Long-Tail Internet Photo Reconstruction

Supplementary Material

Contents

Visualization Webpage

Please refer to our project page for additional visualizations beyond this PDF. The webpage includes: (i) animations of our sparsity-aware sampling procedure on representative scenes; and (ii) comparisons of reconstructions from pretrained π^3 and our finetuned π^3 on long-tail scenes (where COLMAP registers 0 images). We also provide video flythroughs of reconstructed point clouds and additional qualitative results on the local webpage to help visualize performance on diverse, real-world scenes.

1. The MegaDepth-X Dataset

1.1. Data Processing

In this section, we compare COLMAP results with those produced by our proposed data-processing pipeline. Fig. 6 shows reconstructions from COLMAP and our MASt3R-SfM pipeline. COLMAP often fails on ambiguous scenes involving similar-looking objects, visually similar but distinct building facades, symmetric landmarks etc. In contrast, our reconstruction pipeline effectively mitigates these issues and recovers correct geometry. In Fig. 7, we show that our monocular depth–guided dense depthmap filtering strategy prevents background depths from leaking into foreground regions (i.e. the depth-bleeding issue [6]) and removes depth estimates on transient objects, which are often unreliable in COLMAP MVS. Note that we use monocular depth only as guidance, rather than warping it to align with the MVS depth. This is because we prioritize *accurate* depth maps over complete ones. Uncertainty in the relative depth predictions of monocular models can introduce additional noise and inconsistency across views. For example, in the last row of Fig. 7, COLMAP MVS fails to recover the depth of the foreground statue, and we opt to remove the depth values in that region. If we were to warp the monocular depth to match the MVS result, then any inaccuracy in the relative depth between the statue and the background building could produce erroneous and inconsistent cross-view depth estimates.

1.2. Dataset Statistics

We provide an overall comparison between MegaDepth and MegaDepth-X in Tab. 1, including reconstruction statistics as well as several metrics that characterize the spatial distribution of viewpoints. Beyond basic dataset properties such as the number of intact reconstructions, image count, and whether doppelganger filtering or dense depth refinement is applied, we analyze how cameras are positioned and oriented in each scene, as scenes with broad viewpoint coverage allow our sampling strategy to construct more diverse and representative sparse-view subsets. The statistics are computed from Manhattan-aligned COLMAP reconstructions.

Positional coverage. To understand how cameras are placed in the horizontal plane, we compute each camera's azimuth angle relative to the scene centroid (that is, the angle of the direction from the scene centroid to the camera) and divide the full 0-360° range into 36 equal 10° bins. In practice, the scene centroid is derived from the average of the SFM point cloud. A scene with many occupied bins is one where cameras are well-distributed around the object. In the table, the columns "Positional Azimuth Coverage $= 100\%$ / $\geq 75\%$ / $\geq 50\%$ / $\geq 25\%$" report how many scenes achieve at least that percentage of bins($36/36$, $27/36$, $18/36$, $9/36$), with larger thresholds indicating closer to full 360° wrap-around coverage.

Rotational coverage. Position alone does not describe where cameras are looking. We therefore measure the coverage of camera orientations by mapping each camera's forward viewing direction to 36 azimuth bins similar to positional coverage. If cameras face more distinct directions, more bins are occupied; if they face similar directions, only few bins are occupied. We summarize this rotational azimuth coverage using the same percentage thresholds as positional azimuth coverage.

These statistics show that MegaDepth-X contains substantially more scenes with broad camera-position coverage

