St4RTrack: Simultaneous 4D Reconstruction and Tracking in the World

Supplementary Material

Contents

A. Discussion

Despite St4RTrack presents a promising step toward a unified understanding of dynamic scene geometry and motion in a minimalist way, a challenge arises from the per-frame setting. In particular, issues such as scale misalignment, large camera movements, and occlusions are not fully resolved. Incorporating temporal attention across multiple frames would help capture richer motion priors and alleviate these limitations. Another limitation arises from the pretraining dataset's limited diversity and realism in both geometry and motion, necessitating test-time adaptation to improve St4RTrack's robustness in out-of-distribution scenarios. However, it still struggles with highly complex motions. Expanding the training set is therefore a key direction for future work. We envision that large-scale pretraining, when compute permits, could significantly boost St4RTrack's performance and enable it to better handle complex, in-the-wild videos.

B. Differentiable Camera Pose Estimation

We seek to backpropagate the projection loss to the 3D pointmaps through the camera pose. To this end, we build upon the RANSAC-PnP approach from DUSt3R [61], which initially solves for pose $\mathbf{P}^*$ (rotation and translation) by matching per-pixel 2D-3D correspondences in the reconstruction pointmap $\mathbf{X}^j_j$. However, RANSAC is inherently non-differentiable.

To enable end-to-end gradients, we adopt the derivative-based Gauss-Newton (GN) solver inspired by EPro-PnP [5]. Specifically, after obtaining a *detached* solution $\mathbf{P}^*$ from RANSAC-PnP, we refine it using one GN step:

$$\Delta\mathbf{P} \;=\; -\big(J^\top J\big)^{-1} J^\top F(\mathbf{P}^*), \tag{13}$$

where $F(\mathbf{P}^*) = [\, f_1^\top(\mathbf{P}^*), \ldots, f_N^\top(\mathbf{P}^*)]^\top$ is the flattened reprojection error for all N points, and $J = \frac{\partial F(\mathbf{P})}{\partial \mathbf{P}}\big|_{\mathbf{P}=\mathbf{P}^*}$ is its Jacobian. The term $J^\top J$ approximates the Hessian of the negative log-likelihood (NLL), while $J^\top F(\mathbf{P}^*)$ is the gradient of the NLL with respect to the pose. This gradient effectively *pushes* the incremental solution $\Delta\mathbf{P}$ toward reducing the reprojection errors. The final *differentiable* pose estimate is:

$$\mathbf{P} \;=\; \mathbf{P}^* \;+\; \Delta\mathbf{P}. \tag{14}$$

Since $\mathbf{P}^*$ is detached, only the GN increment $\Delta\mathbf{P}$ remains differentiable, allowing the reprojection loss to backpropagate through $\mathbf{P}$ and thus refine the 3D pointmaps.

C. Details on the *WorldTrack* Benchmark

C.1. Datasets

Dataset Preparation. For the two real-world datasets, we adopt the 3D camera coordinate tracking annotation of ADT and Panoptic Studio from the TAPVID-3D dataset. Using the paired camera parameters provided, we transform the camera coordinates to the world coordinate system. For the two synthetic datasets, we use the test sets from Point Odyssey and Dynamic Replica Dataset. We uniformly downsample the query points to approximately 1,000 per sequence. Each sequence contains 128 sampled frames,

Table 3. **World Coordinate 3D Point Tracking (EPE - Global Median) .** We report end-point error (EPE; lower is better) for both all points and dynamic points after global median alignment. The best (lowest) values are in **bold**.

		All Points				Dynamic Points			
Category	**Methods**	PO	DS	ADT	PStudio	PO	DS	ADT	PStudio
Combinational	SpaTracker+RANSAC-Procrustes	0.6408	0.9185	0.5876	0.4266	0.4358	1.0444	0.1600	0.4266
	SpaTracker+MonST3R	0.5917	0.8823	0.5362	0.4837	0.4085	0.9136	0.1511	0.4837
Feed-forward	MonST3R	0.9021	0.4387	0.2721	0.4568	0.6452	0.5313	0.1578	0.4568
	SpaTracker	0.7499	0.9274	0.8530	0.3094	0.4695	1.0828	0.1628	0.3094
	St4RTrack (Ours)	**0.3140**	**0.2682**	**0.2680**	**0.2637**	**0.2970**	**0.2961**	**0.1212**	**0.2637**

Table 4. **World Coordinate 3D Point Tracking (APD/EPE - SIM(3)).** Each cell shows APD_{3D} (higher is better) / EPE (lower is better) after global IM(3) alignment. The best APD (highest) and the best EPE (lowest) in every column are **bold**.

