Emergent Extreme-View Geometry in 3D Foundation Models —Supplementary Material—

Yiwen Zhang[1] Joseph Tung[2] Ruojin Cai[3] David Fouhey[2] Hadar Averbuch-Elor[1]
[1]Cornell University [2]New York University [3]Kempner Institute, Harvard University

We refer readers to the interactive visualizations at our project page that show randomly-selected results for all three 3D foundation models (pre-trained and fine-tuned) on the *relative rotation estimation* and the *dense reconstruction* test sets. In this document, we provide details regarding our proposed benchmark (Section 1), additional implementation details (Section 2) and provide additional experiments and results (Section 3).

1. The MegaUnScene Benchmark

We first provide additional details relating to the curation of MegaUnScene (Section 1.1). We then provide information on how we construct our three test sets: UnScenePairs (Section 1.2), UnScenePairs-t (Section 1.3), and UnSceneRecon (Section 1.4).

1.1. Initial Curation

We provide additional details on the benchmark construction pipeline for MegaUnScene. First, we describe our construction pipeline, organized in four subsections: identifying scenes, sparse reconstruction, obtaining depth maps, and ensuring unseen scenes. Finally, we summarize MegaUnScene's overall scene statistics.

Scene identification. We first identify candidate scenes, each corresponding to an image collection, that we want to reconstruct. As mentioned in the main paper, we follow the MegaScenes [19] dataset curation pipeline to find scenes and their corresponding image collection from Wikimedia Commons and its sister site, Wikidata. In order to avoid scene overlap with the MegaScenes dataset, we query Wikidata with different high-level classes than in those used in MegaScenes. We filter out all Wikimedia Commons categories whose names intersect with MegaScenes, as category names are unique. This results in approximately 340,000 candidate scenes. For each scene, we follow MegaScenes and download images from all Wikimedia Commons subcategories with a max depth of four.

Sparse Reconstruction. We then reconstruct each candidate scene with at least 50 images using Doppelgangers++ (DGPP) [24] integrated with MASt3R-SfM [5] as mentioned in the main paper. In this pipeline, MASt3R [10] is used for image retrieval and matching, followed by match pruning using the DGPP classifier with a threshold of 0.8. We use COLMAP [16] for incremental SfM, manhattan-world alignment, and image undistortion. As Internet photos are noisy, not all images in the scene's image collection are registered to a reconstruction; thus we filter again for reconstructions that contain at least 50 images.

Obtaining Depth Maps. We obtain semi-dense depth maps with COLMAP's stereo fusion [17]. As described in MegaDepth [11], multi-view stereo depth maps typically contain artifacts from reconstruction ambiguities, especially on unconstrained internet photos. We clean these depth maps by following the same depth refinement protocol in MegaDepth. The protocol is slightly modified by replacing the PSPNet [27] segmentation model with the more recent SegFormer [25] in the semantic filtering step. For more details, please refer to Algorithm 1 of MegaDepth's supplementary material.

Ensuring Unseen Scenes from MegaScenes. As a post-processing step after reconstruction, we check for image conflicts between MegaUnScene and MegaScenes's base 9M image set (a superset of MegaScenes' 2M image subset that is reconstructed). This is possible as Wikimedia Commons ensures that each image has a unique filename. We only use MegaUnScene reconstructions with no image conflicts for UnSceneRecon, and reconstructions with less than 10% image conflict for both UnScenePairs and UnScenePairs-t. For UnScenePairs, we only select image pairs where neither image intersects with MegaScenes. For release, we note all conflicting images registered to reconstructions.

Ensuring Unseen Scenes from MegaDepth. We also check for scene overlap with MegaDepth [11], a commonly-used training set for 3DFMs that shares MegaUnScene's source of internet imagery. First, we sample ten images from each MegaUnScene scene. Then, we use MegaLoc [2], an image retrieval model, to find the ten closest MegaDepth images to each MegaUnScene query image. We manually inspect all scenes to filter out overlapping

scenes.

Benchmark statistics. From the dataset curation pipeline, we identify 758 reconstructions across 658 scenes with ≥ 50 images and $< 10\%$ image overlap with MegaScenes. Of these, we release 478 reconstructions across 469 scenes to be used for evaluation in our three new test sets: UnScenePairs, UnscenePairs-t, and UnSceneRecon. A breakdown of dataset statistics is provided in Table 1.

1.2. UnScenePairs Test Set

Prior work [3] introduced the wELP ("in-the-wild" Extreme Landmark Pairs) test set curated from MegaDepth [11]. However, all three 3D foundation models that we fine-tuned were pretrained on MegaDepth. To provide evaluation on a distinct data source but in the same camera-centric distribution, we follow the same pipeline to filter image pairs on MegaUnScene.

As introduced in our paper, the pipeline identifies image pairs with negligible translation and predominant rotation using mutual K-nearest neighbor graphs ($K = 5$) constructed from the distances between camera translation. Mutual neighbors ensure that only pairs that are consistently close in translation space are preserved. Each surviving image pair is then assigned an overlap level (Large, Small, and None) with the following algorithm:

Given the relative rotation matrix $\mathbf{R}$ between two cameras and their respective FoVs, the overlap category o is determined by:

$$o = \begin{cases} \textit{Large} & |\gamma| < \frac{\text{fov}_x^1 + \text{fov}_x^2}{4} \wedge |\beta| < \frac{\text{fov}_y^1 + \text{fov}_y^2}{4} \\ \textit{None} & |\gamma| > \frac{\text{fov}_x^1 + \text{fov}_x^2}{2} \wedge |\beta| > \frac{\text{fov}_y^1 + \text{fov}_y^2}{2} \\ \textit{Small} & \text{otherwise} \end{cases} \quad (1)$$

where γ and β denote the relative yaw and pitch angles extracted from $\mathbf{R}$ using Euler decomposition.

After the pipeline, we manually review all selected pairs and removed those affected by motion blur, occlusions, or insufficient geometric structure. The resulting UnScenePairs statistics is shown in Table 1.

1.3. UnScenePairs-t Test Set

As discussed in our paper, we also construct *UnScenePairs-t* from MegaUnScene with the same pipeline but use $K = 50$ mutual nearest neighbors to evaluate performance on pairs with larger camera translations. We then implement correspondence-based verification using Doppelgangers++ [24] + MASt3R-SfM [5]'s reconstructed geometry and checking the verified inlier matches.

Additionally, with larger camera translations included as a consequence of setting $K = 50$, the same scene structure may appear at vastly different scales depending on camera-to-scene distance, where a telephoto lens from afar and a wide-angle lens nearby could capture the same 3D structure at incompatible image resolutions. To ensure scale consistency, we extend the basic FoV threshold with three criteria: (1) *both* horizontal and vertical FoV differences below 15° independently, (2) focal length ratio $\max(f_x^1, f_x^2)/\min(f_x^1, f_x^2) < 2.5$ to catch zoom differences, and (3) image resolution ratio $\max(w_1h_1, w_2h_2)/\min(w_1h_1, w_2h_2) < 3.0$ to prevent sensor size discrepancies from obscuring focal length mismatches.

Finally, similar to UnScenePairs filtering, we also manually inspected UnScenePairs-t and removed noisy image pairs. Statistics are shown in Table 1.

Table 1. **MegaUnScene Statistics.** Statistics for our benchmark and three MegaUnScene test sets: UnScenePairs, UnScenePairs-t, and UnSceneRecon. For UnScenePairs, and UnScenePairs-t, we report the number of image pairs extracted for each overlap level. For UnSceneRecon, we report the number of reconstructions. At the bottom, we summarize the total number of unique MegaUnScene scenes and reconstructions across the three test sets.