Algorithm 1: One Step of Greedy View Sampling

Input: Current node v
Neighborhood of v: N_v
Set of already sampled nodes S
Community map M (node $\rightarrow$ community)
Camera positions $\mathrm{Pos}(\cdot)$
Output: Next sampled node u^*

```
// Identify communities already covered
```
$S_{\text{comm}} \leftarrow \{M[s] \mid s \in S\}$;
```
// Compute candidate list with
   community novelty and distance
```
$\mathcal{C} \leftarrow \emptyset$;
for ***each*** $u \in N_v$ **do**
  $\mathit{unreached} \leftarrow (M[u] \notin S_{\text{comm}})$;
  $\mathit{dist} \leftarrow \|\mathrm{Pos}(u) - \mathrm{Pos}(v)\|_2$;
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(u, \mathit{unreached}, \mathit{dist})\}$;
end
```
// Sort by unreached, then by distance
```
Sort $\mathcal{C}$ in descending lexicographic order by $(\mathit{unreached}, \mathit{dist})$;
```
// Select the top-ranked candidate
```
$(u^*, \_, \_) \leftarrow$ first element of $\mathcal{C}$;
return u^*;

Algorithm 2: Round-Robin BFS Graph Partitioning

Input: View graph $G = (V, E)$
Number of subgraphs N_{cc}
Output: Subgraphs $\{\mathcal{P}_1, \ldots, \mathcal{P}_{N_{cc}}\}$

Randomly select N_{cc} seed nodes $\{s_1, \ldots, s_{N_{cc}}\} \subseteq V$;
Initialize each $\mathcal{P}_i$ with seed s_i;
Initialize one BFS frontier for each subgraph;
while *there exists a non-empty frontier* **do**
  for ***each*** *subgraph* $\mathcal{P}_i$ **do**
    Expand its frontier by one BFS step;
    Assign each newly reached unassigned node to $\mathcal{P}_i$;
  end
end
return $\{\mathcal{P}_1, \ldots, \mathcal{P}_{N_{cc}}\}$;

and diverse viewing directions, making it better suited for robust sparse-view reconstruction than MegaDepth.

2. Sparsity-aware Sampling

2.1. Greedy Sampling Algorithm

We illustrate one iteration of the greedy view-sampling procedure in Alg. 1. At each step, the algorithm selects the next view based on two criteria:

1. *Community novelty*: prioritizing candidates whose camera-community has not yet been visited by the sampled set. This encourages the trajectory to enter unexplored regions of the view graph and reduces redundancy in viewpoint selection.
2. *Spatial distance*: among candidates with equal novelty, preferring those that are farther from the current camera position. This promotes larger baselines and helps diversify the spatial coverage of the sampled views.

Candidates are lexicographically ranked according to these two criteria, and the highest-ranked node is chosen as the next sampled view.

2.2. Graph Partition

Before sparsity-aware sampling, we partition COLMAP's view graph into N_{cc} subgraphs. Specifically, we randomly select N_{cc} seed nodes and treat each seed as the initial node of one partition. Starting from these seeds, we perform a parallel round-robin breadth-first search(BFS) over the view graph. During each iteration, every subgraph expands from its current frontier to its unassigned neighboring nodes, which are then incorporated into that subgraph. In this way, each node is assigned to the subgraph of the seed that first reaches it, until no further nodes can be expanded.

2.3. Graph Span vs. Search Depth

To understand how greedy search depth D affects the coverage and sparsity of the sampled views, we analyze several statistics on the view-graph. Let G denote the full view-graph of a scene and S the set of sampled nodes. The first two metrics quantify coverage with respect to the *entire* graph G, while the last two measure sparsity *within* the sampled subset S.

k-hop graph coverage. This metric measures how much of the view-graph is reached by the sampled views. Specifically, it computes the fraction of nodes in G that lie within k hops of any sampled node:

$$\mathrm{Cov}_k(G, S) = \frac{1}{|G|} |\{u \in G, v \in S \mid d_G(u, v) \leq k\}|, \quad (1)$$

where S is the subgraph of greedy sampled nodes and $d_G(u, v)$ is the shortest path from u to v on the graph G. A higher Cov_k indicates broader topological coverage, i.e., the sampled set reaches many graph neighborhoods rather than remaining confined to a small region.