		All Points				Dynamic Points			
Category	**Methods**	PO	DR	ADT	PStudio	PO	DR	ADT	PStudio
Combinational	SpaTracker+Procrustes	46.20/0.5670	55.10/0.5292	59.40/0.4027	67.82/0.2660	61.00/0.3338	61.65/0.3720	**88.65**/0.0596	67.82/0.2660
	SpaTracker+MonST3R	48.23/0.5388	56.78/0.5069	60.01/0.3910	64.32/0.2971	61.78/0.3290	61.88/0.3681	87.32/**0.0485**	64.32/0.2971
Feed-forward	MonST3R	37.62/0.8073	64.83/0.3725	79.48/0.1881	64.11/0.3015	48.95/0.4768	55.36/0.3872	84.73/0.0720	64.11/0.3015
	SpaTracker	43.17/0.6079	54.65/0.5324	53.96/0.4963	**80.76/0.1650**	60.49/0.3374	61.32/0.3750	87.68/0.0616	**80.76/0.1650**
	St4RTrack (Ours)	**71.84/0.2774**	**76.28/0.2436**	**83.03/0.1631**	76.97/0.1969	**67.43/0.2870**	**67.90/0.2627**	85.34/0.0688	76.97/0.1969

Table 5. **World Coordinate 3D Reconstruction (APD/EPE - SIM(3)).** Results on Point Odyssey (PO) and TUM-Dynamics after global SIM(3) alignment. Lower is better for EPE, higher is better for APD. The best results are in **bold**.

		Point Odyssey		TUM-Dynamics	
Category	**Methods**	**EPE↓**	**APD↑**	**EPE↓**	**APD↑**
w/ Global Align.	DUSt3R+GA	0.3541	62.42	0.2989	69.23
	MASt3R+GA	0.3717	61.31	0.5294	49.81
	MonST3R+GA	**0.2601**	69.31	0.3173	66.00
Feed-forward	DUSt3R	0.4251	56.70	0.3092	67.48
	MASt3R	0.4473	55.09	0.5862	45.43
	MonST3R	0.3462	62.10	0.3508	62.83
	St4RTrack	0.2741	**69.53**	**0.2413**	**74.14**

though only the first 64 frames are used for evaluation. This results in 160 and 140 sequences from Point Odyssey and Dynamic Replica, respectively. From these, we randomly sample 50 sequences per dataset for evaluation.

Filtering Criteria. To ensure data quality, we apply several filtering strategies: For TUM, we keep the pixels which associated with depth values within 0.1 - 5 meters, as the depth camera is less accurate at long range. For Point Odyssey, we exclude sequences generated in the Kubric style [15] due to their lack of realism. We also remove scenes with ambiguous depth (e.g., heavy foggy conditions), and any frames where the camera intrinsics are dynamic.

C.2. Additional Quantitative Evaluation

Following TAPVid-3D [26], we adopt global median scale alignment, since both our predictions and the ground truth use the first frame's camera coordinate system as the world coordinate. The Average Percent of Points within Distance (APD_{3D}) measures the overall accuracy of the 3D trajectories in world coordinates, while Euclidean endpoint error (EPE) offers a complementary perspective on localization accuracy. Accordingly, we additionally report EPE results on the WorldTrack benchmark. As shown in Table 3, St4RTrack attains state-of-the-art EPE on all sub-test sets, consistent with the APD_{3D} results in the main paper. Beyond alignment to the first camera's pose, we also evaluate under SIM(3) alignment (i.e., SE(3) plus a global scale factor) for both APD_{3D} and EPE to assess performance of 3D tracking (See Tab. 4) and reconstruction (See Tab. 5) under a more flexible registration. Comprehensive evaluations show that St4RTrack achieves state-of-the-art performance in most scenarios.

