Event-aided Dense and Continuous Point Tracking: Everywhere and Anytime – Supplementary Material –

Zhexiong Wan[1,2] Jianqin Luo[1] Yuchao Dai[1] Gim Hee Lee[2]
[1]School of Electronics and Information, Northwestern Polytechnical University &
Shaanxi Key Laboratory of Information Acquisition and Processing
[2]Department of Computer Science, National University of Singapore

1. Appendix

In this appendix, we provide additional details on our methodology and experiments. The former includes B-spline curve modeling and multi-frame trajectory aggregation, and the latter provides additional experimental results as well as a demo video on several datasets, including **real & simulated datasets and dense & sparse point tracking evaluation**.

1.1. Method details

B-spline dense and continuous point trajectories. Given N_c control points $\{\mathbf{P}_i\}^{N_c}$ and basis functions $\{B_{i,p}(t)\}^{N_c}$ with degree p, the continuous point trajectory $\mathbf{T}(t)$ represented by b-spline curve in time variable t is a collection of piecewise polynomial functions:

$$\mathbf{T}(t) = \sum_{i=1}^{N_c} B_{i,p}(t)\mathbf{P}_i. \tag{1}$$

Based on the Cox–de Boor recursion, the detailed derivation of basis functions is:

$$B_{c,0}(t) = \begin{cases} 1 & k_i \leq t < k_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

$$B_{c,p}(t)=\frac{t-k_i}{k_{c+p}-k_i}B_{c,p-1}(t)+\frac{k_{c+p+1}-t}{k_{c+p+1}-k_{c+1}}B_{c+1,p-1}(t), \tag{3}$$

where $k_1, k_2, k_3, \ldots, k_m$ are $m = N_c + p + 1$ knots of the curve with a non-decreasing order that represent the times when the pieces polynomials meet. The internal $N_c - p + 1$ knots $k_{p+1}, k_{p+2}, \ldots, k_{m-p}$ constitute the deformation of the curve. The beginning and the ending remaining knots $k_1, k_2, \ldots, k_p$ and $k_{m-p+1}, k_{m-p+2}, \ldots, k_m$ are usually specified as duplicates of k_{p+1} and k_{m-p}, in order to ensure the curve is tangent to the edges of the first and last control points so that the curve is clamped.

In experiments, we fixed the internal knots to evenly spaced numbers over a specified interval from 0 to 1, and the model only needs to learn the coordinates of control points $\{\mathbf{P}\}^{N_c} \in \mathbb{R}^{2\times N_c\times H\times W}$ to model the continuous trajectory $\mathbf{T}$ of every pixel, where $H \times W$ is the image size. The head and tail of the modeled trajectory coincide with the start and end control points $\mathbf{P}_1$ and $\mathbf{P}_{N_c}$.

Multi-frame optical flow and trajectories accumulation. Existing parametric motion modeling methods are fixed in the number of frames they can handle, *e.g.*, BFlow [7] is limited to between two frames, and CPFlow [12] hard to get benefit for more than 4 frame inputs, resulting in suboptimal long-term trajectory modeling. Inspired by the practice of multi-frame optical flow aggregation [14, 17], we propose a new multi-frame curve trajectories accumulation strategy to handle long-term videos with arbitrary frames.

In optical flow-based frameworks such as AccFlow [17] and MFT [14], multi-frame optical flows are usually combined based on warping operations. Given the the previous global flow $\mathbf{F}_{1\to t}$ and local flow $\mathbf{F}_{t\to t+1}$, representing the motion displacements from time 1 to t and t to $t + 1$, respectively, the aggregated current global flow $\mathbf{F}_{1\to t+1}$ from time 1 to $t + 1$ can be computed as follows:

$$\mathbf{F}_{1\to t+1} = \begin{cases} \mathbf{F}_{1\to t}+\text{Warp}(\mathbf{F}_{t\to t+1},\mathbf{F}_{1\to t}) & \text{if } \mathbf{V}_{1\to t}(\mathbf{x})=1, \\ \mathbf{F}_{1\to t}+\text{Fusion}(\mathbf{F}_{t\to t+1},\mathbf{F}_{1\to t}) & \text{if } \mathbf{V}_{1\to t}(\mathbf{x})=0, \end{cases} \tag{4}$$

where $\mathbf{V}_{1\to t}(\mathbf{x})$ indicates whether the point $\mathbf{x}$ from time 1 is visible at time t. Warp is the backward warping operation [3], which is a fundamental operation in optical flow that allows sampling pixel values or features from one frame to reconstruct another frame using a given flow field. Fusion addresses occlusion issues by integrating additional residual flow prediction where pixels are occluded and cannot be directly aggregated. This process first captures local motion cues from neighborhood features using a sub-network module and then infers global motion through visibility-based $\mathrm{softmax}$ fusion, similar to