			#Pairs			
Subset	# Scenes	K	Large	Small	None	Total
UnScenePairs	451	5	1,855	1,223	776	3,854
UnScenePairs-t	380	50	1,122	520	756	2,398

Subset	# Scenes	# Recons	Notes
UnSceneRecon	96	100	Human-annotated metric scale

Overall	# Scenes	# Recons	Notes
MegaUnScene	469	478	Counts after de-duplication

1.4. UnSceneRecon Test Set

We construct a user interface for human annotators to annotate each reconstruction, as depicted in Figure 1. Annotators are first instructed to visually assess reconstruction quality to see determine if the reconstruction is realistic based on the images; they label the reconstruction "good" or "bad" accordingly. If a reconstruction is good, they are instructed to draw a line on the reconstruction, find the corresponding points on Google Maps in satellite view, and annotate the metric scale (as shown on Figure 1). In practice, we instruct annotators to only label one line to estimate the metric scale. From this process, we label 100 reconstructions across 96 scenes with metric scale annotations, as shown in Table 1.

2. Implementation Details

We first describe how we select backbone layers for fine-tuning (Section 2.1), then outline the construction of the training set (Section 2.2), followed by our training configuration (Section 2.3), and finally provide the full evaluation protocols used across all tasks (Section 2.4).

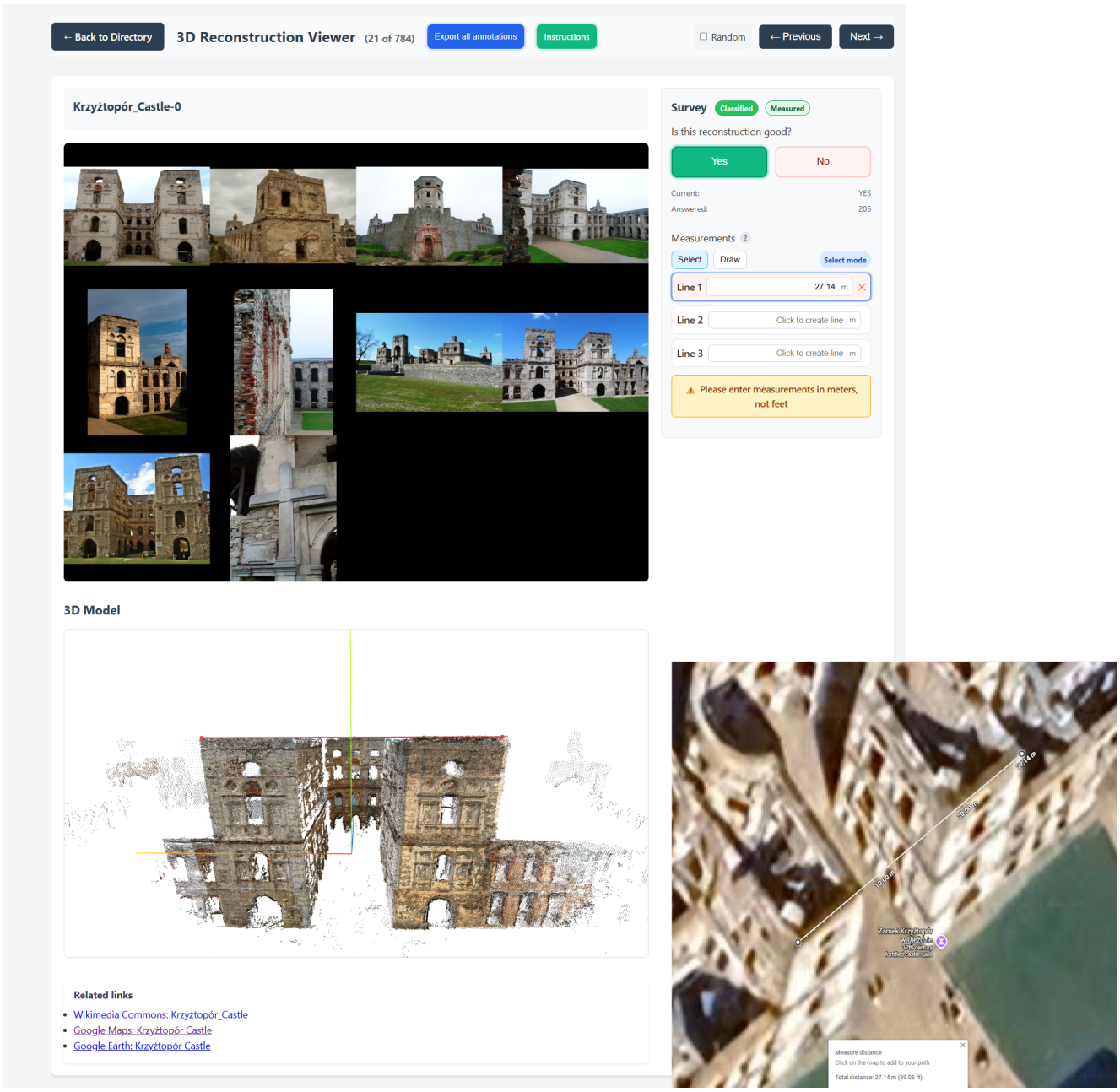

Figure 1. **UnSceneRecon Reconstruction Annotator**. We show the annotator webpage on MegaUnScene's Krzyżtopór Castle scene (left) and its corresponding Google Maps satellite view (bottom right). At the top-left of the viewer, we depict a randomly sampled set of 10 images. On the bottom-left, we show the corresponding 3D model from unprojecting the depths of these 10 images in the global coordinate frame. On the top right of the page, annotators label whether the reconstruction is good or not, and also make a metric estimate of the reconstruction. The metric estimate is done by drawing a line on the reconstruction (shown at the bottom-left, at the top of the building's 3D model in red), measuring the corresponding distance in Google Maps (shown on the bottom right), and pasting the measurement in the corresponding field on the top-right of the viewer. In this example, the annotator labeled "Yes" that the reconstruction is good, and that the annotated red line is 27.14 meters. We provide links to the Wikimedia Commons page, as well as a Google Maps page that searches the scene name, to help annotators identify the correct location on Google Maps.

2.1. Backbone Layer Selection

To quantify the degree of representational change between neighboring backbone layers, we follow the layer similarity pipeline conducted by prior work [7] and run forward passes on ten image pairs, measuring the similarity between the input and output token representations of each layer. The input and output tokens $\mathbf{T}_l^{\text{in}}$ and $\mathbf{T}_l^{\text{out}}$ correspond to either the frame tokens $\mathbf{T}_{\text{frame}}^{(i)}$ or the global tokens $\mathbf{T}_{\text{global}}$, as defined in the method section. For a layer l, we compute the cosine similarity:

$$\text{sim}_l = \frac{\mathbf{T}_l^{\text{in}} \cdot \mathbf{T}_l^{\text{out}}}{\|\mathbf{T}_l^{\text{in}}\| \cdot \|\mathbf{T}_l^{\text{out}}\|}. \tag{2}$$

We run the pipeline on the frame and global attention blocks of VGGT, WorldMirror (WM), and π^3 separately, and the resulting similarity curves are shown in Figure 2. As can be observed in the figure, the curves of WM and VGGT exhibit similar similarity drops, which is expected given that WM inherits both the architecture and weight initialization of VGGT. We also find that the similarity drop regions—where the curve shows a pronounced decline—coincide with the intermediate layers commonly used for dense predictions, namely layers 4, 11, 17, and 23. We therefore adopt this fixed set for these models.