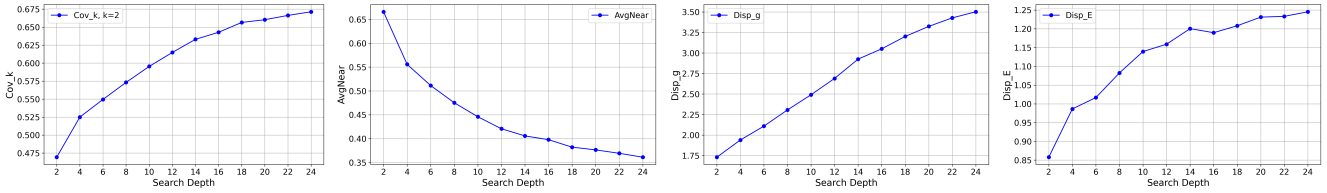
Nearest-sample distance. To evaluate spatial coverage in 3D, we compute the average Euclidean distance from each camera to its closest sampled camera:

$$\mathrm{AvgNear}(G, S) = \frac{1}{|G|} \sum_{u \in G} \min_{v \in S} \|p_u - p_v\|_2, \quad (2)$$

where p_u and p_v are camera positions. Lower values mean the sampled views are spatially well-distributed and lie near many original cameras.

Table 1. **Dataset statistics and viewpoint-distribution metrics.** We report reconstruction statistics and metrics describing camera coverage. *Positional Azimuth Coverage* counts scenes whose camera positions occupy 9–36 (i.e. 25%-100%) of the 36 horizontal azimuth bins (10° per bin, covering the full 360°). *Rotational Azimuth Coverage* represents scenes whose camera forwarding vectors occupy 9–36 (i.e. 25%-100%) of the 36 horizontal azimuth bins (10° per bin, covering the full 360°). For each scene, the more bins covered, the wider the camera distribution is. †Dense depth refinement uses monocular depth–guided filtering.

Dataset	#Recons.	#Images	Doppelganger Check	Dense Depth Refinement	Positional Azimuth Coverage				Rotational Azimuth Coverage			
					= 100% ↑	≥ 75% ↑	≥ 50% ↑	≥ 25% ↑	= 100% ↑	≥ 75% ↑	≥ 50% ↑	≥ 25% ↑
MegaDepth [6]	221	109k	No	Yes	1	7	13	56	14	35	83	196
MegaDepth-X (Ours)	1,865	440k	Yes	Yes†	6	80	223	752	76	490	1123	1816



(a) k-hop Coverage (k=2) (b) Nearest-Sample Distance (c) Graph Dispersion (pairwise hops) (d) Euclidean Dispersion (pairwise distance)

Figure 1. **Coverage and sparsity vs. search depth.** Metrics in (a) and (b) evaluate coverage with respect to the *full* view-graph, while (c) and (d) measure the sparsity of the *sampled* subset. As the search depth increases, the sampled set reaches a larger portion of the view-graph, as shown by the rise in k-hop (graph-distance) coverage in (a). The average distance from each camera to its nearest sampled view decreases in (b), indicating broader spatial coverage. At the same time, both graph dispersion (average pairwise graph distance) in (c) and Euclidean dispersion (average pairwise 3D distance) in (d) increase with depth, showing that the sampled views become more widely separated across the graph and in 3D space.

Graph dispersion and Euclidean dispersion. To understand the sparsity of the sampled views, we calculate the average pairwise distance among sampled views(dispersion) based on graph distances and Euclidean distances:

$$\mathrm{Disp}_{\mathrm{g}}(S) = \frac{1}{|S|(|S|-1)} \sum_{u,v \in S, u \neq v} d_G(u,v), \tag{3}$$

$$\mathrm{Disp}_{\mathrm{E}}(S) = \frac{1}{|S|(|S|-1)} \sum_{u,v \in S, u \neq v} \|p_u - p_v\|_2. \tag{4}$$

Higher dispersion values indicate that the sampled views are more sparsely distributed in both the graph and Euclidean space.

We compute these metrics for the top 100 scenes with the most registered images, evaluating 12 search depths and averaging over 8 sampling runs per depth. The number of sampled views is 24 for all samples. Results are shown in Fig. 1, indicating that deeper searches yield higher coverage on the full graph (a,b) and produce sparser, more widely distributed sampled subsets (c,d).

