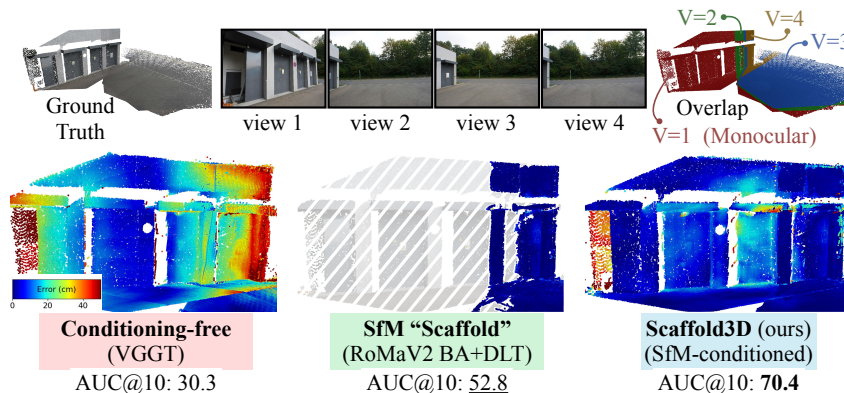


Scaffold3D: SfM-Conditioned Pointmap Prediction for Multi-View 3D Reconstruction

Frano Rajić¹, Yutong Chen¹, Haofei Xu¹, Zador Pataki¹, Marc Pollefeys^{1,2}, Siyu Tang¹
¹ETH Zurich ²Microsoft

Abstract

Multi-view 3D reconstruction is increasingly driven by feed-forward models, which are fast and robust but often exhibit geometric inconsistencies and locally inaccurate predictions because they lack explicit multi-view constraints to enforce consistency across views. Structure-from-motion (SfM) offers a complementary geometric signal through camera poses and 3D points estimated by correspondence filtering and optimization. To propagate this globally consistent *SfM scaffold* into dense geometry, we introduce **Scaffold3D**, an SfM-conditioned reconstruction framework that injects image patch-aligned SfM point tokens into a pairwise feed-forward pointmap predictor. Our approach preserves image-token reasoning while using explicit SfM structure to condition dense prediction. Because Scaffold3D predicts pairwise pointmaps in local frames, we fuse them with global alignment and an SfM anchoring term. Across established benchmarks (ScanNet++, ETH3D, and Tanks&Temples) and out-of-domain data (4D-DRESS and MV-dVRK), Scaffold3D achieves stronger overall performance than feed-forward reconstruction models, point-token propagation, depth-completion baselines, and recent geometry-conditioned models. The gains are especially clear in low- and no-overlap evaluations. Notably, our two-view SfM-conditioned pointmap predictor outperforms several off-the-shelf multi-view geometry-conditioned baselines. Together, these results establish SfM-scaffold conditioning as a practical interface between learned pointmap prediction and classical geometric optimization. Project page: <https://ethz-vlg.github.io/scaffold3d>.



1 Introduction

Multi-view 3D reconstruction is a core component in robotics, AR/VR, digital content creation, and embodied perception. Modern feed-forward models have made dense reconstruction much faster and more robust, and now predict depth, pointmaps, and camera geometry directly from images [16, 18, 21, 35, 37, 39]. However, these models typically do not enforce explicit geometric constraints during prediction to improve multi-view consistency and precision. As a result, despite strong robustness and coverage, their dense outputs often remain geometrically misaligned across views and locally inaccurate, especially on fine structures and out-of-domain scenes.

Structure-from-motion (SfM) pipelines remain strong at enforcing geometric consistency through correspondence filtering and optimization [7, 22, 25, 28, 34]. They estimate camera poses and 3D points, either before dense prediction or as part of a reconstruction pipeline that also uses learned priors. The resulting *SfM scaffold* (i.e., poses, intrinsics, and 3D points) covers only a subset of image regions, but it provides explicit multi-view constraints. We therefore frame dense reconstruction as a propagation problem: given a reliable SfM scaffold, how should a learned model propagate it into dense pointmaps without losing its global consistency?

Feed-forward reconstruction models can use geometric optimization only after dense prediction, for example through bundle adjustment or global alignment [18, 35, 37]. While simple and general, this ignores available SfM cameras and points during dense prediction. Another option is monocular depth completion: complete each view from SfM depth samples and then fuse the views [14, 15, 31, 49]. This uses the scaffold points, but it treats dense prediction mostly as a per-image problem. Multi-view reconstruction should instead exploit cross-view consistency, so that observations from one view constrain another through camera geometry and correspondence.

Recent geometry-conditioned models move in this direction, but use the scaffold through different interfaces. Pow3R [13], Pi3X [39], and MapAnything [16] feed SfM-derived geometry as scaffold depth maps, together with intrinsics or pose conditioning, rather than explicit 3D point observations. GGPT [5] takes the opposite route: it preserves the scaffold as point tokens, but propagates them without image-token decoding, completely discarding appearance and matching cues that are useful for dense pointmap prediction. A useful scaffold interface should therefore integrate SfM geometry into dense image-token pointmap prediction without losing either geometric support or appearance-driven correspondence cues.

Motivated by this insight, we introduce **Scaffold3D**, an SfM-scaffold-conditioned pointmap prediction framework for multi-view reconstruction. Scaffold3D contains a feed-forward image encoder-decoder and augments the decoder with image patch-aligned SfM point tokens. For each view, SfM 3D points whose projections fall into the same image patch are encoded and aggregated into a single point token. Point tokens across patches interact through self-attention and are then injected into the image-token decoder. This allows partial but globally consistent SfM geometry to guide dense pairwise pointmap prediction beyond directly triangulated regions while preserving dense visual correspondence cues.

Since pairwise pointmaps are predicted in local coordinate frames, we further introduce an SfM-anchored global alignment formulation to fuse them into a globally consistent reconstruction. As a result, the SfM scaffold serves two complementary roles: locally, it guides the pairwise pointmap predictor toward geometrically informed dense reconstruction, and globally, it acts as a reliable anchor for fusing pairwise predictions into a coherent 3D scene.

Experiments on standard datasets [17, 27, 43] and out-of-domain data [4, 38] support three conclusions. First, conditioning on estimated SfM scaffolds improves over unconditioned feed-forward baselines [18, 35, 37, 39] and feed-forward baselines that use geometric optimization only after dense prediction [7]. Second, combining image and point tokens improves over point-only propagation [5], especially in low- and no-overlap evaluations. Third, a *two-view* Scaffold3D predictor outperforms existing *multi-view* geometry-conditioned baselines [5, 16, 39]. Controlled Pow3R-style ablations further show that several scaffold-conditioning interfaces perform comparably when trained with the same SfM-conditioned setup, positioning Scaffold3D as one effective instantiation of SfM-conditioned training and alignment. We release training and evaluation code, pre-trained checkpoints, and dataset preprocessing scripts to facilitate future research and reproducibility.

2 Related Work

SfM-Based 3D Reconstruction. COLMAP [28] remains a standard 3D reconstruction approach for views with sufficient overlap. Modern SfM frameworks improve robustness to low overlap, large baselines, or low texture by incorporating learning-based matchers, learned initializations, or depth priors [7, 25, 34]. GGPT [5] similarly builds a learned SfM scaffold from feed-forward pose and point initialization, dense RoMa matches [8, 9], global bundle adjustment (BA), and direct linear triangulation (DLT). Despite being geometrically constrained, these SfM methods cannot triangulate points in textureless or monocular regions, resulting in incomplete reconstructions. Scaffold3D uses these SfM points as partial geometry guidance to predict dense geometry even in sparse-view setups.

Feed-Forward 3D Reconstruction. Recent multi-view transformers [10, 16, 23, 33, 35–37, 39, 44–46] directly predict cameras and depths from RGB images, making reconstruction much faster than SfM. Large-scale training gives strong sparse-view robustness, but without explicit geometric constraints these predictions often exhibit multi-view inconsistency and global drift, especially out of distribution [5]. Given known camera poses, recent multi-view stereo (MVS) methods [1, 12, 32] combine cost volumes with monocular depth priors, but typically require small-baseline inputs and predefined depth ranges. Our method requires only raw RGB inputs and achieves substantially higher reconstruction accuracy than previous feed-forward reconstruction benchmarks.

Monocular Depth Completion (DC). Monocular DC recovers dense geometry from sparse inputs, typically through *injection-based* methods [11, 20, 40] that integrate sparse signals during training or *optimization-based* methods [14, 15, 31] that adapt foundation models at test time. Hybrid approaches [48, 49] predict depth gradients before final optimization. However, monocular DC cannot exploit cross-view constraints for global consistency, and existing benchmarks primarily use uniform or LiDAR-derived points rather than irregular, noisy SfM points from sparse-view settings.

Multi-View Geometry Conditioning. Recent multi-view transformers introduced geometric conditioning interfaces. Pow3R [13] and Pi3X [39] encode depth or intrinsics as patch tokens, while MapAnything [16] fuses ray-geometry features with image embeddings. GGPT [5] propagates partial 3D structures via point tokens alone. Our work keeps image-token reasoning for pairwise pointmap prediction while injecting explicit SfM-derived point tokens into the decoder.

3 Method

Given N RGB input views $\{I_n \in \mathbb{R}^{H \times W \times 3}\}_{n=1}^N$, our goal is to predict dense 3D point maps $\{X_n \in \mathbb{R}^{H \times W \times 3}\}_{n=1}^N$ in a unified global coordinate system. As illustrated in Fig. 2, *Scaffold3D* operates in three stages. First, we leverage an SfM pipeline to estimate a geometric scaffold $\{S_n \in \mathbb{R}^{H \times W \times 3}\}_{n=1}^N$, which provides high-precision, 3D-consistent points but remains incomplete in textureless or non-overlapping regions. Second, an SfM-conditioned pointmap predictor predicts complete two-view pointmaps by conditioning on the RGB images and their respective SfM scaffolds. Finally, we introduce an SfM-anchored global alignment to aggregate these pairwise predictions into a globally consistent and dense 3D reconstruction.

3.1 SfM Scaffold

We adopt the SfM pipeline from GGPT [5] to generate our geometric scaffold. It establishes dense correspondences using RoMaV2 [9] and initializes the reconstruction with cameras and points predicted by a feed-forward reconstruction model, such as VGGT [35]. Using these correspondences and the feed-forward initialization, the pipeline performs a global bundle adjustment (BA) to estimate camera-to-world poses $\{T_n\}_{n=1}^N$ and intrinsics $\{K_n\}_{n=1}^N$. Finally, valid matches are triangulated and filtered for consistency to yield a set of 3D points.

Using the resulting 3D-2D correspondences, we derive a multi-view SfM pointmap $\{S_n \in \mathbb{R}^{H \times W \times 3}\}_{n=1}^N$. Note that S_n contains empty entries in textureless or monocular regions where reliable matches are unavailable. We define the set of pixels with valid SfM support as $\Omega_S = \{(n, p) \mid \text{pixel } p \text{ in view } n \text{ has a valid SfM estimate}\}$. These geometrically consistent yet incomplete points serve as a prior, which we propagate to regions lacking geometry support to achieve an accurate and complete reconstruction.

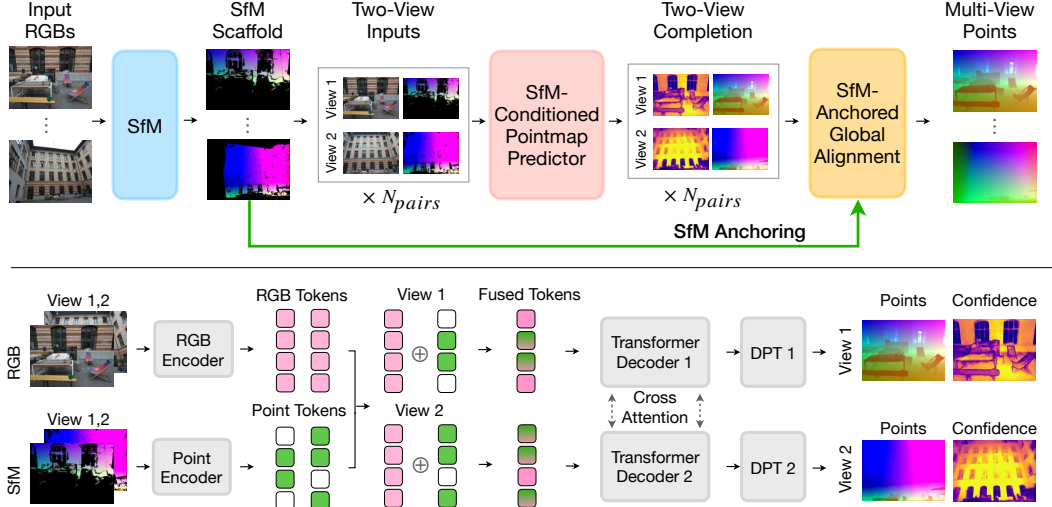


Figure 2: **Scaffold3D overview.** *Top*: Pipeline. First, we run an SfM to triangulate a partial geometry scaffold. Second, we use an SfM-conditioned feed-forward pointmap predictor to predict complete pointmaps for view pairs, conditioning on both image tokens and point tokens. Finally, we fuse pairwise pointmaps with SfM-anchored global alignment. *Bottom*: Architecture of the predictor.