For π^3 which doesn't include such skip connections, we select layers by detecting local minima in the similarity curve (using peak detection on the inverted signal) and expanding around each minimum to include adjacent layers with low similarity scores. This ensures that both the most transformative layers and their contextually relevant neighbors are captured. The selection criterion is defined as:

$$\mathcal{L} = \bigcup_{i \in \mathcal{M}} \left\{ i \pm k : s_{i \pm k} \leq \bar{s} - \frac{\sigma_s}{2},\, k \in [1, \delta] \right\}, \tag{3}$$

where $\mathcal{M}$ is the set of detected local minima, $\bar{s}$ and σ_s are the mean and standard deviation of similarity scores, and δ controls the neighborhood expansion radius. In practice, we use $\delta = 2$. This yields a selected subset which includes frame layers 4, 12–16 and global layers 13–15.

2.2. MegaScenes Train Set

For the train set, we use the same pipeline described in Section 1.2 with $K = 50$ to filter image pairs from scene-level COLMAP reconstructions in MegaScenes [19] and employ a balanced subsampling strategy to ensure uniform pair selection across overlap categories. We first cap each scene at a maximum of 40 pairs to prevent scene-level bias, then subsample to achieve exact balance across the three overlap categories. The final train set contains 64,584 image pairs (21,528 for each overlap category) across 3,284 scenes.

2.3. Training Details

We use the same training configuration for all three models. Training is performed with the AdamW optimizer using a learning rate of 5×10^{-5} and a weight decay of 1×10^{-4}. We train on four NVIDIA RTX A6000 GPUs in a distributed setting with a per-GPU batch size of 1. For each selected layer, we update only the bias parameters in the attention and MLP modules—specifically, the biases of the query–key–value projection (attn.qkv.bias), the attention output projection (attn.proj.bias), and the two MLP fully connected layers (mlp.fc1.bias and mlp.fc2.bias). To stabilize training, we apply gradient clipping with a maximum norm of 1.0 to mitigate gradient explosion.

Additionally, for the ablation checkpoints that unfreezes both weights and biases of a backbone layer or the camera head, we apply LoRA [8] with rank = 24, alpha = 48, and dropout = 0.1 for parameter efficient fine-tuning.

2.4. Evaluation Protocols

In this subsection, we provide additional details on our evaluation protocols and settings.

General Evaluation Settings. For all non-ablation evaluations, the base models we use for VGGT, WorldMirror, and π^3 are the publicly available checkpoints from HuggingFaceHub with the following names:

- `facebook/VGGT-1B`
- `tencent/HunyuanWorld-Mirror`
- `yyfz233/Pi3`

The fine-tuned models are the Layer-Only (LO) and Bias-Only (BO) checkpoints.

Relative Rotation Evaluation Settings. We preprocess input images using each architecture's provided functions: WorldMirror and VGGT crop images to width 518; π^3 proportionally scales with a pixel limit, preserving aspect ratio. All three ensure image width and height are multiples of 14 after preprocessing. The predicted quaternions are converted into rotational matrices for evaluation.

For ExRot [3], we evaluate on their publicly available model on their GitHub page. For all datasets, the images are downsized such that the longer dimension is 256 pixels, then center zero-padded to be size 256x256.

Multiview Pose Estimation, Monocular Depth, and Dense Reconstruction Settings. We also preprocess input images with each architecture's provided function. For multview pose estimation and monocular depth, we follow π^3's [23] protocol and downsize images to a target size of 512 pixels, with dimensions adjusted to be divisible by 14 through rounding. For dense reconstruction, the target size is 518 pixels. As UnSceneRecon is the only dataset with variable aspect ratio, we downsize the longest edge to 518 and center zero-pad to be 518x518 pixels for the dense reconstruction evaluation.

UnSceneRecon Graph-Based Image Sampling for Dense Reconstruction Evaluation. In the main paper, we men-

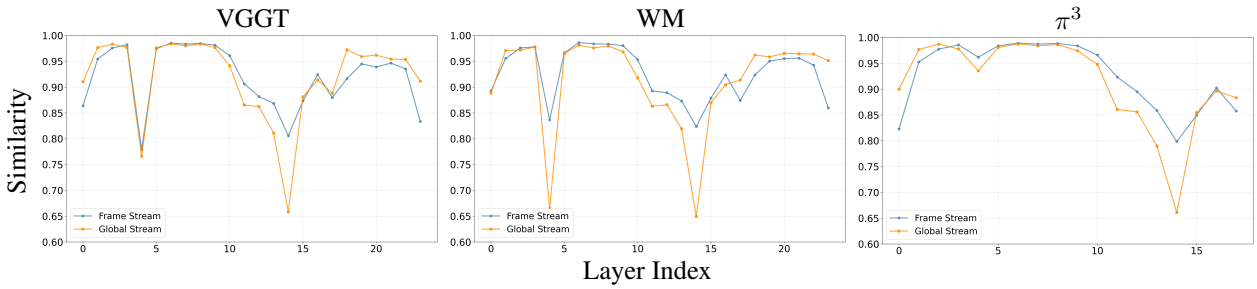

Figure 2. **Layer Analysis.** Cosine similarity curves of the input–output representations for pretrained VGGT, WorldMirror (WM), and π^3. For VGGT and WM, the curves span 24 layers, where each layer corresponds to a frame-global block pair. For π^3, the curve spans 18 layers. Our fine-tuning focuses on layers with pronounced similarity drops; see Section 2.1 for additional details.

tion that we subsample images from UnSceneRecon using a graph-based greedy algorithm for dense reconstruction evaluations. This is because UnSceneRecon scenes typically have widely distributed camera poses that capture different portions of the scene. Random image sampling often leads to poor overlap—*i.e.*, selecting images from disjoint locations on opposite sides of the scene—that are implausible for reconstruction pipelines to reasonably reconstruct. Our graph-based approach ensures connectivity across images while maintaining diversity.

We construct an image connectivity graph for each sparse reconstruction, where nodes represent images and edges connect image pairs with at least 30 shared 3D points and a translation of at least 5 meters. We initialize with a random node in the largest connected component, then greedily sample images based by selecting neighbors that maximize a score combining 80% connectivity and 20% diversity. Here, "connectivity" is the normalized node degree (degree divided by the maximum degree in the graph); "diversity" is the average distance from a candidate to the nodes of all selected images, normalized by the maximum translation across all edges in the graph.

3. Experiments and Results

We begin by evaluating relative camera pose across both overlapping and non-overlapping settings (Section 3.1), then present additional dense reconstruction experiments on multiple benchmarks (Section 3.2), followed by monocular depth evaluations (Section 3.3), and conclude with expanded ablation studies analyzing alternative fine-tuning strategies (Section 3.4).

3.1. Relative Camera Pose

Evaluations on Large/Small Overlapping Pairs. As shown in Table 2, we also evaluate the three fine-tuned models on large and small overlapping image pairs from *s*ELP, UnScenePairs, and UnScenePairs-t. All models achieve comparable, and sometimes slightly improved, rotation accuracy. This demonstrates that our fine-tuning procedure

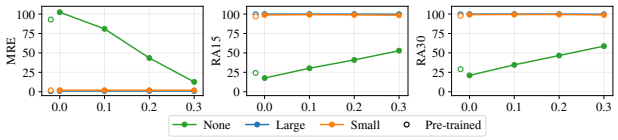

Figure 3. Performance as a function of the fraction of non-overlapping image pairs used for fine-tuning VGGT on sELP. Lower is better for MRE, higher is better for RA15/RA30.

does not compromise performance on overlapping image pairs. Since the pretrained models already produce strong rotation estimates when overlap is present, fine-tuning preserves this capability.

To explicitly examine the trade-off between improving alignment on non-overlapping views and maintaining accuracy on overlapping views, we conduct experiments reducing the fraction of non-overlapping pairs used during fine-tuning (the reported checkpoint uses 1/3 non-overlapping pairs). Fig. 3 shows that increasing this fraction consistently improves performance on non-overlapping cases, while performance on both large- and small-overlapping pairs remains stable (e.g., all models yield RA30 above 99.8% for large-overlapping pairs, even higher than the 99.7% obtained by the pretrained model).