3. Training Details and Additional Results

3.1. Training Setup

We finetune both π^3 and VGGT using their released pretrained checkpoints. All input images are first padded with white borders to a resolution of 518×518. During training, we apply random crops to these padded images, sampling aspect ratios uniformly from $[0.75, 1.0]$. We also apply random color jittering on training images. Each mini-batch contains up to 24 images drawn from MegaDepth-X, with the number of views per batch randomly selected from $[2, 24]$. We process at most 96 images on each GPU. We also augment image orientations during training by randomly rotating images 90° clockwise or counterclockwise with a probability of 0.2.

We use the original loss functions from π^3 [13] and VGGT [10] to finetune the models. To preserve the geometric priors encoded in the pretrained models, we finetune only the Alternating-Attention modules, while keeping the point-cloud and camera decoders frozen. We further include BlendedMVS [15] and TartanAir [12] as additional training data for finetuning. Finetuning is performed for 100 epochs, where each epoch iterates over all scenes in the combined dataset. We use the AdamW optimizer with a peak learning rate of 1×10^{-5}, scheduled with linear warm-up followed by cosine annealing. All experiments are conducted on 4 NVIDIA A6000 GPUs.

3.2. Additional Depth-Estimation Results

We provide monocular and video depth results to complement the main paper. Following [11, 13, 16], we evaluate Absolute Relative Error (Abs Rel) and the accuracy at a threshold of $\delta < 1.25$. For monocular depth, we report

Table 2. Video Depth Estimation on Sintel [1], Bonn [8], and KITTI [5]. We report Absolute Relative Error (Abs Rel, lower is better) and the prediction accuracy at a threshold of $\delta < 1.25$ (higher is better).

Method	Align	Sintel		Bonn		KITTI	
		Abs Rel↓	$\delta < 1.25$ ↑	Abs Rel↓	$\delta < 1.25$ ↑	Abs Rel↓	$\delta < 1.25$ ↑
π^3	*scale*	0.228	0.671	0.051	0.975	**0.038**	**0.986**
π^3-FT	*scale*	**0.213**	**0.713**	**0.047**	**0.978**	0.040	0.985
VGGT	*scale*	0.294	0.649	**0.055**	**0.971**	0.072	0.965
VGGT-FT	*scale*	**0.242**	**0.707**	0.061	0.969	**0.065**	**0.966**
π^3	*scale&shift*	0.207	0.735	0.045	0.976	**0.036**	**0.986**
π^3-FT	*scale&shift*	**0.188**	**0.739**	**0.043**	**0.978**	0.038	0.985
VGGT	*scale&shift*	0.226	0.683	**0.049**	**0.974**	0.059	0.961
VGGT-FT	*scale&shift*	**0.197**	**0.728**	0.056	0.973	**0.056**	**0.964**

Table 3. Monocular Depth Estimation on Sintel [1], Bonn [8], KITTI [5], and NYU-v2 [7]. We report Absolute Relative Error (Abs Rel, lower is better) and threshold accuracy $\delta < 1.25$ (higher is better).

Method	Sintel		Bonn		KITTI		NTU-v2	
	Abs Rel↓	$\delta < 1.25$ ↑	Abs Rel↓	$\delta < 1.25$ ↑	Abs Rel↓	$\delta < 1.25$ ↑	Abs Rel↓	$\delta < 1.25$ ↑
π^3	**0.277**	0.621	0.052	0.971	0.059	**0.972**	0.054	0.956
π^3-FT	0.284	**0.629**	**0.049**	**0.977**	**0.056**	**0.972**	**0.052**	**0.958**
VGGT	0.331	0.600	**0.051**	**0.974**	**0.089**	0.939	0.055	0.953
VGGT-FT	**0.311**	**0.628**	0.056	**0.974**	0.092	**0.941**	**0.053**	**0.955**

performance on Sintel [1], Bonn [8], KITTI [5], and NYU-v2 [7]. For video depth, we evaluate on Sintel [1], Bonn [8], and KITTI [5] under both *scale* and *scale&shift* alignment settings. Our finetuned models maintain competitive performance across all datasets, demonstrating that the adaptation to in-the-wild imagery does not degrade their depth-estimation ability.