3.2 SfM-Conditioned Pointmap Predictor

To densify the partial reconstruction from Ω_S across the entire image, Scaffold3D uses a two-view conditioned pointmap predictor. For a given image pair (a, b) , the network takes the RGB images and normalized SfM points as input to predict complete 3D pointmaps $(\hat{X}_{a,a}, \hat{X}_{b,a})$ in the coordinate system of view a . We describe the architecture, normalization, and training objectives below.

3.2.1 Network Architecture

The architecture of our SfM-conditioned pointmap predictor (Fig. 2) comprises an RGB encoder and a point encoder to produce view-specific tokens. Spatially aligned RGB and point tokens are fused via element-wise addition before the first decoder layer at each view. These fused tokens are then processed by two cross-attended decoders to predict points and confidence maps. We instantiate the RGB encoder and decoders using the MAST3R architecture [18] and introduce a novel point encoder for aggregating SfM observations into patch-aligned conditioning tokens.

RGB encoder. Following [18, 37], each input image I_v ($v \in \{a, b\}$) is patchified and encoded by a ViT encoder [6]. The resulting RGB tokens are denoted as $f_v^0 \in \mathbb{R}^{L \times d}$, where $L = HW/P_{\text{patch}}^2$ and P_{patch} is the patch size.

Point encoder. We map the SfM inputs to a sequence of point tokens spatially aligned with the RGB tokens. Given the normalized SfM input $\hat{S}_v \in \mathbb{R}^{H \times W \times 3}$, we partition the map into patches of size P_{patch} . For each pixel $(v, p) \in \Omega_S$ with a valid SfM estimate, an MLP ϕ_{pt} encodes its 3D coordinates $\hat{S}_v(p)$ and its relative 2D location within the patch $\delta(p)$:

$$h_{v,p} = \phi_{\text{pt}}(\hat{S}_v(p), \delta(p)). \quad (1)$$

For each patch q , we average the features $h_{v,p}$ within the patch to compute a patch-level feature $g_{v,q}$:

$$g_{v,q} = \frac{1}{|\mathcal{K}_{v,q}|} \sum_{p \in \mathcal{K}_{v,q}} h_{v,p}, \quad \mathcal{K}_{v,q} = \{p \mid (v, p) \in \Omega_S, p \text{ is in patch } q\}. \quad (2)$$

We collect features $g_{v,q} \in \mathbb{R}^d$ for all patches with valid SfM support to form a sequence $g_v \in \mathbb{R}^{L' \times d}$, where $L' \leq L$. Lightweight self-attention blocks then let these patch-level point tokens exchange information across image patches:

$$Z_v^{\ell+1} = \text{PointBlock}^\ell(Z_v^\ell), \quad Z_v^0 = \{g_{v,q} \mid |\mathcal{K}_{v,q}| > 0\}. \quad (3)$$

After processing, we scatter refined features onto the image-token grid as $\bar{g}_v \in \mathbb{R}^{L \times d}$, setting $\bar{g}_{v,q} = 0$ for empty patches. This ensures that point tokens $\bar{g}_v$ are spatially aligned with image tokens f_v^0 .

RGB and point token fusion. We fuse the RGB tokens (f_a^0, f_b^0) and point tokens $(\bar{g}_a, \bar{g}_b)$ for each view via element-wise addition:

$$\tilde{f}_a^0 = f_a^0 + \bar{g}_a, \quad \tilde{f}_b^0 = f_b^0 + \bar{g}_b. \quad (4)$$

Decoders. The fused tokens $\tilde{f}_a^0$ and $\tilde{f}_b^0$ are fed into decoders Dec_a and Dec_b , which employ mutual cross-attention:

$$\tilde{f}_a^{\ell+1} = \text{Dec}_a^\ell(\tilde{f}_a^\ell, \tilde{f}_b^\ell), \quad \tilde{f}_b^{\ell+1} = \text{Dec}_b^\ell(\tilde{f}_b^\ell, \tilde{f}_a^\ell). \quad (5)$$

The features from the final attention layers are processed by two point heads to predict the dense geometry and associated confidence:

$$\left(\hat{X}_{a,a}, \hat{C}_{a,a} \right) = \psi_a(\{\tilde{f}_a^\ell\}_\ell), \quad \left(\hat{X}_{b,a}, \hat{C}_{b,a} \right) = \psi_b(\{\tilde{f}_b^\ell\}_\ell). \quad (6)$$

Here, $\hat{X}_{a,a}$ and $\hat{X}_{b,a}$ represent 3D coordinates in the camera coordinate system of view a , following the convention in MAST3R [18].

Input normalization. We transform the original SfM points (S_a, S_b) , initially in a global coordinate system with an arbitrary scale, into normalized local coordinates $(\hat{S}_a, \hat{S}_b)$ for view a . We first perform a conditioning-free pass (by setting $\bar{g}_a = \bar{g}_b = 0$) to obtain unconditional point predictions $(\hat{X}_{a,a}^{\text{uncond}}, \hat{X}_{b,a}^{\text{uncond}})$. Additionally, we transform S_a and S_b into camera a 's coordinate system as $S_{a \rightarrow a}$ and $S_{b \rightarrow a}$ using the inverse SfM pose T_a^{-1} . The scale factor s is determined by the median depth ratio across all SfM-supported pixels in the pair:

$$s = \text{median} \left\{ \frac{\left[\hat{X}_{v,a}^{\text{uncond}}(p) \right]_z}{[S_{v \rightarrow a}(p)]_z} \mid v \in \{a, b\}, (v, p) \in \Omega_S \right\}. \quad (7)$$

The normalized SfM inputs for the pair (a, b) are then calculated as $\hat{S}_a = s \cdot S_{a \rightarrow a}$ and $\hat{S}_b = s \cdot S_{b \rightarrow a}$.

3.2.2 Training Loss

We supervise both the unconditional and SfM-conditioned forward passes with equal weight. Following MAST3R [18], each pass uses a confidence-weighted point map regression loss $\mathcal{L}_{\text{pm}}$ and an auxiliary InfoNCE matching loss $\mathcal{L}_{\text{match}}$:

$$\mathcal{L}_{\text{train}} = \frac{1}{2} \sum_{t \in \{\text{uncond}, \text{cond}\}} (\mathcal{L}_{\text{pm}}^t + 0.075 \mathcal{L}_{\text{match}}^t). \quad (8)$$

Training begins with ground-truth scaffold points sampled from overlapping regions and later transitions to real SfM scaffold points using the inference-time scale alignment. We randomly drop point tokens during training to ensure the decoder remains robust to varying scaffold densities.

3.3 SfM-Anchored Global Alignment

To aggregate pairwise predictions into a global 3D scene, we extend the global alignment framework from DUST3R [37] by introducing an SfM-anchoring term. Let $\mathcal{E}$ be the set of selected image-pair edges. We fix the camera parameters $\{K_n, T_n\}_{n=1}^N$ provided by the SfM scaffold and optimize a dense depth map D_v for each view along with an edge-local similarity transform (P_e, σ_e) for each pair. Here, $P_e \in SE(3)$ maps the pairwise frame to the SfM world frame, and $\sigma_e > 0$ scales the transformed point. The scene-frame point for pixel p is defined as:

$$\chi_v(p; D_v) = T_v \Pi^{-1}(p, D_v(p); K_v), \quad (9)$$

where Π^{-1} denotes pinhole back-projection. The pairwise loss $\mathcal{L}_{\text{pair}}$ enforces consistency between optimized scene points and pairwise predictions:

$$\begin{aligned} \mathcal{L}_{\text{pair}} = & \sum_{e=(a,b) \in \mathcal{E}} \sum_p \hat{C}_{a,a}^e(p) \left\| \chi_a(p; D_a) - \sigma_e P_e \hat{X}_{a,a}^e(p) \right\| \\ & + \hat{C}_{b,a}^e(p) \left\| \chi_b(p; D_b) - \sigma_e P_e \hat{X}_{b,a}^e(p) \right\|. \end{aligned} \quad (10)$$

To ensure the dense maps remain anchored to the high-precision scaffold, we add a weighted anchoring term for SfM-supported pixels:

$$\mathcal{L}_{\text{anc}} = \frac{1}{\sum_{(v,p) \in \Omega_S} w_{v,p}} \sum_{(v,p) \in \Omega_S} w_{v,p} \|\chi_v(p; D_v) - S_v(p)\|_2. \quad (11)$$

The weights are DLT confidence scores $w_{v,p} = 1/(1 + e_{v,p}^{\text{reproj}})$, where $e_{v,p}^{\text{reproj}}$ is the reprojection error averaged over the visible 2D observations of the DLT track rasterized at pixel p . The final alignment is the result of:

$$\min_{\{D_v\}, \{P_e\}, \{\sigma_e > 0\}} \mathcal{L}_{\text{pair}} + \lambda_{\text{anc}} \mathcal{L}_{\text{anc}}. \quad (12)$$

At inference, the output is a set of dense point maps in the SfM world frame, with camera parameters preserved from the upstream SfM pipeline.

4 Experiments

Architecture and training. We train Scaffold3D on the 11 datasets as summarized in Table B2; full architecture, optimizer, schedule, and compute details are provided in Section C.

Evaluation metrics. We align each predicted reconstruction to the metric ground truth with robust Umeyama alignment [28], compute Euclidean point errors on ground-truth-covered pixels, and report the area under the accuracy curve (AUC) at the specified cm error thresholds. Our main tables use AUC@1/5/10, and AUC@M, where M is AUC@10 on monocular pixels visible in exactly one input view. All methods are evaluated on the same 512×384 image canvas. For MV-dVRK, the same numeric thresholds are measured in millimeters.

Evaluation datasets. We evaluate multi-view 3D reconstruction on five benchmarks. ScanNet++ [43] is treated as an in-domain indoor benchmark, while ETH3D [27] and Tanks&Temples [17] test cross-domain generalization to real captures with different scene statistics. We further evaluate on two out-of-domain datasets: 4D-DRESS [38], which contains clothed humans with masked white background, and MV-dVRK [4], which contains synthetic robotic surgical scenes. Table B1 summarizes view count, ground-truth coverage, view-overlap, and SfM-scaffold support statistics.

Baselines. We compare against four groups of methods. Feed-forward predictors include DUST3R [37], MAST3R [18], MAST3R-SfM [7], VGGT [35], Pi3/Pi3X [39], Pow3R [13], MapAnything [16], and our pointmap predictor without SfM condition inputs. Monocular depth-completion baselines, OMNI-DC [49] and Marigold-DC [31], complete each view independently from SfM scaffold depth samples. Two-view pointmap predictors, including DUST3R, MAST3R, Pow3R, and Scaffold3D, output pairwise geometry, so we fuse their predictions into scene-level reconstructions with SfM-anchored global alignment. For point-token propagation, we evaluate GGPT [5] with different feed-forward initializers. Geometry-conditioned predictors include MVSA anywhere [12], MapAnything, Pi3X, and Pow3R conditioned on the same dataset-specific SfM scaffold as our method. MVSA anywhere uses the scaffold cameras and depth ranges derived from the scaffold points, but does not consume point tokens. Camera diagnostics for all methods are provided in Table E3. All geometry-conditioned results use a fixed dataset-specific SfM scaffold, built as described in Section 3.1. We follow the SfM pipeline introduced by GGPT [5], running RoMaV2 dense matching [9] on the corresponding feed-forward image grid (see Table A5 for full-resolution matching).

4.1 Main Results

Comparison to feed-forward and conditioned baselines. Table 1 shows that Scaffold3D is strongest on all-pixel AUC@10 for the cross-domain ETH3D and Tanks&Temples benchmarks, reaching 66.4 and 60.9, respectively, and outperforming feed-forward predictors, point-token propagation, monocular depth completion, and geometry-conditioned baselines. On the in-distribution ScanNet++ benchmark, it achieves the best AUC@1 and reaches 69.6 AUC@10. On the cross-domain benchmarks, the gains are especially pronounced on monocular pixels, where dense geometry must be inferred beyond direct multi-view triangulation. On Tanks&Temples₄, it improves AUC@M from 24.0 for the best feed-forward model and 23.4 for GGPT(Pi3) to 26.3.

On out-of-domain data, behavior depends on the quality and coverage of the SfM scaffold. Scaffold3D is best on most MV-dVRK metrics, including AUC@M, AUC@10, and AUC@5. On 4D-DRESS, it

Table 1: **Multi-view 3D reconstruction.** Scaffold3D achieves strongest overall AUC@10 on the **cross-domain** benchmarks and MV-dVRK and best AUC@1 on in-distribution ScanNet++, while remaining competitive on 4D-DRESS where the SfM scaffold already covers most evaluated pixels. Gains are especially clear on low-overlap and monocular metrics where dense geometry must be completed beyond direct multi-view triangulation. Each dataset block reports AUC@1/5/10/M. M denotes the 10 cm AUC on monocular pixels, i.e., pixels observed by exactly one view. Thresholds are in centimeters, except MV-dVRK, which uses same numeric thresholds in millimeters. IT, PT, and PD indicate image tokens, point tokens, and depth-map conditioning. Geometry-conditioned baselines use the same dataset-specific SfM scaffold. Best and second-best results per metric and dataset are **bold** and underlined. [†] denotes fine-tuning under our training setup. [§]MVSAnywhere uses scaffold cameras only, with scaffold points used only to set depth ranges.