AccFlow [17]. By combining both local and global motion information, Fusion ensures a more accurate and coherent optical flow estimation in occluded regions. Notably, the backward warping operation has an inherent error as it requires integer sampling with floating-point coordinates, *i.e.*, $\text{Warp}(\mathbf{a}, \mathbf{b})(\mathbf{x}) = \mathbf{a}(\mathbf{x} + \mathbf{b}(\mathbf{x}))$. Besides this, optical flow and visible area estimates with insufficient accuracy also affect the reliability of multi-frame aggregation. Therefore, an additional post-refinement is still necessary even in unoccluded areas [1, 17]. As a result, we additionally introduce a refinement sub-network to estimate the flow update $\Delta\mathbf{F}_t$ and compute $\text{Warp}(\mathbf{F}_{t\to t+1}, \mathbf{F}_{1\to t}) + \Delta\mathbf{F}_t$ or $\text{Fusion}(\mathbf{F}_{t\to t+1}, \mathbf{F}_{1\to t}) + \Delta\mathbf{F}_t$ to reduce the above errors.

In contrast, multi-frame curve aggregation also considers how to keep the shape of the subcurves while aggregating the curves. Denote the previous global curve as $\mathbf{T}_{1\to t}$ with $(t-1)\times N_c$ control points, which represents the aggregation of $t-1$ sub-curves $\mathbf{T}_{1\to 2}, ..., \mathbf{T}_{t-1\to t}$ from time 1 to t. If we get the local sub-curve piece as $\mathbf{T}_{t\to t+1}$ with N_c control points from time t to $t+1$, we can propagate the current global trajectory $\mathbf{T}_{1\to t+1}$ with $t \times N_c$ control points from time 1 to $t+1$ by:

$$\mathbf{T}' = \begin{cases} \text{Warp}(\mathbf{T}_{t\to t+1}, \mathbf{T}_{1\to t}, \mathbf{O}_t) + \Delta\mathbf{T}_t & \text{if } \mathbf{V}_{1\to t}(\mathbf{x}) = 1, \\ \text{Fusion}(\mathbf{T}_{t\to t+1}, \mathbf{T}_{1\to t}, \mathbf{M}^{global}_{1\to t}) + \Delta\mathbf{T}_t & \text{if } \mathbf{V}_{1\to t}(\mathbf{x}) = 0, \end{cases}$$
$$\mathbf{T}_{1\to t+1}(\mathbf{x}) = \big[\mathbf{T}_{1\to t}(\mathbf{x}), \mathbf{T}'(\mathbf{x})\big], \tag{5}$$

where $[,]$ aggregates the control points of two sub-curves to create a more complex smooth curve. $\mathbf{V}_{1\to t}$ is the visible mask of each point from the initial frame to the t-th frame, and $\delta\mathbf{T}_t$ is the trajectory update used for uniformly refinement in the global trajectory accumulation process, both of which are estimated by the trajectory decoder.

Taking two curves $\mathbf{T}_1$ and $\mathbf{T}_2$ with N_1 and N_2 control points $\{\mathbf{P}_i\}^{N_1}$ and $\{\mathbf{Q}_i\}^{N_2}$ respectively as an example, the aggregation process smoothly connects the two curves while ensuring the resulting curve goes through the endpoints of the sub-curves, *i.e.*, the first start point $\mathbf{P}_1$, the first endpoint $\mathbf{P}_{N_1}$ (overlapped with the second start point $\mathbf{Q}_1$), and the end point of $\mathbf{Q}_{N_2}$. To achieve this, we need to ensure that both the position, tangent and curvature (0th, 1st, 2nd order derivatives) are continuous at the position of the connected points, *i.e.*, $\mathbf{Q}'_1 = \mathbf{P}_{N_1}$, $\mathbf{Q}'_2 - \mathbf{Q}'_1 = s_1(\mathbf{P}_{N_1} - \mathbf{P}_{N_1-1})$, and $\mathbf{Q}'_3 - \mathbf{Q}'_1 = s_2(\mathbf{P}_{N_1} - 2\mathbf{P}_{N_1-1} + \mathbf{P}_{N_1-2})$, where s_1, s_2 are the scaling factors usually set to 1, $\mathbf{Q}'$ represent the updated control points of the second curve. This process is included in the Align operation along with the trajectory updates ΔT_t prediction. As a result, the aggregation process can be expressed as:

$$[\mathbf{T}_1, \mathbf{T}_2] = \Big\{\{\mathbf{P}_i\}^{N_1}, \text{Align}\big(\{\mathbf{Q}_i\}^{N_2}\big)\Big\}, \tag{6}$$

where the control points of the original first curve and the control points of the updated second curve are concatenated together to get $N_1 + N_2$ control points. Then the corresponding modifications get $N_1 + N_2 + p + 1$ knots, which gives the aggregated long-term global trajectory. We simplify the expression of the above procedure in Sec. 3.1, *i.e.*, Align consists of the third-order alignment and residual ΔT_t update from two sub-curves to a global curve.

Framework. We chose DOT as our code base, and the encoder structure is the same as RAFT's. In Fig. 1 and L299-305 of the main paper, the image and event features are not fused at the extraction step but are merged to obtain the local motion representation M^{local} after the local correlation (consistent with RAFT) is computed. Depending on the network depth, the intermediate representations have dimensions from 32 to 256, and both local and global motion representations are 128. In the local motion estimation step, the trajectory decoder takes M^{local} along with reference frame and event features as input and outputs a tensor of size $(N_c{\times}3){\times}\text{H}{\times}\text{W}$. The first two channels are the control points and the rest channel is the visibility map. Warp and Align are differentiable parameter-free pure transformations, and Fusion is a fusion subnetwork.

