Matching Every Pair to Track Every Point: PairFormer for All-Pairs Tracking and Video Trajectory Fields

Anonymous CVPR submission

Abstract

Tracking-any-point (TAP) answers query-conditioned correspondence but leaves the dense, all-pairs structure of a video implicit. We formulate All-Pairs Tracking (APT): given a video, predict dense displacement and visibility for every source-target frame pair, from which per-pixel trajectories can be read out. To this end, we propose PairFormer, a feed-forward transformer that addresses APT in a single pass. A spatio-temporal patch encoder computes temporally conditioned features for all frames. CorrBank constructs a learnable correlation memory for each frame pair to obtain pairwise motion tokens. A broadcast motion mixer aggregates information across space and time and refines these tokens with global context. A trajectory head then predicts full-resolution displacement for each pair, yielding a coherent all-pairs trajectory field. To support APT at scale, we develop PAIRender, a data platform that synthesizes photo-realistic dynamic scenes with dense annotations. From PAIRender we derive a training set (π-R10K) and a benchmark (APT-Bench) with an all-to-all evaluation protocol. Experiments show that PairFormer achieves strong performance on APT-Bench and competitive results on standard TAP benchmarks. Code and dataset will be released upon publication.

1. Introduction

Understanding motion in video is more than linking a few queried points [12, 13, 18, 22] or estimating pairwise optical flow [20, 27, 37, 40]; it requires a representation that exposes how *every* image location relates to *every* frame across the sequence. TAP delivers long-range, occlusion-aware but *sparse*, query-conditioned tracks [12, 22], whereas optical flow focuses on adjacent-frame correspondences [20, 27, 37, 40]; in contrast, APT requires an explicit *all-pairs* dense field, enabling uniform reasoning and temporal consistency across the entire sequence.

We advocate *All-Pairs Tracking (APT)*: given a video, predict dense correspondences for *every* ordered frame pair, together with a visibility probability that marks traceable mappings. APT strictly generalizes two-frame optical flow [20, 27, 37, 40] and subsumes TAP-style point tracking [12, 22]. From an explicit all-pairs field, trajectories for any pixel can be read out on demand rather than obtained by propagation from a single start frame. Framed this way, matching at any temporal offset is handled uniformly, and sequence-wide temporal consistency becomes a first-class modeling objective.

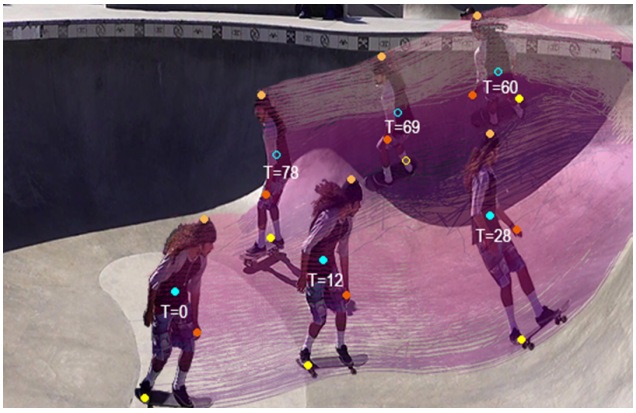

Figure 1. **All-Pairs Tracking (APT) illustration.** APT estimates a dense trajectory field for all pixels; here we visualize trajectories of pixels on the person region over the entire sequence, where hollow markers indicate occluded points.

Realizing APT poses several challenges. First, the number of frame pairs grows quadratically with sequence length. Second, the relevant context for each correspondence is often spread across distant frames and spatial locations. Third, repeated textures, large motions, and occlusions introduce substantial ambiguity. Existing handcrafted cost volumes and local propagation schemes [5, 28, 34, 37] struggle to aggregate such non-local context and to maintain consistency at the scale of an entire sequence.

To address these challenges, we introduce *PairFormer*, a transformer architecture guided by a simple principle: *first construct pairwise correspondences, then enforce global*

consistency across the sequence. The model comprises four components. A **Spatio-Temporal Patch Encoder** first produces temporally conditioned patch representations for all frames. **CorrBank** then converts each ordered frame pair into a learnable correlation memory and performs memory-augmented matching to yield *pairwise motion tokens*. Next, a **Broadcast Motion Mixer** aggregates information along trajectories (per-trajectory), within frames (per-frame), and across the sequence (all-frames), and at selected depths *broadcasts* the aggregated context back to the pairwise tokens to resolve ambiguities, including occlusion and reappearance. Finally, a **Trajectory Field Decoder** maps the refined tokens to per-pair displacement and visibility, yielding an explicit all-pairs trajectory field.