Method	IT	PT	PD	ScanNet++ ₄				ETH3D ₄				Tanks&Temples ₄				4D-DRESS				MV-dVRK																		
				AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M																		
				<i>M pixels: 24.0%</i>				<i>M pixels: 26.5%</i>				<i>M pixels: 10.2%</i>				<i>M pixels: 0.9%</i>				<i>M pixels: 0.4%</i>																		
<i>Feed-forward predictors</i>																																						
VGGT-Omega [35]	✓	✗	✗	3.2	21.9	38.8	26.6	7.0	35.4	51.8	43.1	7.5	30.9	45.7	23.1	34.0	77.5	88.2	71.1	16.1	51.6	68.1	29.1															
MAS3R-SfM [7]	✓	✗	✗	12.3	32.8	43.9	12.1	13.9	36.8	49.3	24.4	19.6	41.0	52.9	13.4	56.6	72.0	76.9	27.7	51.4	68.7	75.8	24.7															
MAS3R [18]	✓	✗	✗	8.0	33.6	49.8	37.9	6.7	24.5	36.7	23.5	8.5	31.0	45.0	18.3	22.2	52.4	64.7	32.1	10.0	33.1	47.4	13.4															
DUS3R [37]	✓	✗	✗	9.4	34.9	50.4	39.6	5.1	21.1	33.3	21.7	5.8	24.3	37.5	14.2	7.8	31.6	46.1	19.1	14.0	41.6	56.4	19.1															
Pow3R [13]	✓	✗	✗	10.2	36.7	52.4	42.9	6.3	20.4	31.8	18.4	5.2	22.1	34.8	12.1	7.6	32.2	45.9	20.6	9.8	32.4	47.1	15.5															
VGGT [35]	✓	✗	✗	8.2	38.5	55.6	44.7	6.4	26.3	41.4	28.5	6.6	29.8	43.3	20.3	13.1	54.2	73.3	45.9	9.7	33.7	49.1	15.0															
MapAnything [16]	✓	✗	✗	11.7	45.2	61.0	53.2	1.7	11.2	21.2	17.1	2.9	17.2	30.2	11.8	3.7	22.3	40.2	25.8	2.6	16.5	30.8	5.1															
MAS3R [†]	✓	✗	✗	17.1	50.2	65.6	54.9	6.2	26.4	40.6	28.1	9.2	30.8	44.4	19.5	3.0	15.0	26.5	7.8	26.0	56.5	70.1	28.7															
Pi3X [39]	✓	✗	✗	15.0	50.8	66.5	<u>56.6</u>	9.5	33.3	47.3	35.6	5.4	28.9	44.8	22.3	3.7	15.4	26.5	17.5	4.4	21.0	35.3	8.3															
Pi3 [39]	✓	✗	✗	16.7	52.0	68.0	57.8	6.2	29.2	44.1	31.5	5.6	31.0	46.6	<u>24.0</u>	10.5	50.4	71.4	49.4	18.3	51.9	67.0	27.2															
<i>Monocular depth completion</i>																																						
OMNI-DC [49]	✓	✗	✓	22.5	44.5	56.1	20.0	<u>25.6</u>	47.0	57.1	12.7	26.1	48.8	59.5	11.0	74.8	88.4	92.7	45.5	66.1	82.9	88.0	32.9															
Marigold-DC [31]	✓	✗	✓	21.8	44.4	56.3	27.7	<u>25.6</u>	48.5	59.6	20.8	25.2	47.1	58.1	15.3	76.2	<u>89.1</u>	<u>93.1</u>	50.4	<u>68.6</u>	<u>84.7</u>	<u>89.7</u>	40.4															
<i>Point-token propagation</i>																																						
GGPT-DLT (VGGT-Omega) [5]	✗	✓	✗	21.4	39.8	46.8	0.9	24.0	46.9	55.6	0.8	27.2	49.2	59.2	0.8	<u>76.5</u>	85.7	88.3	4.2	65.1	79.4	83.3	11.4															
GGPT (VGGT) [5]	✗	✓	✗	16.2	48.0	63.2	44.4	14.0	42.2	56.6	26.7	14.8	42.6	55.1	17.2	69.3	86.4	91.9	49.7	56.6	76.3	83.0	27.5															
GGPT (Pi3X) [5]	✗	✓	✗	16.9	50.0	65.3	51.0	15.5	45.4	60.0	29.0	17.0	43.7	56.1	20.8	63.7	82.6	87.6	21.8	55.7	74.3	81.0	23.0															
GGPT (Pi3) [5]	✗	✓	✗	14.0	50.2	66.1	54.9	7.4	33.3	50.0	27.3	10.4	36.6	51.9	23.4	72.4	87.8	92.7	53.4	61.7	81.4	87.9	35.8															
GGPT (MAS3R [†]) [5]	✗	✓	✗	<u>23.5</u>	<u>55.3</u>	<u>69.0</u>	52.4	16.0	44.9	58.8	22.5	20.5	47.3	<u>59.9</u>	17.2	49.0	59.9	64.4	11.2	68.8	84.1	89.1	38.8															
<i>Geometry-conditioned predictors</i>																																						
MVSAnywhere [12] [§]	✓	✗	✗	10.2	27.5	40.1	23.6	11.3	30.3	41.8	21.5	9.9	29.8	42.9	16.4	30.2	63.1	77.4	<u>57.9</u>	48.9	71.0	79.8	31.2															
MapAnything [16]	✓	✗	✓	12.4	37.9	50.4	30.9	6.7	26.3	39.5	21.4	9.3	31.1	44.8	15.5	39.0	74.5	82.9	37.3	49.8	72.9	79.2	17.6															
Pow3R [13] (SfM-anchored DGA)	✓	✗	✓	23.4	49.4	61.7	42.1	25.7	50.9	62.3	27.2	26.1	48.1	59.3	21.0	76.9	89.2	93.2	41.2	65.2	83.5	88.6	36.5															
Pow3R [13] (DGA)	✓	✗	✓	20.0	49.1	62.2	42.5	18.7	45.3	58.4	25.7	18.9	43.0	56.0	20.8	59.2	84.6	90.9	39.5	67.6	83.4	88.3	36.7															
Pi3X [39]	✓	✗	✓	20.4	54.5	68.5	55.9	12.4	42.1	57.8	<u>40.9</u>	11.6	37.7	52.7	23.6	64.7	86.4	91.9	57.6	66.7	84.2	89.5	<u>42.9</u>															
Scaffold3D (ours)	✓	✓	✗	25.7	56.5	69.6	56.3	<u>25.6</u>	53.6	66.4	40.3	<u>26.4</u>	49.2	60.9	26.3	74.4	88.6	92.9	51.2	68.4	85.4	90.4	44.0															

remains competitive, but the dataset has very few monocular pixels and high scaffold support, leaving less room for geometry propagation beyond the scaffold. This is reflected in Table B1, where only 0.9% of evaluated 4D-DRESS pixels are monocular and 92.6% are already supported by the scaffold. Additional out-of-domain experiments in Table E1 further show a limitation of SfM-conditioned reconstruction: when matching and BA+DLT produce unreliable scaffolds on articulated humans or texture-poor regions, strong feed-forward priors are more robust than scaffold-conditioned methods.

Effect of the number of views. Table 2 evaluates the same scenes with 4, 8, and 16 views. Increasing the number of views reduces the monocular fraction substantially: from 24.0% to 2.9% on ScanNet++, from 26.5% to 3.9% on ETH3D, and from 10.2% to 2.7% on Tanks&Temples. Scaffold3D remains best or second-best on nearly all all-pixel metrics and is particularly strong on monocular regions. This suggests that SfM conditioning is useful both in the sparse four-view setting and when additional views reduce the no-overlap area but do not eliminate the need for learned completion.

Qualitative comparison. Figure 3 compares representative reconstructions across in-domain, cross-domain, and out-of-domain scenes. The visualizations show that Scaffold3D preserves accurate scaffold-supported geometry while improving dense prediction beyond directly triangulated regions.

4.2 Ablations and Analysis

How should the scaffold be used? Table 3 compares where to use the SfM scaffold. The scaffold-only row evaluates the triangulated points themselves, and therefore has 0.0 AUC@N by construction because N is outside SfM support. Naive depth completion uses scaffold depth samples to complete each view independently; on ETH3D₄, it reaches 39.7 AUC@N, 9.4 points below Scaffold3D. SfM-anchored DGA on top of the non-conditioned MAS3R[†] prediction uses the scaffold only after

Table 2: **View-count scaling.** We evaluate the same scenes with 4, 8, and 16 input views. Each dataset block reports AUC@10 on all evaluated pixels and on monocular pixels. Additional views reduce the fraction of no-overlap pixels. Scaffold3D remains best or comparable as the number of views varies.

Method	ScanNet++						ETH3D						Tanks&Temples					
	All points			Monocular			All points			Monocular			All points			Monocular		
	4	8	16	4	8	16	4	8	16	4	8	16	4	8	16	4	8	16
Monocular pixels (%)	–	–	–	24.0	9.1	2.9	–	–	–	26.5	12.9	3.9	–	–	–	10.2	4.7	2.7
<i>Feed-forward predictors</i>																		
DUST3R [37]	50.4	49.6	51.0	39.6	36.7	38.9	33.3	27.6	25.9	21.7	14.5	9.2	37.5	34.6	30.9	14.2	6.1	5.5
MASt3R [18]	49.8	49.8	50.0	37.9	40.7	39.3	36.7	31.1	28.9	23.5	15.3	19.0	45.0	43.4	40.8	18.3	8.9	9.8
Pow3R [13]	52.4	52.3	53.6	42.9	39.6	42.2	31.8	27.4	26.6	18.4	13.3	12.8	34.8	32.5	28.6	12.1	7.0	7.8
MapAnything [16]	61.0	60.0	61.3	53.2	53.1	52.1	21.2	17.0	13.5	17.1	12.4	9.9	30.2	26.9	24.9	11.8	8.0	5.2
VGGT [35]	55.6	56.6	57.9	44.7	48.1	44.4	41.4	37.1	31.2	28.5	19.9	19.8	43.3	44.5	41.1	20.3	14.7	12.9
VGGT-Omega [35]	38.8	38.2	37.6	26.6	25.3	28.1	51.8	48.3	42.9	43.1	35.3	29.7	45.7	45.6	41.3	23.1	18.6	13.9
Pi3 [39]	68.0	70.2	72.4	57.8	61.7	61.8	44.1	35.7	34.9	31.5	21.2	20.3	46.6	47.0	44.0	<u>24.0</u>	18.1	15.8
Pi3X [39]	66.5	70.1	72.0	<u>56.6</u>	62.1	61.7	47.3	42.3	38.9	35.6	27.5	23.0	44.8	44.6	42.6	22.3	18.6	16.4
MASt3R [†]	65.6	66.4	66.8	54.9	57.0	52.8	40.6	36.4	34.8	28.1	19.8	15.9	44.4	44.7	39.7	19.5	10.2	13.2
<i>Point-token propagation</i>																		
GGPT (VGGT) [5]	63.2	70.2	74.8	44.4	51.6	48.6	56.6	58.0	62.8	26.7	20.0	22.8	55.1	61.2	59.7	17.2	17.2	17.7
GGPT (Pi3) [5]	66.1	71.0	74.6	54.9	58.7	58.6	50.0	49.5	52.4	27.3	22.6	28.4	51.9	57.4	57.9	23.4	19.6	19.7
GGPT (Pi3X) [5]	65.3	71.2	75.4	51.0	53.3	52.0	60.0	60.6	63.8	29.0	24.0	25.3	56.1	61.7	61.1	20.8	20.6	21.7
GGPT (MASt3R [†]) [5]	<u>69.4</u>	<u>74.9</u>	<u>78.8</u>	53.5	55.7	54.6	58.6	60.2	66.9	21.6	19.9	25.9	<u>60.1</u>	<u>67.6</u>	<u>67.0</u>	17.3	17.0	19.2
<i>Geometry-conditioned predictors</i>																		
MVSA anywhere [12]	40.1	51.5	58.1	23.6	31.5	35.7	41.8	48.4	52.5	21.5	21.5	27.5	42.9	47.2	47.7	16.4	17.7	20.2
MapAnything [16]	50.4	63.5	69.3	30.9	35.6	40.8	39.5	43.5	40.0	21.4	16.0	15.3	44.8	53.5	51.0	15.5	15.1	14.4
Pi3X [39] (submission)	61.9	66.3	70.9	50.0	53.3	54.6	51.5	46.3	45.3	37.9	29.7	27.2	48.9	49.6	46.2	21.1	17.4	19.0
Pi3X [39] (pred. cameras)	67.1	69.9	74.1	56.4	57.1	58.0	54.5	49.3	48.2	40.4	30.6	30.6	51.2	52.8	49.2	22.7	20.0	20.7
Pi3X [39]	68.5	73.8	77.4	55.9	61.7	61.3	57.8	56.1	59.2	<u>40.9</u>	34.3	37.3	52.7	56.7	55.3	23.6	23.0	21.8
Pow3R [13]	61.7	72.3	77.1	42.1	49.8	50.4	<u>62.3</u>	<u>64.9</u>	67.6	27.2	24.3	28.0	59.3	66.9	66.4	21.0	15.2	20.0
Scaffold3D (ours)	69.6	75.7	79.0	56.3	60.6	62.7	66.4	67.4	<u>67.0</u>	40.3	36.1	<u>36.6</u>	60.9	69.5	68.2	26.3	<u>21.3</u>	25.1