Translation Evaluation on UnScenePairs-t. For UnScenePairs-t, ground-truth relative translations are also available. Following prior work [20], we evaluate translation accuracy using the angular error. Let $\mathbf{t}_{21}$ and $\mathbf{t}^{\star}_{21}$ denote the predicted and ground-truth translation vectors from camera 2 to camera 1. The error is defined as

$$\mathrm{err}_T = \arccos\left(\frac{\left|\mathbf{t}_{21}^{\top}\mathbf{t}^{\star}_{21}\right|}{\|\mathbf{t}_{21}\|\,\|\mathbf{t}^{\star}_{21}\|}\right). \tag{4}$$

As shown in Figure 2, we report the Median Translation Error (MTE) and the Translation Accuracy (TA) at thresholds of 15° and 30°, denoted TA_{15} and TA_{30}. Since our loss only supervises over the predicted rotation, the translation error and accuracy stay the roughly the same after fine-tuning, with small improvement on the non-overlapping pairs for all three models. As discussed in the paper, this

Table 2. Expanded comparison of sELP, UnScenePairs, and UnScenePairs-t benchmarks across VGGT, WorldMirror (WM), π^3, and their fine-tuned variants. MRE and MTE report the median rotation and translation errors in degrees. RA_{15}/RA_{30} and TA_{15}/TA_{30} indicate the percentage of predictions whose rotation or translation errors are below 15° or 30°, respectively.

		sELP			UnScenePairs			UnScenePairs-t					
Overlap	Method	MRE	RA_{15}	RA_{30}	MRE	RA_{15}	RA_{30}	MRE	RA_{15}	RA_{30}	MTE	TA_{15}	TA_{30}
Large	VGGT	0.75	99.7	99.7	0.99	99.7	99.8	1.07	99.9	100.0	2.44	92.7	97.0
	$VGGT_{FT}$	0.95	99.9	100.0	1.04	99.6	99.7	1.08	99.9	99.9	2.24	94.0	97.6
	WM	0.58	99.7	99.7	1.23	96.8	99.7	1.20	99.0	99.7	3.08	87.4	94.3
	WM_{FT}	0.82	99.7	99.7	1.40	98.2	99.5	1.15	99.4	99.7	3.34	88.7	94.6
	π^3	0.68	99.8	99.8	1.24	96.8	99.2	1.09	99.2	99.8	3.39	86.7	94.1
	π^3_{FT}	0.93	99.9	99.9	1.23	99.4	99.8	0.99	99.9	100.0	3.05	88.7	95.9
Small	VGGT	1.86	96.9	97.8	1.93	96.7	98.0	2.01	98.8	99.8	9.69	62.7	74.0
	$VGGT_{FT}$	2.12	98.8	99.4	1.95	97.5	98.8	2.04	100.0	100.0	9.18	61.4	75.1
	WM	1.20	97.7	98.3	2.37	95.4	98.6	1.98	97.7	99.8	11.15	55.8	72.8
	WM_{FT}	1.92	98.1	98.2	2.65	96.5	98.9	2.39	97.9	99.6	11.75	56.4	72.1
	π^3	1.50	97.3	97.7	2.31	94.4	98.5	1.77	98.1	99.8	9.68	60.6	75.3
	π^3_{FT}	2.29	98.4	98.7	2.43	96.7	99.1	1.99	99.0	100.0	10.91	57.5	75.5
None	VGGT	92.92	24.2	29.1	31.64	33.8	48.8	46.65	29.1	42.1	37.28	25.3	42.2
	$VGGT_{FT}$	14.21	50.9	56.5	12.71	53.6	67.9	14.48	50.6	62.1	35.79	26.5	44.0
	WM	68.96	36.3	42.5	19.25	44.1	58.9	21.52	42.6	57.4	33.83	27.6	45.2
	WM_{FT}	9.74	56.9	63.5	11.75	56.2	68.1	13.13	53.3	64.5	33.42	27.8	46.7
	π^3	45.24	43.8	48.3	17.66	46.5	59.4	21.62	43.5	56.8	33.16	29.4	45.7
	π^3_{FT}	11.96	53.7	60.0	12.92	54.0	69.2	13.31	53.1	65.5	32.05	31.4	47.7

suggests that all fine-tuning framework performs minimal updates to the pretrained weights without overfitting to predicting extreme relative rotation.

Supervising over Full Pose. We also experiment with adding a loss term to supervise relative translation in addition to rotation. Each of the three models addresses scale ambiguity differently: VGGT normalizes all camera translations in the training data, WorldMirror provides a normalized camera pose through prior prompting, and π^3 computes a scale factor by aligning its predicted point map with the ground truth. We follow VGGT by normalizing each translation vector using the mean distance from the ground truth sparse 3D points to the point-cloud center, noting that VGGT measures distances to the origin while our dataset does not anchor the first image. Additionally for π^3, we adopt a similar scaling strategy. However, with only two input images, aligning a partial predicted point map to the full ground-truth scene is unstable. We instead apply a simple scaling heuristic. For each image pair i, let $\mathbf{t}^i_{\text{pred}}$ and $\mathbf{t}^i_{\text{gt}}$ denote the predicted and ground-truth relative translations in world coordinate. We compute a scale factor

$$s_i = \frac{\left\|\mathbf{t}^i_{\text{gt}}\right\|_2}{\left\|\mathbf{t}^i_{\text{pred}}\right\|_2}, \tag{5}$$

and rescale the prediction as

$$\tilde{\mathbf{t}}^i_{\text{pred}} = s_i\, \mathbf{t}^i_{\text{pred}}, \tag{6}$$

We add an L1 loss on the translation vectors and keep the same geodesic loss for rotations. For VGGT and WorldMirror, the translation loss anchors the first image as the reference frame and compute the loss on the two absolute predicted translations where the first image's translation should be [0,0,0] in world coordinate. For π^3, we instead compute the loss on the relative translation vectors. Results in Table 4 show that this additional translation supervision provides essentially no improvement in translational or rotational accuracy, and occasionally will lead to worse performances. This illustrates that our proposed rotation-based objective can better align models to extreme-view geometries, in comparison to the full pose objective used by prior work. As mentioned in the paper, our finding also shows

Table 3. **Dense Reconstruction** on the NRGBD [1] dataset in sparse and dense image settings. Non-negligible differences ($>5\%$ relative) between the base and fine-tuned models are in **bold**.

Method	NRGBD-sparse				NRGBD-dense			
	ACC ↓		CMP ↓		ACC ↓		CMP ↓	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
VGGT [21]	0.051	**0.025**	0.066	**0.035**	0.015	0.007	0.016	0.006
VGGT_{FT}	**0.047**	0.029	0.066	0.044	0.015	0.007	0.016	0.006
WM [12]	0.037	0.016	0.042	0.016	**0.014**	**0.007**	0.017	**0.006**
WM_{FT}	0.036	0.016	0.040	0.015	0.016	0.009	0.017	0.007
π^3 [23]	**0.024**	**0.014**	**0.028**	**0.014**	**0.013**	**0.007**	**0.014**	**0.006**
π^3_{FT}	0.029	0.017	0.033	0.017	0.017	0.010	0.015	0.006

that predicting large translational displacements between two images is intrinsically hard and remains to be an important future line of work.

3.2. Dense Reconstruction

Additional Results: DTU, 7Scenes, and NRGBD. We provide additional dense reconstruction experiments on the object-centric DTU [9] and indoor 7Scenes [18] and NRGBD [1] datasets. We follow π^3 [23] and sample every 5th image from DTU (10 images per scene). For 7Scenes, we sample every 40th image, corresponding to π^3's dense-view evaluation setting (16 scenes with 25 images, 2 scenes with 13 images). For NRGBD, we sample every 500th image and 100th image for the sparse and dense-view settings, respectively following π^3. We report accuracy (ACC) and completion (COMP) as in the main paper.

