Uni4D: Unifying Visual Foundation Models for 4D Modeling from a Single Video

Supplementary Material

A. Additional Qualitative Results

We provide extensive qualitative results of Uni4D and other baselines on all datasets in the attached webpage (accessed by `index.html`). We ran MonST3R using their provided hyperparameters from their respective official codebase on all datasets. In our qualitative comparison, we use the *estimated* dynamic masks from MonST3R. This ensures a fair comparison, as the qualitative results for ALL competing algorithms, including ours and all the baselines, do not use privileged information. We generate dynamic masks from CasualSAM by thresholding its uncertainty prediction, using their estimated video depth maps and camera pose to output 4D reconstruction. For Uni4D, we use the same set of hyperparameters throughout our pipeline for all videos for each respective dataset.

All reconstructions are performed with depth estimates resized back to original input resolutions, and with background point clouds downsampled 5 times for efficiency using uniform downsampling. We render final (point-cloud) reconstructions using Open3D, manually picking similar viewpoints for all methods since the reconstructions are neither axis nor scale aligned. We provide visualizations for DAVIS [8], Sintel [2], TUM-dynamics [11], Bonn [7], and KITTI [3], including failure cases. We include sampled frames of our visualizations in Fig. 5, 6, 7, though we strongly encourage viewing the attached webpage for the best visualization experience of our results.

B. Quantitative Evaluation Procedures

For all quantitative evaluation results of pose and video depth maps, we follow MonST3R's evaluation script. We ran all of our baselines using their official codebase and default hyperparameters on all datasets. We use the same depth map alignment, based on least squares in disparity space, for all our depth map evaluations. This is slightly different from the evaluation in MonST3R, where after confirming with MonST3R author, different alignment methods were used for different baselines. This accounts for the different quantitative results in our study and MonST3R's for overlapping baselines (Particularly, we found that CasualSAM [13] and DepthCrafter [5] achieves better reported performance than in the MonST3R paper (see Table 2 in main paper)).

C. Runtime Breakdown

Figure 1 presents a detailed runtime breakdown of Uni4D's preprocessing and optimization stages. Runtimes are averaged across videos of the same length from our evaluation datasets. The reliance on foundation models significantly contributes to the preprocessing time, particularly due to Unidepth [10] and CotrackerV3 [6]. Stage 1 initialization, which estimates poses from scratch, accounts for the majority of the optimization runtime. Overall, runtime scales linearly. Further improvements through advanced optimizers and parallelization are left for future work.

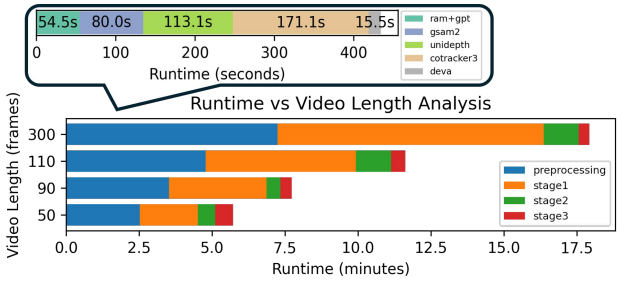


Figure 1. **Runtime Breakdown of preprocessing and optimization**

D. Densification Details

During fusion, we wish to densify the sparse depth obtained from our point trajectories to obtain full-resolution depth maps. Naively interpolating our projected depth in image space leads to poor results, especially across edges and boundaries. Fortunately, flickers in predicted depth maps are usually constant across each scene element. Using this observation, we perform a scale interpolation derived in 3D to obtain a scaling correction $s(\mathbf{x})$ for pixel coordinates $\mathbf{x}$ for every pixel in the depth maps using the following interpolation formula:

$$s(\mathbf{x}) = \sum_{\mathbf{p_i} \in \mathbf{n}(\mathbf{x})} \mathbf{w_i} \frac{z(\mathbf{p_i}, \xi_\mathbf{t})}{\mathbf{D_t}(\pi_\mathbf{K}(\mathbf{p_i}, \xi_\mathbf{t}))} \tag{1}$$

where $\mathbf{n}(\mathbf{x})$ are the 3 nearest point trajectories in 3D of the unprojection of $\mathbf{x}$, $\pi_\mathbf{K}^{-1}(\mathbf{x}, \xi_t)$. $\mathbf{w_i}$ is simply $\frac{1}{d_i}$ where d_i is the euclidean distance between unprojection of $\mathbf{x}$ and each corresponding $\mathbf{p_i}$. $z(\mathbf{p_i}, \xi_\mathbf{t})$ returns the z-component of $\mathbf{p_i}$ after transforming to camera coordinates at time $\mathbf{t}$, and $\mathbf{D_t}()$ returns the depth value from our estimated video depth at the given pixel coordinate at time $\mathbf{t}$. We get our final depth value at pixel $\mathbf{x}$ through $s(\mathbf{x}) \cdot \mathbf{D_t}(\mathbf{x})$. Note that our interpolation is tracklet-aware and searches for nearest neighbors within our preprocessed dynamic object masks. Intuitively, this performs depth map alignment by aligning the original temporally inconsistent depth predictions with our point trajectories to achieve consistent and stable video

Method	Sintel		
	ATE ↓	RPE trans ↓	RPE rot ↓
Uni4D (Metric3D [4])	0.135	0.033	**0.347**
Uni4D (Depth-Pro [1])	0.143	**0.032**	0.451
Uni4D (Depthanythingv2-outdoor [12])	0.112	0.040	0.556
Uni4D (Unidepth)	**0.109**	**0.032**	**0.347**

Table 1. **Performance with different depth models.** We evaluate pose estimation performance on Sintel using different metric depth estimation models.

depth.