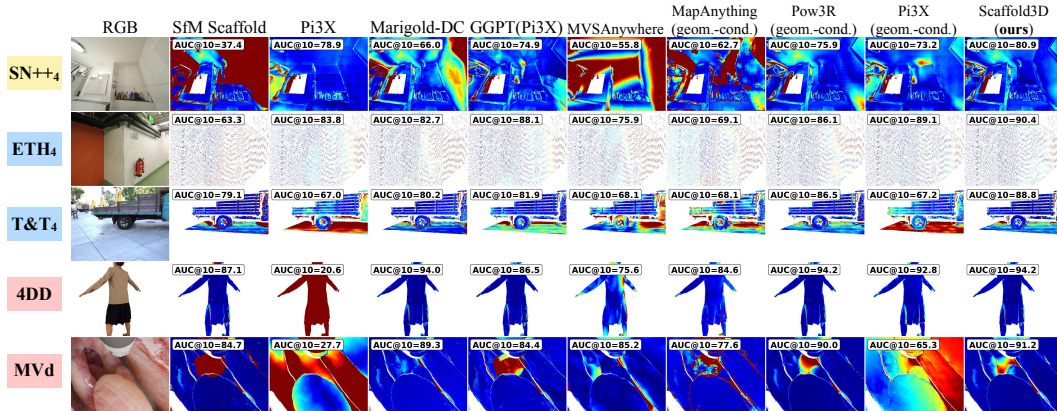


Figure 3: **Qualitative reconstruction comparison.** Each row shows one example from ScanNet++, ETH3D, Tanks&Temples, 4D-DRESS, and MV-dVRK, with only the first input view visualized due to space constraints. Columns compare the input RGB view, the SfM scaffold, a feed-forward Pi3X prediction, monocular depth completion, point-token propagation, and geometry-conditioned baselines using the same scaffold. AUC@10 is shown in each reconstruction tile.

dense prediction by fitting dense depth maps to fixed scaffold cameras; it reaches a similar 39.8 AUC@N, still 9.3 points below Scaffold3D. GGPT uses the scaffold as point tokens but propagates them without refining image-token features; on ETH3D₄, it reaches 22.9 AUC@N, 26.2 points below Scaffold3D. The same pattern holds on the other standard benchmarks: Scaffold3D improves over the best baseline that does not condition the decoder by 1.7 AUC@N points on ScanNet++₄ and 4.5 points on Tanks&Temples₄. The all-pixel AUC@10 follows the same pattern, with Scaffold3D reaching 66.4 vs. 61.9 on ETH3D₄ and 60.9 vs. 60.1 on Tanks&Temples₄ against the strongest alternatives. These results indicate that, among these alternatives, injecting SfM point tokens into image-token decoding is more effective than using the scaffold only before or after dense prediction.

Table 3: **SfM conditioning ablation.** We compare simple, training-free ways to use the SfM scaffold with MAST3R[†], the conditioning-free feed-forward predictor used by Scaffold3D, against our learned Scaffold3D predictor. Using the scaffold only for point propagation, naive depth completion, or global alignment after feed-forward prediction is weaker overall than injecting SfM point tokens into the image decoder. Each dataset block reports AUC@10 overall, outside SfM support (N), on monocular pixels (M), and on non-monocular pixels (NM). M/NM labels are approximated from GT reprojection. N/A is used for methods not using SfM. DGA denotes DUST3R global alignment; Unanchored DGA removes the final SfM anchoring term. See Table B1 for the overlap and SfM-support ratios.

Method	ScanNet++ ₄				ETH3D ₄				Tanks&Temples ₄				4D-DRESS				MV-dVRK			
	AUC@10/N/M/NM				AUC@10/N/M/NM				AUC@10/N/M/NM				AUC@10/N/M/NM				AUC@10/N/M/NM			
MASt3R [†] feed-forward	65.6	N/A	54.9	68.9	40.6	N/A	28.1	44.1	44.4	N/A	19.5	46.2	26.5	N/A	7.8	26.9	70.1	N/A	28.7	70.3
SfM scaffold	25.2	0.0	0.2	33.1	42.0	0.0	0.3	56.7	52.0	0.0	0.4	56.2	87.6	0.0	4.8	88.7	83.1	0.0	11.3	83.4
GGPT (MASt3R [†])	69.4	58.9	53.5	74.0	58.6	22.9	21.6	70.3	<u>60.1</u>	18.4	17.3	<u>63.8</u>	<u>92.7</u>	<u>71.9</u>	<u>53.4</u>	<u>93.1</u>	<u>89.1</u>	56.5	38.8	<u>89.4</u>
Naive depth completion (MASt3R [†])	68.4	64.1	57.1	72.1	61.4	39.7	33.6	69.9	58.5	27.2	20.9	61.8	92.6	66.6	57.5	93.0	88.1	45.9	33.6	88.3
MASt3R [†] + SfM-anchored DGA	68.0	63.8	56.4	71.7	<u>61.9</u>	39.8	32.5	<u>70.5</u>	59.8	31.0	24.0	62.8	86.7	45.8	34.2	87.3	85.7	49.0	34.5	86.0
Scaffold3D (Unanchored DGA)	69.1	<u>65.3</u>	54.5	<u>73.7</u>	61.2	47.8	39.2	68.0	55.2	34.9	<u>26.2</u>	57.9	61.5	55.3	39.3	61.8	88.4	81.0	44.3	88.6
Scaffold3D (ours)	69.6	65.8	56.3	74.0	66.4	49.1	40.3	74.7	60.9	35.5	26.3	64.1	92.9	72.7	51.2	93.5	90.4	66.4	<u>44.0</u>	90.6

Role of SfM anchoring. The unanchored-DGA ablation isolates the final anchoring term in Eq. (12). Adding the SfM anchor improves the final AUC@10 on every main dataset and is especially important for keeping non-monocular regions tied to the optimized scaffold. On MV-dVRK, unanchored alignment is stronger on the no-SfM subset, but the anchored model achieves better overall reconstruction, reflecting a tradeoff between free dense fitting and respecting the reliable scaffold.

Architecture and training choices. Additional ablations in Table A1 support three design choices: (1) point tokens are most effective when fused by element-wise addition before decoding rather than appended, masked, replaced, or gated; (2) aggregating all SfM observations per image patch and refining the resulting point tokens with self-attention gives a small gain; and (3) pretraining with GT scaffolds before switching to real SfM improves over training with real SfM from the start. Table A3 shows that different decoder-side conditioning mechanisms (either Pow3R-style or pointmap decoder fusion) are comparable under our chosen training and SfM-anchored alignment. This positions Scaffold3D as one effective instantiation of SfM-conditioned training and alignment. Finally, subsampling dense DLT scaffold observations gives similar results after a moderate budget, whereas using only bundle-adjusted tracks is weaker on several OOD and monocular metrics (Table A4).

4.3 Discussion and Future Work

Our results indicate that SfM is most useful as an explicit scaffold for dense prediction, rather than only as a post-processing constraint or incomplete triangulated reconstruction. This also defines the main limitation of Scaffold3D: it assumes the SfM scaffold is accurate but incomplete. When matching, bundle adjustment, or DLT fail, scaffold errors propagate, as shown on articulated-human and hand-object data in Table E1. Better scaffold validation or fallback mechanisms could reduce such errors. Even with a usable scaffold, some examples show limited propagation from scaffold-supported to unsupported regions, and predicted pointmaps do not always exactly preserve the scaffold-implied relative poses and intrinsics. Larger training mixtures, stronger models, or explicit consistency constraints may improve these aspects. The method is also not real time because it builds an SfM scaffold, runs two forward passes, and runs SfM-anchored global alignment (Table D1).

5 Conclusion

Scaffold3D uses SfM cameras and triangulated points as explicit conditioning signals, injects their observations as patch-aligned point tokens into image-token pointmap prediction, and fuses pairwise outputs with SfM-anchored global alignment. On standard sparse-view benchmarks, this improves over using the scaffold only as post-processing, depth-completion input, or point-token propagation, showing that learned reconstruction can use classical geometric optimization as a globally consistent scaffold for completing dense geometry.

Acknowledgements

This study was conducted within the national “Proficiency” research project (No. PFFS-21-19), funded by the Swiss Innovation Agency Innosuisse in 2021 as one of 15 flagship initiatives. This work was supported by the Swiss AI Initiative under project IDs A43, A44, and A129, with computational resources provided by the Swiss National Supercomputing Centre (CSCS) on the Alps infrastructure. We thank Luigi Piccinelli, Yiming Wang, Sergey Prokudin, Nikita Karaev, and Xudong Jiang for valuable discussions and support.

References

- [1] L. Bartolomei, F. Tosi, M. Poggi, and S. Mattoccia. Stereo anywhere: Robust zero-shot deep stereo matching even where either stereo or mono fail. In *CVPR*, 2025.
- [2] G. Baruch, Z. Chen, A. Dehghan, Y. Feigin, P. Fu, T. Gebauer, D. Kurz, T. Dimry, B. Joffe, A. Schwartz, and E. Shulman. ARKitScenes: A diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *NeurIPS*, 2021.
- [3] Y. Cabon, N. Murray, and M. Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020.
- [4] G. Caccianiga, S. Prokudin, B. Javot, Y. Chen, O. B. Aladag, R. L’Orsa, Y. Sharon, J. Rolinger, I. Capobianco, S. Tang, A. Deguet, and K. J. Kuchenbecker. MV-dVRK: Integrated methods for multi-viewpoint surgical telerobotics. <https://mv-dvrk.is.mpg.de/>, 2026.
- [5] Y. Chen, Y. Wang, X. Zhang, S. Prokudin, and S. Tang. GGPT: Geometry-grounded point transformer. In *CVPR*, 2026.
- [6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [7] B. P. Duisterhof, L. Zust, P. Weinzaepfel, V. Leroy, Y. Cabon, and J. Revaud. MAST3R-SfM: a fully-integrated solution for unconstrained structure-from-motion. In *3DV*, 2025.
- [8] J. Edstedt, Q. Sun, G. Bökman, M. Wadenbäck, and M. Felsberg. RoMa: Robust Dense Feature Matching. In *CVPR*, 2024.
- [9] J. Edstedt, D. Nordström, Y. Zhang, G. Bökman, J. Astermark, V. Larsson, A. Heyden, F. Kahl, M. Wadenbäck, and M. Felsberg. RoMa v2: Harder Better Faster Denser Feature Matching. *arXiv preprint arXiv:2511.15706*, 2025.
- [10] X. Fang, J. Gao, Z. Wang, Z. Chen, X. Ren, J. Lyu, Q. Ren, Z. Yang, X. Yang, Y. Yan, and C. Lyu. Dens3R: A foundation model for 3D geometry prediction. In *ICLR*, 2026.
- [11] H. Guo, H. Zhu, S. Peng, H. Lin, Y. Yan, T. Xie, W. Wang, X. Zhou, and H. Bao. Multi-view reconstruction via SfM-guided monocular depth estimation. In *CVPR*, 2025.
- [12] S. Izquierdo, M. Sayed, M. Firman, G. Garcia-Hernando, D. Turmukhambetov, J. Civera, O. Mac Aodha, G. J. Brostow, and J. Watson. MVSAnywhere: Zero shot multi-view stereo. In *CVPR*, 2025.
- [13] W. Jang, P. Weinzaepfel, V. Leroy, L. Agapito, and J. Revaud. Pow3R: Empowering unconstrained 3D reconstruction with camera and scene priors. In *CVPR*, 2025.
- [14] C. Jeong, I. Bae, J.-H. Park, and H.-G. Jeon. Test-time prompt tuning for zero-shot depth completion. In *ICCV*, 2025.
- [15] B. Ke, Q. Zhou, J. Huang, X. Ren, T. Shen, K. Schindler, L. Leal-Taixé, and S. Huang. Depth completion as parameter-efficient test-time adaptation. *arXiv preprint arXiv:2602.14751*, 2026.
- [16] N. Keetha, N. Müller, J. Schönberger, L. Porzi, Y. Zhang, T. Fischer, A. Knapitsch, D. Zauss, E. Weber, N. Antunes, J. Luiten, M. Lopez-Antequera, S. R. Bulò, C. Richardt, D. Ramanan, S. Scherer, and P. Kotschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *3DV*, 2026.
- [17] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM ToG*, 2017.
- [18] V. Leroy, Y. Cabon, and J. Revaud. Grounding image matching in 3D with MAST3R. In *ECCV*, 2024.