1.2. Comparison with Feature Tracking

Our task of spatially dense point tracking is fundamentally more challenging than conventional sparse feature tracking. Unlike feature tracking, which focuses only on selected salient points, our method aims to track all image points across time, including those in low-texture or occluded regions. This dense requirement makes a direct comparison with event-based sparse feature tracking methods (typically evaluated on ED or EDS datasets) infeasible. Instead, we focus on more appropriate baselines such as dense TAP [2, 13] and optical flow [14, 17] methods.

Several prior event-based works propose related motion estimation methods, but face limitations in terms of applicability and evaluation. BFlow [7] models local curve trajectories and conducts experiments only on synthetic MultiFlow and real DSEC datasets. BlinkFlow and BlinkTrack are entirely based on synthetic Blender data (for both images and events), and the BlinkTrack dataset has not been released publicly, making fair evaluation impossible. FE-TAP [11] aims to combine events and TAP, but is not open-source at the time of writing. Currently, DSEC remains the only real-world dataset that provides high-quality events with dense motion annotations suitable for our task. Other real event datasets used for sparse feature tracking are not directly applicable.

In contrast, our evaluation strategy leverages real RGB frames from the TAP benchmark to simulate events, rather than using entirely synthetic data. Such Sim&Real evaluation protocols are widely adopted in prior works like BFlow [7], AccFlow [17], and DOT [13]. In addition to