To support large-scale training and evaluation, we develop a Blender-based platform (*PAIRender*) featuring diverse environments, moving characters, and natural camera trajectories, in the spirit of recent synthetic data generators for correspondence tasks [17, 43]. The platform generates synchronized RGB frames, per-pixel 2D/3D trajectories, depth, semantic labels, visibility, and camera parameters, enabling dense supervision and controlled experimentation. From PAIRender we derive a training dataset (π*-R10K*) and a held-out benchmark (*APT-Bench*) with an *all-to-all* evaluation protocol that queries points sampled from every frame, encouraging sequence-wide reasoning rather than first-frame propagation as in TAP-style benchmarks [12].

In summary, our contributions are:

- We introduce **APT** as an explicit, dense, visibility-aware correspondence formulation that generalizes optical flow and TAP while targeting sequence-wide consistency.
- We propose **PairFormer**, a feed-forward transformer architecture that constructs a globally consistent all-pairs trajectory field in a single forward pass.
- We develop **PAIRender** and release π*-R10K* and *APT-Bench*, providing dense all-pairs supervision and an all-to-all evaluation protocol tailored to APT.

2. Related Work

Optical flow. The notion of optic flow in perception dates back to Gibson's work [16], and we refer to Niehorster et al. [30] for a historical overview. In computer vision, optical flow typically denotes the dense 2D motion field relating two consecutive video frames, mapping each pixel in the first frame to a location in the second [20, 27]. Most classical and modern approaches share a similar optimization pattern: an initial flow estimate is obtained (for example, assuming zero motion or a simple prior), a matching cost is computed from appearance features, this cost is regularized by smoothness assumptions or learned priors, and the flow is iteratively refined in a local neighborhood [4, 21, 34, 37, 42]. Recent learning-based methods such as RAFT and SEA-RAFT push this paradigm further [37, 40]. They construct a correlation volume between feature maps extracted by a deep backbone [19], maintain a low-resolution flow field that is updated recurrently by convolutional networks using both flow and correlations, and finally upsample the prediction to full resolution using specialized upsampling modules [15, 21, 33]. Our formulation of All-Pairs Tracking (APT) is related in spirit—dense correspondences remain central—but differs in two key ways: (i) we target *all* ordered frame pairs rather than only adjacent ones, and (ii) we replace handcrafted correlation volumes and recurrent conv refinements with a transformer-based architecture that builds a learnable correlation memory (CorrBank) and a global motion field over the entire sequence.

Flow-based point tracking. Dense optical flow is often used as a primitive for constructing longer trajectories. A common strategy is to link per-frame flow estimates over time, forming tracks by chaining displacement vectors and detecting occlusions to prevent drift [32, 35]. More recent systems extend this idea by using multi-step flows [7–9, 28] or by incorporating temporal priors learned from data [5], allowing trajectories to bridge over occlusions instead of terminating. These approaches can yield dense multi-frame tracks, but they rely on a separate optical flow model as a pre-processing stage and treat tracking as a post-hoc operation on its outputs.

Point trackers without flow. A second line of work tracks points directly, without explicitly predicting dense optical flow. Classical methods such as the Lucas–Kanade family [27, 38] track sparse features using local image alignment. More recently, Harley et al. [18] proposed a deep point tracker with a sliding multi-frame window that allows tracks to pass through occlusions, and Doersch et al. [12] introduced the "Tracking Any Point" (TAP) benchmark, which catalyzed a wave of follow-up methods. Subsequent work has explored multi-point interactions [22, 23], improved initialization and re-initialization [13], expanded correlation neighborhoods [2, 6], transformer-based designs [25, 26], and post-hoc densification from sparse tracks [24]. Among these, CoTracker3 [22] represents a strong state of the art for TAP-style tracking. However, CoTracker-type models operate on a *sparse* set of user-chosen queries, and performance depends on how these queries are distributed [23]. Memory and compute constraints limit the number of simultaneously tracked points to a few thousand.

Training dataset. For point tracking, common training sources include Kubric [17], FlyingThings++ [18], and PointOdyssey [43]. Performance can be sensitive to how these datasets are combined and scheduled [40], and different works adopt different curricula.

