Appendix

A BADJA Benchmark

BADJA is a dataset of animal joint tracking, including 7 videos from the DAVIS dataset and 2 extra ones. Since it approximately fivefold downsamples the video, the movement of the objects in the video is very fast, and the movements of the target tracking points are also intense accordingly, making them hard to track. Although seven of them are collected from DAVIS, the videos' FPS and annotations are different from the TAP-Vid-DAVIS dataset.

To provide a more detailed comparison, follow both the settings in PIPs and CoTracker. The first one is the "δ^{seg}" one following PIPs. This setting only considers the 7 videos collected from DAVIS and calculates the proportion of points with discrepancies less than $0.2\sqrt{A}$ compared to the ground truth (GT) as its metric, where A indicates the area of the tracking animal's mask. Also, this setting requires comparing the performance of each video in detail. The second one is the "δ^{3px}" one proposed by CoTracker. This setting considers all of the 9 videos and calculates the proportion of points with discrepancies less than 3 pixels compared to the GT as its metric. This setting requires comparing the overall performance.

As shown in Table 7, TAPTR obtains the best overall performance on both settings. Note that, due to the difference between the requirement of tracking physical surface points in TAP, and the requirement of tracking joints inside the animal in BADJA benchmark, the performance on BADJA can only be considered as a reference.

Table 7: Avg.-7 indicates the average performance of the 7 videos collected from DAVIS. Avg.-All indicates the average performance of all videos, including the extra 2.

Method	δ^{seg}								δ^{3px}
	bear	camel	cows	dog-a	dog	horse-h	horse-l	Avg.-7	Avg.-All
DINO [37]	75.0	59.2	70.6	10.3	<u>47.1</u>	35.1	56.0	50.5	–
ImageNet ResNet [12]	65.4	53.4	52.4	0.0	23.0	19.2	27.2	34.4	–
CRW [16]	66.1	67.2	64.7	6.9	33.9	25.8	27.2	41.7	–
VFS [59]	64.3	62.7	71.9	10.3	35.6	33.8	33.5	44.6	–
MAST [21]	51.8	52.0	57.5	3.4	5.7	7.3	34.0	30.2	–
RAFT [48]	64.6	65.6	69.5	13.8	39.1	37.1	29.3	45.6	7.6
PIPs [11]	<u>76.3</u>	<u>81.6</u>	<u>83.2</u>	**34.2**	44.0	**57.4**	<u>59.5</u>	<u>62.3</u>	13.5
TAP-Net [5]	–	–	–	–	–	–	–	–	6.3
TAPIR [7]	–	–	–	–	–	–	–	–	15.2
CoTracker [19]	–	–	–	–	–	–	–	–	<u>18.0</u>
TAPTR(ours)	**81.8**	**86.8**	**89.8**	<u>26.9</u>	**52.6**	<u>47.1</u>	**63.2**	**64.0**	**18.2**

B TAPTR in Trajectory Prediction

Here we show the handwriting trajectory prediction of TAPTR for an example. For more details please refer to the videos in our code repository. Corresponding video names are provided in Sec. **??**.

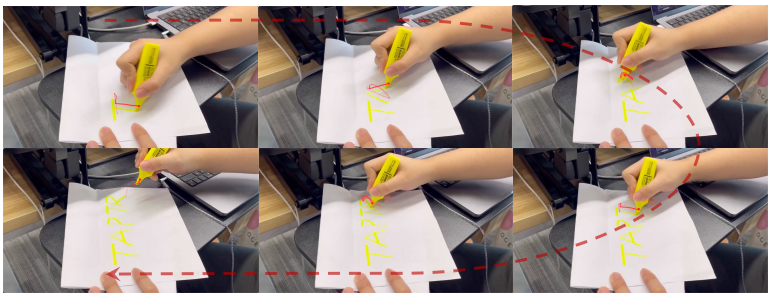

Fig. 5: The trajectory of handwriting predicted by TAPTR.

C TAPTR in Video Editing

Here we show the results of the video editing using TAPTR. We sample points in the editing area of the first frame and track these points across the whole video. For more details please refer to the videos in our code repository. Corresponding video names are provided in Sec. **??**.