We depict DTU and 7Scenes results in Table 6. As shown on DTU, despite fine-tuning on Internet photos, all our finetuned models are able to generalize to an object-centric dataset: VGGT shows minimal change from fine-tuning, while WM and π^3 show minimal performance loss. Furthermore, on 7Scenes, the reconstruction performance for all models is extremely similar before and after fine-tuning. This is validated by the qualitative results in Figure 4. Overall, our finetuned models still generalize to indoor scenes with minimal impact on dense image inputs.

We separately show the NRGBD sparse and dense setting results in Table 3. VGGT performs nearly identically after finetuning in both settings; WM also performs similarly after finetuning in the sparse setting. For WM in the dense setting and π^3 in both settings, we observe a slight performance decrease in the thousandth place across the metrics. Note that while the 5% bolding threshold is used to remain consistent with other tables, it is extremely sensitive when quantitative values are small in magnitude (*e.g.*, there is a $> 5\%$ difference between π^3 and π^3_{FT}'s median CMP in NRGBD-dense, even if both round to 0.006). The minor deltas in these metrics indicate that there is no significant change in reconstruction quality. Overall, all finetuned models maintain their reconstruction capabilities on indoor NRGBD scenes in both sparse and dense image settings.

Additional Results: UnSceneRecon. We show additional qualitative results on UnSceneRecon in Figure 5; input images and 3D model visualizations for π^3 are shown on the project webpage. These scenes are selected from the 100 reconstructions in UnSceneRecon using a random sampler. As shown, there is negligible difference in reconstruction quality between the base and finetuned models. Note that when the base model reconstructs poorly, the fine-tuned model does too, as exemplified by VGGT and WorldMirror on Kamerlengo-0 and Naubat Khana (Red Fort)-0. The scale alignment of VGGT_{FT} appears much worse than base VGGT on Kamerlengo-0, since automatic alignment has greater variance when aligning incorrect point clouds to the ground truth. These reconstructions demonstrate the difficulty that current 3DFMs have in accurately reconstructing Internet photos, emphasizing the importance of a test set of real-world, unconstrained settings.

Additional Results: RE10K and CO3D. We show qualitative results on the scene-centric RealEstate10K [28] and object-centric CO3Dv2 [15] datasets in Figure 6. Each scene is reconstructed from the same 10 subsampled images as in π^3's [23] camera pose estimation evaluation. For all three models, the finetuned versions's reconstructions largely maintain the same quality as the base versions. One small difference is the CO3Dv2 cake scene (374_42274_84517) for WM, where the top of the cake is slightly noisier. In this case, both the base WM and WM_{FT} models struggle to reconstruct the top of the cake, with the finetuned model amplifying the error. Besides this, our finetuning generally retains each model's reconstruction capabilities, even on object-centric datasets that are out-of-domain from our finetuning data.

3.3. Monocular Depth Estimation

We additionally evaluate monocular depth to determine if fine-tuning alters the performance of the 3DFM's dense prediction heads, as it directly alters the internal representations they decode.

Experimental Details. We test four datasets: Sintel [4], Bonn [14], KITTI [6], and NYU-v2 [13]. Sintel and Bonn contain synthetic scenes; KITTI contains real outdoor driving scenes; NYU-v2 contains real indoor scenes. Following prior work [22, 23, 26], we align each predicted depth to the ground truth with per-frame median scaling. For VGGT and WorldMirror, we directly use the depth head outputs for evaluation. For π^3, since the model does not have a depth head, we obtain the depth by taking the z-values of the model's point map prediction.

Metrics. We report absolute relative error (AbsRel) and threshold accuracy below 1.25 (δ_1) like in [22, 23, 26].

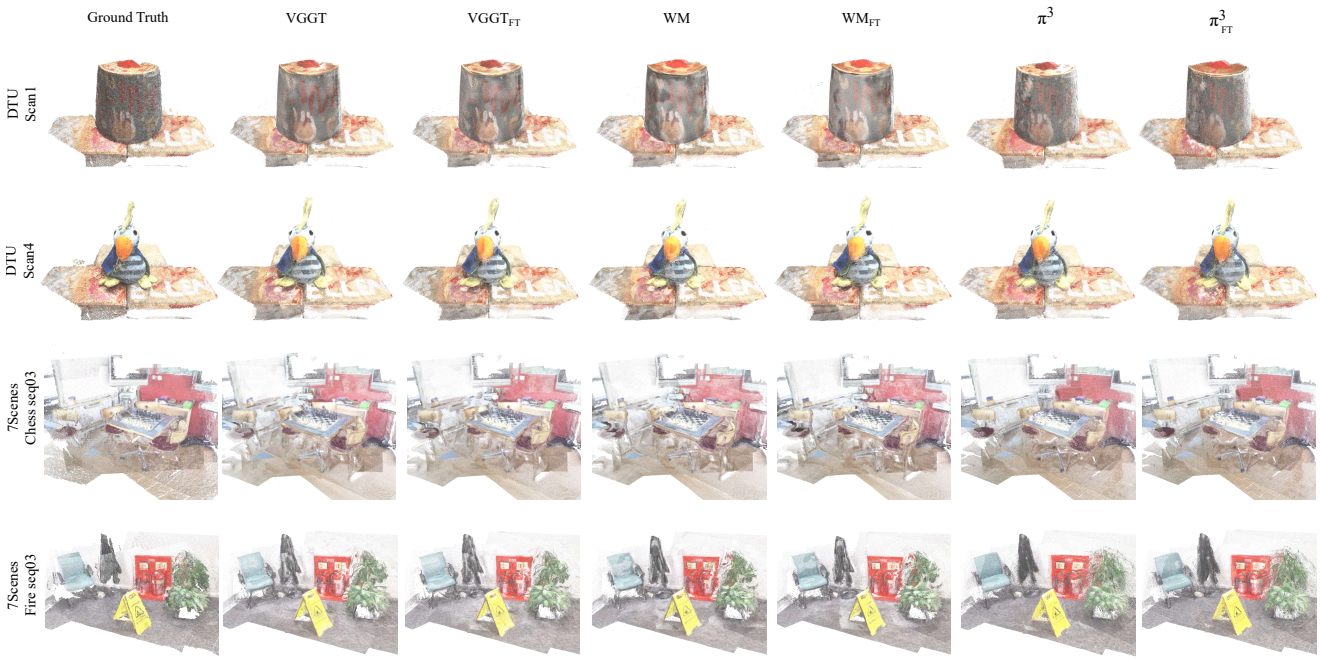

Figure 4. **DTU and 7Scenes Examples.** We show reconstruction results from the base and finetuned VGGT [21], WorldMirror (WM) [12], and π^3 [23] models on DTU's [9] scan1 and scan4 and 7Scenes's [18] chess-seq03 and fire-seq03. The ground-truth reconstruction, obtained using Doppelgangers++ [24] and MASt3R-SfM [5] as further detailed in the text, are shown in the first column. The predicted scenes are automatically aligned to the ground truth per the evaluation protocol discussed in the main paper.

Results. We show monocular depth results in Table 5. Remarkably, all fine-tuned models perform similarly to their base counterparts across all datasets, with VGGT demonstrating minor improvements. This indicates that the frozen dense prediction heads remain effective at decoding the altered internal representations, despite fine-tuning with only rotation loss and no depth supervision.