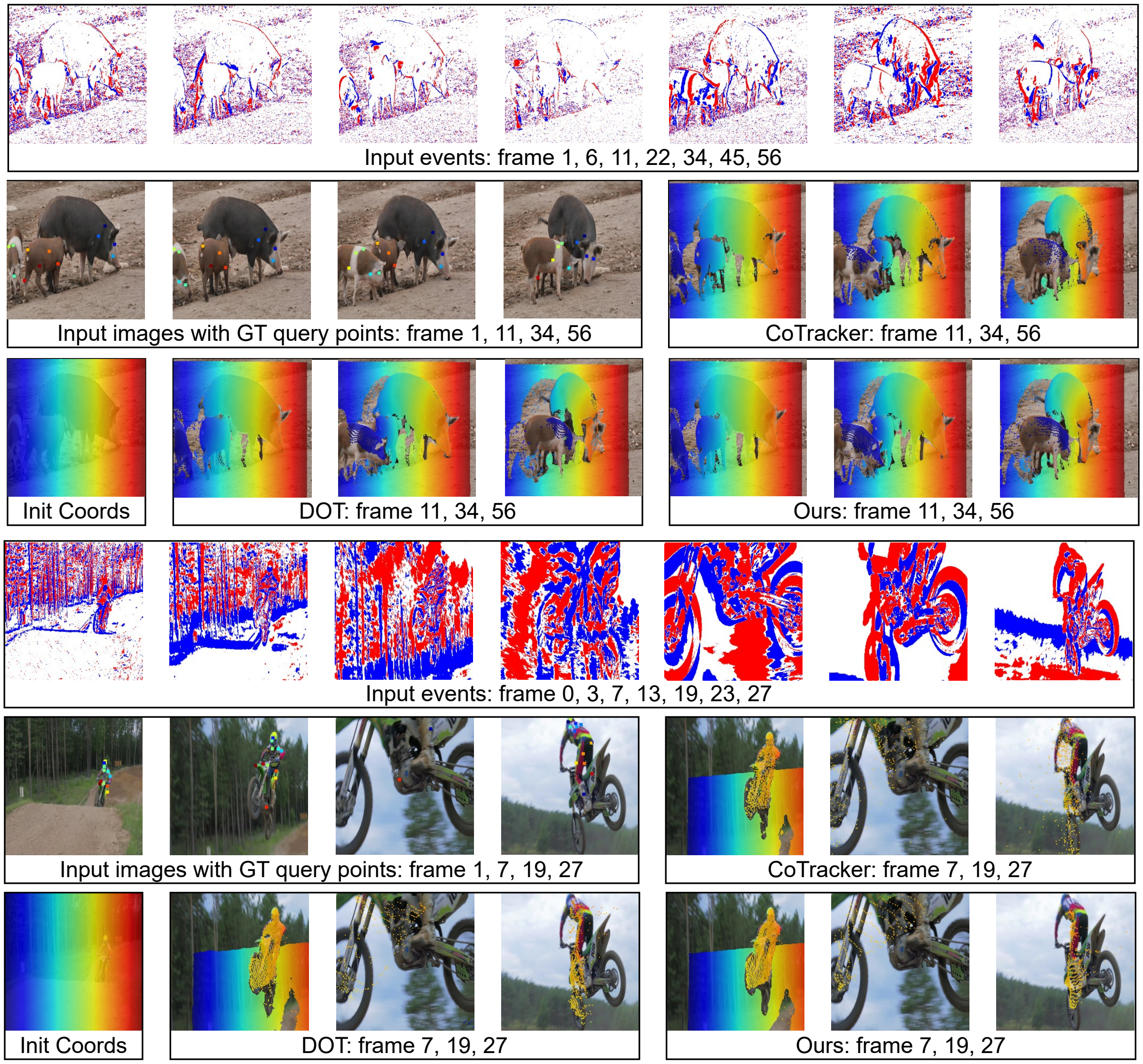

Input events: frame 1, 6, 11, 22, 34, 45, 56
Input images with GT query points: frame 1, 11, 34, 56
CoTracker: frame 11, 34, 56
Init Coords
DOT: frame 11, 34, 56
Ours: frame 11, 34, 56
Input events: frame 0, 3, 7, 13, 19, 23, 27
Input images with GT query points: frame 1, 7, 19, 27
CoTracker: frame 7, 19, 27
Init Coords
DOT: frame 7, 19, 27
Ours: frame 7, 19, 27

Figure 1. Visual comparisons of long-term dense point tracking on the *pigs* and *motocross-jump* sequences of TAP-Vid-DAVIS [2] dataset, with the ground-truth sparse query points of input images.

the real DSEC dataset, we also include qualitative results on the real-world ERF-X170FPS dataset to further support our conclusions. We believe that our use of both simulated (CVO, TAP-Vid) and real (DSEC, ERF-X170FPS) event data aligns with standard practices in dense point tracking, and sufficiently demonstrates the effectiveness and practicality of our proposed framework.

1.3. Experimental details

Datasets. We follow the common evaluation practices in CoTracker [9] and DOT [13]. The training set MOVI-F [8] contains over 10,000 videos with 7 frames each. The CVO test [17] and extended [13] sets contain $\sim$500 videos with 7 and 48 frames respectively. The real test TAP-Vid-DAVIS benchmark [2] includes 30 videos with $\sim$100 frames each. We simulate events for these three training and evaluation datasets using the vid2e [4] simulator. For the dense CVO dataset, we report the dense absolute error $\mathrm{EPE}_{\mathrm{all/vis/occ}}$ for all, visible and occluded points, as well as occlusion accuracy OA for estimated visible mask computed with IoU metric. For the sparse TAP-Vid-DAVIS dataset, we follow TAPNet [2] by reporting average Jaccard AJ, position accuracy $<\delta^x_{\mathrm{avg}}$, and occlusion accuracy OA. Additionally, we adopt the real-captured event-based optical flow dataset DSEC [5, 6] to verify the adaptation capacity, which contains 18 videos with $\sim$700 frames each. We follow the DSEC benchmark procedure and report the average of the endpoint error EPE and the angular error AE to measure the optical flow accuracy.