E. Depth Model Ablation Study

A key strength of Uni4D is that its modular pipeline allows for the interchangeability of its underlying pre-trained components. We try different depth estimation models and evaluate their pose and depth estimation results on the Sintel [2] dataset in Tab. 1. We find that currently, Unidepth [9] provides the best results.

F. Ablation on tracker and segmentation choice

Method	ATE↓	RPE-t↓	RPE-r↓	AbsRel↓	$\delta_{1.25}$ ↑
Uni4D (TAPIR)	0.131	0.048	1.483	0.224	71.7
Uni4D (BootsTAPIR)	0.135	**0.027**	0.403	0.219	72.5
Uni4D (CTv2)	0.111	0.032	**0.309**	**0.214**	**72.7**
Uni4D (original, CTv3)	**0.110**	0.031	0.338	0.216	72.5
Uni4D (Mask-RCNN)	**0.107**	**0.028**	0.498	0.269	68.2
Uni4D (original, DEVA)	0.110	0.031	**0.338**	**0.216**	**72.5**

Table 2. **Ablation on different trackers and segmentators** We compare both pose and geometry performance on Sintel using different tracklet and segmentation models.

We compare different trackers and segmentors in Tab. 2. TAPIR and BootsTAPIR lead to worse camera pose and depth. CTv2 (CotrackerV2) performs similarly to CTv3 (CotrackerV3), though we found CTv3 to have better dynamic correspondences qualitatively. Mask-RCNN tends to have false positives, leading to over filtering of static tracklets. Due to our dense tracklet initialization, this does not necessarily harm pose estimation. However, it harms depth estimation due to our tracklet-aware densification.

G. Dynamic Regularization Ablation Study

We ablate our different energy terms for dynamic objects in Tab. 3, demonstrating depth map improvements in dynamic regions with each additional dynamic energy term. Note that dynamic segmentations are particularly difficult on Sintel dataset due to large camera motions and close-ups of dynamic elements. Despite the challenging setting, our method produces better dynamic depth maps under the δ<1.25 metric with estimated dynamic segmentations.

Method	Sintel	
	Abs Rel ↓	δ<1.25 ↑
Unidepth [9]	0.178	78.4
Uni4D (no dynamic opt.)	0.253	75.1
Uni4D (+$\mathbf{E}_{\text{smooth}}$)	0.228	77.0
Uni4D (+$\mathbf{E}_{\text{smooth}}$ + $\mathbf{E}_{\text{arap}}$)	0.226	77.1
Uni4D (+$\mathbf{E}_{\text{smooth}}$ + $\mathbf{E}_{\text{arap}}$ + $\mathbf{E}_{\text{NR}}$)	0.220	78.8
Uni4D with gt seg (+$\mathbf{E}_{\text{smooth}}$ + $\mathbf{E}_{\text{arap}}$ + $\mathbf{E}_{\text{NR}}$)	**0.169**	**79.4**

Table 3. **Ablation on $\mathbf{E}_{\text{motion}}$ ($\mathbf{E}_{\text{arap}}$, $\mathbf{E}_{\text{smooth}}$) and $\mathbf{E}_{\text{NR}}$.** We ablate on our different dynamic element energy terms $\mathbf{E}_{\text{motion}}$ and $\mathbf{E}_{\text{NR}}$ through depth map accuracy on Sintel (only considering dynamic elements as defined by ground truth dynamic masks).

With ground truth dynamic masks, our dynamic regularization improves on depth map estimation in dynamic regions over Unidepth [9].

H. Qualitative Results on Camera Pose Evaluation

For a thorough breakdown and visualization of our camera pose evaluations, we plot our Average Translation Error (ATE) results on all camera pose datasets in Fig. 2 3 4. Despite the highly dynamic nature of the Sintel dataset [2], Uni4D provides accurate estimations for most videos thanks to accurate dynamic segmentation, with failure cases in Cave 2 and Temple 3 as seen in Fig.2. Both videos have large dynamic objects that make them challenging among other baselines as well. For real-world datasets TUM-Dynamics [11] and Bonn [7], Uni4D consistently produces the best camera pose estimates with minimal failure cases. Note that across the diverse settings in TUM-Dynamics, including purely translational, rotational, and static camera motion, Uni4D nearly always provides the best pose estimates as seen in (Fig. 3). Our camera smoothness regularization also results in the smoothest trajectories, as shown in Fig. 4.