3.4. Extra Ablations

We show an expanded version of our ablation results in Table 7, with metrics for VGGT [21], WorldMirror (WM) [12], and π^3 [23]. For clarity, we show columns for whether models are trained on select layers only (**LO**) and bias only (**BO**); we additionally denote whether the reconstruction metrics use the point head or not (using fused point maps from unprojected depth) in the **PH** (point head) column. We use the same $\triangle$**REC**$_{\text{PH}}$ and $\triangle$**REC**$_{\text{Fused}}$ metrics as in the main paper's ablation table. We show the same rotation metrics: **MRE**, **RA**$_{15}$, and **RA**$_{30}$, as well as the median reconstruction metrics: **ACC** and **COMP**, as discussed in the main paper. **REC** is the average of **ACC** and **COMP**.

As discussed in the main paper, π^3 performs similarly to WorldMirror (WM) in the ablations: regarding how to finetune the backbone, using select-layers only and bias-only provides a good trade-off between rotation and reconstruction performance (-26.8 for $\triangle$**ROT** and 9.2 for $\triangle$**REC**). Switching the former option to *all layers* or the latter option to *weights and biases* leads to a performance degradation in **REC** (from a 9.2 $\triangle$**ROT** to 14.0 and 15.8, respectively). Interestingly, our fine-tuned π^3 model on all layers and weights and biases exhibits abnormally strong performance (a $\triangle$**ROT** of -37.7 and $\triangle$**REC** of 9.9), and is an outlier in the trends we see across all three models.

We also show π^3's fine-tuning results when unfreezing only the camera head $\mathcal{D}_c$. Unlike VGGT [21] and WM [12], which directly infer 3D points in global space, π^3 uses the extrinsic predictions of the camera head to transform predicted points in local coordinates to global coordinates. Consequently, we see that fine-tuning the camera head leads to worse $\triangle$**REC** metrics (567.4) compared other fine-tuning schemes; this indicates that the performance of $\mathcal{D}_c$ is destroyed. At the same time, we see that $\mathcal{D}_c$ does not have much capacity to align to our target task of extreme rotation estimation, reflected by a negligible $\triangle$**ROT** of 4.1.

3.5. Limitations

A prominent failure mode of the finetuned models, also seen in prior work, is opposite-yaw prediction on non-overlapping image pairs, where the predicted rotation has a similar magnitude but flipped yaw direction (e.g., left 80° vs. right 80°; see Fig. 7). We quantify this over VGGT$_{\text{FT}}$ by checking if flipping the yaw improves the error: on sELP non-overlapping pairs, 26.5% of the cases with error $\geq$30° are substantially improved (error reduction $\geq$90°).

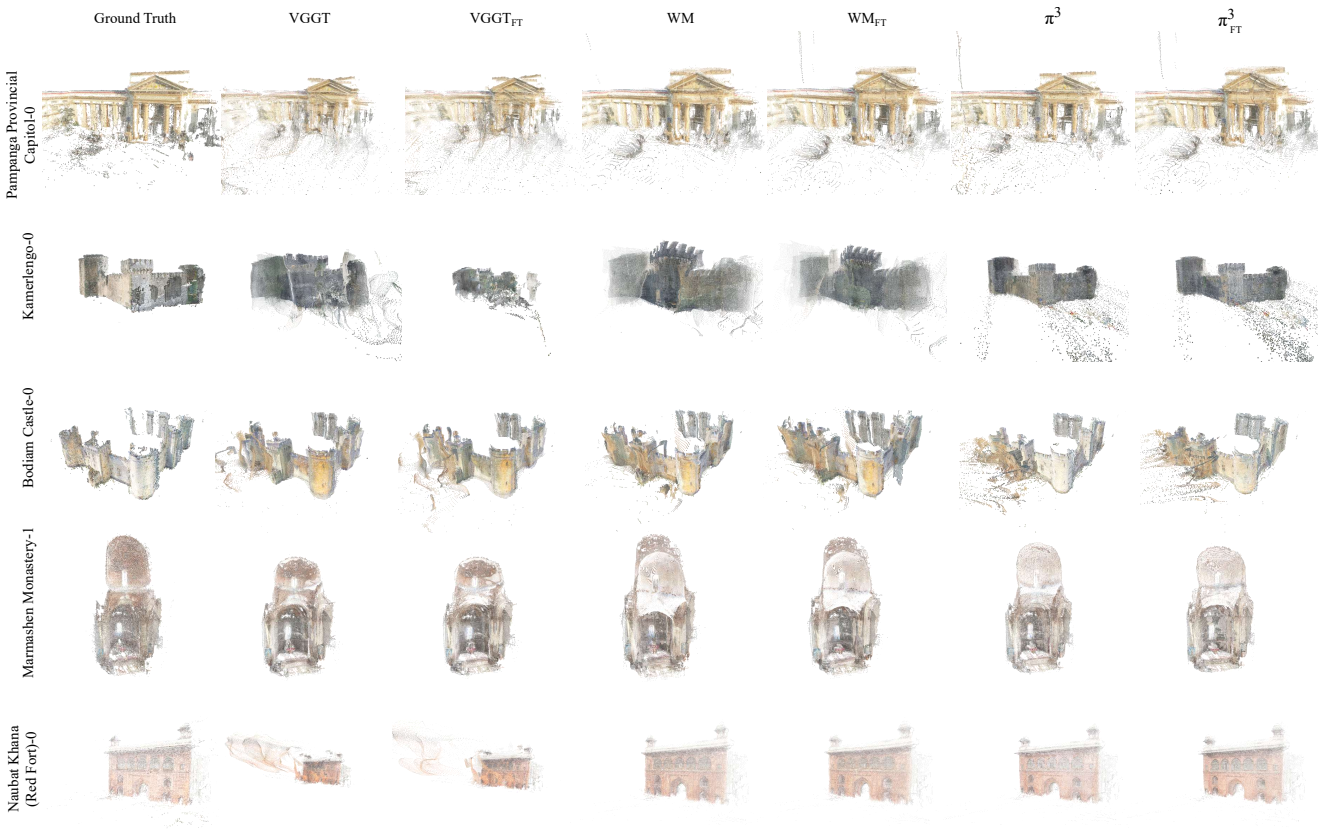

Figure 5. **UnSceneRecon Examples.** We show reconstruction results from the base and finetuned VGGT [21], WorldMirror (WM) [12], and π^3 [23] models on five randomly selected UnSceneRecon scenes. The selected scenes (top to bottom) are Pampanga Provincial Capitol-0, Kamerlengo-0, Bodiam Castle-0, Marmashen Monastery-1, and Naubat Khana (Red Fort)-0. The ground-truth reconstruction, obtained using Doppelgangers++ [24] and MASt3R-SfM [5] as further detailed in the text, are shown in the first column. The predicted scenes are automatically aligned to the ground truth per the evaluation protocol discussed in the main paper.

Pretrained VGGT has a different failure mode: in approximately 15% of non-overlapping pairs, it predicts near-parallel camera poses despite large true rotations. We attribute both failure modes to incorrect cross-view attention. In large-error cases, the model's attention is often diffuse or focused on mismatched regions, unlike successful cases where it concentrates on corresponding areas.

