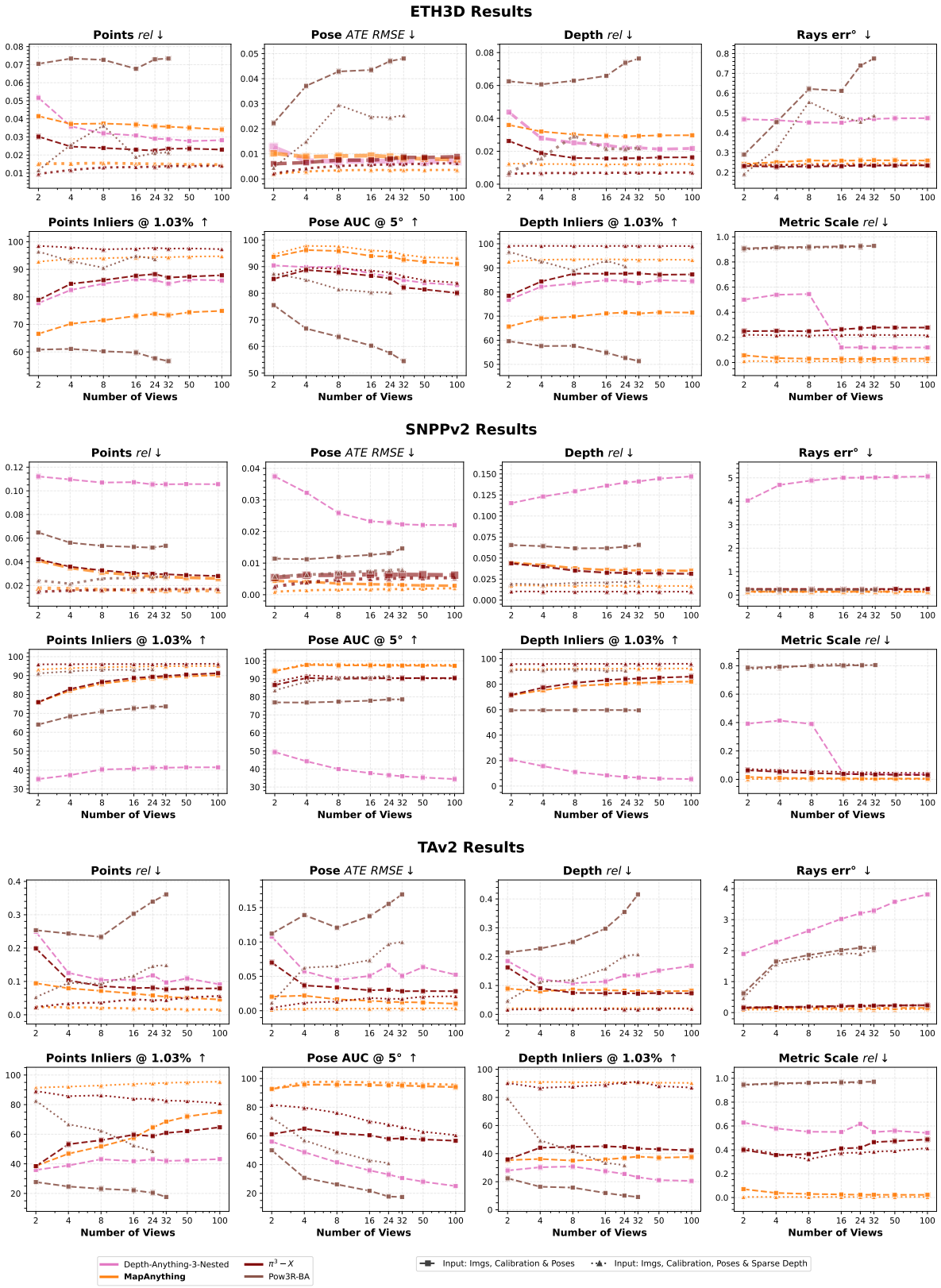

Figure S.8. **Dense multi-view reconstruction benchmarking across individual datasets using multi-modal inputs.** We also include results for latest public concurrent state-of-the-art models as of January 20th 2026. Please see Figure S.6 for details & the averaged version.

