Complet4R: Geometric Complete 4D Reconstruction

Supplementary Material

1. Camera and Depth Supervision

Our loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{point}} + \lambda \mathcal{L}_{\text{camera}} + \mathcal{L}_{\text{depth}}. \tag{1}$$

In the following, we provide a detailed description of the camera loss and the depth loss included in the loss function.

The camera loss $\mathcal{L}_{\text{camera}}$ supervises the predicted camera parameters $\hat{\mathbf{g}}$ by comparing them with the ground-truth parameters $\mathbf{g}$ using the Huber loss $\|\cdot\|_\epsilon$. Specifically, it is defined as:

$$\mathcal{L}_{\text{camera}} = \sum_{i=1}^{N} \|\hat{\mathbf{g}}_i - \mathbf{g}_i\|_\epsilon . \tag{2}$$

The depth loss $\mathcal{L}_{\text{depth}}$ follows VGGT and implements the aleatoric-uncertainty loss, weighting the discrepancy between the predicted depth $\hat{D}_i$ and the ground-truth depth D_i with the predicted uncertainty map $\hat{\Sigma}_i^D$:

$$\begin{aligned}\mathcal{L}_{\text{depth}} = \sum_{i=1}^{N} \Big(&\|\Sigma_i^D \odot (\hat{D}_i - D_i)\| \\ &+ \|\Sigma_i^D \odot (\nabla \hat{D}_i - \nabla D_i)\| - \alpha \log \Sigma_i^D \Big)\end{aligned} \tag{3}$$

where $\odot$ is the channel-broadcast element-wise product.

2. Details of the Ablation Study

We introduce three variants in our ablation study, focusing on the training loss, the aggregation representation, and the way aggregation tokens are processed.

Dynamic-Weighted Point Loss. First, we detail the design of the *dynamic-weighted point loss*, which aims to emphasize points that exhibit significant motion across timestamps in the sequence.

Given the predicted points $\hat{\mathbf{P}}_i^a$ warped to a target timestamp a and the corresponding ground-truth points $\mathbf{P}_i^a$, the dynamic-weighted point loss function is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{point}} = \sum_{i=1}^{N} \Big(&\|\hat{\Sigma}_{i,a}^P \odot \mathbf{w}_i^a \odot (\hat{\mathbf{P}}_i^a - \mathbf{P}_i^a)\| \\ &+ \|\hat{\Sigma}_{i,a}^P \odot (\nabla \hat{\mathbf{P}}_i^a - \nabla \mathbf{P}_i^a)\| - \alpha \log \hat{\Sigma}_{i,a}^P \Big),\end{aligned} \tag{4}$$

where $\hat{\Sigma}_{i,a}^P$ is the predicted uncertainty map for aleatoric weighting, α is a regularization coefficient, and the weights $\mathbf{w}_i^a$ are computed based on whether a point is dynamic:

$$\mathbf{w}_i^a = \begin{cases} 1000, & \text{dynamic point}, \\ 1, & \text{static point}, \end{cases} \tag{5}$$

The method for distinguishing dynamic points from static points is as follows. For each point, we first compute its offsets across time, defined as the Euclidean distance between the point's coordinate at any timestamp and its coordinate at the target timestamp a:

$$\Delta \mathbf{P}_i^t = \|\mathbf{P}_i^t - \mathbf{P}_i^a\|_2, \qquad t = 0, \ldots, N-1, \tag{6}$$

where $\mathbf{P}_i^t$ denotes the coordinate of point i at timestamp t, and $\mathbf{P}_i^a$ is its coordinate at the target timestamp a. A point is considered *dynamic* if its motion exceeds a dataset-specific threshold:

$$\|\Delta \mathbf{P}_i\| > \delta, \qquad \delta = \text{threshold}_{\text{dataset}}. \tag{7}$$

Specifically, we set $\delta = 0.03$ for the Point Odyssey and Dynamic Replica datasets, and $\delta = 0.01$ for the SAIL-VOS 3D dataset.

Aggregation Representation. We consider two types of aggregation representations. The first is the predicted 3D coordinates of points at the target timestamp a, referred to as the *endpoint*. The second is the predicted offsets from points at each timestamp to their corresponding coordinates at the target timestamp a, referred to as the *offset*. We describe the supervision method for the *offset* representation below.

The ground-truth offsets are computed as follows:

$$\mathbf{O}_i^t = \mathbf{P}_i^a - \mathbf{P}_i^t, \qquad t = 0, \ldots, N-1, \tag{8}$$

where $\mathbf{O}_i^t$ represents the ground-truth offset of point i from timestamp t to the target timestamp a, $\mathbf{P}_i^t$ and $\mathbf{P}_i^a$ denote the 3D coordinates of point i at timestamp t and the target timestamp a, respectively.