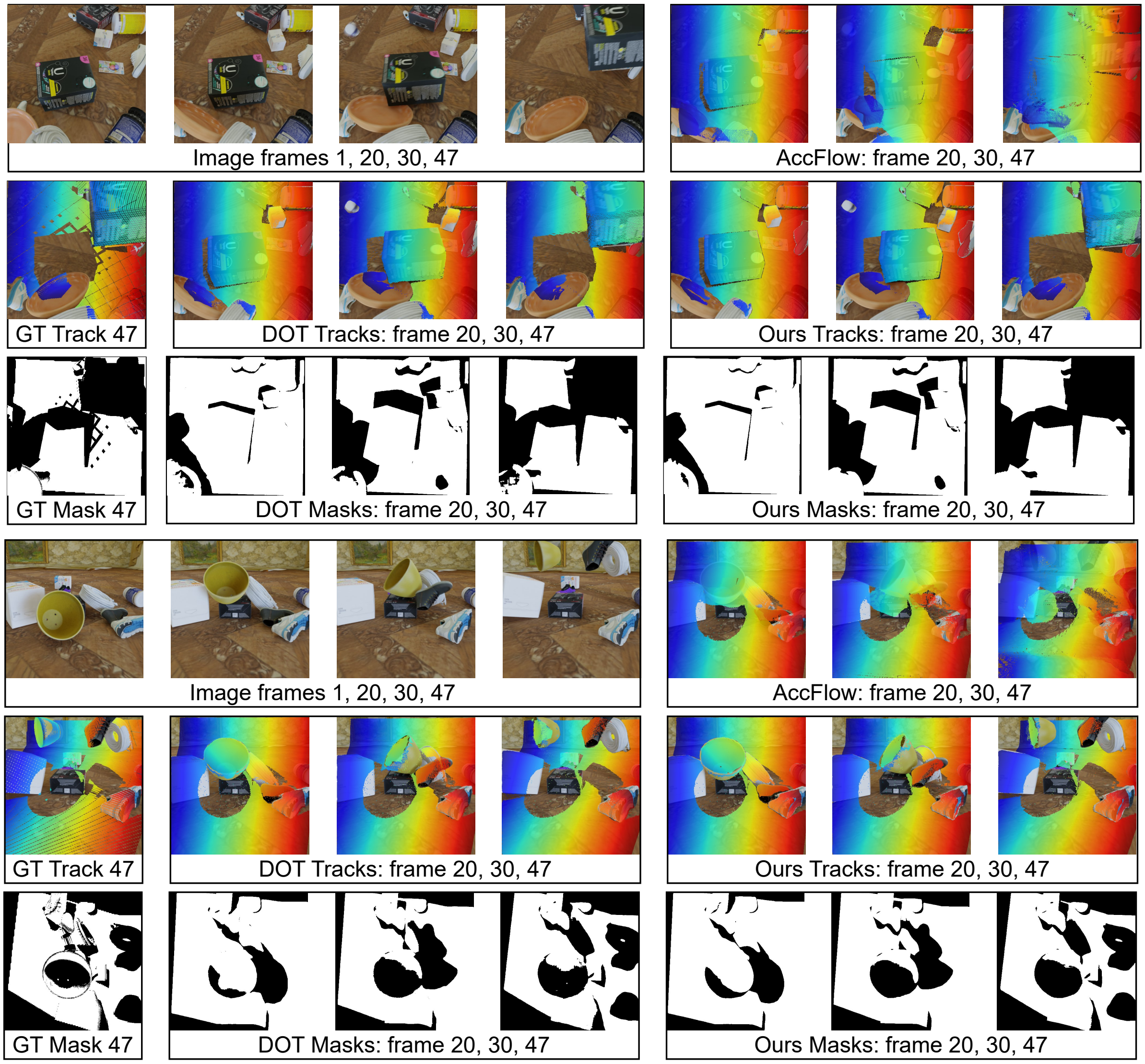


Image frames 1, 20, 30, 47
AccFlow: frame 20, 30, 47
GT Track 47
DOT Tracks: frame 20, 30, 47
Ours Tracks: frame 20, 30, 47
GT Mask 47
DOT Masks: frame 20, 30, 47
Ours Masks: frame 20, 30, 47
Image frames 1, 20, 30, 47
AccFlow: frame 20, 30, 47
GT Track 47
DOT Tracks: frame 20, 30, 47
Ours Tracks: frame 20, 30, 47
GT Mask 47
DOT Masks: frame 20, 30, 47
Ours Masks: frame 20, 30, 47

Figure 2. Visual comparisons of dense point tracking on the CVO extended set [13] with the ground-truth dense point coordinates and visible mask at the last (48-th) frame.

Implementation details. We implement our model with PyTorch, train it on MOVI-F and directly evaluate it on CVO and TAP-Vid-DAVIS datasets without any fine-tuning. Following DOT [13], our model is trained for 500k steps on 4 × NVIDIA L40 48G GPUs, using the Adam optimizer and OneCycle learning rate decay with a maximum of 10^{-4}. We also adopt the strategy of upgrading from multi-frame sparse to dense tracking in DOT to ensure temporal consistency. Following the practice of existing methods [16, 18], we convert the raw events into a grid representation as the model input, with the temporal bins set to $B = 5$. We choose 3 frame intervals as the randomly selected training samples, along with the random selection of up to 10 frames in different frame intervals. The loss hyperparameters are set to 1.0, 0.1, 0.1. Unless specifically mentioned, we evaluate our models and competitors on the same PC with a single RTX 3090 GPU.

Qualitative visual comparisons. Due to the length limitation, we provide more visualization results of point tracking in this appendix. Fig. 1 and Fig. 2 show the results on the TAP-Vid-DAVIS and CVO datasets, where we achieve better point tracking performance compared to recent competitive methods It is worth noting that the TAP-Vid-DAVIS dataset [2] only provides sparse query point trajectories for each frame, so we plot the positions of the ground-truth query points directly on the input image, while initial point coordinates (*Init Coords*) represent the initial coordinates of dense point tracking. In contrast, the CVO extended set [13] has only the last frame of the dense point motion vectors, so