3.3. Results on Doppelganger Scenes

Doppelganger cases often cause both classical SfM pipelines and pretrained feed-forward models to fail, merging distinct structures into a single incorrect reconstruction. As shown in Fig. 2, our fine-tuned π^3 model correctly distinguishes visually similar but distinct structures within each landmark and recovers geometry consistent with reference aerial imagery, indicating improved reconstruction of global scene layout.

To evaluate the effectiveness of different sampling strategies on doppelganger scenes, we evaluate the pretrained π^3 and finetuned π^3 on doppelganger scenes and show results in fig.3. Results indicate that pretrained models and dense-only fine-tuning are less robust to ambiguity, while finetuning with sparsity-aware sampling (e.g., mixed or sparse) tends to improve disambiguation, suggesting sparsity-aware sampling helps.

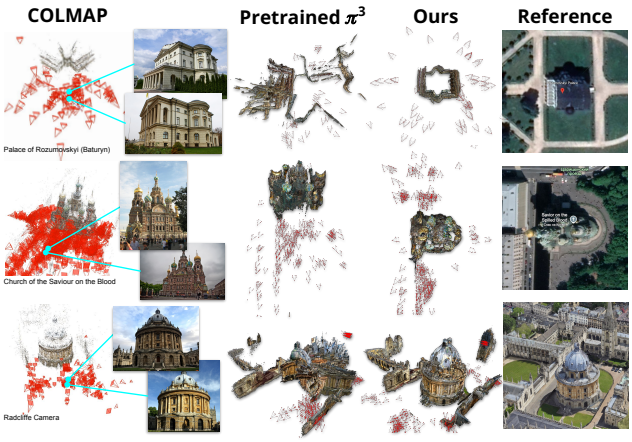


Figure 2. **Disambiguation of doppelganger scenes.** Each example shows a pair of visually similar structures that cause classical SfM (COLMAP) and pretrained π^3 to collapse into incorrect or merged reconstructions. In contrast, our finetuned model correctly distinguishes the symmetric or repetitive sides of the same building, reconstructing consistent geometry for each viewpoint. Reference views from Google Earth are provided for comparison, confirming that our model resolves these ambiguities and recovers accurate global structure under challenging visual similarity.

3.4. Quantitative results on Long-tail scenes

To enable quantitative evaluation on long-tail scenes, we augment MegaScenes with additional observations from external cultural heritage datasets [2, 3, 9] and jointly register all images using COLMAP. The quantitative and qualitative

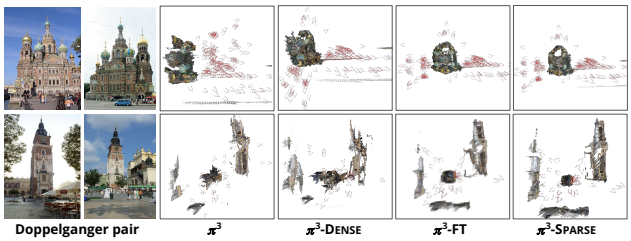


Figure 3. **Comparison of ablated models on doppelganger scenes** We show predictions from the pre-trained model and ablated models on two doppelganger scenes. Disambiguation behavior holds across fine-tuned variants with sparsity-aware sampling, while the pre-trained model and model finetuned with densely sampled views are less robust to doppelgangers.

results of this long-tail evaluation are shown in Fig. 4. Our model consistently reduces the mean relative rotation and translation errors across all scenes, while also producing more complete point clouds.