Additionally, we've implemented a 2D tracking baseline,"CoTracker3+MonST3R (w/ GA)", for the completeness. It achieved only 48.18%, 59.10% APD and 0.676m, 0.382m EPE on the PO/PStudio datasets, respectively, notably worse than the 60.71%, 66.14%, and 0.342m, 0.281m achieved by our method. The 2D-based tracking methods cannot be estimate points when occluded, as they do not reason in 3D space.

Moreover, We evaluate St4RTrack on camera pose and

Table 6. **Depth and Camera Pose Evaluation.** We compare with MonST3R global alignment and fully feed-forward version for depth prediction and camera pose evalution on two datasets.

	Sintel		Bonn		Sintel	TUM
	Abs Rel↓	$\delta<1.25$ ↑	Abs Rel↓	$\delta<1.25$ ↑	ATE↓	ATE↓
MonST3R (w/ GA)	0.335	58.5	**0.063**	**96.4**	**0.108**	0.074
MonST3R (FFW)	0.443	56.6	0.066	95.6	0.416	0.071
St4RTrack (FFW)	**0.318**	**62.2**	0.065	95.6	0.348	**0.045**

Figure 5. **Qualitative Results of Camera Pose Evaluation.**

monocular depth evaluation following MonST3R's setup on Sintel, Bonn, and TUM (see table below). As in Tab. 6, our feed-forward method outperforms MonST3R-FFW across all metrics and approaches MonST3R+GA in several cases.

C.3. Qualitative Evaluation

We present the qualitative results of our fully feed-forward approach on WorldTrack benchmark. Specifically, we show the reconstruction results in Fig. 7 (TUM-Dynamics) and Fig. 8 (Point Odyssey). We show the tracking results of all four datasets in Fig. 9.

D. Details on Test-Time Adaptation

D.1. Implementation Details

We set the weights of different loss factors in Eq. (11) to $\lambda_{\text{traj}} = 1$, $\lambda_{\text{depth}} = 10$, and $\lambda_{\text{align}} = 5$. For WorldTrack evaluation, the two test-time adaptations are set up as follows: **Sequence-Level (Instance) Adaptation:** Fine-tune a separate model for each of the 50 sequences. We sample 300 frames per epoch, train for 3 epochs, and use a batch size of 4. **Dataset-Level (Domain) Adaptation:** Fine-tune a single model on the entire dataset. We sample 100 frames per epoch, train for 15 epochs, and use a batch size of 4.

D.2. Ablation Studies

We perform an ablation study to evaluate two key design choices of our method and present qualitative results in Fig. 6. First, we assess the effectiveness of our pretraining stage by directly applying test-time adaptation to a pretrained checkpoint from MonST3R [67], without finetuning the base model on our training datasets. As shown in Fig. 6 (column 2), the baseline exhibits unaligned pointmaps between the tracking and reconstruction branches, underscoring the importance of pretraining on synthetic data—even in the presence of a domain gap with real-world data.

Second, we evaluate the impact of our proposed test-time adaptation. As demonstrated in Fig. 6 (column 3), the adapted model successfully corrects drifting points, ensuring that points consistently trace back to their original spatial locations in the first frame. This finding supports our analysis that small-scale training data alone is insufficient for fine-grained prediction, particularly at the boundaries of moving objects. In contrast, St4RTrack produces spatially aligned pointmaps with significantly fewer drifting points. The colorful tails in the visualization indicate the long-term trajectories, while the accurately predicted geometry in dynamic regions results in a crisp and precise rendering.

Table 7. **World Coordinate 3D Tracking (Median-Scale).** End-point error (EPE ↓) and APD_{3D} ↑ for DR and PStudio after global median scaling. Best (lowest EPE / highest APD_{3D}) in each column is shown in **bold**.

	DR		PStudio	
Methods	**EPE↓**	**APD↑**	**EPE↓**	**APD↑**
Spatialtracker+Procrustes-RANSAC	0.9185	55.01	0.4266	52.05
St4RTrack	0.2682	73.74	0.2637	69.67
St4RTrack + TTA (per-sequence)	**0.2472**	**76.07**	**0.2243**	**73.71**
St4RTrack + TTA (per-dataset)	0.2547	74.86	0.2280	73.30
w/o trajectory loss	0.2767	72.75	0.2421	72.50
w/o depth loss	0.5524	48.22	0.2975	66.50
w/o alignment loss	0.3263	66.65	0.3357	60.07
w/o pre-training	0.3377	65.50	0.3801	57.71

Furthermore, we ablate (1) the performance gain from the feed-forward St4RTrack, instance-level adaptation, and domain-level adaptation, and (2) the contribution of each TTA component by omitting individual elements. Table 7 summarizes our findings. First, both TTA variants yield substantial improvements over the feed-forward mode, with instance-level adaptation achieving the highest accuracy, as it fully specializes to each test sequence. Second, removing any single TTA component—trajectory loss, depth loss, alignment loss, or synthetic pretraining—causes a performance drop in all scenarios, underscoring the necessity of each component.