- [19] Z. Li and N. Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018.
- [20] H. Lin, S. Peng, J. Chen, S. Peng, J. Sun, M. Liu, H. Bao, J. Feng, X. Zhou, and B. Kang. Prompting Depth Anything for 4K resolution accurate metric depth estimation. In *CVPR*, 2025.
- [21] H. Lin, S. Chen, J. H. Liew, D. Y. Chen, Z. Li, Y. Zhao, S. Peng, H. Guo, X. Zhou, G. Shi, J. Feng, and B. Kang. Depth Anything 3: Recovering the visual space from any views. In *ICLR*, 2026.
- [22] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, 2021.
- [23] J. Lu, T. Huang, P. Li, Z. Dou, C. Lin, Z. Cui, Z. Dong, S.-K. Yeung, W. Wang, and Y. Liu. Align3R: Aligned monocular depth estimation for dynamic videos. In *CVPR*, 2025.
- [24] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.
- [25] Z. Pataki, P.-E. Sarlin, J. L. Schönberger, and M. Pollefeys. MP-SfM: Monocular Surface Priors for Robust Structure-from-Motion. In *CVPR*, 2025.
- [26] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *ICCV*, 2021.
- [27] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017.
- [28] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016.
- [29] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.
- [30] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeurIPS*, 2021.
- [31] M. Viola, K. Qu, N. Metzger, B. Ke, A. Becker, K. Schindler, and A. Obukhov. Marigold-DC: Zero-shot monocular depth completion with guided diffusion. In *ICCV*, 2025.
- [32] F. Wang, Q. Xu, Y.-S. Ong, and M. Pollefeys. Lightweight and accurate multi-view stereo with confidence-aware diffusion model. *TPAMI*, 2025.
- [33] H. Wang and L. Agapito. 3D reconstruction with spatial memory. In *3DV*, 2025.
- [34] J. Wang, N. Karaev, C. Rupprecht, and D. Novotny. VGGSfM: Visual geometry grounded deep structure from motion. In *CVPR*, 2024.
- [35] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025.
- [36] Q. Wang*, Y. Zhang*, A. Holynski, A. A. Efros, and A. Kanazawa. Continuous 3D perception model with persistent state. In *CVPR*, 2025.
- [37] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. DUSf3R: Geometric 3D vision made easy. In *CVPR*, 2024.
- [38] W. Wang, H.-I. Ho, C. Guo, B. Rong, A. Grigorev, J. Song, J. J. Zarate, and O. Hilliges. 4D-DRESS: A 4D dataset of real-world human clothing with semantic annotations. In *CVPR*, 2024.
- [39] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He. π^3 : Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.
- [40] Z. Wang, S. Chen, L. Yang, J. Wang, Z. Zhang, H. Zhao, and Z. Zhao. Depth Anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025.
- [41] H. Xia, Y. Fu, S. Liu, and X. Wang. RGBD objects in the wild: Scaling real-world 3D object learning from RGB-D videos. *arXiv preprint arXiv:2401.12592*, 2024.

- [42] Y. Yao, Z. Luo, S. Li, J. Zhang, Y. Ren, L. Zhou, T. Fang, and L. Quan. BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020.
- [43] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *CVPR*, 2023.
- [44] C. Zhang, G. Le Moing, S. Koppula, I. Rocco, L. Momeni, J. Xie, S. Sun, R. Sukthankar, J. K. Barral, R. Hadsell, Z. Ghahramani, A. Zisserman, J. Zhang, and M. S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. *arXiv preprint*, 2025.
- [45] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. In *ICLR*, 2025.
- [46] S. Zhang, J. Wang, Y. Xu, N. Xue, C. Rupprecht, X. Zhou, Y. Shen, and G. Wetzstein. FLARE: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *CVPR*, 2025.
- [47] Y. Zhou, Y. Wang, J. Zhou, W. Chang, H. Guo, Z. Li, K. Ma, X. Li, Y. Wang, H. Zhu, M. Liu, D. Liu, J. Yang, Z. Fu, J. Chen, C. Shen, J. Pang, K. Zhang, and T. He. OmniWorld: A multi-domain and multi-modal dataset for 4D world modeling. In *ICLR*, 2026.
- [48] Y. Zuo and J. Deng. OGNI-DC: Robust depth completion with optimization-guided neural iterations. In *ECCV*, 2024.
- [49] Y. Zuo, W. Yang, Z. Ma, and J. Deng. OMNI-DC: Highly robust depth completion with multiresolution depth integration. In *ICCV*, 2025.

A Additional Ablations

Table A1 reports development runs that motivated the final design. Each block is a controlled comparison that isolates one design decision under comparable training conditions.

Table A1: **Model-development ablations.** We report development runs that isolate key choices in Scaffold3D. Each dataset block reports AUC at 1, 5, and 10 cm. The controlled blocks motivate adding point tokens to image tokens, aggregating point tokens so the model can use all available SfM points, and pretraining with GT-overlap points before real SfM. These ablations use earlier training configurations than the final model (e.g., smaller training and data scale, different source of SfM scaffold), but are comparable within each block. Cyan rows indicate the selected design choices. Best and second-best results within each block are **bold** and underlined.

Variant	ScanNet++ ₄			ETH3D ₄			T&T ₄		
	AUC@1/5/10			AUC@1/5/10			AUC@1/5/10		
<i>Token fusion (first sweep)</i>									
Append point tokens as extra tokens	20.3	49.5	64.6	5.6	21.3	34.0	4.8	20.3	33.7
Mask image tokens at point locations	15.5	43.7	59.4	6.5	21.9	35.3	4.1	17.5	30.3
Replace image tokens with point tokens	17.6	49.3	65.5	7.4	26.5	40.6	5.9	24.1	38.9
Add point tokens to image tokens	25.7	59.4	72.8	11.6	37.1	53.6	11.7	35.2	50.6
<i>Token fusion (second sweep)</i>									
Replace image tokens with point tokens	27.1	59.4	71.9	12.5	37.0	52.6	14.6	38.9	54.2
Replace image tokens + point-token dropout	25.0	58.8	72.3	12.6	36.8	52.7	13.4	38.3	53.7
Gate image/point tokens	20.8	56.4	71.4	18.1	47.1	62.3	16.4	43.2	58.6
Add point tokens to image tokens	29.8	61.3	74.0	20.3	50.1	64.6	18.5	45.9	60.5
<i>Point-token capacity</i>									
Unaggregated point tokens (768 points)	5.2	31.3	52.3	4.2	30.3	52.4	9.8	46.9	66.8
Patch-aggregated point tokens (all points)	5.6	32.0	52.9	4.0	30.9	52.8	10.7	47.0	66.7
Patch aggregation + point self-attention	5.3	32.6	53.6	4.9	32.8	54.1	11.5	48.8	67.9
<i>Sparse-input curriculum</i>									
Real SfM from start	17.4	48.1	64.2	8.3	28.2	41.8	6.1	23.2	36.6
GT-overlap pretrain → real SfM	20.3	51.9	67.2	11.5	33.7	48.3	11.7	34.0	48.9

Table A2 studies Pi3X fine-tuning under our setup, comparing training duration and recipe choices before and after conditioning on the same SfM scaffold.

Table A3 compares scaffold-conditioning interfaces under our training and alignment pipeline. The Pow3R-pretrained rows show that our SfM-anchored DGA improves over plain DGA. The MAST3R-initialized rows use the same scaffold construction, SfM-conditioned training setup, and SfM-anchored final alignment, and show that several conditioning interfaces can turn an SfM scaffold into dense geometry.

Table A4 evaluates how many SfM points are needed for conditioning. The results saturate with a modest number of sampled observations on most datasets, suggesting diminishing returns from using every available scaffold observation. The zero-point row corresponds to running the same architecture without usable scaffold points, isolating the contribution of SfM point conditioning.

Table A5 checks whether using full-resolution 4:3 center crops for RoMaV2 matching changes the scaffold-conditioned comparison under the same evaluation protocol. The main effect is a stronger ETH3D scaffold, while the relative trends remain similar across methods.

Table A6 replaces the estimated SfM scaffold points with ground-truth points at the same scaffold-supported pixels, leaving unsupported pixels unsupported. This separates downstream use of the scaffold from errors in point triangulation: the table measures how much each method benefits when the conditioning points are accurate without increasing scaffold support.

Table A2: **Pi3X fine-tuning ablations.** We compare released Pi3X [39] with variants fine-tuned under our SfM-conditioned training setup. The first numeric block reports mean AUC@10 over the five datasets. Each dataset block reports all-pixel AUC@10 for the conditioning-free feed-forward output (FF), conditioning on the same dataset-specific SfM scaffold with predicted cameras (Pred.), and conditioning with the scaffold cameras fixed (Fixed). Thresholds are in centimeters except MV-dVRK, which uses millimeters. Unless stated otherwise, variants use 67K steps, learning rate 10^{-5} , bfloat16, and a frozen image encoder. Cyan marks the final 123K-step checkpoint. Best and second-best results within each section are **bold** and underlined.

Variant	Overall		ScanNet++ ₄		ETH3D ₄		T&T ₄		4D-DRESS		MV-dVRK							
	FF	Pred. Fixed	FF	Pred. Fixed	FF	Pred. Fixed	FF	Pred. Fixed	FF	Pred. Fixed	FF	Pred. Fixed						
Training length																		
Released Pi3X (no fine-tuning)	44.1	67.4	72.1	66.5	67.1	68.5	47.3	54.5	57.8	44.8	51.2	52.7	26.5	90.0	91.9	35.3	74.1	89.5
34K steps (1 day)	47.4	68.1	72.4	69.2	69.7	71.6	38.8	53.0	58.6	40.2	46.2	49.6	35.9	88.8	91.9	53.0	83.0	90.2
67K steps (2 days)	51.6	68.1	72.3	70.5	70.5	72.0	39.4	52.5	57.9	39.8	45.2	49.3	55.5	89.6	91.8	52.8	82.5	90.6
123K steps (4 days)	52.8	68.7	72.3	71.4	72.0	72.7	39.2	52.3	57.4	41.9	45.4	48.6	59.3	90.3	92.3	52.3	83.5	90.5
Training choices (67K steps)																		
Base recipe	51.6	68.1	72.3	70.5	70.5	72.0	39.4	52.5	57.9	39.8	45.2	49.3	55.5	89.6	91.8	52.8	82.5	90.6
Learning rate 10^{-4}	45.5	63.5	70.3	61.8	63.5	67.9	32.1	46.1	55.4	33.4	42.3	46.7	47.0	82.1	91.6	52.9	83.7	89.7
Scaffold-quality gating	53.9	68.6	72.4	70.3	69.3	71.6	40.7	53.9	57.7	40.7	46.2	50.3	59.6	89.1	92.0	58.1	84.3	90.3
GT-scaffold probability 0.1	49.6	67.9	71.9	70.9	71.2	72.1	40.7	50.9	56.8	40.4	45.3	48.7	44.7	89.3	91.7	51.2	82.9	90.2
No confidence loss	52.8	67.8	72.2	70.9	70.4	72.1	40.1	52.7	57.7	40.8	45.4	49.1	56.1	89.5	92.1	56.1	81.2	90.3
Unfrozen image encoder	56.1	68.7	72.2	67.3	70.9	71.7	38.1	52.9	57.9	40.1	45.2	48.9	69.2	88.9	92.0	66.0	85.6	90.4
Conditioning probability $2.5\times$	53.7	69.3	72.2	68.9	70.7	72.4	39.5	53.7	57.4	40.5	44.9	48.7	61.8	90.5	92.3	57.7	86.8	90.3

Table A3: **Pow3R-style conditioning.** We compare scaffold-conditioning interfaces under the same SfM inputs and evaluation protocol. The first numeric column reports mean AUC@10 over the five dataset blocks. Each dataset block reports AUC at 1, 5, and 10 in the dataset unit: centimeters except MV-dVRK, which uses millimeters. Conditioning input icons denote : scaffold as patchified depth map, : scaffold as 3D point tokens or patchified pointmaps, : scaffold cameras or relative pose, and : scaffold intrinsics/rays. Cyan marks our final Scaffold3D checkpoint trained with the staged curriculum. Best and second-best results are **bold** and underlined. The pretrained Pow3R using our SfM-anchored alignment improves over plain DGA by 3.9 AUC@10 on ETH3D and 3.3 on Tanks&Temples. Variants trained from the same MAST3R checkpoint suggest three takeaways: (1) in the simplified depth/pointmap-only interface, decoder fusion improves over encoder fusion by 0.5 AUC@10 for  inputs and 1.7 for  inputs on average, (2)  pointmap decoder fusion improves over  depth decoder fusion by 0.9 mean AUC@10, and (3) different conditioning mechanisms perform comparably well (either Pow3R-style or pointmap decoder fusion), so Scaffold3D should be interpreted as **one effective instantiation of SfM-conditioned training and alignment**. As usual for deep-learning ablations, these conclusions are tied to the dataset mixture, training length, hyperparameters, and SfM construction used here. With different training scales, tuning budgets, or scaffold construction details, the ordering of conditioning interfaces could change.