I. Failure Cases

We provide full visualization of failure cases in our webpage, and sampled frames in Fig. 8. Failure cases include erroneous dynamic masks, depth map estimations, and localization. These errors stem from the underlying models used for segmentation, depth map estimation, and pixel tracking respectively. As the various models are improved upon in the future, we can expect the performance of Uni4D to improve as well.

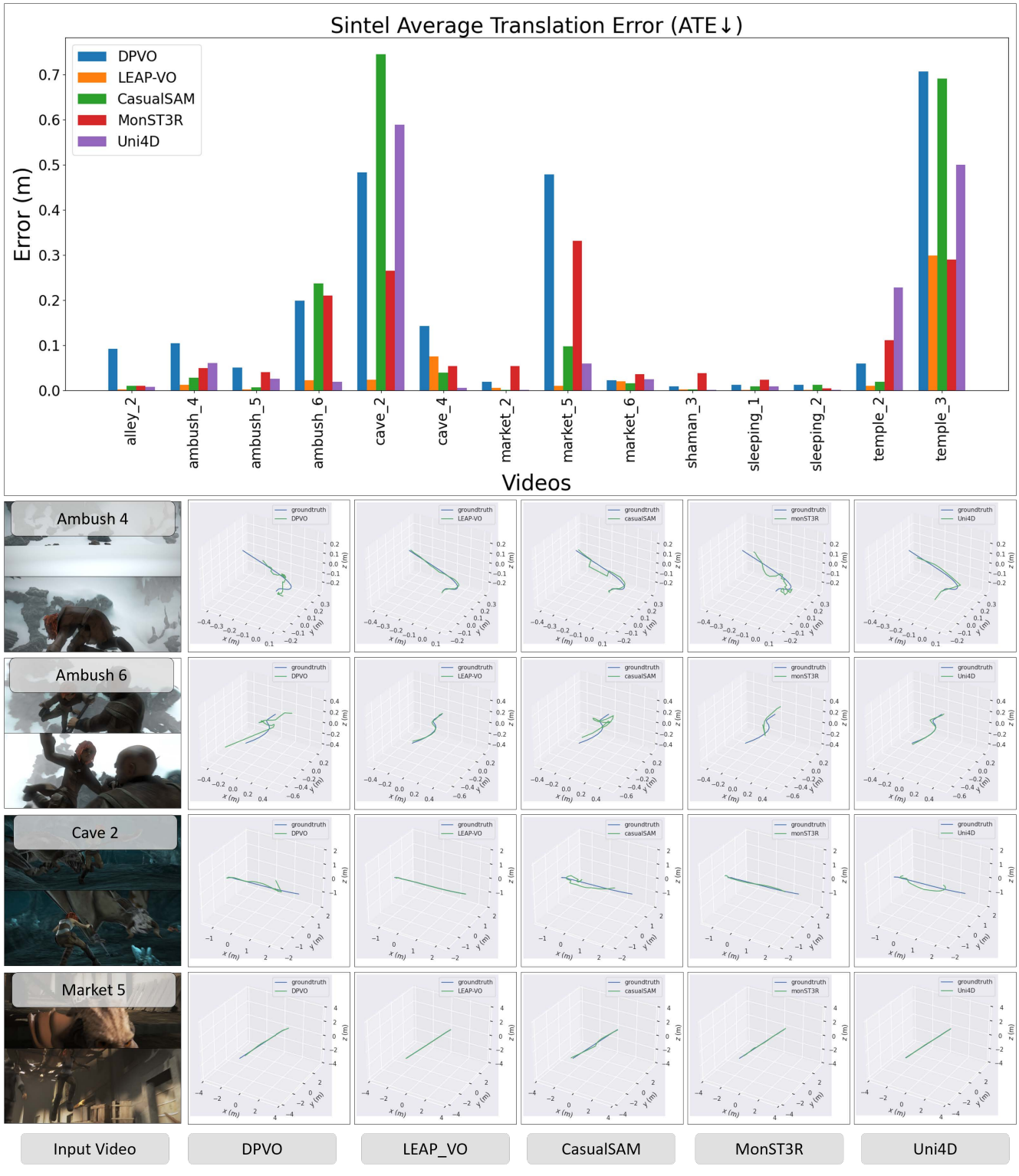


Figure 2. **Qualitative Pose Results on Sintel** Uni4D provides accurate pose estimate on Sintel which contains highly dynamic elements which takes up much of the frame, with 2 failure cases in cave 2 and temple 3. Cave 2 full visualization can be seen from our webpage under "failure cases". Other pose estimates are competitive and even outperform baselines in certain scenes.