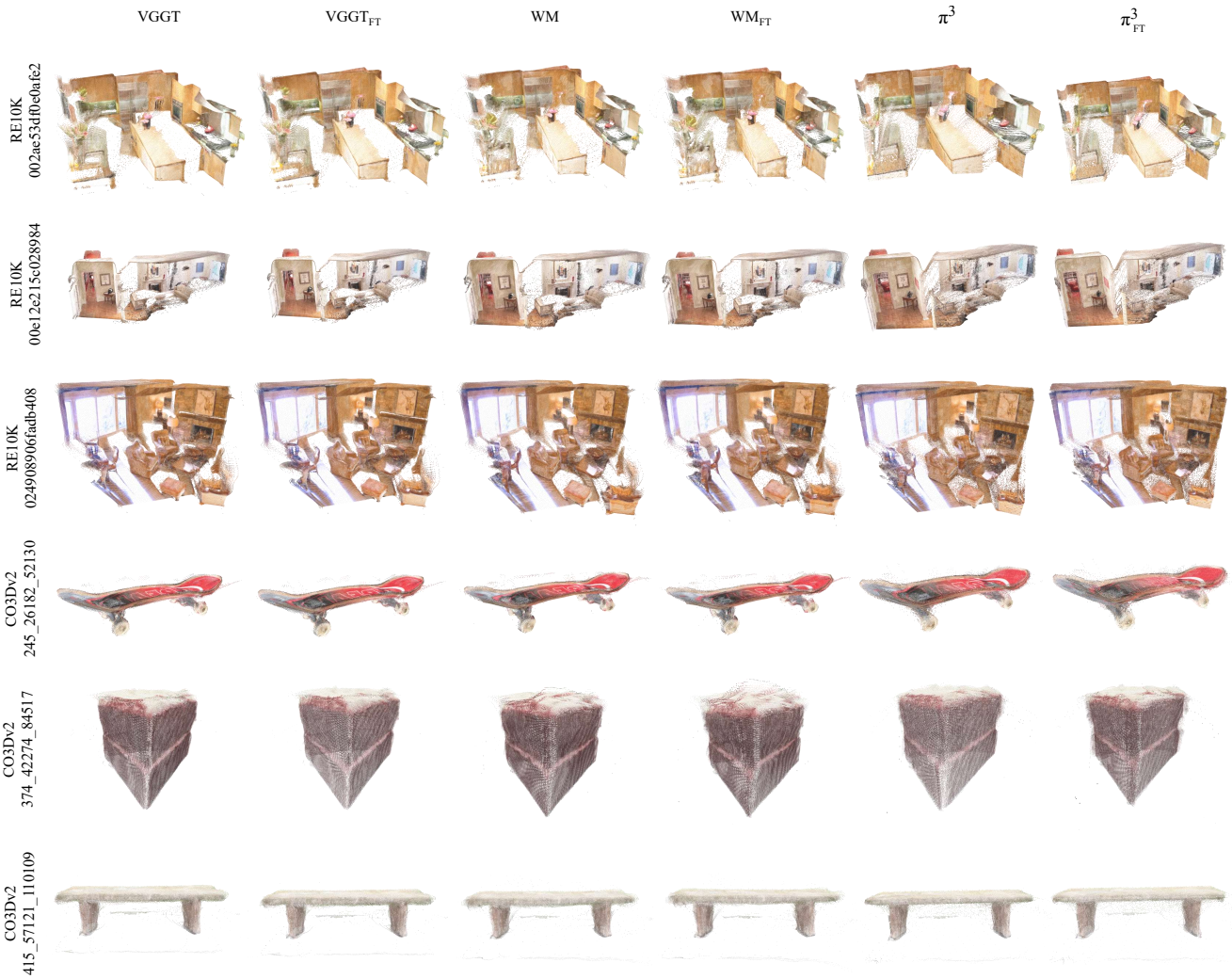

Figure 6. **RealEstate10K and CO3Dv2 Examples.** We show reconstruction results from the base and finetuned VGGT [21], WorldMirror (WM) [12], and π^3 [23] models on three RealEstate10K (RE10K) [28] scenes and three CO3Dv2 [15] scenes. Each scene is reconstructed using the same 10 images as π^3's camera pose estimation evaluation.

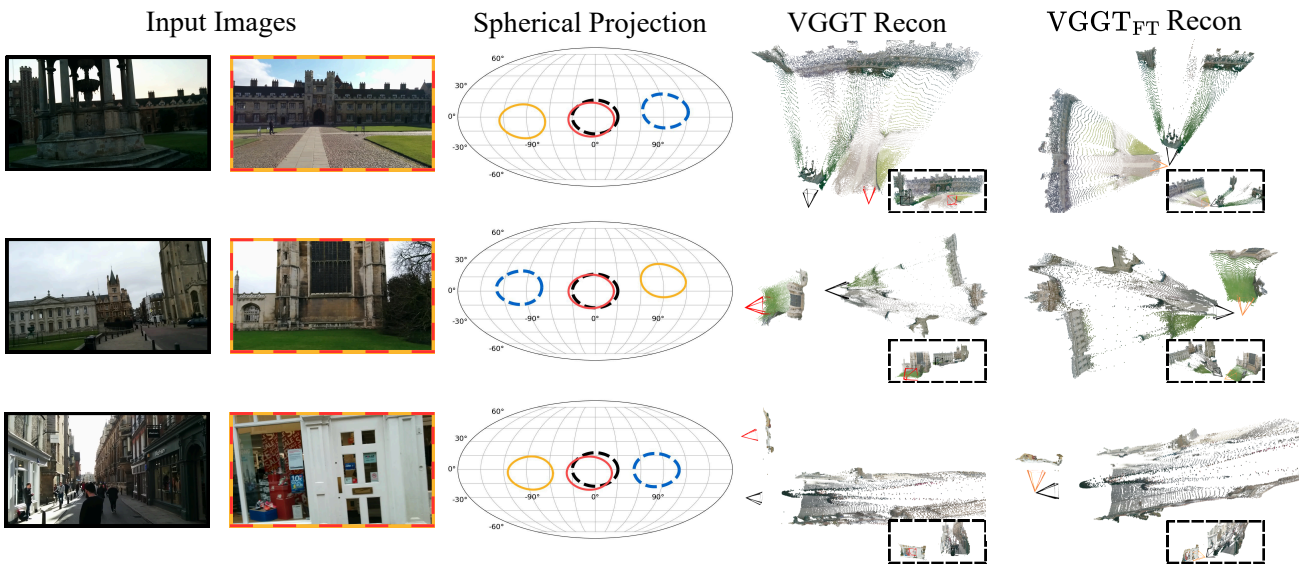

Figure 7. **Limitations**. From left to right, we show the input image pair, spherical projection of relative rotations (black: reference view, blue: ground truth, red: pretrained VGGT, yellow: fine-tuned VGGT), and the corresponding reconstructions (dense pretrained and fine-tuned). These examples illustrate common failure modes: pretrained VGGT often collapses to near-parallel camera poses despite large true rotations, while fine-tuned VGGT predicts an opposite-yaw rotation.

Table 4. Evaluation of fine-tuned VGGT, WM, and π^3 models trained with additional translation loss (TL) on the UnScenePairs-t test set. Comparing with our final fine-tuned checkpoints (FT), the results show that translation supervision offers no improvement in relative translation and rotation accuracy.

	Method	UnScenePairs-t					
		MRE	RA_{15}	RA_{30}	MTE	TA_{15}	TA_{30}
Large	$VGGT_{FT}$	1.08	99.9	99.9	2.24	94.0	97.6
	$VGGT_{TL}$	1.15	99.8	99.8	3.08	93.0	96.9
	WM_{FT}	1.15	99.4	99.7	3.34	88.7	94.6
	WM_{TL}	1.22	99.4	99.7	3.85	88.0	94.2
	π^3_{FT}	0.99	99.9	100.0	3.05	88.7	95.9
	π^3_{TL}	1.52	99.8	100.0	6.93	80.6	92.8
Small	$VGGT_{FT}$	2.04	100.0	100.0	9.18	61.4	75.1
	$VGGT_{TL}$	2.36	99.8	100.0	10.34	59.1	73.6
	WM_{FT}	2.39	97.9	99.6	11.75	56.4	72.1
	WM_{TL}	2.43	97.3	99.8	12.47	54.3	72.1
	π^3_{FT}	1.99	99.0	100.0	10.91	57.5	75.5
	π^3_{TL}	2.38	99.0	100.0	14.71	51.8	70.0
None	$VGGT_{FT}$	14.48	50.6	62.1	35.79	26.5	44.0
	$VGGT_{TL}$	14.75	50.2	61.6	37.75	22.2	39.5
	WM_{FT}	13.13	53.3	64.5	33.42	27.8	46.7
	WM_{TL}	13.36	52.0	63.0	38.50	23.2	40.2
	π^3_{FT}	13.31	53.1	65.5	32.05	31.4	47.7
	π^3_{TL}	15.50	49.0	63.8	33.24	27.8	46.0

Table 5. **Monocular Depth Estimation** on Sintel [4], Bonn [14], KITTI [6], and NYU-v2 [13] datasets. Non-negligible differences ($> 5\%$ relative) between the base and fine-tuned models are in **bold**.