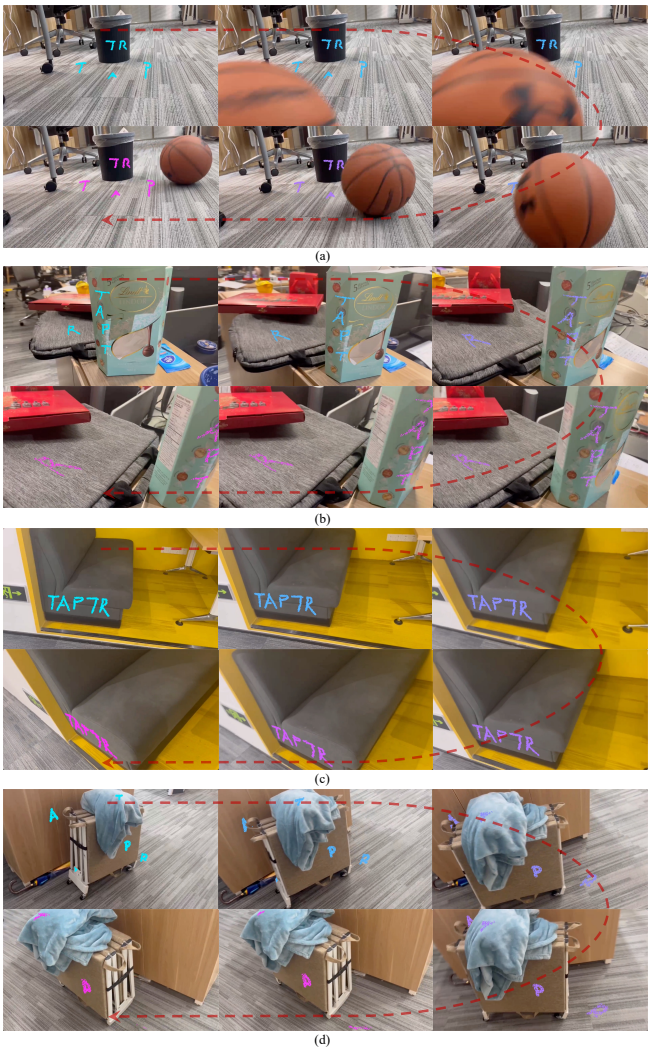

Fig. 6: Edit video with TAPTR. The color of the editing area changes over time.

D More Comparisons

Here we show more comparisons between the current state-of-the-art method and TAPTR. For more details please refer to the videos in our code repository. Corresponding video names are provided in Sec. **??**.

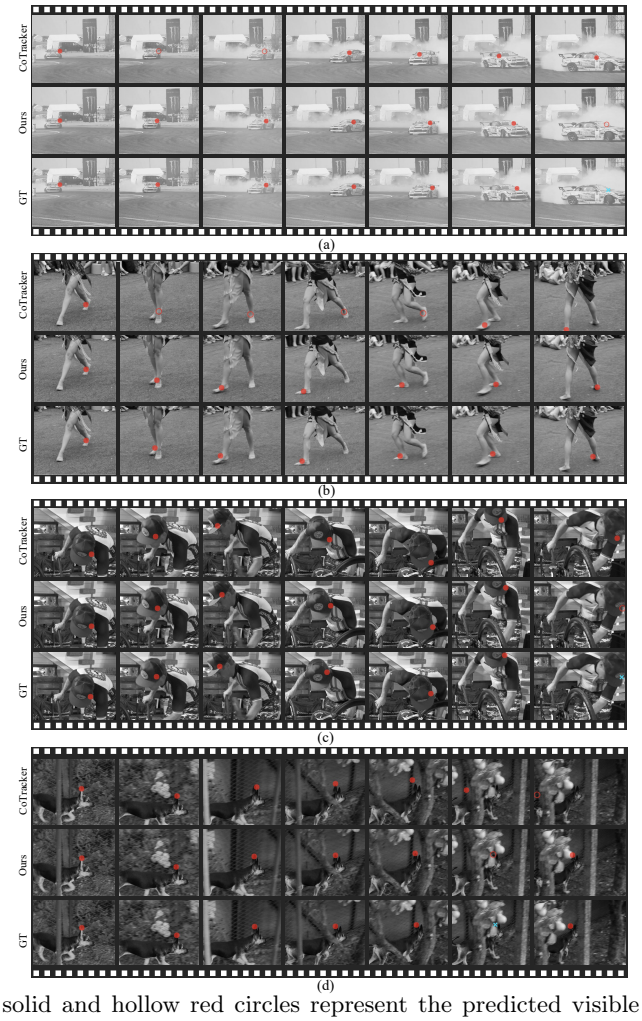

Fig. 7: The solid and hollow red circles represent the predicted visible and invisible points. We manually supplement the GT locations of invisible points for better comparison. Best view in electronic version