E. Additional Results

We also present additional results for both feed-forward only (Fig. 10) and test-time adaptation (Fig. 11) below.

F. Discussion and Future Work

While the key insight of St4RTrack is that unified 4D reconstruction and world-coordinate tracking can be achieved without modifying the original DUSt3R architecture, but rather by simply redefining the pointmap outputs, we found the long sequence non-overlapping cases remain challenging. This is mostly due to the lack of explicit temporal modeling—*i.e.*, incorporating temporal context via global alignment (as in DUSt3R/MonST3R) or temporal attention (as utilized in video models). It remains a limitation of St4RTrack and thus represents a promising future direction.

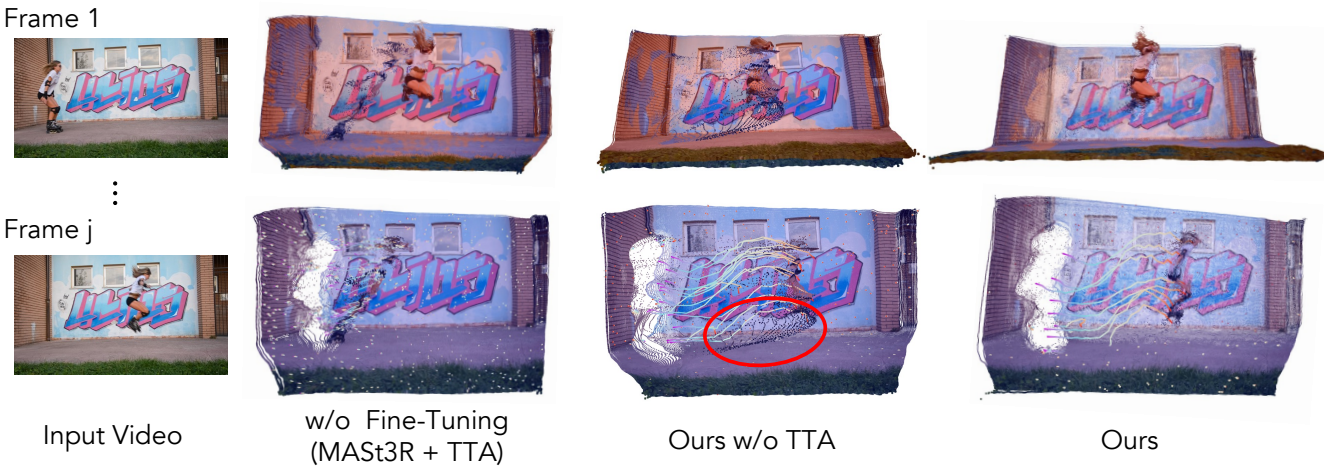

Figure 6. **Ablation Study.** We show the qualitative comparison of our full method and variants that do not pretrain or do not adapt in test time. Predicted pointmaps from two heads are visualized together.