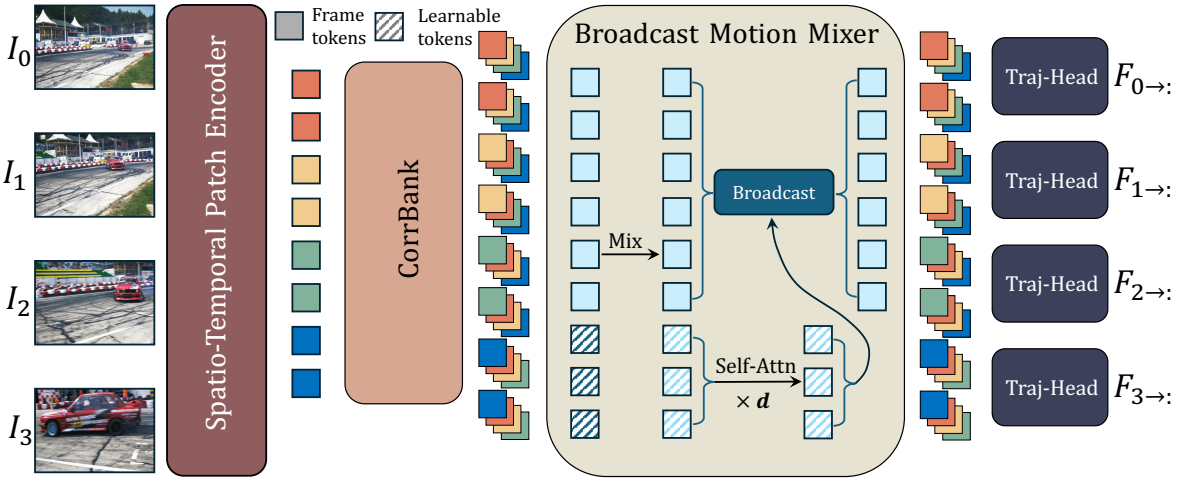

Figure 2. **PairFormer architecture for All-Pairs Tracking (APT).** Given a video, the **Spatio-Temporal Patch Encoder** maps all frames to temporally conditioned patch features on a regular grid. **CorrBank** performs memory-augmented matching to produce *pairwise motion tokens*. The **Broadcast Motion Mixer** applies trajectory-wise mixing, followed by per-frame and all-frames context aggregation and a broadcast step that conditions each motion token on global context. Finally, the **Trajectory Field Decoder** (Traj-Head) applies a DPT-style head to each pair and outputs full-resolution displacement. $F_{s\to:}$ denotes displacement field from frame s to all other frames.

3. Method

3.1. Problem Formulation

Let $\mathcal{V} = \{I_t\}_{t=1}^{T}$ be a video, where each frame I_t is defined on an image domain $\Omega \subset \mathbb{R}^2$. *All-Pairs Tracking (APT)* aims to estimate, for every ordered frame pair (s, t), a dense forward correspondence from frame s to frame t.

We represent the correspondence from a pixel $x \in \Omega$ in frame s to its location in frame t by

$$\phi_{s\to t}(x) = x + F_{s\to t}(x), \tag{1}$$

where $F_{s\to t} : \Omega \to \mathbb{R}^2$ is a displacement field. To handle occlusion and out-of-view cases, we further predict a *visibility map*

$$V_{s\to t} : \Omega \to [0, 1], \tag{2}$$

which we interpret as the probability that $\phi_{s\to t}(x)$ is visible and corresponds to the same physical point in frame t. Given the family of mappings $\{\phi_{s\to t}\}_{t=1}^{T}$ for a fixed source frame s, the trajectory of a pixel x is the sequence $\{\phi_{s\to t}(x)\}_{t=1}^{T}$.

3.2. Network Architecture

We propose *PairFormer*, a transformer architecture tailored to APT and guided by a simple principle: *first construct pairwise correspondences, then enforce global consistency across the sequence*. As illustrated in Fig. 2, the model comprises four components. A **Spatio-Temporal Patch Encoder** (ST-Patch Encoder) maps each frame to temporally conditioned patch representations. **CorrBank** converts each ordered frame pair into a learnable *correlation memory* and performs memory-augmented matching to produce *pairwise motion tokens*. The **Broadcast Motion Mixer** (BMM) performs trajectory-wise and spatio-temporal context mixing, and *broadcasts* the aggregated context back to refine the pairwise motion tokens. Finally, a **Trajectory Field Decoder** applies a DPT-style head [31] independently to each pair, mapping refined tokens to full-resolution displacement $F_{s\to t}$, visibility $V_{s\to t}$, and confidence $C_{s\to t}$, forming an explicit all-pairs trajectory field.

ST-Patch Encoder. Given video frames $\{I_t\}_{t=1}^{T}$, the ST-Patch Encoder splits each frame into a regular grid of patches and maps each patch to a feature vector. We then add a temporal embedding (shared across patches within a frame) and a 2D positional encoding on the patch grid, and apply a stack of self-attention blocks over all frames jointly. This yields temporally conditioned patch representations for all frames, which serve as the input to CorrBank and the subsequent motion modeling modules.