Given these ground-truth offsets, the point-wise loss is defined to supervise both the positional and smoothness discrepancies, while also incorporating the predicted aleatoric uncertainty $\hat{\Sigma}_{i,a}^O$ and the focal-style point weighting $\mathbf{w}_i^a$:

$$\begin{aligned}\mathcal{L}_{\text{point}} = \sum_{i=1}^{N} \Big(&\|\hat{\Sigma}_{i,a}^O \odot \mathbf{w}_i^a \odot (\hat{\mathbf{O}}_i^a - \mathbf{O}_i^a)\| \\ &+ \|\hat{\Sigma}_{i,a}^O \odot (\nabla \hat{\mathbf{O}}_i^a - \nabla \mathbf{O}_i^a)\| - \alpha \log \hat{\Sigma}_{i,a}^O \Big).\end{aligned} \tag{9}$$

Category	Methods	PO		DR		ADT		PStudio	
		APD↑	EPE↓	APD↑	EPE↓	APD↑	EPE↓	APD↑	EPE↓
Combinational	SpaTracker+RANSAC-Procrustes	61.00	33,38	61.65	37.20	**88.65**	5.96	67.82	26.60
	SpaTracker+MonST3R	61.78	32.90	61.88	36.81	87.32	**4.85**	64.32	29.71
Feed-forward	MonST3R	48.95	47.68	55.36	38.72	84.73	7.20	64.11	30.15
	SpaTracker	60.49	33.74	61.32	37.50	87.68	6.16	80.76	16.50
	St4RTrack	67.43	28.70	67.90	26.27	85.34	6.88	76.97	19.69
	Complet4R (Ours)	**85.82**	**12.39**	**85.30**	**12.78**	87.58	5.39	**85.62**	**12.49**

Table 1. **World Coordinate 3D Point Tracking on Dynamic Points with Global SIM(3) Alignment.**

	Sintel		BONN		KITTI	
Method	Abs Rel ↓	δ<1.25 ↑	Abs Rel ↓	δ<1.25 ↑	Abs Rel ↓	δ<1.25 ↑
VGGT	**0.298**	**68.1**	**0.057**	**96.8**	**0.061**	**97.0**
CUT3R	0.421	47.9	0.078	93.7	0.118	88.1
Ours	0.353	63.9	0.066	95.9	0.079	94.9

Table 2. Evaluation on Video Depth Estimation.

	Sintel			ScanNet		
Method	ATE↓	RPE trans ↓	RPE rot ↓	ATE↓	RPE trans ↓	RPE rot ↓
VGGT	**0.168**	**0.065**	**0.477**	**0.035**	**0.015**	**0.380**
CUT3R	0.213	0.066	0.621	0.099	0.022	0.600
Ours	0.215	0.089	1.081	0.050	0.023	0.549

Table 3. Evaluation on Camera Pose Estimation.

Aggregation Tokens. We design two strategies for handling the aggregation tokens. The first strategy concatenates the aggregation tokens with the image tokens. The second strategy broadcasts the aggregation tokens and adds them to all image tokens.

3. Additional Experiments

3.1. Additional 3D Point Tracking Evaluation

We additionally test the 3D point tracking on dynamic points in world coordinates with global SIM(3) alignment (SE(3) plus a global scale factor), and the results reported in Tab. 1. They are also tested on the four datasets described in the 3D Point Tracking Section. Complet4R outperforms all competing approaches by a substantial margin on most datasets, except on ADT, where it achieves performance close to the best method. This demonstrates the strong potential tracking ability of our work.

3.2. Depth and Camera Pose Estimation

We evaluate our method and prior methods on multi-frame depth estimation and camera pose estimation, with results reported in Tables 2 and 3. Although our method does not outperform VGGT, it retains the core capabilities after fine-tuning for 4D complete reconstruction, while achieving competitive or superior performance compared to CUT3R.

Frames	2	4	8	16	32	64	128	256
Time (s)	0.10	0.15	0.26	0.51	1.16	2.95	8.53	27.86
Mem. (GB)	7.0	7.6	9.0	10.2	12.6	15.1	18.0	28.6

Table 4. Runtime and GPU memory consumption for different sequence lengths. Runtime is measured in seconds, and GPU memory usage is reported in gigabytes.

3.3. Runtime and Memory

As shown in Tab. 4, we evaluate the inference runtime and peak GPU memory usage of Complet4R under different numbers of input frames. All experiments are conducted on a single NVIDIA A100 GPU, with an image resolution of 294×518.