References

[1] Omar Alama, Avigyan Bhattacharya, Haoyang He, Seungchan Kim, Yuheng Qiu, Wenshan Wang, Cherie Ho, Nikhil Keetha, and Sebastian Scherer. RayFronts: Open-set semantic ray frontiers for online scene understanding and exploration. In *IROS*, 2025. 2

[2] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. SceneScript: Reconstructing scenes with an autoregressive structured language model. In *ECCV*, 2024. 5

[3] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022. 3

[4] Jonathan T. Barron. A general and adaptive robust loss function. In *CVPR*, 2019. 5

[5] 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. In *ICLR*, 2025. 8

[6] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. MUSt3R: Multi-view network for stereo 3D reconstruction. In *CVPR*, 2025. 2, 7, 8

[7] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *CVPR*, 2024. 4

[8] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In *CVPR*, 2025. 1, 2

[9] Bardienus P. Duisterhof, Jan Oberst, Bowen Wen, Stan Birchfield, Deva Ramanan, and Jeffrey Ichnowski. RaySt3R: Predicting novel depth maps for zero-shot object completion. In *NeurIPS*, 2025. 3

[10] 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 *3DV*, 2025. 2

[11] Sven Elflein, Qunjie Zhou, Sérgio Agostinho, and Laura Leal-Taixé. Light3R-SfM: Towards feed-forward structure-from-motion. In *CVPR*, 2025. 2

[12] Michael D. Grossberg and Shree K. Nayar. A general imaging model and a method for finding its parameters. In *ICCV*, 2001. 4

[13] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *Int. J. Comput. Vis.*, 103(3):267–305, 2013. 1

[14] Greg Heinrich, Mike Ranzinger, Hongxu, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIOv2.5: Improved baselines for agglomerative vision foundation models. In *CVPR*, 2025. 4

[15] Cherie Ho, Jiaye Zou, Omar Alama, Sai M Kumar, Benjamin Chiang, Taneesh Gupta, Chen Wang, Nikhil Keetha, Katia Sycara, and Sebastian Scherer. Map it anywhere: Empowering bev map prediction using large-scale public datasets. In *NeurIPS*, 2024. 2

[16] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, 2005. 1

[17] Berthold KP Horn. Obtaining shape from shading information. In *Shape from shading*, pages 123–171. MIT Press, 1989. 1

[18] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3D v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46 (12):10579–10596, 2024. 8

[19] Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, Shibo Zhao, Shayegan Omidshafiei, Dong-Ki Kim, Ali akbar Agha-mohammadi, Katia Sycara, Matthew Johnson-Roberson, Dhruv Batra, Xiaolong Wang, Sebastian Scherer, Chen Wang, Zsolt Kira, Fei Xia, and Yonatan Bisk. Toward general-purpose robots via foundation models: A survey and meta-analysis. arXiv:2312.08782, 2023. 2

[20] Yuan-Ting Hu, Jiahong Wang, Raymond A. Yeh, and Alexander G. Schwing. SAIL-VOS 3D: A synthetic dataset and baselines for object detection and 3D mesh reconstruction from video data. In *CVPR*, 2021. 5

[21] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *CVPR*, 2018. 5

[22] Sergio Izquierdo, Mohamed Sayed, Michael Firman, Guillermo Garcia-Hernando, Daniyar Turmukhambetov, Javier Civera, Oisin Mac Aodha, Gabriel Brostow, and Jamie Watson. MVSAnywhere: Zero-shot multi-view stereo. In *CVPR*, 2025. 1, 7, 8

[23] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3R: Empowering unconstrained 3D reconstruction with camera and scene priors. In *CVPR*, 2025. 3, 6, 7

[24] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *ICLR*, 2025. 3

[25] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. DynamicStereo: Consistent dynamic depth from stereo videos. In *CVPR*, 2023. 5

[26] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4D: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025. 8

[27] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM. In *CVPR*, 2024. 2

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 9

[29] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *ECCV*, 2024. 1, 2, 7, 8

[30] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 5

[31] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 4

[32] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *CVPR*, 2024. 5

[33] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004. 1

[34] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3R: Aligned monocular depth estimation for dynamic videos. In *CVPR*, 2025. 3

[35] Yuanxun Lu, Jingyang Zhang, Tian Fang, Jean-Daniel Nahmias, Yanghai Tsin, Long Quan, Xun Cao, Yao Yao, and Shiwei Li. Matrix3D: Large photogrammetry model all-in-one. In *CVPR*, 2025. 3

[36] Manuel López Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *ECCV*, 2020. 5