CorrBank. Let $Z_s, Z_t \in \mathbb{R}^{P\times D}$ denote ST-Patch Encoder features for frames s and t on a patch grid with P locations. For each ordered pair (s, t), we first form a residual feature map

$$R_{s,t} = h(Z_t - Z_s), \tag{3}$$

where $h(\cdot)$ is a small convolutional network that emphasizes local motion cues. CorrBank maintains a set of learnable *correlation tokens* $A \in \mathbb{R}^{K\times D}$, shared across all pairs. We use cross-attention to let the residual features query these tokens and obtain a *pair-specific correlation memory* on the

patch grid $M_{s,t} \in \mathbb{R}^{P\times D}$:

$$M_{s,t} = \mathrm{Attn}\big(Q = R_{s,t},\, K = A,\, V = A\big). \quad (4)$$

In a second step, we perform *memory-augmented matching*: source features Z_s act as queries, target features Z_t provide keys, and the correlation memory $M_{s,t}$ supplies values to produce *pairwise motion tokens* $Y_{s,t} \in \mathbb{R}^{P\times D}$:

$$Y_{s,t} = \mathrm{Attn}\big(Q = Z_s,\, K = Z_t,\, V = M_{s,t}\big). \quad (5)$$

Thus, CorrBank replaces an explicit 4D cost volume with a learnable, token-based correlation module that can leverage efficient kernels such as FlashAttention [10].

Broadcast Motion Mixer (BMM). Let T denote the number of frames and P the number of patches per frame. Given pairwise motion tokens $Y_{s,t} \in \mathbb{R}^{P\times D}$ for each source frame s and target frame $t \in \{1,\dots,T\}$, BMM refines them in two stages.

Trajectory-wise mixing. For each source frame s and patch index p, we consider the temporal sequence of motion tokens $\{Y_{s,\tau}(p)\}_{\tau=1}^{T} \in \mathbb{R}^{T\times D}$. BMM introduces K learnable context tokens $U \in \mathbb{R}^{K\times D}$, shared across all trajectories, and concatenates them with this sequence:

$$X_{s,p} = \big[U;\, Y_{s,1}(p), \dots, Y_{s,T}(p)\big] \in \mathbb{R}^{(K+T)\times D}. \quad (6)$$

Several self-attention blocks are applied to $X_{s,p}$. We keep the first K output positions as *trajectory context tokens* $H_{s,p} \in \mathbb{R}^{K\times D}$, and the remaining T positions as updated motion tokens $\tilde{Y}_{s,\tau}(p)$. In this way, the learnable tokens $H_{s,p}$ absorb information from all time steps and form a compact summary of the motion along the trajectory for patch (s,p).

Context mixing (per-frame and all-frames). For each source frame s, we feed the P trajectory context tokens $\{H_{s,p}\}_{p=1}^{P}$ into a stack of d per-frame self-attention blocks (with s as the batch index), yielding per-frame features $F_s \in \mathbb{R}^{P\times D}$ that encode spatial structure and context within frame s. We then concatenate $\{F_s\}_{s=1}^{T}$ along the token dimension and pass the resulting $T \times P$ spatio-temporal tokens through another stack of d self-attention blocks, obtaining scene-level features $G \in \mathbb{R}^{T\times P\times D}$ that mix information across both space and time.

Broadcast refinement. At selected depths, we construct, for each (s,p), a small context set $S_{s,p} = \big[F_s(p),\, G_{s,p}\big]$, where $F_s(p)$ and $G_{s,p}$ denote the per-frame and scene-level features at that patch, respectively. We then refine the motion tokens at patch (s,p) via cross-attention:

$$\tilde{Y}^{*}_{s,\cdot}(p) = \mathrm{Attn}\big(Q = \tilde{Y}_{s,\cdot}(p),\, K = S_{s,p},\, V = S_{s,p}\big). \quad (7)$$

The updated sequence $\tilde{Y}^{*}_{s,\cdot}(p) \in \mathbb{R}^{T\times D}$ is finally reshaped back to refined pairwise motion tokens $Y^{*}_{s,t} \in \mathbb{R}^{P\times D}$. This broadcast step conditions each pairwise motion token on both its own trajectory summary and the global spatio-temporal context, while preserving its dependence on the specific frame pair (s,t).

Trajectory Field Decoder. The final stage maps the refined pairwise motion tokens to dense per-pixel outputs. Given $Y^{*}_{s,t} \in \mathbb{R}^{P\times D}$ on the patch grid for each ordered pair (s,t), we apply a DPT-style decoder head [31] with learned upsampling independently to each pair. This head takes $Y^{*}_{s,t}$ and predicts a four-channel full-resolution map, comprising 2D displacement $F_{s\to t} \in \mathbb{R}^{H_{\mathrm{im}}\times W_{\mathrm{im}}\times 2}$, a visibility field $V_{s\to t} \in [0,1]^{H_{\mathrm{im}}\times W_{\mathrm{im}}}$, and a confidence field $C_{s\to t} \in \mathbb{R}^{H_{\mathrm{im}}\times W_{\mathrm{im}}}$.