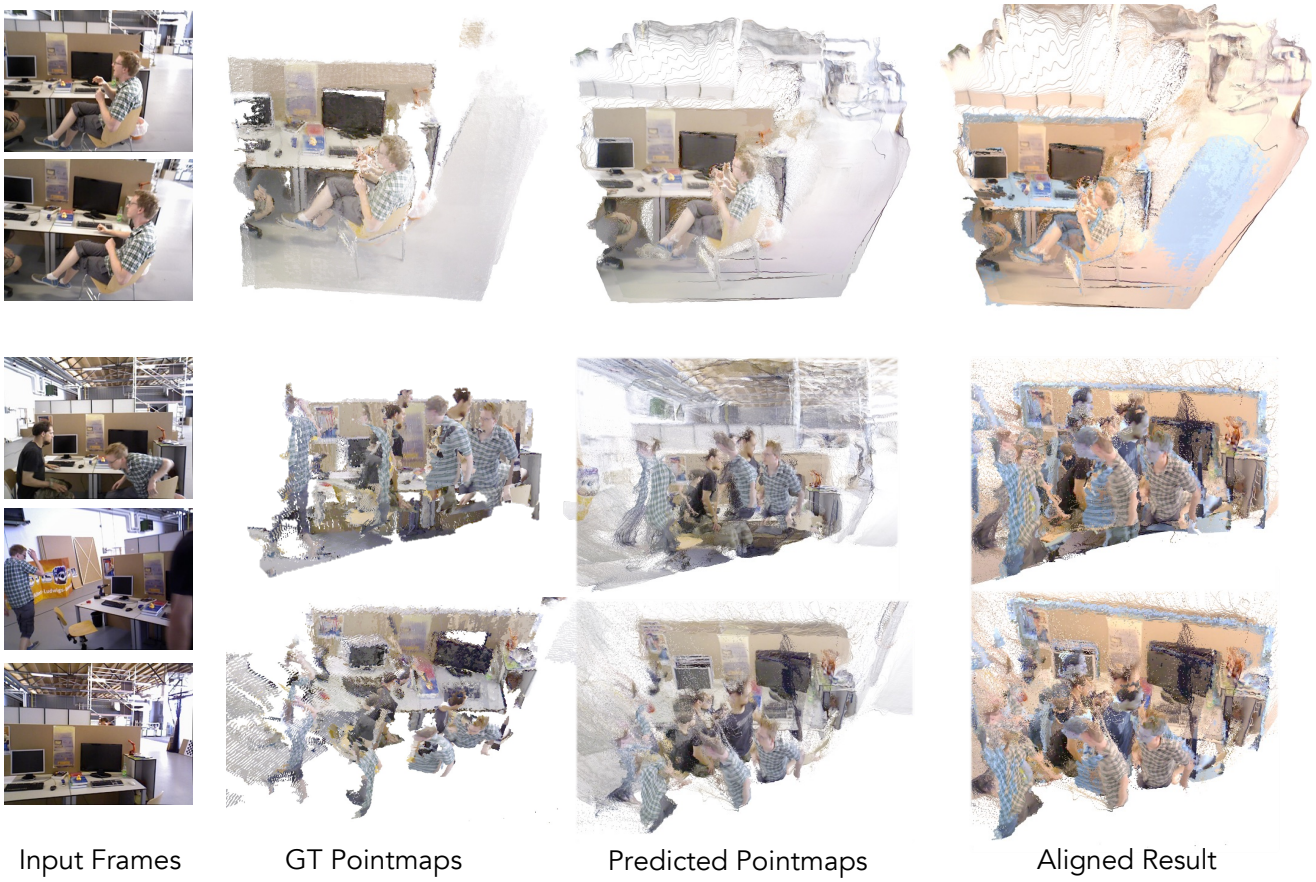

Figure 7. **Reconstruction Results of St4RTrack on TUM-Dynamics Dataset.** From left to right, we show 1) the sampled frames from the input sequence of 64 frames, 2) the subsampled ground truth pointmaps, 3) the predicted pointmaps of our method, and 4) the aligned results of the predicted and GT pointmaps with median-scale. Note that the reconstruction result is inferred in a feed-forward way.