Variant	Input	Overall	ScanNet++ ₄	ETH3D ₄	T&T ₄	4D-DRESS	MV-dVRK
		AUC@10	AUC@1/5/10	AUC@1/5/10	AUC@1/5/10	AUC@1/5/10	AUC@1/5/10
Pow3R-pretrained references							
Pow3R [13] (DGA)		71.2	20.0 49.1 62.2	18.7 45.3 58.4	18.9 43.0 56.0	59.2 84.6 90.9	67.6 83.4 88.3
Pow3R [13]		73.0	23.4 49.4 61.7	25.7 50.9 62.3	26.1 48.1 59.3	76.9 89.2 93.2	65.2 83.5 88.6
Pow3R-pretrained + SfM fine-tune		75.6	25.9 56.2 69.1	25.6 52.0 64.5	27.0 48.9 59.9	75.3 89.2 93.4	69.9 86.1 91.0
Variants trained from the MAST3R checkpoint							
Pointmap encoder fusion		74.1	24.7 53.9 67.5	23.0 49.7 62.3	25.3 48.1 59.4	69.8 86.8 91.8	65.4 83.9 89.5
Depth encoder fusion		74.4	24.5 54.2 67.3	25.3 51.1 63.3	26.4 48.8 60.3	71.0 87.3 92.1	62.7 83.1 89.1
Depth decoder fusion		74.9	26.5 57.0 69.9	25.9 52.8 65.2	26.9 49.3 60.9	61.3 84.0 90.3	60.1 82.1 88.4
Pow3R with pointmap decoder fusion		75.4	26.3 55.6 68.9	25.3 52.7 65.5	24.1 47.4 59.2	76.0 88.9 93.1	69.8 85.3 90.3
Pow3R (native: depthmap encoder fusion)		75.7	27.3 56.7 69.4	24.4 52.7 66.0	25.2 48.0 59.5	76.3 89.4 93.4	68.8 85.2 90.3
Pow3R with depthmap decoder fusion		75.8	25.8 55.1 68.9	25.1 53.7 66.6	26.3 48.3 59.7	75.6 89.2 93.3	71.6 85.9 90.6
Pointmap decoder fusion		75.8	25.9 57.3 70.1	24.7 53.2 65.9	26.6 49.0 60.6	72.2 87.9 92.5	64.9 84.2 89.9
Final checkpoint trained with the staged Scaffold3D curriculum							
Scaffold3D: pointmap decoder fusion		76.0	25.7 56.5 69.6	25.6 53.6 66.4	26.4 49.2 60.9	74.4 88.6 92.9	68.4 85.4 90.4

Table A4: **SfM point-budget sensitivity.** We subsample the SfM scaffold before conditioning Scaffold3D. Each dataset block reports AUC@1, AUC@5, AUC@10, and AUC@M. M denotes AUC@10 on monocular pixels. Performance saturates after a moderate point budget in this subsampling experiment, suggesting diminishing returns from using every available scaffold observation. The separate BA-tracks row (4316 tracks on average) uses only bundle-adjusted tracks and omits the DLT-triangulated scaffold observations.

#Points	ScanNet++ ₄				ETH3D ₄				T&T ₄				4D-DRESS				MV-dVRK			
	AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M			
Subsampled SfM scaffold observations																				
0	24.7	54.8	68.0	56.1	24.2	49.4	62.1	32.1	26.1	48.6	60.0	23.8	56.6	79.4	86.6	34.3	57.1	79.0	85.7	34.5
512	25.7	56.6	69.6	58.7	23.2	51.2	64.8	40.3	25.6	48.6	60.2	22.7	74.0	88.4	92.8	51.6	64.6	83.6	89.2	40.9
1024	<u>26.1</u>	56.6	69.5	<u>58.2</u>	23.0	51.3	65.0	<u>40.7</u>	26.4	48.8	60.1	23.5	74.2	88.5	<u>92.9</u>	52.8	65.3	83.7	89.3	41.7
2048	26.3	56.7	69.6	57.7	23.3	51.6	65.1	39.8	24.1	48.2	59.9	24.7	75.1	88.7	93.0	<u>53.7</u>	67.1	84.4	89.7	42.2
4096	25.8	<u>56.8</u>	69.7	57.8	22.7	52.0	65.7	40.2	25.5	48.7	60.2	24.8	72.4	88.1	92.7	53.5	<u>69.0</u>	84.9	90.0	42.8
8192	<u>26.1</u>	56.9	69.8	57.9	<u>24.5</u>	52.3	65.7	40.6	25.9	<u>49.3</u>	<u>60.9</u>	25.2	73.6	88.4	92.8	53.9	68.0	84.8	90.0	43.1
16384	25.5	56.6	69.7	57.1	24.2	52.3	65.9	40.8	25.2	48.9	60.4	25.2	73.8	88.5	92.8	52.6	63.9	84.1	89.7	42.8
32768	26.0	56.9	70.0	57.1	24.1	52.0	65.6	39.6	25.9	49.2	60.6	25.7	<u>74.8</u>	88.7	93.0	50.8	69.4	<u>85.3</u>	<u>90.2</u>	43.6
65536	25.3	56.2	69.3	56.4	23.8	52.6	65.9	40.5	26.9	49.9	61.3	26.6	73.8	88.5	<u>92.9</u>	51.1	68.3	85.2	<u>90.2</u>	44.0
131072	<u>26.1</u>	56.5	69.6	56.1	24.6	53.3	66.3	40.4	<u>26.6</u>	<u>49.3</u>	60.7	25.9	74.2	<u>88.6</u>	<u>92.9</u>	51.1	65.8	84.7	90.0	44.0
all	25.9	<u>56.8</u>	<u>69.9</u>	56.2	24.0	<u>53.1</u>	<u>66.1</u>	40.0	25.8	49.2	<u>60.9</u>	<u>26.1</u>	73.3	88.4	92.8	51.2	69.4	85.6	90.5	<u>43.9</u>
BA tracks only, no DLT triangulation																				
4316	26.0	56.3	69.3	57.0	24.0	52.6	65.5	40.8	26.6	48.9	60.5	24.6	66.7	85.1	90.2	37.1	64.8	83.0	88.5	35.8

Table A5: **Full-resolution RoMaV2 scaffold matching.** We rerun scaffold-conditioned methods using full-resolution 4:3 center crops as RoMaV2 matcher inputs. Matches are expressed on the 512×384 grid used for BA/DLT, conditioning, and evaluation. After DLT, visible track observations are rasterized to this grid by rounding their 2D coordinates and averaging collisions. Each cell reports AUC@10 with the change relative to the 512×384 RoMaV2 scaffold. Full-resolution matching strengthens the scaffold most clearly on ETH3D, but the relative trends remain similar, suggesting that the main conclusions are not an artifact of low-resolution matching.

Method	ScanNet++ ₄	ETH3D ₄	Tanks&Temples ₄	4D-DRESS	MV-dVRK
<i>Monocular depth completion</i>					
OMNI-DC [49]	55.6 (-0.5)	63.6 (+6.4)	59.1 (-0.4)	93.6 (+1.0)	87.3 (-0.7)
Marigold-DC [31]	56.8 (+0.5)	66.0 (+6.4)	58.1 (+0.0)	93.8 (+0.7)	<u>88.8</u> (-0.9)
<i>Point-token propagation</i>					
GGPT (VGGT) [5]	63.5 (+0.3)	58.7 (+2.1)	53.8 (-1.2)	92.7 (+0.8)	83.6 (+0.6)
GGPT (Pi3) [5]	<u>67.1</u> (+1.0)	50.5 (+0.5)	51.2 (-0.7)	93.1 (+0.4)	88.3 (+0.4)
GGPT (Pi3X) [5]	65.7 (+0.3)	63.0 (+3.0)	54.8 (-1.3)	89.3 (+1.7)	77.9 (-3.1)
GGPT (MASt3R [†]) [5]	69.5 (+0.5)	62.9 (+4.1)	<u>59.4</u> (-0.6)	64.4 (+0.0)	88.7 (-0.4)
<i>Geometry-conditioned predictors</i>					
MVSAnywhere [12] [§]	40.9 (+0.8)	41.6 (-0.2)	42.6 (-0.2)	77.2 (-0.3)	79.1 (-0.6)
MapAnything [16]	51.2 (+0.8)	42.7 (+3.2)	44.4 (-0.5)	82.5 (-0.4)	78.9 (-0.2)
Pi3X [39] (submission)	62.3 (+0.4)	53.3 (+1.8)	47.5 (-1.4)	90.2 (+0.6)	73.0 (+0.6)
Pow3R [13]	61.9 (+0.2)	<u>68.0</u> (+5.7)	58.5 (-0.8)	93.8 (+0.6)	87.9 (-0.7)
Scaffold3D (ours)	69.5 (-0.1)	71.0 (+4.5)	60.0 (-0.9)	93.5 (+0.6)	90.5 (+0.1)

B Dataset Details

Table B1 reports how much of each dataset is covered by ground truth, how often pixels are visible in one or multiple input views, and how much of the evaluation region is supported by the SfM scaffold. These statistics explain several trends in the main paper. ScanNet++, ETH3D, and Tanks&Temples contain substantial monocular or no-scaffold regions in the four-view setting, leaving room for learned geometry propagation. In contrast, 4D-DRESS and MV-dVRK have high scaffold coverage and few

Table A6: **Ground-truth scaffold points.** We replace the estimated SfM scaffold points by ground-truth points at the same scaffold-supported pixels, while keeping the same cameras and evaluation protocol as Table 1. Each cell reports AUC@10 with the change relative to the estimated-SfM scaffold. This isolates the effect of scaffold point accuracy from the downstream conditioning mechanism.

Method	ScanNet++ ₄	ETH3D ₄	Tanks&Temples ₄	4D-DRESS	MV-dVRK
<i>Monocular depth completion</i>					
OMNI-DC [49]	58.8 (+2.7)	67.0 (+9.8)	59.8 (+0.3)	<u>95.8</u> (+3.2)	<u>94.6</u> (+6.6)
Marigold-DC [31]	59.5 (+3.2)	65.2 (+5.6)	58.8 (+0.7)	X	X
<i>Point-token propagation</i>					
GGPT (VGGT) [5]	63.3 (+0.2)	56.6 (+0.0)	54.5 (-0.6)	91.7 (-0.2)	83.0 (+0.0)
GGPT (Pi3) [5]	66.3 (+0.2)	49.5 (-0.5)	52.0 (+0.1)	92.6 (-0.0)	87.9 (+0.0)
GGPT (Pi3X) [5]	65.6 (+0.3)	59.6 (-0.4)	56.1 (-0.0)	87.9 (+0.3)	80.9 (-0.1)
GGPT (MASt3R [†]) [5]	<u>69.3</u> (+0.4)	59.0 (+0.2)	<u>60.2</u> (+0.3)	64.2 (-0.2)	89.0 (-0.1)
<i>Geometry-conditioned predictors</i>					
MVSAnywhere [12] [§]	39.7 (-0.3)	40.2 (-1.6)	40.8 (-2.0)	77.2 (-0.2)	79.7 (-0.1)
MapAnything [16]	52.4 (+1.9)	27.4 (-12.2)	38.6 (-6.3)	83.4 (+0.5)	79.6 (+0.5)
Pi3X [39] (submission)	61.9 (+0.0)	51.5 (+0.0)	48.9 (+0.0)	89.6 (+0.0)	72.4 (+0.0)
Pi3X [39] (pred. cameras)	69.9 (+2.9)	55.6 (+1.1)	49.7 (-1.5)	91.3 (+1.3)	75.7 (+1.6)
Pow3R [13]	68.3 (+6.6)	<u>69.6</u> (+7.3)	<u>60.2</u> (+0.9)	95.9 (+2.7)	94.3 (+5.7)
Scaffold3D (ours)	74.8 (+5.2)	70.7 (+4.2)	60.7 (-0.2)	95.3 (+2.3)	95.1 (+4.7)

Table B1: **Dataset support statistics.** Views denotes the number of input views. GT is the fraction of pixels covered by the evaluation ground truth. The view-count columns split GT-covered pixels by the estimated number of views they are visible in. S and NS denote pixels with and without SfM-scaffold support, respectively. 4D-DRESS and MV-dVRK have few monocular pixels and high SfM-scaffold coverage, so they leave less room for gains from geometry propagation beyond SfM support. DexYCB-4A/B/C use camera subsets 1–4, 3–6, and 5–8, respectively; DexYCB-8 uses all eight views. DexYCB and EgoBody have much lower scaffold support due to difficult matching and triangulation. **Black** marks the easiest extremes in overlap and SfM-scaffold coverage, whereas **red** marks the most difficult extremes.

Dataset	Views	GT	Visible in						Coverage	
			1	>1	2	3	4	≥4	S	NS
ScanNet++ ₄	4	96.6	<u>24.0</u>	76.0	39.7	27.1	9.2	9.2	30.8	69.2
ScanNet++ ₈	8	96.5	9.1	90.9	19.8	22.6	21.7	48.5	78.8	21.2
ScanNet++ ₁₆	16	96.3	2.9	97.1	6.0	8.4	10.4	82.7	90.1	9.9
ETH3D ₄	4	22.6	26.5	73.5	30.7	32.5	10.4	10.4	52.0	48.0
ETH3D ₈	8	21.5	12.9	87.1	18.7	22.0	17.9	46.4	83.3	16.7
ETH3D ₁₆	16	21.4	3.9	96.1	9.6	13.8	13.8	72.7	92.4	7.6
Tanks&Temples ₄	4	34.6	10.2	89.8	22.6	28.0	39.2	39.2	71.0	29.0
Tanks&Temples ₈	8	36.4	4.7	95.3	7.1	10.6	12.6	77.6	<u>93.5</u>	6.5
Tanks&Temples ₁₆	16	34.6	2.7	97.3	4.6	5.3	5.3	87.4	95.7	4.3
4D-DRESS	4–12	17.4	0.9	<u>99.1</u>	4.4	9.9	16.8	84.8	92.6	7.4
MV-dVRK	4–12	83.5	0.4	99.6	13.2	17.5	17.5	68.8	88.7	11.3
DexYCB-4A	4	56.2	19.6	80.4	32.4	32.3	15.7	15.7	5.6	<u>94.4</u>
DexYCB-4B	4	69.2	21.6	78.4	31.9	38.3	8.2	8.2	5.5	94.5
DexYCB-4C	4	75.5	13.2	86.8	40.2	31.5	15.1	15.1	13.4	86.6
DexYCB-8	8	66.6	7.2	92.8	11.6	13.7	18.9	67.5	7.7	92.3
EgoBody-Exo	3/5	11.7	11.4	88.6	19.5	30.0	16.6	39.1	31.0	69.0

monocular pixels, while DexYCB and EgoBody expose failure modes of the scaffold construction itself.