3.3. Training Scheme

We denote ground-truth displacement and visibility by $F^{\star}_{s\to t}$ and $V^{\star}_{s\to t}$ when available. For a pixel $x \in \Omega$ in source frame s and a target frame t, we define the per-pixel residual

$$r_{s\to t}(x) = \big\|F_{s\to t}(x) - F^{\star}_{s\to t}(x)\big\|_1. \quad (8)$$

The overall training objective is a weighted sum

$$\mathcal{L} = \lambda_{\mathrm{traj}}\,\mathcal{L}_{\mathrm{traj}} + \lambda_{\mathrm{conf}}\,\mathcal{L}_{\mathrm{conf}} + \lambda_{\mathrm{vis}}\,\mathcal{L}_{\mathrm{vis}} + \lambda_{\mathrm{corr}}\,\mathcal{L}_{\mathrm{corr}}. \quad (9)$$

Trajectory loss. We supervise displacements with a standard per-pixel ℓ_1 loss:

$$\mathcal{L}_{\mathrm{traj}} = \mathbb{E}_{(s,t),x}\big[r_{s\to t}(x)\big], \quad (10)$$

where the expectation is taken over all supervised frame pairs and all valid pixels.

Confidence adjustment. PairFormer predicts an auxiliary confidence field $C_{s\to t}(x) > 0$ alongside the displacement $F_{s\to t}(x)$. We treat $C_{s\to t}(x)$ as a learned weight on the residual, regularized by a log term. For a single pixel (s,t,x), the confidence-adjusted penalty is

$$\ell_{\mathrm{conf}}(s,t,x) = r_{s\to t}(x)\, C_{s\to t}(x) - \alpha \log C_{s\to t}(x), \quad (11)$$

with $\alpha = 0.3$ controlling the strength of regularization. The overall confidence loss is then

$$\mathcal{L}_{\mathrm{conf}} = \mathbb{E}_{(s,t),x}\big[\ell_{\mathrm{conf}}(s,t,x)\big]. \quad (12)$$

The first term encourages larger confidence on pixels with small residuals, while the negative log term prevents the model from driving $C_{s\to t}(x)$ to zero.

Figure 3. Sample image and dense flow field rendered from our PAIRender data platform.

Visibility loss. In addition to displacement, PairFormer predicts a visibility field $V_{s\to t}(x)$ that should match the ground-truth visibility $V^{\star}_{s\to t}(x)$. We supervise visibility with a standard pixel-wise binary cross-entropy loss:

$$\mathcal{L}_{\text{vis}} = \mathbb{E}_{(s,t),x}\left[\ell_{\text{BCE}}\big(V_{s\to t}(x),\, V^{\star}_{s\to t}(x)\big)\right], \tag{13}$$

where ℓ_{BCE} denotes the binary cross-entropy between a predicted probability and a binary label.

Correspondence regularization. Synthetic supervision from PAIRender provides temporally consistent trajectories, which allows us to identify pixels that correspond to the same physical point across frames. Let Ω_{corr} denote a set of matched pixel pairs $((s,x),(t,x'))$ lying on the same ground-truth trajectory. For such pairs, the predicted all-pairs trajectory fields starting from x and x' should agree. We define a trajectory descriptor $\mathbf{q}_s(x) \in \mathbb{R}^{T\times 2}$ as the stack of forward-mapped positions

$$\mathbf{q}_s(x) = \big[\phi_{s\to 1}(x),\, \ldots,\, \phi_{s\to T}(x)\big], \tag{14}$$

and similarly $\mathbf{q}_t(x')$ for frame t. The correspondence regularizer is

$$\mathcal{L}_{\text{corr}} = \mathbb{E}_{((s,x),(t,x'))\in\Omega_{\text{corr}}}\left[\big\|\mathbf{q}_s(x) - \mathbf{q}_t(x')\big\|_1\right]. \tag{15}$$

This term encourages points on the same trajectory to share a consistent all-pairs prediction, beyond what is enforced by per-pair supervision alone.

Together, Eqs. (10)–(15) provide complementary supervision: $\mathcal{L}_{\text{traj}}$ anchors displacements to ground-truth flow, $\mathcal{L}_{\text{conf}}$ adaptively reweights and regularizes per-pixel residuals, $\mathcal{L}_{\text{vis}}$ trains the visibility field with a probabilistic interpretation, and $\mathcal{L}_{\text{corr}}$ ties together predictions for pixels along the same physical trajectory. Their weighted combination in Eq. (9) is used for all experiments. Empirically, we set $(\lambda_{\text{traj}}, \lambda_{\text{conf}}, \lambda_{\text{vis}}, \lambda_{\text{corr}}) = (1, 1, 1, 0.3)$ in all experiments, which we find yields stable training.