[37] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*, 2023. 5

[38] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2I-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI*, 2024. 3, 4

[39] Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. In *CVPR*, 2025. 2

[40] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26 (06):756–777, 2004. 1

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 4

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 4, 9

[43] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *ECCV*, 2024. 1

[44] Zador Pataki, Paul-Edouard Sarlin, Johannes L. Schönberger, and Marc Pollefeys. MP-SfM: Monocular surface priors for robust structure-from-motion. In *CVPR*, 2025. 2

[45] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2026. 8

[46] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 4, 10

[47] 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. Pattern Anal. Mach. Intell.*, 44(3):1623–1637, 2022. 5

[48] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. AM-RADIO: Agglomerative vision foundation model – reduce all domains into one. In *CVPR*, 2024. 4

[49] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1

[50] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *ICCV*, 2011. 1

[51] Philipp Schröppel, Jan Bechtold, Artemij Amiranashvili, and Thomas Brox. A benchmark and a baseline for robust multi-view depth estimation. In *3DV*, 2022. 8

[52] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[53] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1

[54] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 5, 6

[55] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568(C), 2024. 4

[56] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. MV-DUSt3R+: Single-stage scene reconstruction from sparse views in 2 seconds. In *CVPR*, 2025. 2

[57] Zachary Teed and Jia Deng. DeepV2D: Video to depth with differentiable structure from motion. In *ICLR*, 2020. 2, 8

[58] Javier Tirado-Garín and Javier Civera. AnyCalib: On-manifold learning for model-agnostic single-view camera calibration. In *ICCV*, 2025. 7

[59] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. SMD-nets: Stereo mixture density networks. In *CVPR*, pages 8942–8952, 2021. 5

[60] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew Fitzgibbon. Bundle adjustment – a modern synthesis. In *ICCV*, pages 298–372, 2000. 1

[61] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. DeMoN: Depth and motion network for learning monocular stereo. In *CVPR*, 2017. 2, 8

[62] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *ECCV*, 2024. 5

[63] Igor Vasiljevic, Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Wolfram Burgard, Greg Shakhnarovich, and Adrien Gaidon. Neural ray surfaces for self-supervised learning of depth and ego-motion. In *3DV*, 2020. 4

[64] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Single-image calibration with geometric optimization. In *ECCV*, 2024. 1

[65] Hengyi Wang and Lourdes Agapito. 3D reconstruction with spatial memory. In *3DV*, 2025. 2

[66] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Visual geometry grounded deep structure from motion. In *CVPR*, 2024. 2

[67] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025. 2, 4, 6, 7, 8, 9, 12

[68] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. PF-LRM: Pose-free large reconstruction model for joint pose and shape prediction. In *ICLR*, 2024. 1, 2

[69] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *CVPR*, 2025. 2

[70] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, 2025. 5, 7, 8

[71] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. In *NeurIPS*, 2025. 7, 8

[72] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024. 1, 2, 4, 5, 7

[73] 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 *IROS*, 2020. 5, 6

[74] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning. arXiv:2507.13347, 2025. 2, 3, 8

[75] Ethan Weber, Norman Müller, Yash Kant, Vasu Agrawal, Michael Zollhöfer, Angjoo Kanazawa, and Christian Richardt. Fillerbuster: Multi-view scene completion for casual captures. In *3DV*, 2026. 3

[76] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jerome Revaud. CroCo v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *ICCV*, 2023. 4

[77] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980. 1

[78] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In *CVPR*, 2025. 2, 12

[79] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In *NeurIPS*, 2024. 5, 8

[80] 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 *CVPR*, 2020. 5

[81] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, 2023. 5, 6

[82] Zehao Yu and Shenghua Gao. Fast-MVSNet: Sparse-to-dense multi-view stereo with learned propagation and Gauss-Newton refinement. In *CVPR*, 2020. 8

[83] Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722, 2018. 3

[84] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *ICLR*, 2024. 1, 2, 4

[85] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 3

[86] Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, Sebastian Scherer, and Wenshan Wang. UFM: A simple path towards unified dense correspondence with flow. In *NeurIPS*, 2025. 5, 6

[87] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. DeepTAM: Deep tracking and mapping with convolutional neural networks. *Int. J. Comput. Vis.*, 128:756–769, 2020. 2

[88] Jinghao (Jensen) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *ICCV*, 2025. 3