Table B2: **Training data mixture.** We train Scaffold3D on this training mixture. Dataset selection and approximate ratios are mainly inspired by MAST3R [18]. The table reports the number of preprocessed training samples per source dataset after filtering and the input-view range sampled during preprocessing.

Dataset	Samples	Share (%)	Views
BlendedMVS [42]	27,324	15.2	3–6
MegaDepth [19]	27,267	15.2	3–6
ARKitScenes [2]	21,656	12.1	3–6
OmniWorld-Game [47]	20,480	11.4	3–6
ScanNet++ v2 [43]	18,122	10.1	3–6
Waymo [29]	17,719	9.9	3–6
Habitat [30]	16,959	9.4	3–5
Co3D-v2 [26]	11,445	6.4	3–6
WildRGBD [41]	9,091	5.1	3–6
StaticThings3D [24]	9,089	5.1	3–6
Virtual KITTI 2 [3]	464	0.3	3–6
Total	179,616	100.0	3–6

C Implementation Details

The SfM-conditioned pointmap predictor follows the MAST3R-Large image model configuration [18]: patch size 16, a 24-layer encoder with width 1024 and 16 heads, two 12-layer decoders with width 768 and 12 heads, and DPT-style dense pointmap prediction heads. In point MLP ϕ_{pt} , we use a four-layer ReLU MLP and a three-layer ReLU MLP with width 512 to encode 3D coordinates and within-patch 2D offset separately, then a four-layer ReLU MLP to project both features into a single 768-dimensional feature. The patch point feature sequence is processed by four PointBlock blocks with 12-head self-attention. We freeze the MAST3R image encoder throughout training. We train the remaining parameters with a staged curriculum designed for fast model development. We first train with ground-truth scaffold points sampled from multi-view-overlap regions, then continue with mixed ground-truth and real SfM scaffolds before a six-epoch fine-tuning stage on the 179K-sample mixture in Table B2. For computational reasons, we precompute the SfM scaffolds for all training samples before training starts. We use point-token dropout throughout the SfM-conditioned stages. The final training stage uses batch size 2 per GPU on 8 GPUs, resulting in global batch size 16 and 67K optimizer updates. We train and evaluate on NVIDIA GH200 nodes with four 96GB GPUs per node; the training runs that produced the final Scaffold3D checkpoint used less than 1K GPU-hours, while the full model-development and evaluation effort, including scaffold precomputation, baseline evaluations, ablations, and preliminary or failed runs, used on the order of tens of thousands of GPU-hours. We optimize trainable parameters with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.05, no weight decay for biases and normalization layers). The learning rate is linearly warmed up to $2 \cdot 10^{-5}$ over 1.2 epochs and then follows a half-cycle cosine decay to $5 \cdot 10^{-7}$.

D Runtime

Table D1 breaks the inference time into scaffold construction, the two network forwards, and SfM-anchored global alignment. The network passes are relatively small compared with the final alignment at higher view counts.

E Additional Evaluation Results

Table E1 extends the main results to hand-object and human-centric datasets. These datasets are useful stress tests because their surfaces, masks, sparse-view capture, and texture-poor regions make matching and triangulation unreliable. The results show that geometry conditioning is not automatically beneficial when the upstream scaffold is inaccurate: methods that treat failed scaffold points as reliable degrade, while feed-forward priors are more robust in these cases. Table E2 provides

Table D1: **Runtime breakdown.** Wall-clock timings for Scaffold3D on ETH3D, in seconds. The final SfM-anchored DGA dominates runtime at higher view counts, while the two network forwards are comparatively small. The initial conditioning-free feed-forward pass sets the scale for the injected SfM scaffold points.

Stage	2 views	4 views	8 views
SfM scaffold	4.3	15.8	26.0
Initial conditioning-free forward	0.1	0.6	2.7
Geometry-conditioned forward	0.1	1.3	6.2
SfM-anchored DGA	4.3	11.9	37.7
Total (including SfM scaffold)	9.8	30.6	74.8
Total	5.5	14.7	48.8

the corresponding camera diagnostics for the DexYCB and EgoBody results. Figure E1 visualizes this failure mode on representative DexYCB and EgoBody sequences.

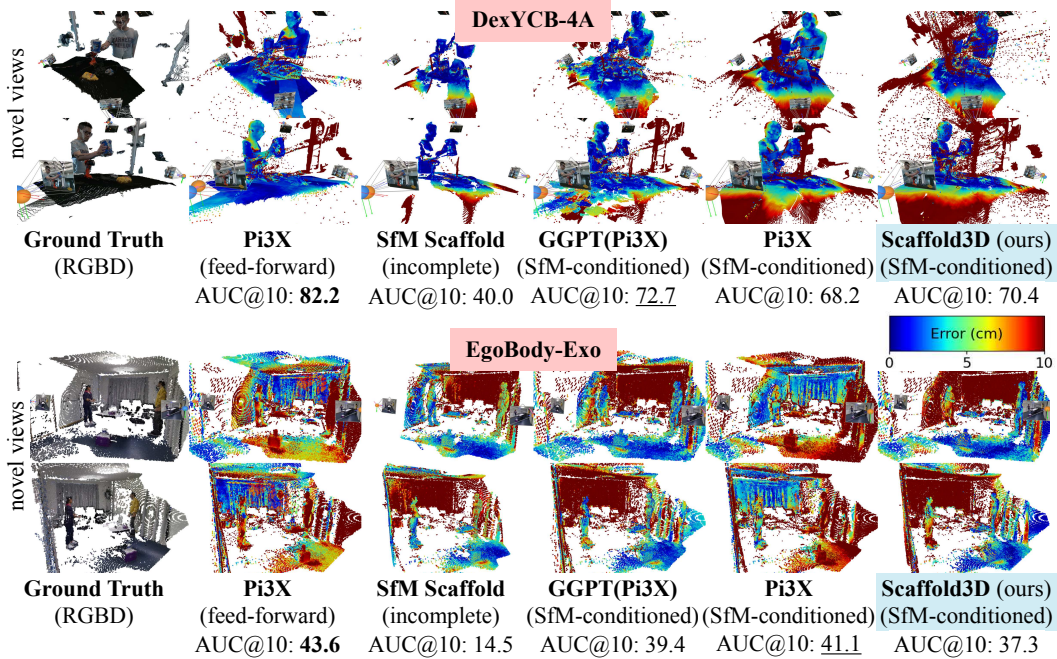


Figure E1: **Failure mode under inaccurate SfM scaffolds.** We visualize DexYCB-4A and EgoBody-Exo examples from two viewpoints each. Columns show ground truth RGB-D geometry, feed-forward Pi3X, the incomplete SfM scaffold, GGPT(Pi3X), Pi3X conditioned on the scaffold, and Scaffold3D. Predicted points are colored by metric 3D error, clipped at 10 cm, where dark blue indicates low error and red indicates high error. On these out-of-domain examples, matching and BA+DLT produce inaccurate scaffold geometry. Methods conditioned on this scaffold inherit and propagate these errors, leading scaffold-conditioned methods to underperform their feed-forward initialization when the upstream SfM result is unreliable.

The main reconstruction tables compare methods under a shared scaffold whenever geometry conditioning is used. Table E3 separates camera quality from dense reconstruction quality: many conditioned rows inherit the same cameras, so differences in point accuracy are primarily due to how methods use the scaffold rather than due to different camera estimates.

Table E1: **Additional out-of-domain datasets.** We report the same AUC@1/5/10/M format as Table 1 on hand-object and human-scene datasets. These out-of-domain datasets contain texture-poor regions and articulated humans, highlighting a limitation of SfM-conditioned reconstruction when matching and BA+DLT produce unreliable triangulated points. Geometry-conditioned methods then degrade because they treat this scaffold as reliable, while strong feed-forward priors are more robust. DexYCB-4A/B/C use camera subsets 1–4, 3–6, and 5–8, respectively; DexYCB-8 uses all eight views. EgoBody-Exo uses 3–5 exocentric cameras.

	DexYCB-4A				DexYCB-4B				DexYCB-4C				DexYCB-8				EgoBody-Exo			
Method	AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M				AUC@1/5/10/M			
	M pixels: 19.6%				M pixels: 21.6%				M pixels: 13.2%				M pixels: 7.2%				M pixels: 11.4%			
Feed-forward predictors																				
MASt3R-SfM [7]	6.6	16.9	22.7	10.1	13.9	29.9	36.9	36.2	33.9	54.0	61.2	35.7	23.4	38.3	45.1	20.5	2.5	18.2	33.5	16.5
Pow3R [13]	12.0	24.7	30.0	10.4	11.1	34.8	46.8	22.6	6.8	33.5	47.7	13.4	7.4	32.2	43.9	6.1	2.5	18.4	34.3	20.3
DUSf3R [37]	14.1	28.4	35.8	14.6	12.8	37.2	47.5	25.8	9.5	39.8	52.0	17.0	9.6	38.0	50.7	9.6	3.2	19.5	34.8	17.8
MapAnything [16]	12.4	43.4	57.9	27.9	17.3	57.1	70.2	41.8	12.6	58.7	72.0	33.0	12.1	54.2	69.0	19.2	2.1	12.4	23.6	21.2
MASt3R [†]	13.3	46.3	59.3	28.7	20.0	53.5	66.5	43.0	25.6	57.1	68.0	31.3	29.6	61.8	73.0	24.1	2.3	19.0	35.1	15.7
MASt3R [18]	17.4	50.2	64.2	33.3	21.9	53.8	66.3	45.3	31.8	64.8	75.2	37.1	30.8	63.6	75.0	32.6	2.7	19.1	34.9	18.0
VGGT [35]	11.1	48.1	65.4	34.9	28.6	68.0	79.8	58.0	17.8	57.8	73.3	37.2	15.6	59.3	75.0	31.1	2.8	22.2	40.3	<u>29.1</u>
VGGT-Omega [35]	25.7	66.7	78.7	52.1	46.7	77.4	84.9	66.2	32.5	71.9	81.7	46.6	38.5	<u>76.2</u>	84.9	<u>45.6</u>	2.8	18.1	34.9	22.1
Pi3X [39]	<u>27.5</u>	68.5	<u>78.8</u>	44.1	15.6	63.9	76.9	58.8	34.5	73.1	82.9	48.1	35.4	75.5	84.8	41.9	3.0	<u>23.4</u>	<u>42.9</u>	29.3
Pi3 [39]	27.6	<u>68.2</u>	78.9	<u>46.3</u>	<u>34.7</u>	<u>73.1</u>	<u>82.4</u>	<u>63.8</u>	42.3	<u>75.9</u>	<u>84.5</u>	<u>51.6</u>	40.9	75.8	<u>85.0</u>	43.0	2.8	23.1	41.8	23.5
Monocular depth completion																				
OMNI-DC [49]	22.8	50.6	62.8	27.2	23.8	52.8	64.8	25.5	30.0	66.3	77.6	38.3	33.7	68.0	78.2	23.3	2.3	16.4	31.4	6.8
Marigold-DC [31]	22.8	51.9	63.7	31.2	26.0	58.2	70.1	38.9	36.0	72.0	81.3	42.8	<u>41.7</u>	72.9	81.3	30.5	2.6	19.3	35.8	11.3
Point-token propagation																				
GGPT (MASt3R [†]) [5]	17.7	50.7	62.8	28.2	25.1	57.8	69.9	39.0	32.0	67.4	76.6	35.3	38.3	70.8	79.3	25.4	2.7	21.2	39.0	16.6
GGPT (VGGT) [5]	20.5	54.6	68.7	36.4	29.2	64.2	75.8	50.8	37.1	73.3	82.7	45.1	38.2	73.2	82.6	36.6	2.6	23.5	43.3	28.4
GGPT (Pi3X) [5]	24.3	59.9	71.3	37.9	27.3	62.9	74.9	50.7	<u>41.2</u>	<u>75.9</u>	84.2	50.4	40.7	74.8	83.3	38.3	2.5	22.9	41.8	25.3
GGPT (Pi3) [5]	26.3	61.5	73.1	41.9	32.5	66.7	77.0	53.4	42.3	76.3	84.7	52.7	42.2	75.8	84.3	42.8	2.0	21.2	40.6	21.8
Geometry-conditioned predictors																				
MVSAnywhere [12]	6.5	20.0	29.6	9.1	14.0	37.6	50.6	30.2	18.9	50.6	64.9	32.1	16.7	43.9	58.2	16.9	2.5	16.0	30.3	11.2
Pow3R [13]	14.6	39.4	51.6	20.8	19.4	44.5	56.5	34.0	29.1	64.1	75.3	40.0	27.0	57.4	68.8	25.7	2.9	20.5	36.6	14.2
MapAnything [16]	17.1	43.5	53.7	20.3	20.7	52.7	64.1	37.2	30.4	59.3	67.8	30.3	36.7	63.7	71.1	19.1	2.2	18.2	34.5	14.9
Pi3X [39] (submission)	17.3	54.1	67.5	35.1	18.4	58.8	71.9	46.1	35.8	73.9	83.1	49.9	26.0	70.2	81.1	40.7	2.6	21.6	39.4	19.2
Pi3X [39] (pred. cameras)	20.7	60.1	73.9	42.7	19.1	66.9	78.9	62.1	33.7	73.7	83.0	49.5	24.5	72.9	83.9	48.1	2.4	19.7	37.3	20.7
Pi3X [39]	23.2	61.2	74.3	43.3	29.1	69.3	80.2	62.4	40.1	75.5	83.9	50.4	39.8	77.2	86.1	49.3	2.8	21.2	38.8	21.6
Scaffold3D (ours)	22.1	53.3	65.7	30.7	30.2	61.8	72.5	48.6	34.8	66.9	76.3	39.5	40.2	70.2	79.3	31.7	<u>3.1</u>	21.2	38.4	21.8

Table E2: **Supplemental camera diagnostics.** We report camera-pose AUC at 1, 5, and 10 degrees and F° , the mean field-of-view error in degrees, for the datasets in Table E1. These diagnostics separate camera accuracy from dense reconstruction quality on the hand-object and human-centric stress tests.