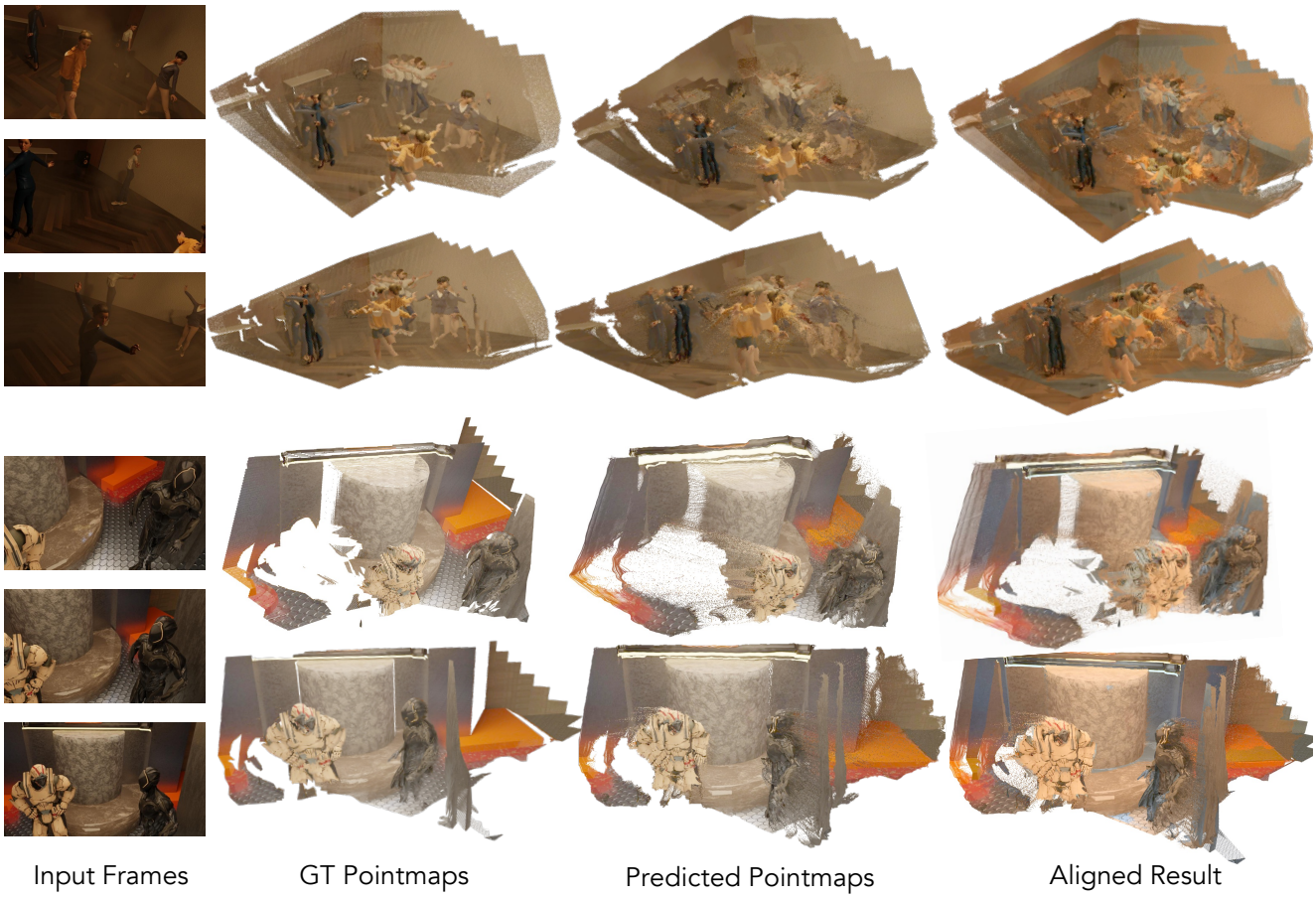

Figure 8. **Reconstruction Results of St4RTrack on Point Odyssey Dataset.** From left to right, we show 1) the sampled frames from the input sequence of 64 frames, 2) the subsampled ground truth pointmaps, 3) the predicted pointmaps of our method, and 4) the aligned results of the predicted and GT (yellow) pointmaps with median-scale. Note that the reconstruction result is inferred in a feed-forward way.