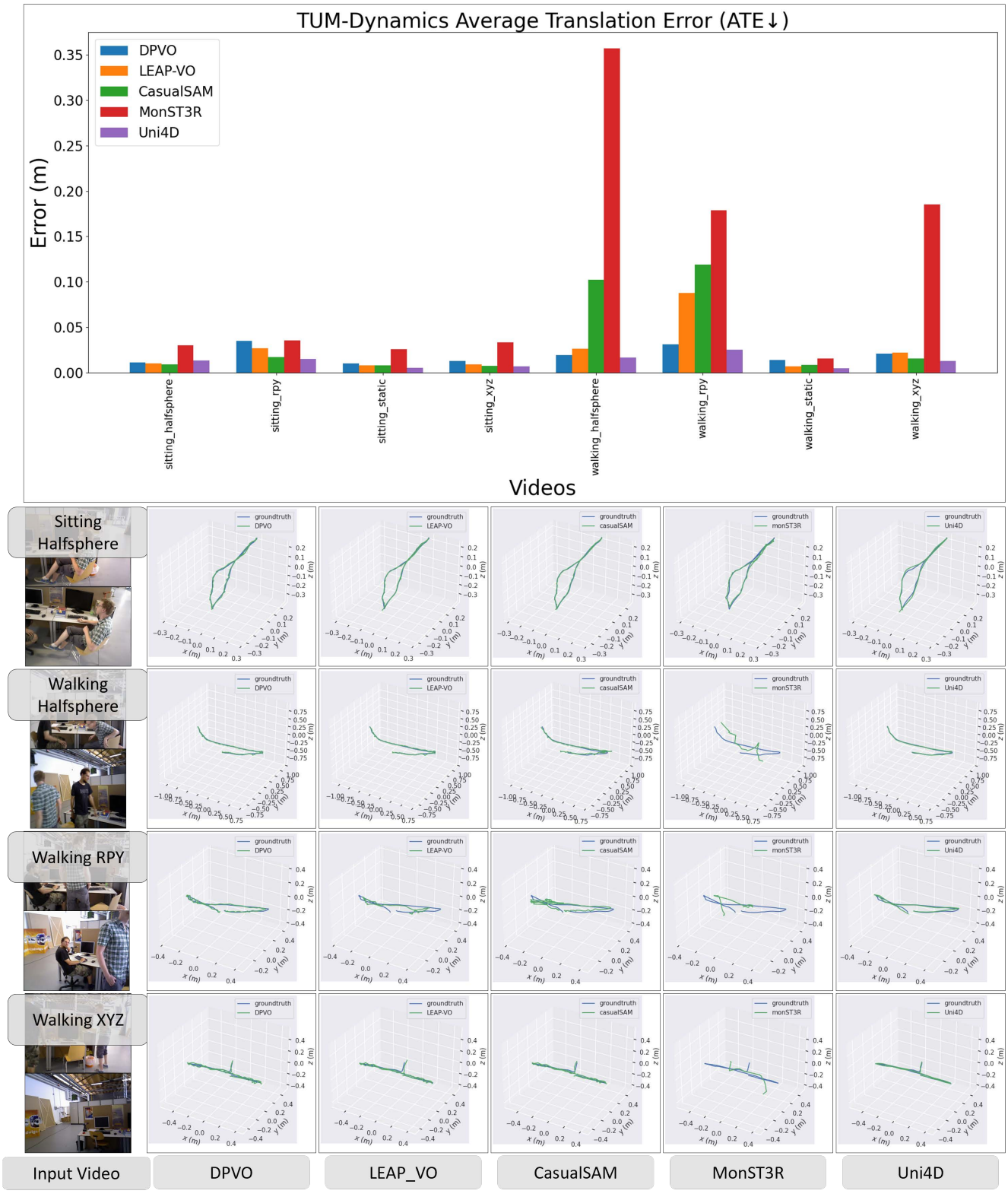


Figure 3. **Qualitative Pose Results on TUM-Dynamics** Uni4D performs well in real-world datasets due to its leverage of big models. Across varied settings where camera motion is mainly rotations (rpy videos), static (static videos), and contains highly dynamic elements (walking videos), Uni4D surpasses other baselines in estimating accurate camera pose.