	DexYCB-4A				DexYCB-4B				DexYCB-4C				DexYCB-8				EgoBody-Exo			
Method	AUC@1/5/10/F°				AUC@1/5/10/F°				AUC@1/5/10/F°				AUC@1/5/10/F°				AUC@1/5/10/F°			
Feed-forward predictors																				
DUST3R [37]	0.0	11.0	20.6	1.9	0.0	12.5	34.2	2.6	0.8	21.5	44.4	2.6	0.0	17.3	42.5	2.1	27.5	68.0	82.7	1.1
MASt3R [18]	3.3	54.7	77.2	1.3	1.7	41.5	62.8	1.8	7.5	56.7	76.5	1.5	13.4	64.3	80.5	1.3	38.9	83.4	91.7	0.9
MASt3R-SfM [7]	16.7	22.3	27.5	0.8	23.4	31.3	32.4	<u>1.1</u>	16.7	82.6	91.4	0.5	48.6	58.4	60.6	0.5	36.4	83.2	91.6	0.6
Pow3R [13]	0.0	5.3	10.3	3.6	0.0	7.0	27.8	2.8	0.0	10.2	36.7	2.2	0.0	10.9	27.9	2.5	26.0	74.9	87.2	1.7
MapAnything [16]	5.8	52.5	74.5	1.3	1.7	41.8	69.3	1.4	4.2	58.7	78.5	1.6	2.7	54.3	76.2	1.5	19.6	66.8	83.1	4.2
VGGT [35]	8.3	59.5	78.9	1.1	9.2	66.7	82.9	1.0	2.5	38.0	61.9	2.0	13.4	65.5	82.2	<u>0.7</u>	49.7	83.1	91.5	0.9
Pi3 [39]	26.7	77.8	88.9	1.4	4.2	58.0	78.5	1.0	18.3	67.8	83.8	<u>0.7</u>	15.2	72.0	85.8	0.9	48.9	83.8	91.9	0.8
Pi3X [39]	<u>17.5</u>	<u>73.0</u>	<u>86.5</u>	1.0	6.7	57.8	75.8	1.7	15.8	74.3	87.2	1.8	16.4	76.8	88.4	<u>0.7</u>	42.9	85.1	92.5	0.6
MASt3R [†]	5.0	45.3	69.0	1.3	4.2	36.7	63.5	<u>1.1</u>	5.0	48.5	72.1	1.3	6.6	57.7	77.2	1.4	37.0	79.5	89.8	1.5
Monocular depth completion																				
OMNI-DC [49]	12.5	62.7	80.5	<u>0.9</u>	14.2	73.8	86.9	2.1	46.7	83.2	91.6	1.5	66.8	92.6	96.3	1.0	55.7	86.6	93.3	<u>0.8</u>
Marigold-DC [31]	12.5	62.7	80.5	<u>0.9</u>	14.2	73.8	86.9	2.1	46.7	83.2	91.6	1.5	66.8	92.6	96.3	1.0	55.7	86.6	93.3	<u>0.8</u>
Point-token propagation																				
GGPT (VGGT) [5]	10.0	63.3	81.2	1.1	14.2	70.7	85.3	2.2	52.5	<u>85.3</u>	<u>92.7</u>	1.5	48.2	88.5	94.2	1.0	54.2	86.7	93.3	0.9
GGPT (Pi3) [5]	14.2	60.7	79.0	1.0	14.2	<u>75.0</u>	<u>87.5</u>	2.1	42.5	83.3	91.7	1.7	52.0	89.5	94.8	1.1	<u>56.0</u>	<u>86.8</u>	<u>93.4</u>	<u>0.8</u>
GGPT (Pi3X) [5]	15.0	66.3	82.1	<u>0.9</u>	<u>15.8</u>	75.5	87.8	2.1	<u>51.7</u>	86.0	93.0	1.4	51.2	89.2	94.6	1.0	53.8	<u>86.8</u>	<u>93.4</u>	<u>0.8</u>
GGPT (MASt3R [†]) [5]	12.5	62.7	80.5	<u>0.9</u>	14.2	73.8	86.9	2.1	46.7	83.2	91.6	1.5	66.8	92.6	96.3	1.0	55.7	86.6	93.3	<u>0.8</u>
Geometry-conditioned predictors																				
MVSAnywhere [12]	12.5	62.7	80.5	<u>0.9</u>	14.2	73.8	86.9	2.1	46.7	83.2	91.6	1.5	66.8	92.6	96.3	1.0	55.7	86.6	93.3	<u>0.8</u>
MapAnything [16]	15.0	65.0	81.6	1.1	10.0	71.8	85.9	2.3	45.0	83.7	91.8	1.6	<u>58.0</u>	<u>90.8</u>	<u>95.4</u>	1.4	58.3	87.1	93.6	0.9
Pi3X [39] (submission)	19.2	68.0	83.3	0.9	10.8	73.3	86.7	2.1	43.3	84.7	92.3	1.4	40.9	86.2	93.1	1.0	61.1	88.5	94.2	0.8
Pi3X [39] (pred. cameras)	16.7	67.8	83.5	0.9	9.2	70.2	85.1	2.1	30.8	80.3	90.2	1.5	45.7	88.2	94.1	1.0	61.3	88.5	94.3	0.8
Pi3X [39]	12.5	62.7	80.5	<u>0.9</u>	14.2	73.8	86.9	2.1	46.7	83.2	91.6	1.5	66.8	92.6	96.3	1.0	55.7	86.6	93.3	<u>0.8</u>
Pow3R [13]	12.5	62.7	80.5	<u>0.9</u>	14.2	73.8	86.9	2.1	46.7	83.2	91.6	1.5	66.8	92.6	96.3	1.0	55.7	86.6	93.3	<u>0.8</u>
Scaffold3D (ours)	12.5	62.7	80.5	<u>0.9</u>	14.2	73.8	86.9	2.1	46.7	83.2	91.6	1.5	66.8	92.6	96.3	1.0	55.7	86.6	93.3	<u>0.8</u>

Table E3: **Camera diagnostics.** We report camera-pose AUC at 1, 5, and 10 degrees and F° , the mean field-of-view error in degrees. Most geometry-conditioned rows inherit the same SfM cameras, so the reconstruction gaps in Table 1 cannot be explained by camera accuracy alone. MapAnything receives the same camera conditioning as Scaffold3D, but nevertheless predicts camera updates rather than keeping the input cameras fixed.

Method	ScanNet++ ₄				ETH3D ₄				Tanks&Temples ₄				4D-DRESS				MV-dVRK			
	AUC@1/5/10/ F°				AUC@1/5/10/ F°				AUC@1/5/10/ F°				AUC@1/5/10/ F°				AUC@1/5/10/ F°			
<i>Feed-forward predictors</i>																				
DUST3R [37]	5.8	48.6	67.6	2.8	7.7	39.4	58.3	3.4	15.6	51.8	69.3	2.6	0.0	1.1	8.1	27.2	0.7	16.7	36.9	4.1
MASt3R [18]	19.5	53.3	69.1	3.4	12.8	47.3	61.3	3.2	18.8	59.0	74.4	2.7	0.1	10.3	24.0	4.8	0.3	3.9	19.7	19.9
MASt3R-SfM [7]	40.2	77.6	86.1	1.2	52.6	76.3	81.3	0.5	68.8	90.0	93.4	0.7	39.1	67.5	73.2	1.0	58.8	78.9	85.2	1.9
Pow3R [13]	10.9	55.4	69.4	2.1	9.0	40.8	60.5	3.4	18.8	56.8	71.4	1.8	0.0	2.5	9.7	17.8	0.2	4.3	19.6	18.5
MapAnything [16]	46.5	78.3	86.1	0.6	5.8	33.8	51.5	6.0	22.4	69.5	82.3	3.2	0.0	3.9	16.9	30.9	0.0	2.1	8.5	23.7
VGGT [35]	6.9	40.3	60.5	1.4	26.9	65.0	77.4	2.0	65.1	84.0	88.1	0.6	0.0	9.3	28.9	28.4	0.0	9.2	25.3	14.5
Pi3 [39]	60.9	84.0	89.7	0.7	39.1	67.6	78.4	2.3	55.2	81.8	87.6	0.8	1.7	36.8	66.3	5.2	2.3	30.3	51.5	3.4
Pi3X [39]	55.7	81.4	88.2	0.7	47.4	78.8	87.1	0.9	66.7	86.4	90.5	0.4	0.1	10.6	33.2	23.9	1.2	3.9	18.5	19.2
MASt3R [†]	37.9	74.5	83.6	1.5	8.3	43.1	62.2	3.0	29.2	61.0	74.8	1.5	0.0	0.3	1.3	29.2	2.2	29.0	52.5	2.5
<i>Monocular depth completion</i>																				
OMNI-DC [49]	85.6	92.8	94.1	0.1	91.7	96.3	<u>97.8</u>	0.2	92.7	97.0	97.7	0.4	93.5	96.9	97.3	0.5	91.5	96.3	<u>97.1</u>	0.2
Marigold-DC [31]	85.6	92.8	94.1	0.1	91.7	96.3	<u>97.8</u>	0.2	92.7	97.0	97.7	0.4	93.5	96.9	97.3	0.5	91.5	96.3	<u>97.1</u>	0.2
<i>Point-token propagation</i>																				
GGPT (VGGT) [5]	<u>79.9</u>	<u>91.4</u>	93.4	<u>0.3</u>	87.8	94.5	97.0	0.2	89.1	91.8	92.8	0.4	<u>92.2</u>	93.3	94.4	1.0	76.8	92.0	94.9	0.6
GGPT (Pi3) [5]	<u>79.9</u>	<u>91.4</u>	93.4	<u>0.3</u>	87.8	94.9	97.4	<u>0.3</u>	89.1	93.1	94.2	0.4	93.5	96.9	97.3	0.5	<u>79.8</u>	<u>94.6</u>	97.3	<u>0.4</u>
GGPT (Pi3X) [5]	79.3	91.2	<u>93.5</u>	0.4	<u>88.5</u>	<u>95.9</u>	97.9	<u>0.3</u>	<u>91.1</u>	93.4	94.4	<u>0.5</u>	91.2	<u>96.5</u>	<u>97.1</u>	0.5	77.8	92.6	95.5	<u>0.4</u>
GGPT (MASt3R [†]) [5]	85.6	92.8	94.1	0.1	91.7	96.3	<u>97.8</u>	0.2	92.7	97.0	97.7	0.4	55.2	56.9	57.2	2.8	91.5	96.3	<u>97.1</u>	0.2
<i>Geometry-conditioned predictors</i>																				
MVSAanywhere [12]	85.6	92.8	94.1	0.1	91.7	96.3	<u>97.8</u>	0.2	92.7	97.0	97.7	0.4	93.5	96.9	97.3	0.5	91.5	96.3	<u>97.1</u>	0.2
MapAnything [16]	79.3	88.6	91.4	<u>0.3</u>	73.7	89.0	93.4	0.5	90.1	<u>95.5</u>	<u>96.5</u>	0.6	83.4	94.6	96.2	<u>0.9</u>	62.5	87.6	92.6	0.9
Pi3X [39] (submission)	64.9	84.1	89.5	0.4	72.4	86.2	91.7	0.3	82.8	90.8	92.9	0.5	57.5	89.3	93.5	<u>0.5</u>	9.5	43.3	59.5	0.4
Pi3X [39] (pred. cameras)	66.7	85.3	89.6	0.1	75.0	87.2	92.5	0.2	83.8	93.8	95.5	0.4	59.5	89.7	93.7	0.5	13.4	46.0	61.3	0.2
Pi3X [39]	85.6	92.8	94.1	0.1	91.7	96.3	<u>97.8</u>	0.2	92.7	97.0	97.7	0.4	93.5	96.9	97.3	0.5	91.5	96.3	<u>97.1</u>	0.2
Pi3X [†] [39]	85.6	92.8	94.1	0.1	91.7	96.3	<u>97.8</u>	0.2	92.7	97.0	97.7	0.4	93.5	96.9	97.3	0.5	91.5	96.3	<u>97.1</u>	0.2
Pow3R [13]	85.6	92.8	94.1	0.1	91.7	96.3	<u>97.8</u>	0.2	92.7	97.0	97.7	0.4	93.5	96.9	97.3	0.5	91.5	96.3	<u>97.1</u>	0.2
Scaffold3D (ours)	85.6	92.8	94.1	0.1	91.7	96.3	<u>97.8</u>	0.2	92.7	97.0	97.7	0.4	93.5	96.9	97.3	0.5	91.5	96.3	<u>97.1</u>	0.2