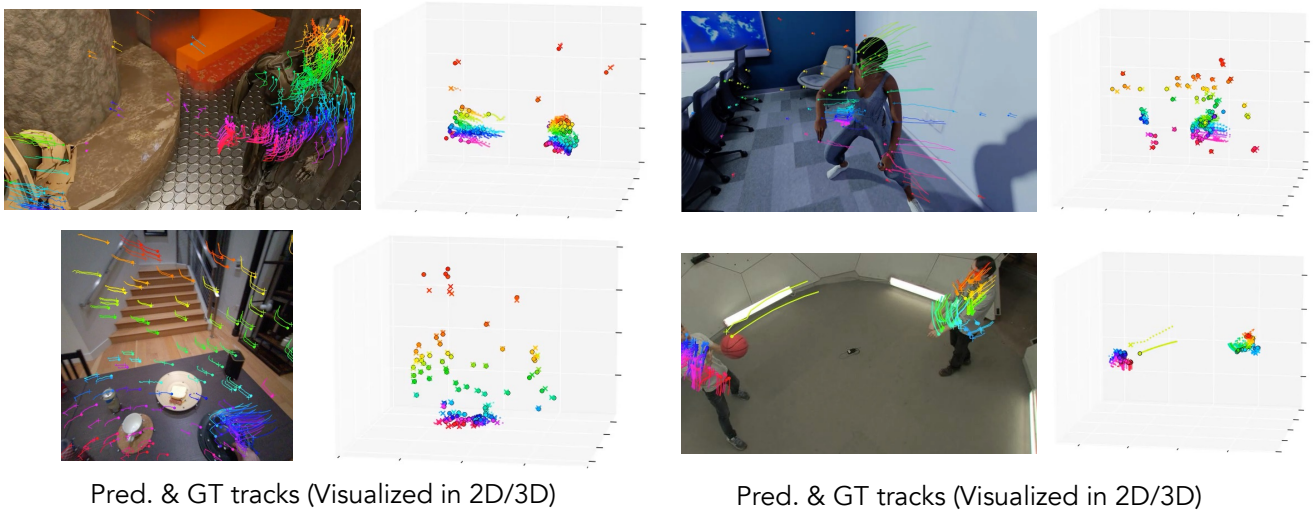

Figure 9. **Tracking Results of St4RTrack on WorldTrack Benchmark.** We show the 2D and 3D visualized results of the predicted tracks (visualized as "+") aligned with the ground truth tracks (visualized as "•"). The corresponding datasets are Point Odyssey (top left), Dynamic Replica (top right), Arial Digital Twin (bottom left), and Pnapotic Studio (bottom right).