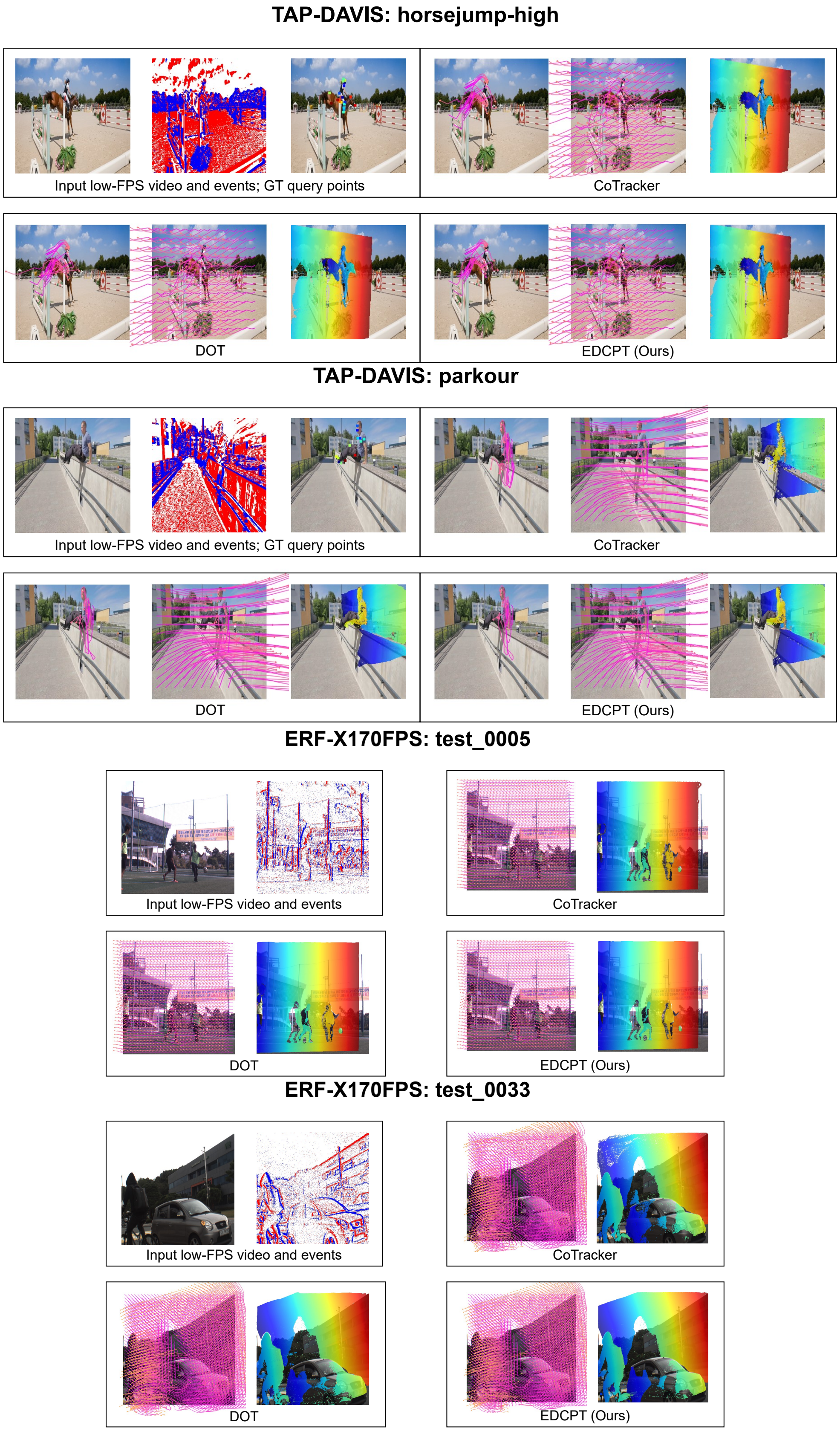
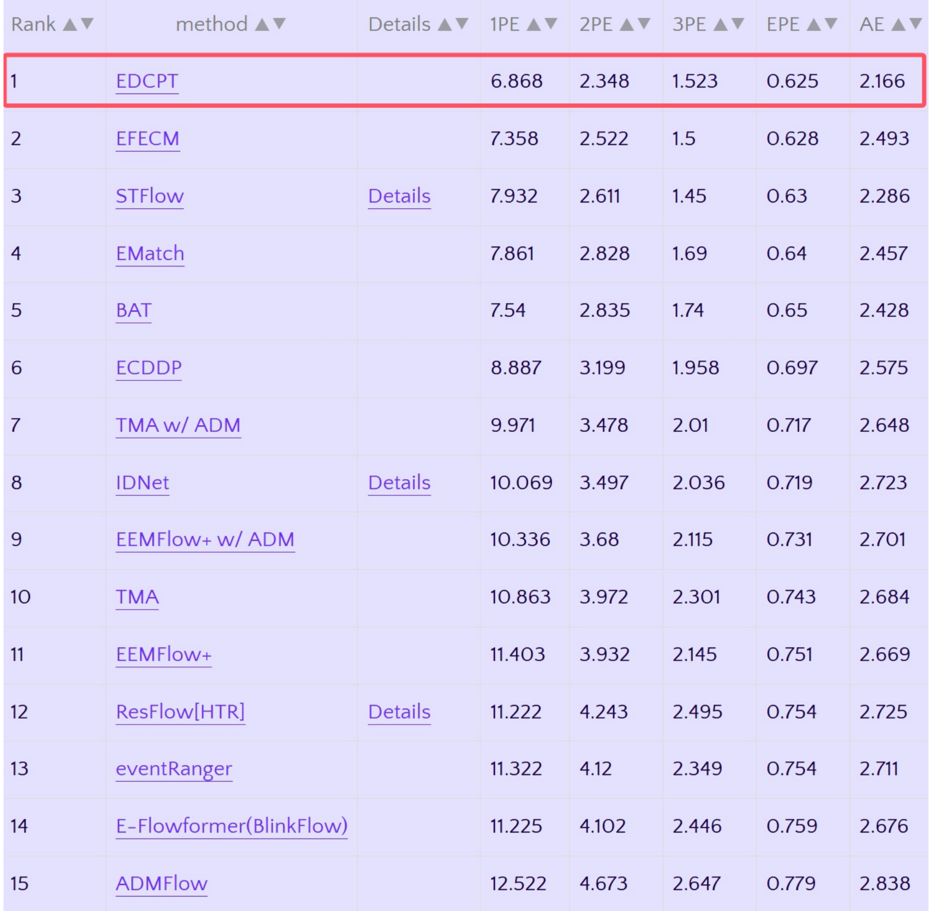