Despite incorporating additional functionalities and a more complex architecture than VGGT, Complet4R maintains competitive efficiency. For example, inference on 64 frames takes less than 3 seconds with only 15 GB of peak GPU memory usage, demonstrating the practicality of our model for real-world applications.

Moreover, due to the use of global attention, the runtime does not scale linearly with the number of frames, which poses challenges for long-video inference. A potential solution is to adopt Flash Attention to further optimize the attention layers, which can reduce both computation time and memory consumption.

4. Additional Qualitative Results

We present additional reconstruction results, as shown in Fig. 1, and tracking results, as shown in Fig. 3, demonstrating that our model achieves excellent 4D complete reconstruction and 3D point tracking performance.

5. Additional Discussion

Existing NeRF/Gaussian-based scene generation methods address related problems. Here, we select the representative Gaussian-based generative approach DreamScene4D for comparison. Such methods aim to obtain visually pleasing renderings through iterative optimization, which differs

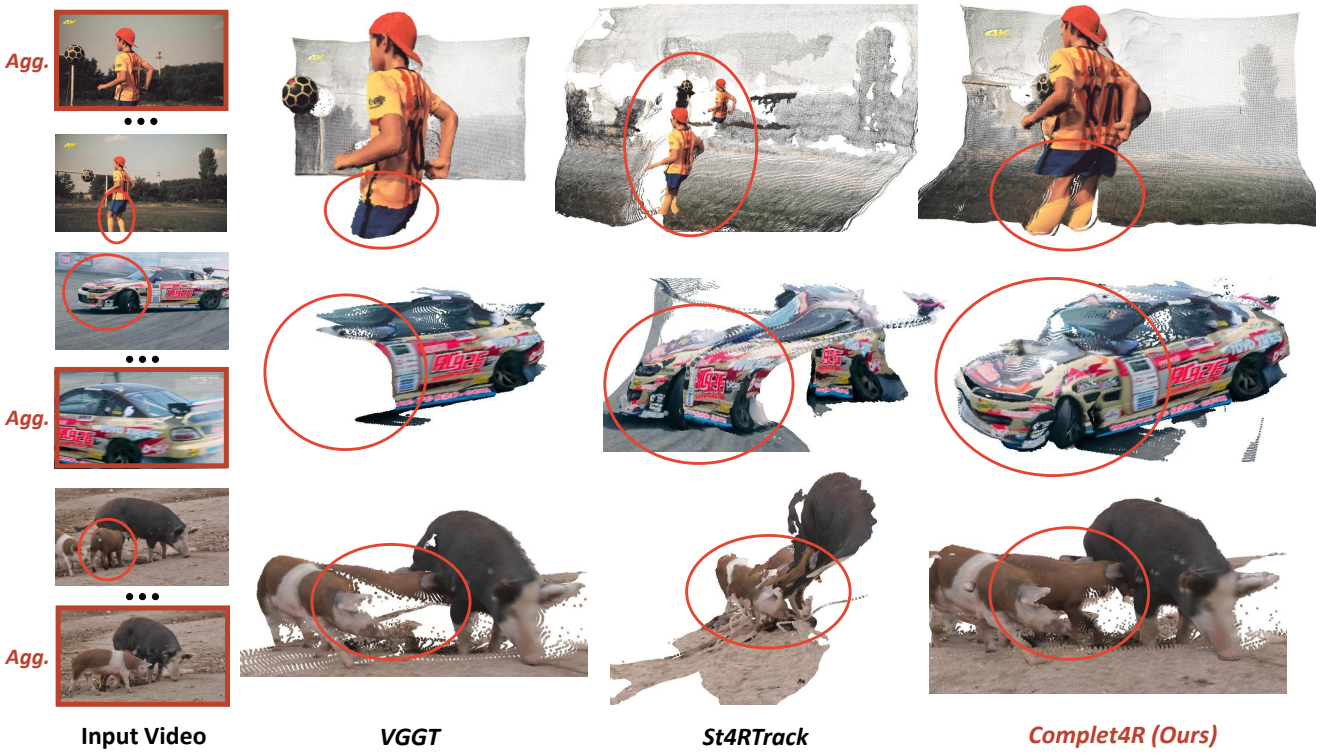

Figure 1. **More Qualitative Results for 4D Complete Reconstruction.** The first column shows the video inputs, with red boxes indicating the target aggregation timestamp for each sequence (*Agg.*: aggregation). The subsequent columns present the outputs of different models. Our method successfully reconstructs the complete geometry at the target timestamp highlighted by the red ellipses, whereas other methods produce incomplete or geometrically inconsistent reconstructions.