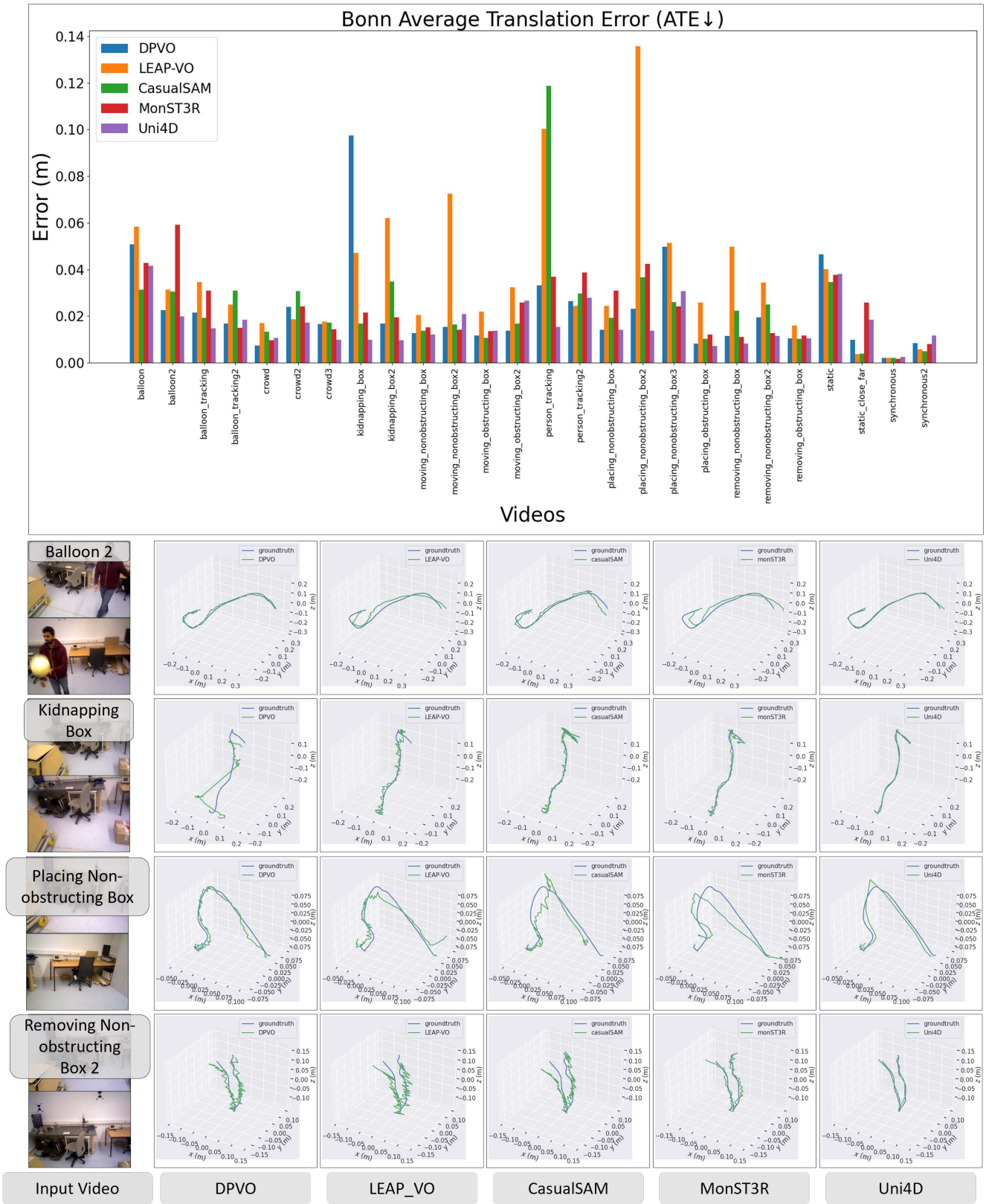


Figure 4. **Qualitative Pose Results on Bonn** Uni4D performs well in real-world datasets, with minimal trajectory errors across all videos in Bonn dataset, successfully estimating trajectories in difficult videos such as 'kidnapping box' and 'placing non-obstructing box' where other baselines face difficulties in.