TAP-DAVIS: horsejump-high
Input low-FPS video and events; GT query points
CoTracker
DOT
EDCPT (Ours)
TAP-DAVIS: parkour
Input low-FPS video and events; GT query points
CoTracker
DOT
EDCPT (Ours)
ERF-X170FPS: test_0005
Input low-FPS video and events
CoTracker
DOT
EDCPT (Ours)
ERF-X170FPS: test_0033
Input low-FPS video and events
CoTracker
DOT
EDCPT (Ours)

Figure 3. Screenshots from our demo video, including comparisons of dense and continuous point tracking trajectories on the commonly used TAP-Vid-DAVIS benchmark [2] and the real-world ERF-X170FPS dataset [10].



Rank ▲▼	method ▲▼	Details ▲▼	1PE ▲▼	2PE ▲▼	3PE ▲▼	EPE ▲▼	AE ▲▼
1	EDCPT		6.868	2.348	1.523	0.625	2.166
2	EFECM		7.358	2.522	1.5	0.628	2.493
3	STFlow	Details	7.932	2.611	1.45	0.63	2.286
4	EMatch		7.861	2.828	1.69	0.64	2.457
5	BAT		7.54	2.835	1.74	0.65	2.428
6	ECDDP		8.887	3.199	1.958	0.697	2.575
7	TMA w/ ADM		9.971	3.478	2.01	0.717	2.648
8	IDNet	Details	10.069	3.497	2.036	0.719	2.723
9	EEMFlow+ w/ ADM		10.336	3.68	2.115	0.731	2.701
10	TMA		10.863	3.972	2.301	0.743	2.684
11	EEMFlow+		11.403	3.932	2.145	0.751	2.669
12	ResFlow[HTR]	Details	11.222	4.243	2.495	0.754	2.725
13	eventRanger		11.322	4.12	2.349	0.754	2.711
14	E-Flowformer(BlinkFlow)		11.225	4.102	2.446	0.759	2.676
15	ADMFlow		12.522	4.673	2.647	0.779	2.838

Figure 4. Screenshot of the DSEC optical flow leaderboard [5] on Mar. 7, 2025 from https://dsec.ifi.uzh.ch/uzh/dsec-flow-optical-flow-benchmark. Our EDCPT achieves the first rank in the DSEC optical flow benchmark.

we provide the visualization of the ground-truth points (*GT points*) from the Init Coords of the first frame to last frame.

Demo video on the real-captured event dataset. The demo video is uploaded to https://figshare.com/articles/media/EDCPT_demo_video/29656805. In this appendix, we provide screenshots of the demo videos. Fig. 3 shows the video screenshots for the comparison results of dense and continuous point tracking in four scenes, including the *horsejump-high* and *parkou* sequences on the TAP-Vid-DAVIS dataset [2], and the *test_0005* and *test_0033* sequences on the real-captured ERF-X170FPS dataset [10]. We chose to compare with two recent SOTA methods, CoTracker [9] and DOT [13]. The visualization of dense and continuous point tracking trajectories is shown in three separate forms: query point trajectories, grid trajectories, and dense point coordinate shifts.

In particular, the ERF-X170FPS dataset is proposed in CBMNet [10] originally for video frame interpolation in highly dynamic scenarios. Both its image and event data are real-captured and of high quality, we utilize it to further validate the applicability of our framework on real-world data. Since this dataset lacks motion annotations and query point coordinates, we only show grid trajectories and dense point coordinate shifts. As shown in the demo video and screenshots in Fig. 3, our framework achieves better point tracking performance compared to Cotracker and DOT for small objects (soccer ball in *test_0005*) and curve motion (camera rotation in *test_0033*). In particular, in Fig. 5 we provide a typical case of non-linear motion with players kicking a soccer ball. We can find that the curvilinear motion representation is more suitable for modeling real-world

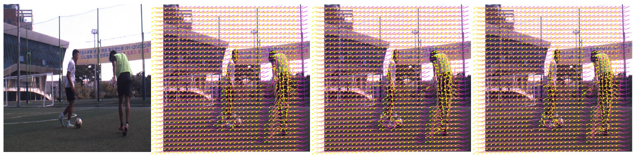


Image CoTracker DOT Ours

Figure 5. Comparisons of non-linear motion modeling capabilities on typical real-captured data.

Table 1. Ablations at standard frame rate.

Model	CVO $\text{EPE}_{\text{all/vis/occ}}$ ↓	DAVIS AJ / $<\delta^x_{\text{avg}}$ ↑
CoTracker	1.89 / 0.63 / 7.05	61.1 / 74.6
DOT	1.83 / 0.59 / 6.95	61.6 / 75.5
DOT+Events	1.79 / 0.57 / 6.80	62.8 / 75.9
DOT+Curve (image only Abl.)	1.88 / 0.61 / 6.84	62.1 / 75.7
Ours (full model)	1.76 / 0.55 / 6.73	63.8 / 76.3

non-linear motion.