3.5. Limitations

Long-tail scenes often contain fragmented viewpoints, where different subsets of images capture disjoint parts of the scene (e.g., indoor and outdoor areas) without overlapping views to connect them. When such mixed collections are fed into the models at once, both pretrained and finetuned π^3 may blend these unrelated regions into a single 3D structure, as illustrated in Fig.5. While our finetuned model handles these mixtures more robustly than the pretrained baseline, enabling the model to reason robustly about disconnected components and produce reasonable overall layouts still remains a challenge.

References

[1] 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. 4

[2] Filiberto Chiabrando, Loren Clark, John Driscoll, Scott McAvoy, Dominique Rissolo, Alessandra Spreafico, and Beatrice Tanduo. Salvation mountain - photogrammetry - terrestrial, photogrammetry - aerial, lidar - terrestrial, lidar - mobile, survey data, 2023. Distributed by Open Heritage 3D. 4

[3] CyArk. Great mosque - kilwa kisiwani - lidar - terrestrial, photogrammetry - terrestrial, photogrammetry - aerial, 2020. Distributed by Open Heritage 3D. 4

[4] 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. 7

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 4

[6] 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, 3

[7] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 4

[8] 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. 4

[9] Ashley Richter, Michael Hess, Vid Petrovic, Falko Kuester, Cultural Heritage Engineering Initiative (CHEI), Architecture Center of Interdisciplinary Science for Art, and Archaeology (CISA3). Torre dei baldovinetti - florence - lidar - terrestrial, photogrammetry - terrestrial, 2023. Distributed by Open Heritage 3D. 4

[10] 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. 3

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

[12] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 3

[13] 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. 3

[14] Yuanbo Xiangli, Ruojin Cai, Hanyu Chen, Jeffrey Byrne, and Noah Snavely. Doppelgangers++: Improved visual disambiguation with geometric 3d features, 2025. 7