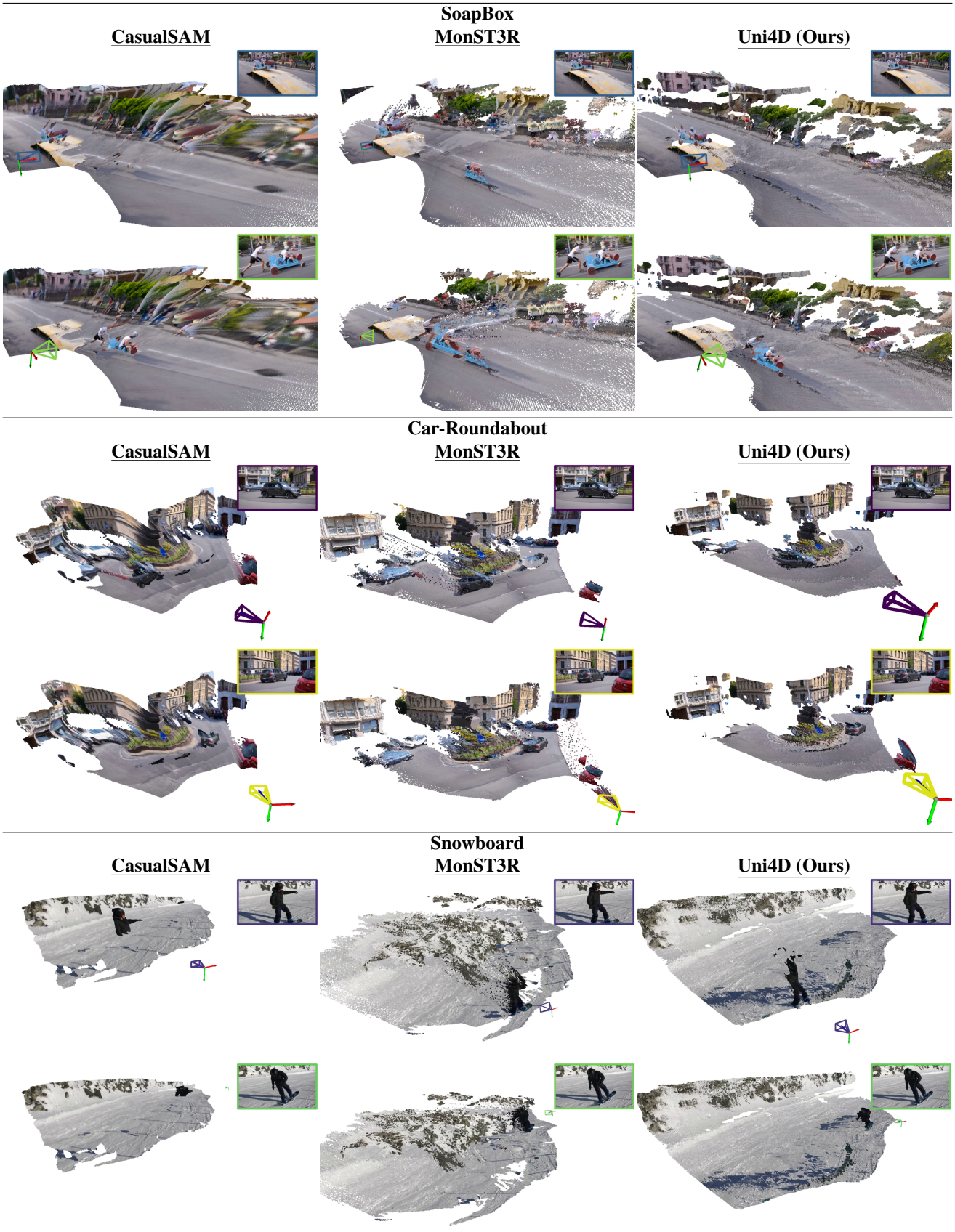


Figure 5. **Qualitative Results on DAVIS dataset** We show qualitatively some of our reconstruction results on the DAVIS dataset compared with other baselines. We visualize here two temporally separate frames and their reconstructions. For full reconstruction, please refer to our attached supplementary webpage.

