Table 1: **Tracking Error**. The table demonstrates our point track error handling and contains four row sections. **1. Ours**: our results when evaluating on the original CoTracker point tracks from the pet test set. **2.** $\sigma = \mathbf{1, 5, 10}$: our results using the original CoTracker point tracks with added Gaussian noise with corresponding standard deviation in pixels. **3. 10,20,50% outliers**: our results using the original CoTracker point tracks where respectively 10,20,50 percent of the tracks are replaced with uniformly sampled pixels. **4. 10,20,50% occlusions**: same as the outlier setup, but the outlier points are marked as occluded by setting $o = 0$ (defined in L97 in the main paper), which improves outlier handling. Overall we see that our method can robustly handle the noisy CoTracker inputs, and can further tolerate a significant level of synthetically added noise.

	Abs Rel ↓		$\delta < 1.25$ ↑		$\delta < 1.25^2$ ↑		$\delta < 1.25^3$ ↑		ATE ↓	RPE Trans ↓	RPE Rot ↓
	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.	All	(mm)	(mm)	(deg)
Ours	0.11	0.08	0.88	0.92	0.99	0.98	1.00	1.00	8.96	3.79	0.23
$\sigma = 1$	0.11	0.08	0.88	0.92	0.99	0.98	1.00	1.00	9.07	3.87	0.24
$\sigma = 5$	0.11	0.08	0.88	0.92	0.99	0.98	1.00	1.00	10.96	5.05	0.30
$\sigma = 10$	0.13	0.10	0.84	0.90	0.98	0.98	1.00	0.99	13.18	6.75	0.42
10% outliers	0.26	0.19	0.74	0.74	0.89	0.90	0.94	0.96	14.45	5.92	0.45
20% outliers	0.35	0.23	0.62	0.65	0.82	0.85	0.90	0.94	15.26	5.96	0.46
50% outliers	0.61	0.32	0.36	0.51	0.60	0.75	0.78	0.88	19.68	11.41	0.69
10% occlusions	0.22	0.16	0.80	0.80	0.92	0.93	0.95	0.97	11.68	4.58	0.30
20% occlusions	0.30	0.20	0.70	0.71	0.85	0.89	0.92	0.95	13.15	5.04	0.35
50% occlusions	0.58	0.33	0.38	0.51	0.64	0.76	0.82	0.88	16.18	5.82	0.51

Table 2: **PointOdyssey**. Average results on 9 test samples from the PointOdyssey dataset. Metrics are the same as in the main paper. Note that since the input point tracks are provided by the dataset, our running times are computed without considering point track extraction time. As can be seen, our model trained only on cats, can efficiently and accurately handle sequences from the PointOdyssey dataset.

	Abs Rel ↓		$\delta < 1.25$ ↑		$\delta < 1.25^2$ ↑		$\delta < 1.25^3$ ↑		ATE ↓	RPE Trans ↓	RPE Rot ↓	Time ↓
	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.	All			(deg)	(minutes)
Ours (C)	**0.09**	**0.09**	**0.91**	**0.92**	**1.00**	**1.00**	**1.00**	**1.00**	0.05	0.02	0.18	**0.003**
Ours (C) + BA	**0.09**	**0.09**	**0.91**	**0.92**	**1.00**	**1.00**	**1.00**	**1.00**	**0.02**	**0.01**	0.06	0.007
CasualSAM (59)	0.12	0.11	0.85	0.86	0.98	0.99	0.99	1.00	**0.02**	**0.01**	**0.03**	62

Table 3: **Comparisons with Marigold**. Comparison with the Marigold depth prediction method on the pet test set. As can be seen, our depth accuracy is much higher.

	Abs Rel ↓		$\delta < 1.25$ ↑		$\delta < 1.25^2$ ↑		$\delta < 1.25^3$ ↑	
	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.	All
Ours (C)	**0.11**	**0.08**	**0.88**	**0.92**	**0.99**	**0.98**	**1.00**	**1.00**
Marigold	0.48	0.31	0.28	0.51	0.50	0.75	0.64	0.86

Table 4: **TAPIR point tracks**. Comparison of using our method with TAPIR(10) for point tracking at inference time versus using CoTracker(18), which was used for training our method, on the pet test set. As can be seen, our method is robust to the point tracks obtained by TAPIR, and the results are further improved when applying test time finetuning (FT).

	Abs Rel ↓		$\delta < 1.25$ ↑		$\delta < 1.25^2$ ↑		$\delta < 1.25^3$ ↑		ATE ↓	RPE Trans ↓	RPE Rot ↓
	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.	All	(mm)	(mm)	(deg)
Ours (C) CoTracker	0.11	0.08	0.88	0.92	0.99	0.98	1.00	1.00	8.96	3.79	0.23
Ours (C) TAPIR	0.14	0.11	0.82	0.85	0.96	0.96	0.99	0.99	11.86	5.00	0.36
Ours (C) TAPIR + BA	0.14	0.11	0.82	0.85	0.96	0.96	0.99	0.99	6.08	4.14	0.26
Ours (C) TAPIR + FT	0.09	0.06	0.88	0.94	0.99	0.99	1.00	0.99	6.66	3.88	0.22