[15] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020. 3

[16] 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. 3

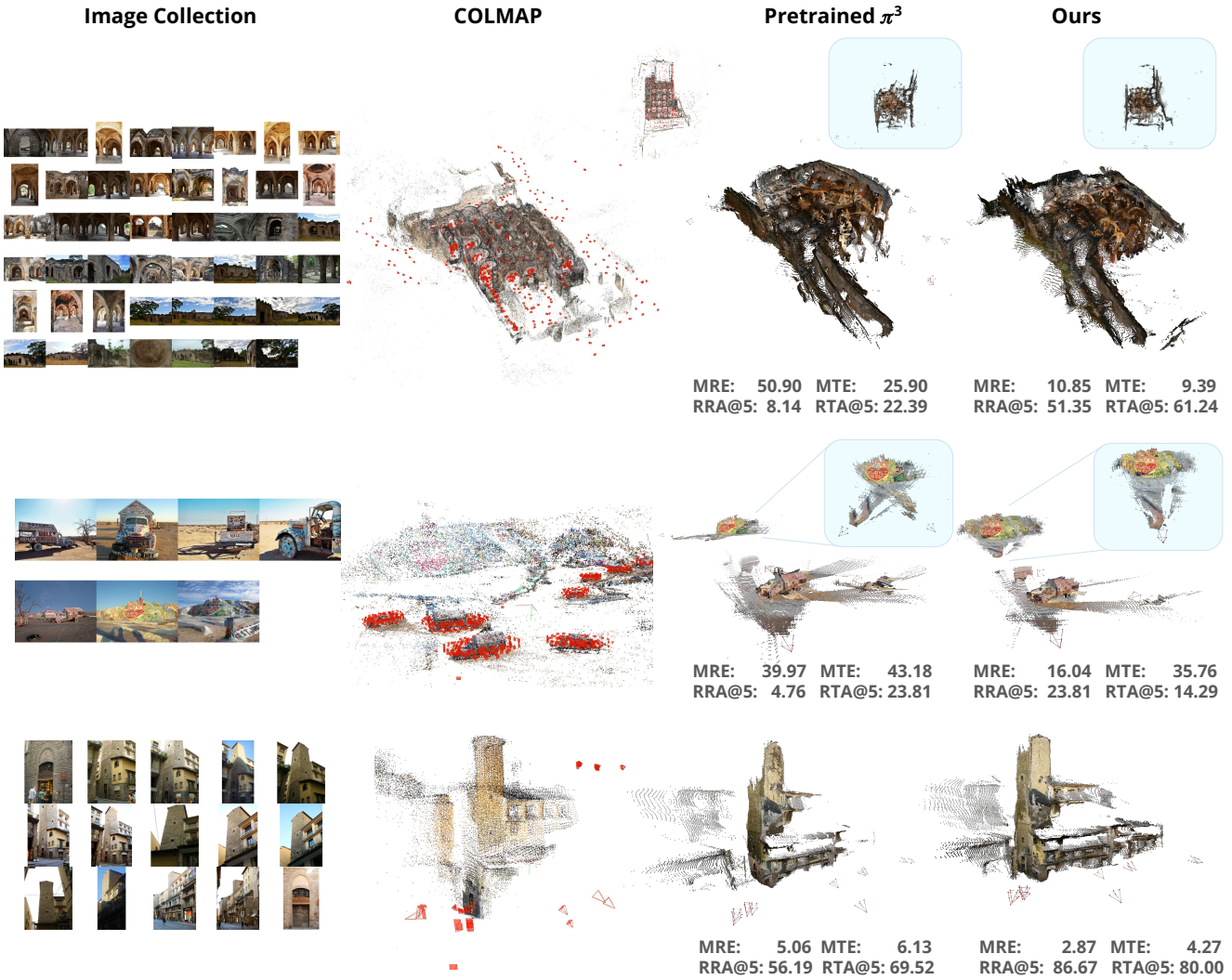


Figure 4. **Quantitative results on Long-tail scenes.** Our model performs better on scenes with strong ambiguities (first row) and on scenes with minimal overlap across different scene components (second row). For a more densely photographed scene that still exhibits large viewpoint variation (third row), our model not only reduces pose error but also reconstructs a more complete point cloud.

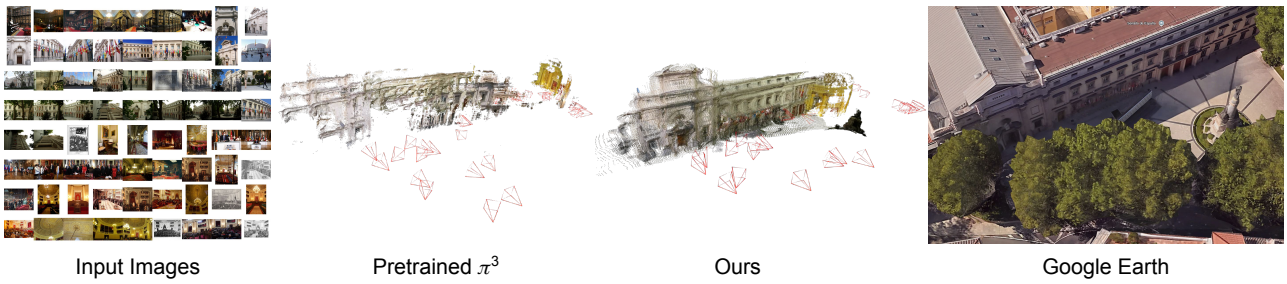


Figure 5. **Limitations.** This example contains images from two disjoint parts of the scene: indoor photos with warm lighting (producing a yellowish point cloud) and outdoor photos (producing a white point cloud). Pretrained π^3 struggles to handle such mixed inputs and produces inconsistent geometry. Our finetuned model is more robust in this setting, but both models still fuse the indoor and outdoor structures into a single reconstruction without separating them.