Figure 2. **Comparison with DreamScene4D.** The left shows the video inputs, while the right presents the reconstruction results: (a) from DreamScene4D, and (b) from our method. Our approach demonstrates higher reconstruction quality.

from our feed-forward regression framework that focuses on geometric consistency. In complex scenarios, such as the occlusion case shown in 2, DreamScene4D exhibits geometric inconsistencies and blurry artifacts, whereas Complet4R demonstrates strong geometric consistency. Moreover, benefiting from its feed-forward design, Complet4R runs significantly faster than DreamScene4D (0.5 s vs. 28 min for 16 frames).

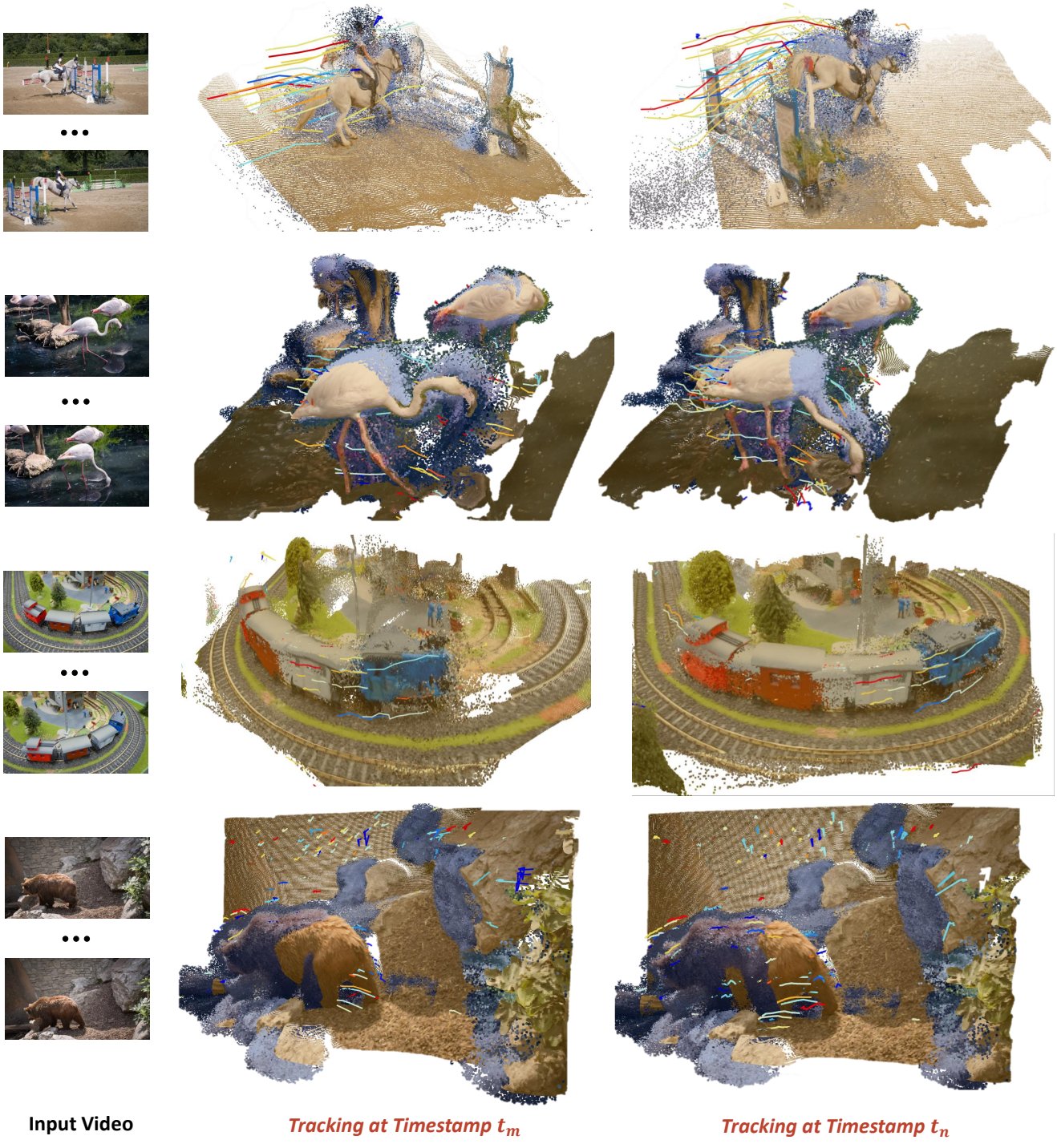

Figure 3. **More Qualitative Results for 3D Dynamic Point Tracking.** The first column shows the input images; the second and third columns display the tracking trajectories produced by our method at successive time steps. The smooth trajectories demonstrate strong spatiotemporal geometric consistency.