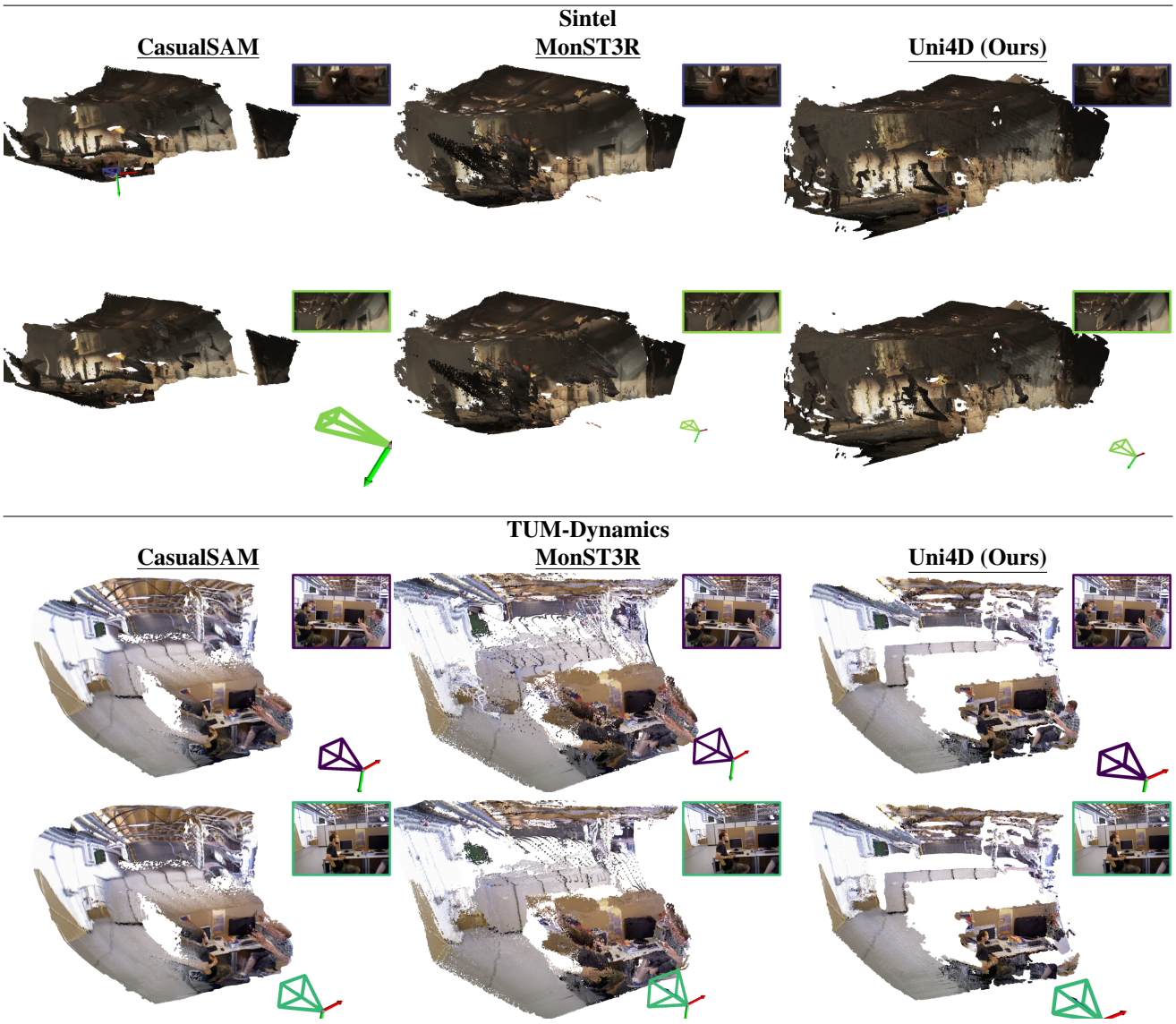


Figure 6. **Qualitative Results on Sintel and TUM-Dynamics dataset** We show qualitatively some of our reconstruction results on Sintel and TUM-Dynamics dataset compared with other baselines. We visualize here 2 temporally separate frames and their reconstructions. For full reconstruction, please refer to our attached supplementary webpage.

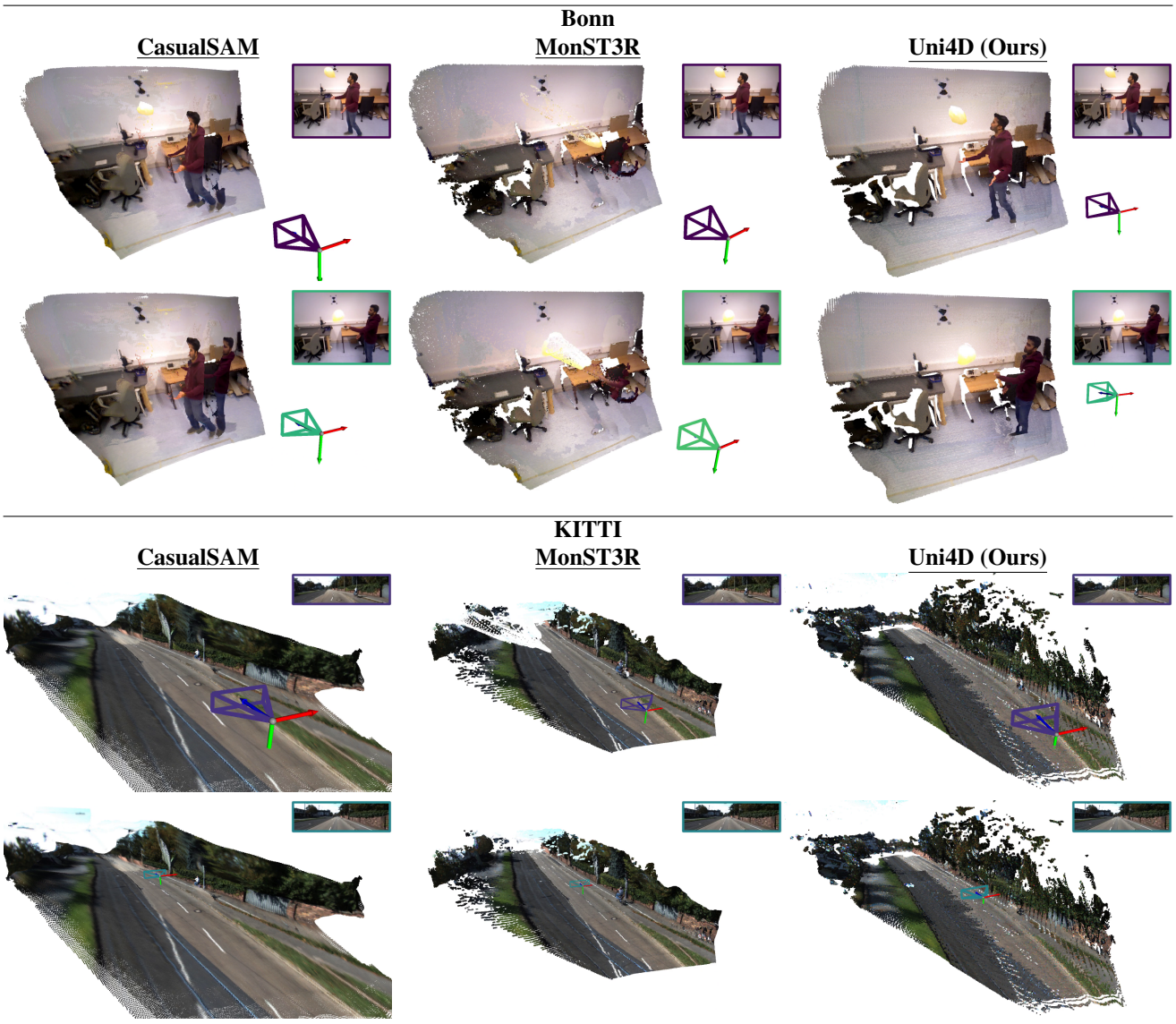
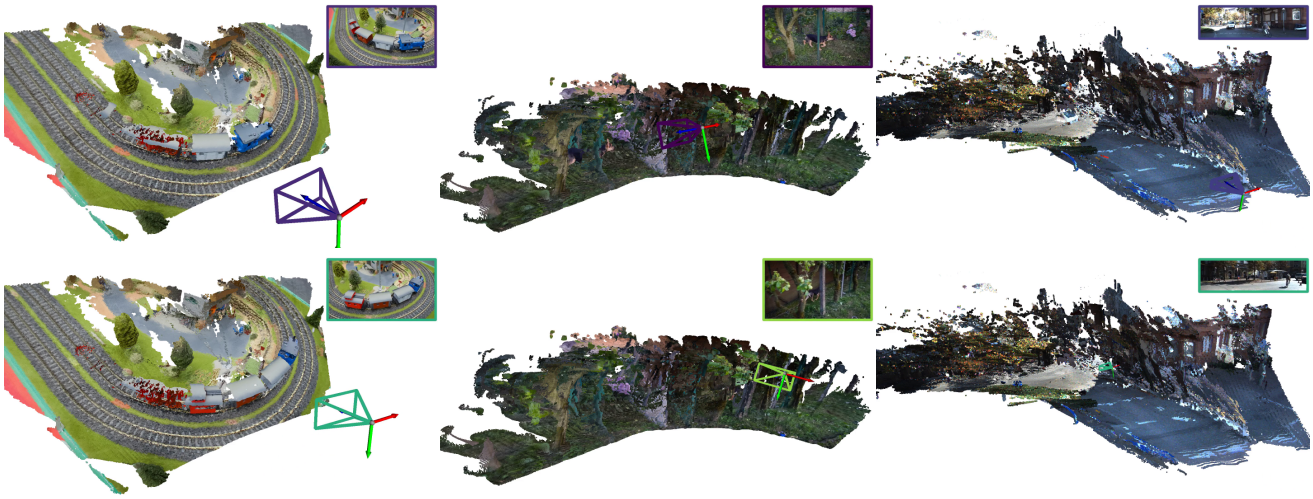
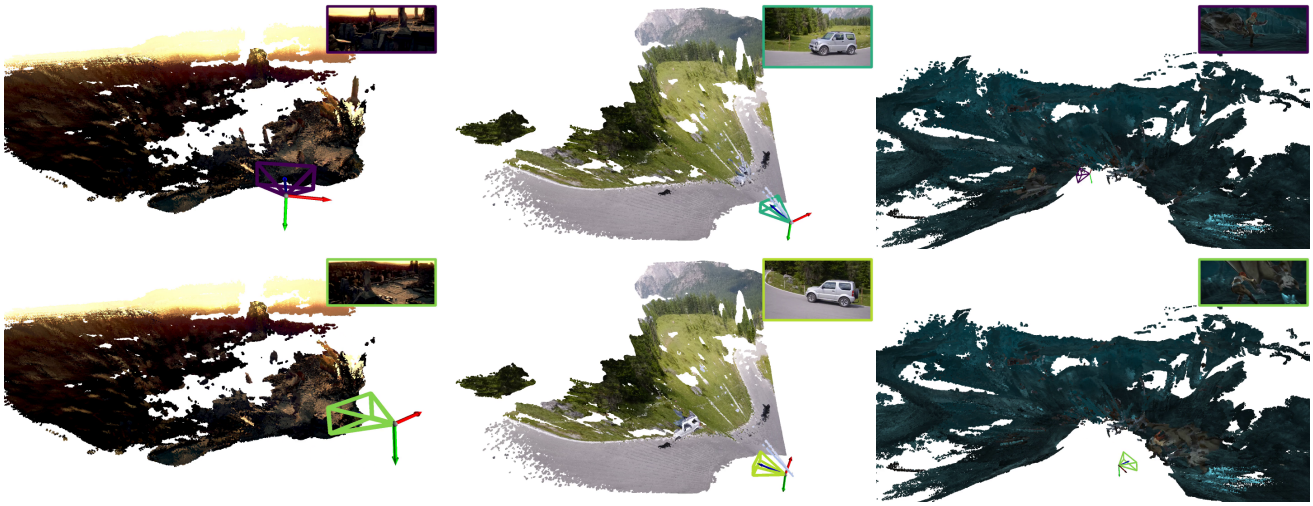


Figure 7. **Qualitative Results on Bonn and KITTI dataset**We show qualitatively some of our reconstruction results on Bonn and KITTI dataset compared with other baselines. We visualize here 2 temporally separate frames and their reconstructions. For full reconstruction, please refer to our attached supplementary webpage.



Incomplete Dynamic Mask | **Trailing pixels around thin structures** | **Poor Camera Pose**



Incorrect Dynamic Masks | **Trailing pixels around dynamic objects + Incorrect Dynamic Trajectory** | **Poor Camera Pose + Dynamic Masks**

Figure 8. **Failure Cases** We visualize several failure cases of Uni4D on various datasets. We visualize here 2 temporally separate frames and their reconstructions. For full reconstruction, please refer to our attached supplementary webpage.

References

[1] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 2

[2] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 611–625. Springer, 2012. 1, 2

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1

[4] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 2

[5] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024. 1

[6] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 1

[7] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7855–7862. IEEE, 2019. 1, 2

[8] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 1

[9] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024. 2

[10] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025. 1

[11] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 1, 2

[12] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 2

[13] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022. 1