4. Data Platform: PAIRender

Data-driven modeling of dynamic scenes is limited by the lack of large-scale datasets that provide *dense*, *reliable*, and *long-range* supervision. Existing synthetic corpora are modest in scale and biased toward rigid or short-term motion, with sparse annotations that are insufficient for learning all-pairs correspondence at the sequence level. To address this, we develop *PAIRender*, a scalable Blender-based platform that generates photorealistic dynamic videos with per-pixel ground truth tailored to APT.

π-R10K dataset. Using PAIRender, we assemble *π-R10K*, a training dataset for all-pairs trajectory field estimation. The collection (Fig. 3) contains 10,000 scenes, each a sequence of 60 frames at 512×512 resolution. Diversity is characterized along three aspects: (i) **Environment**: indoor and outdoor layouts from public asset libraries and procedural generation; (ii) **Dynamics**: articulated humans and everyday objects undergoing both rigid and non-rigid motion, with frequent occlusions; (iii) **Camera**: smooth trajectories centered on regions with significant motion, obtained by retargeting real-world camera motions. Each sequence provides synchronized RGB, dense 2D/3D trajectories, depth, visibility masks, semantic labels, and camera parameters, enabling the supervision scheme in Sec. 3. PAIRender is fully programmable, so scene count and motion patterns can be easily scaled.

APT-Bench benchmark. To evaluate APT models, we construct *APT-Bench*, a held-out test set of 100 sequences curated from PAIRender. Each sequence contains 120 frames at 1920×1080 resolution. Unlike established point-tracking benchmarks that evaluate only points sampled from the first frame, APT-Bench uses an *all-to-all* protocol that samples points from every frame and evaluates trajectories across the entire sequence.

Table 1. TAP benchmark results in terms of δ_{avg} (higher is better).

Method	Bad.	Cro.	Dav.	Dri.	Ego.	Kin.	Rgb.	Rob.	Avg.
RAFT	23.7	29.3	48.5	44.8	41.0	64.3	82.8	72.2	50.8
SEA-RAFT	23.9	21.9	48.7	49.4	44.0	64.3	85.7	67.6	50.7
AccFlow	10.3	22.2	23.5	26.4	4.0	38.8	63.2	57.9	30.8
PIPs++	34.1	27.5	62.5	51.3	38.5	64.2	70.4	73.4	52.7
LocoTrack	41.4	43.1	68.0	66.5	58.4	70.0	80.3	76.9	63.1
BootsTAPIR	42.7	34.9	67.9	66.9	56.8	70.6	81.0	78.2	62.4
DELTA	44.6	42.9	75.3	67.8	40.3	66.5	83.0	74.8	61.9
CoTracker2	40.0	31.7	70.9	67.8	43.2	65.8	73.4	73.0	58.2
CoTracker3	48.3	**44.5**	**77.1**	**69.8**	60.4	71.8	84.2	81.6	67.2
PairFormer	**49.7**	44.1	75.8	66.9	**63.2**	**72.1**	**89.1**	**83.8**	**68.1**

Table 2. Average Jaccard (AJ, higher is better).

Method	Dav.	Kin.	Rgb.	Rob.	Avg.
CoTracker3	62.9	53.5	69.5	66.3	63.1
PairFormer	**63.1**	**56.1**	**79.0**	**69.9**	**65.9**

Table 3. Occlusion Accuracy (OA, higher is better).

Method	Dav.	Kin.	Rgb.	Rob.	Avg.
CoTracker3	90.1	85.6	91.6	89.8	89.3
PairFormer	**90.2**	**89.5**	**92.5**	**90.2**	**91.1**

5. Experiments

5.1. Training Details

We train PairFormer on a 1:1 mixture of π-R10K and Kubric [17] training dataset. Video clips have random sequence length (uniformly sampled from 30 to 60 frames) in fixed size of 384×512. We adopt standard data augmentations used in point tracking and optical flow, including random cropping and scaling, horizontal flips, color jitter, and synthetic occluders [22]. We train the model with 8 NVIDIA H100-80G GPUs for 50,000 iterations. We optimize with AdamW (weight decay 0.01); the learning rate is 2×10^{-4} with cosine decay and 1,000-step linear warmup.

5.2. Evaluation on Benchmarks

Baseline methods. We compare our PairFormer with both point trackers and optical flow models: RAFT [37], SEA-RAFT [40], AccFlow [41], PIPs++ [43], LocoTrack [6], DELTA [29], BootsTAPIR [14], DOT [24], and CoTracker [22, 23]. For fair-comparison, we use their publicly released checkpoints for evaluation.