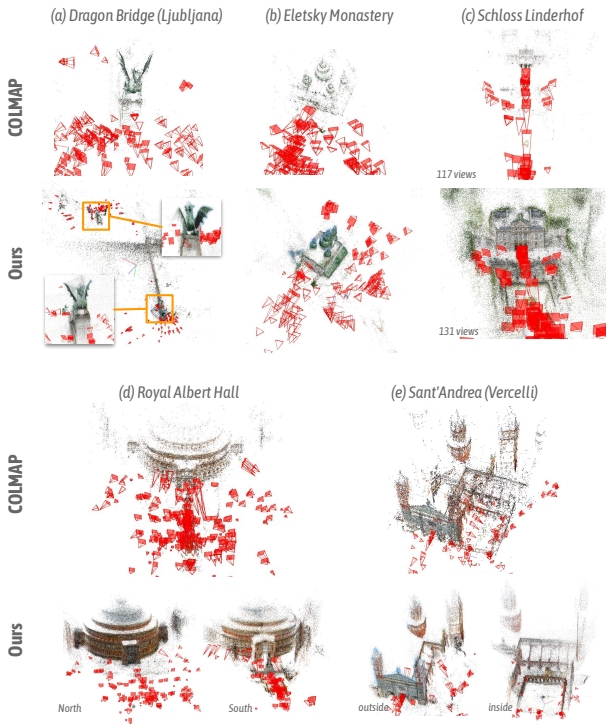


Figure 6. **Comparison of COLMAP and our reconstruction pipeline.**We replace COLMAP with MASt3R-SfM [4] combined with the doppelganger++ classifier [14] to obtain sparse reconstructions, allowing effective disambiguation of doppelganger scenes. (a) The bridge has two similar dragon statues, one at each end. COLMAP incorrectly treats them as the same statue and registers them together, whereas our method correctly separates them. (b), (d), and (e) illustrate additional doppelganger cases, in which different sides or parts of a landmark are mistakenly merged. (c) In this low-texture scene, our pipeline also succeeds in registering more images.

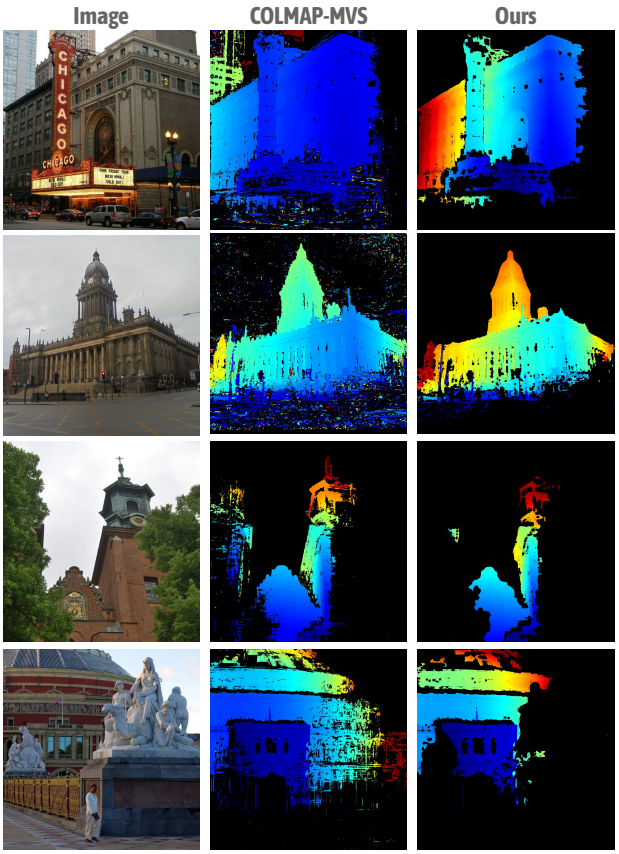


Figure 7. **Comparison of COLMAP MVS and our filtered dense depth results.** COLMAP MVS suffers from depth bleeding and struggles to correctly estimate the depth of transient objects. Our strategy mitigates these issues by leveraging ordering priors from monocular depth predictions. Note that we prioritize *accurate* depth maps over complete ones. In the last row, COLMAP fails to recover the depth of the foreground statue, and we opt to remove the depth values in that region. If we were to warp the monocular depth to match the MVS result, then any inaccuracy in the relative depth between the statue and the background building could produce erroneous and inconsistent cross-view depth estimates.