Experimental result on the DSEC benchmark. To qualitatively validate the applicability of our scheme on real captured events data, we conduct experiments on DSEC [5] , a widely used benchmark for optical flow estimation, and submit the results on the test set to the DSEC online server. In Tab. 3, we compare the performance of various SOTA methods under different training and input settings, here we also provide a screenshot of the DSEC online leaderboard in Fig. 4. Our proposed EDCPT achieves the first rank in the DSEC optical flow benchmark.

Zero-shot (Sim2Real) evaluation. We need to clarify that the standard event simulation pipeline [4], video interpolation (VFI) + event simulation (ESIM [15]), introduces new information through additional motion priors from the VFI model, which is pretrained on large-scale video datasets. So we pre-train the proposed model on large-scale simulated data with simulation events and verify the generalization performance of the model by zero-shot evaluation across datasets. In addition to the zero-shot experimental results on TAP-Vid-DAVIS and CVO in the main paper, our TAP results on ERF-X170FPS (Fig. 3 and Fig. 5) are also zero-shot. We also submit the pre-trained model before finetuning to the DSEC benchmark, achieving EPE/AEE: 0.82/2.52, outperforming GMA's 0.94/2.66.

Ablations at standard frame rate. In Tab. 3 and Tab. 6 of the main paper, we provide ablations without input events and curve representations at low frame rate (skip 2 or 3 frames), focusing on evaluating the ability to model nonlinear motion. We have provided ablations of our model by interpolating with *linear* and *quad*ratic motion assumptions in Tab. 6. On the other hand, the results for the baselines (including DOT [13]) in Tab. 6 are obtained by taking the linear motion assumption. Here we reevaluate with the quadratic assumption, and get a slight raise of DOT's DAVIS results to 50.9 / 65.7, but it is not as strong as learning curves such as ours.

Moreover, we provide more results at standard frame rate (no skip frames) in Tab. 1. The results consistently show that our model benefits not only from the ability of the curve representation to handle complex nonlinear motion, but also from the concise and valuable motion information in events, allowing for more accurate point tracking.

Longer sequences. We provide more qualitative comparisons of the MOVi-F pre-trained model on another two TAP-Vid sub-benchmark in Tab. 2, verifying the advantages of ECDPT on longer sequences. Note that for efficient inference, num_tracks is set to 1024 for DOT and ECDPT. This observation is consistent with that in the main paper on TAP-DAVIS.

Table 2. More quantitative results on TAP-Vid benchmark.

Method	Kinetics (First)			RGB-S. (Strided)		
	AJ ↑	$<\delta^x_{\text{avg}}$ ↑	OA ↑	AJ ↑	$<\delta^x_{\text{avg}}$ ↑	OA ↑
TAP-Net	38.5	54.4	80.6	59.9	72.8	90.4
CoTracker	44.8	63.2	81.2	74.1	85.2	92.3
DOT	48.3	61.1	83.7	81.2	90.0	94.1
EDCPT (Ours)	**49.7**	**63.2**	**84.5**	**83.3**	**91.4**	**94.9**

References

[1] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. Flowtrack: Revisiting optical flow for long-range dense tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19268–19277, 2024. 2

[2] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, pages 13610–13626, 2022. 2, 3, 4, 5, 6

[3] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 1

[4] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3586–3595, 2020. 3, 7

[5] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters (RA-L)*, 2021. 3, 6, 7

[6] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-RAFT: Dense optical flow from event cameras. In *International Conference on 3D Vision (3DV)*, pages 197–206, 2021. 3

[7] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense continuous-time optical flow from event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(7):4736–4746, 2024. 1, 2

[8] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3749–3761, 2022. 3

[9] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, 2024. 3, 6

[10] Taewoo Kim, Yujeong Chae, Hyun-Kurl Jang, and Kuk-Jin Yoon. Event-based video frame interpolation with cross-modal asymmetric bidirectional motion fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18032–18042, 2023. 5, 6

[11] Jiaxiong Liu, Bo Wang, Zhen Tan, Jinpu Zhang, Hui Shen, and Dewen Hu. Tracking any point with frame-event fusion network at high frame rate. *arXiv preprint arXiv:2409.11953*, 2024. 2

[12] Jianqin Luo, Zhexiong Wan, Yuxin Mao, Bo Li, and Yuchao Dai. Continuous parametric optical flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 23520–23532, 2023. 1

[13] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: Connecting the dots. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 4, 6, 7

[14] Michal Neoral, Jonáš Šerỳch, and Jiří Matas. Mft: Long-term tracking of every pixel. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 6837–6847, 2024. 1, 2

[15] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: an open event camera simulator. In *Conference on Robot Learning*, pages 969–982, 2018. 7

[16] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3857–3866, 2019. 4

[17] Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. Accflow: backward accumulation for long-range optical flow. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12119–12128, 2023. 1, 2, 3

[18] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 989–997, 2019. 4