Evaluation on TAP benchmarks. We report results on public point-tracking datasets spanning varied domains: *BADJA* [3] (animals), *CroHD* [36] (surveillance), *TAPVid–DAVIS* [12] (YouTube-like), *DriveTrack* [1] (driving), *EgoPoints* [11] (egocentric), *TAPVid–Kinetics* [12] (YouTube), *RGB-Stacking* [12] (robotic manipulation), and *RoboTAP* [39] (robotics). For long videos we cap the length to at most 600 frames for tractability, and follow the official TAP-Vid protocol for sampling query trajectories and visibility labels. On these benchmarks we use standard TAP metrics: (1) δ**-accuracy**, averaged over pixel thresholds $k \in \{1, 2, 4, 8, 16\}$ at a canonical 256×256 scale, measuring the fraction of query points whose estimated positions fall within k pixels of ground truth; (2) **Average Jaccard (AJ)**, which jointly evaluates localization and visibility on TAP-style splits; and (3) **Occlusion Accuracy (OA)**, the fraction of correctly predicted visible/occluded states. All quantities are computed at official query points and reported per dataset; AJ, and OA are additionally summarized on the TAP-Vid subsets (DAVIS, Kinetics, RGB-Stacking, RoboTAP).

In Tab. 1–3, we report performance on each TAP-style benchmark. Specifically, we use offline version of CoTracker3 model for comparison. From Tab. 1 we observe that PairFormer attains the best δ_{avg} on 6/8 datasets, with particularly strong gains on Rgb. and Rob., where dense spatial context and long-range reasoning are crucial. On Bad., Ego., and Kin. our method also slightly outperforms CoTracker3. Nevertheless, when averaged over all benchmarks, PairFormer achieves the highest δ_{avg} (68.1 vs. 67.2), indicating better overall robustness across diverse domains. Tables 2 and 3 summarize AJ and OA on TAP-Vid datasets. Here PairFormer consistently improves over CoTracker3. These results show that our model provides more precise trajectories while also yielding more reliable occlusion decisions.

Evaluation on APT benchmarks. For evaluating APT, we utilize three datasets with dense pairwise annotations: *CVO* [41], *CVO-Extended* [24], and *APT-Bench*. The dense scenario can be formulated as densely sampled query points and their corresponding trajectories, thus enabling the direct adoption of metrics designed for TAP-style benchmarks. Additionally, we introduce *strided-EPE* for a comprehensive analysis. Specifically, for temporal gaps $g \in [1, T-1]$, we compute average EPE (Endpoint Error) of displacement field $F_{s\to t}$ when $|s - t| = g$, denoted as Δ_g^{epe}. Notably, CVO and CVO-Extended only provide first-to-any dense flow, while APT-Bench supports all-pairs evaluation.

In Table 4, we summarize δ_{avg}, EPE, AJ and strided-EPE with stride $g = 1, 5, 25$ across three datasets. Our PairFormer attains the best performance on all reported metrics, outperforming DOT and the CoTracker3 variants in terms of both accuracy and calibration. The perfor-

Table 4. APT benchmark results and Strided-EPE on APT benchmarks. Δ_g^{epe} denotes average EPE on frame stride g (lower is better).

Method	CVO [41]					CVO-Ext [24]						APT-Bench					
	$\delta_{\text{avg}}\uparrow$	EPE$\downarrow$	AJ$\uparrow$	$\Delta_1^{\text{epe}}\downarrow$	$\Delta_5^{\text{epe}}\downarrow$	$\delta_{\text{avg}}\uparrow$	EPE$\downarrow$	AJ$\uparrow$	$\Delta_1^{\text{epe}}\downarrow$	$\Delta_5^{\text{epe}}\downarrow$	$\Delta_{25}^{\text{epe}}\downarrow$	$\delta_{\text{avg}}\uparrow$	EPE$\downarrow$	AJ$\uparrow$	$\Delta_1^{\text{epe}}\downarrow$	$\Delta_5^{\text{epe}}\downarrow$	$\Delta_{25}^{\text{epe}}\downarrow$
CoTracker3_Offline	78.9	1.53	75.5	1.23	1.63	71.5	5.30	70.4	3.66	5.59	8.95	75.2	4.25	72.5	2.47	4.18	6.48
CoTracker3_Online	77.2	1.55	74.3	1.27	1.71	70.8	5.35	69.2	3.79	5.82	9.21	73.9	4.31	73.2	2.59	4.32	6.73
DOT	81.2	1.40	79.8	1.12	1.39	72.7	5.28	70.8	3.52	5.38	8.77	76.7	3.68	75.4	2.18	3.69	6.02
PairFormer	**81.8**	**1.32**	**80.3**	**1.04**	**1.31**	**73.2**	**5.13**	**71.2**	**3.34**	**5.09**	**8.26**	**77.5**	**3.26**	**76.3**	**2.03**	**3.21**	**5.39**