Method	Sintel		Bonn		KITTI		NYU-v2	
	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑
VGGT [21]	0.335	0.597	0.053	0.970	0.082	0.947	0.056	0.951
$VGGT_{FT}$	**0.316**	0.621	**0.050**	0.974	**0.077**	0.952	**0.052**	0.955
WM [12]	0.340	0.625	0.066	0.963	0.093	0.930	0.053	0.957
WM_{FT}	0.336	0.633	0.064	0.964	0.089	0.933	0.053	0.957
π^3 [23]	0.280	0.617	0.048	0.974	0.059	0.971	0.054	0.956
π^3_{FT}	0.287	0.596	0.048	0.974	0.061	0.969	0.054	0.956

Table 6. **Dense Reconstruction** on DTU [9] and 7Scenes [18] datasets. Non-negligible differences ($> 5\%$ relative) between the base and fine-tuned models are in **bold**.

Method	DTU				7Scenes			
	ACC ↓		CMP ↓		ACC ↓		CMP ↓	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
VGGT [21]	1.185	0.716	2.215	1.302	0.020	0.007	0.030	0.014
$VGGT_{FT}$	1.178	0.711	2.189	1.256	**0.018**	0.007	0.029	0.014
WM [12]	**1.033**	**0.573**	1.759	0.790	**0.016**	0.007	0.028	0.013
WM_{FT}	1.121	0.614	1.761	0.783	0.017	0.007	0.027	**0.012**
π^3 [23]	**1.152**	**0.622**	1.797	0.631	0.016	0.007	0.022	0.011
π^3_{FT}	1.396	0.748	1.768	0.634	0.016	0.007	**0.020**	**0.008**

Table 7. **Full Ablation Table.** The full table for the ablation evaluating rotation (△**ROT**) and reconstruction (△**REC**) changes relative to pretrained models. The camera head is denoted as $\mathcal{D}_c$ and the backbone as AA. We show all fine-tuned results for layer-only (LO), bias-only (BO), or both (LO+BO) updates. We denote whether we use the point head or not (using fused unprojected depths) in the **PH** column for reconstruction metrics. **REC** is the average of **COMP** and **ACC**; we report median values only. The reconstruction delta △**REC** is **REC**'s percent change compared to the base model. Significant improvements ($> 10\%$ error drop) for △**ROT** and △**REC** are shown in green, and degradations in red.

Model	Comp.	LO	BO	PH	△ROT	MRE	RA_{15}	RA_{30}	$\triangle REC_{PH}$	$\triangle REC_{Fused}$	REC	ACC	COMP	#Params
VGGT	$\mathcal{D}_c$	×	×	✓	6.8	33.79	28.2	46.4	0.0	(N/A)	0.889	1.049	0.729	216.2M
	AA+$\mathcal{D}_c$	×	×	×	-74.3	8.14	65.5	78.5	(N/A)	90.3	1.692	1.384	2.000	820.9M
	AA	×	×	×	-69.8	9.57	60.4	73.9	(N/A)	-9.2	0.808	0.961	0.654	604.7M
	AA	×	×	✓	-69.8	9.57	60.4	73.9	-33.7	(N/A)	0.590	0.687	0.493	604.7M
	AA	✓	×	✓	-66.7	10.55	57.5	70.8	-33.3	(N/A)	0.593	0.727	0.459	100.8M
	AA	×	✓	✓	-69.7	9.60	61.3	75.6	-16.7	(N/A)	0.741	0.912	0.570	0.4M
	AA	✓	✓	✓	-59.8	12.71	53.6	67.9	-12.4	(N/A)	0.779	0.908	0.650	0.07M
WM	$\mathcal{D}_c$	×	×	✓	-1.4	18.98	43.8	60.7	0.0	(N/A)	0.500	0.612	0.387	216.2M
	AA+$\mathcal{D}_c$	×	×	×	-47.5	10.10	60.7	75.8	(N/A)	81.5	0.907	1.087	0.727	820.9M
	AA	×	×	×	-41.6	11.24	58.0	71.3	(N/A)	13.6	0.567	0.704	0.431	604.7M
	AA	×	×	✓	-41.6	11.24	58.0	71.3	14.0	(N/A)	0.570	0.716	0.423	604.7M
	AA	✓	×	✓	-40.4	11.48	55.9	70.0	18.3	(N/A)	0.591	0.719	0.463	100.8M
	AA	×	✓	✓	-42.3	11.11	56.9	70.6	12.9	(N/A)	0.564	0.702	0.426	0.4M
	AA	✓	✓	✓	-39.0	11.75	56.2	68.1	2.9	(N/A)	0.514	0.660	0.368	0.07M
π^3	$\mathcal{D}_c$	×	×	✓	4.1	18.39	44.0	59.3	567.4	(N/A)	2.812	2.755	2.869	2.1M
	AA	×	×	✓	-37.7	11.00	59.5	73.0	9.9	(N/A)	0.463	0.501	0.425	453.5M
	AA	✓	×	✓	-31.3	12.13	55.0	69.8	14.0	(N/A)	0.480	0.562	0.398	113.4M
	AA	×	✓	✓	-39.6	10.66	60.2	73.8	15.8	(N/A)	0.488	0.555	0.421	0.3M
	AA	✓	✓	✓	-26.8	12.92	54.0	69.2	9.2	(N/A)	0.460	0.517	0.403	0.08M

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, 2022. 7

[2] Gabriele Berton and Carlo Masone. Megaloc: One retrieval to place them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2861–2867, 2025. 1

[3] Hana Bezalel, Dotan Ankri, Ruojin Cai, and Hadar Averbuch-Elor. Extreme rotation estimation in the wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1061–1070, 2025. 2, 4

[4] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, page 611–625, Berlin, Heidelberg, 2012. Springer-Verlag. 7, 12

[5] 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 *International Conference on 3D Vision 2025*, 2025. 1, 2, 8, 9

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 7, 12

[7] Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed, 2024. 4

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 4

[9] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 7, 8, 12

[10] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. 1

[11] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 1, 2

[12] Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting, 2025. 7, 8, 9, 10, 12

[13] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 7, 12

[14] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. *arXiv*, 2019. 7, 12

[15] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 7, 10

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

[17] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[18] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images, 2013. 7, 8, 12

[19] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. *arXiv preprint arXiv:2406.11819*, 2024. 1, 4

[20] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023. 5

[21] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 7, 8, 9, 10, 12

[22] Qianqian Wang*, Yifei Zhang*, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 7

[23] 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, 2025. 4, 7, 8, 9, 10, 12

[24] Yuanbo Xiangli, Ruojin Cai, Hanyu Chen, Jeffrey Byrne, and Noah Snavely. Doppelgangers++: Improved visual disambiguation with geometric 3d features, 2025. 1, 2, 8, 9

[25] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021. 1

[26] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024. 7

[27] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[28] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37, 2018. 7, 10