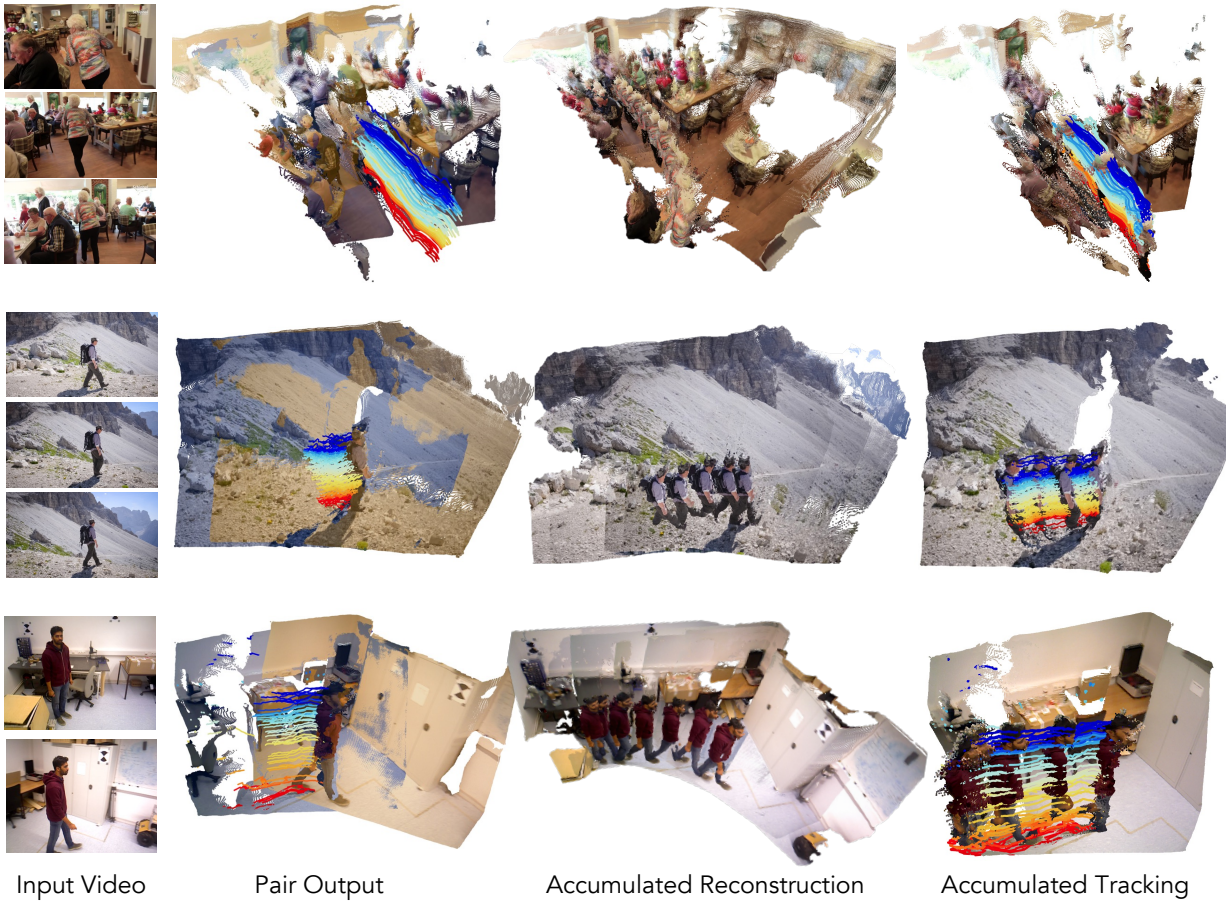

Figure 10. **Fully *Feed-Forward* Inference Results of St4RTrack.** We show from left to right: 1) the input video, 2) the pairwise output for tracking (in blue) and reconstruction (in yellow) of the same frame, 3) the accumulated results of the reconstruction pointmaps, and 4) the accumulated results of the tracking pointmaps. Note that we anchor the *middle frame* as the reference frame for point tracking.

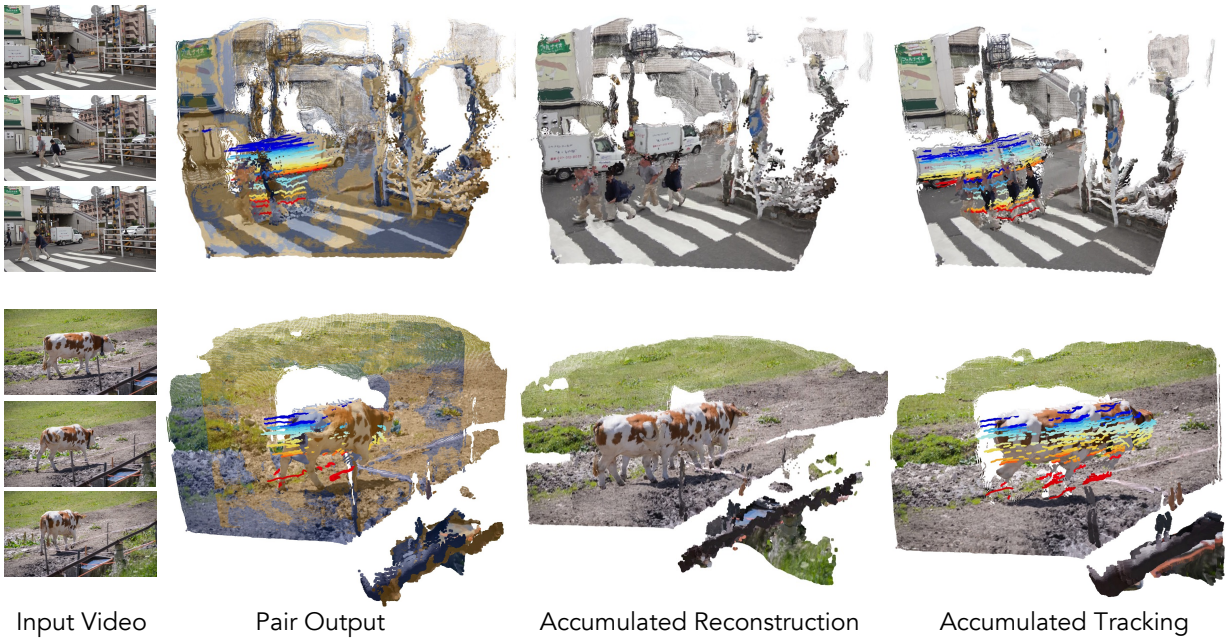

Figure 11. **Test-Time Adaptation Results of St4RTrack.** The first frame is set to be the reference frame.