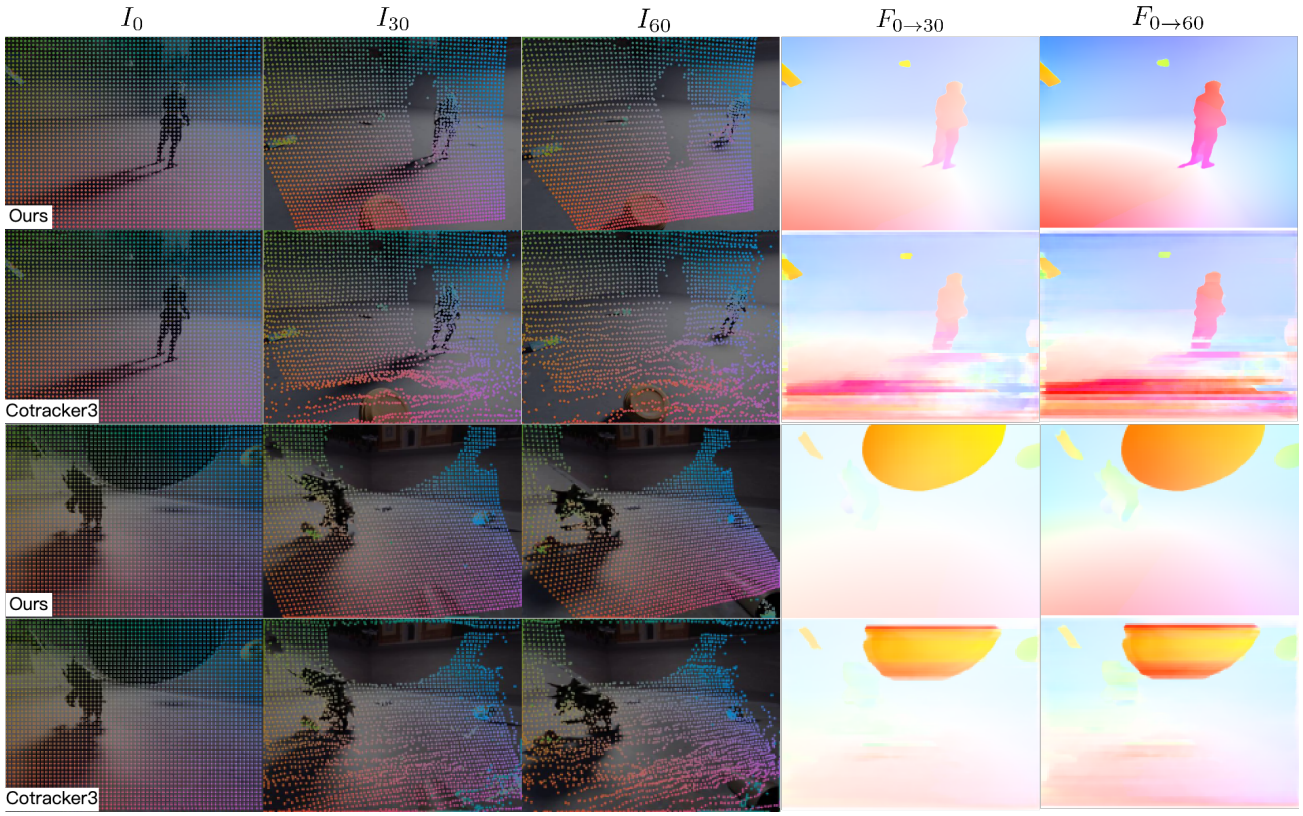

Figure 4. Flow and trajectory visualization across long temporal gaps. We visualize $F_{s\to t}$ for $s = 0$ and $t = 30, 60$ on a representative APT-Bench sequence, comparing CoTracker3_Offline and PairFormer.

mance gains are particularly notable on APT-Bench, which consists of 100-frame-long videos. On CVO and CVO-Extended (comprising $<$ 50-frame video clips) the performance margins are smaller yet consistent. This indicates that our method achieves more substantial improvements on long-sequence video benchmarks while maintaining steady advantages over shorter video clips. This is further evidenced by strided-EPE values with stride $g = 1, 5, 25$ on three dataset. Since CVO only contains 7-frame video clips, Δ_{25}^{epe} values are ommited for this dataset. For all methods and datasets, EPE increases monotonically as g grows from 1 to 25, which matches the intuition that longer temporal distances lead to larger displacements, more occlusions, and higher ambiguity. However, the growth rate for PairFormer is consistently lower. These trends indicate that our model accumulates less error as temporal distance increases, yielding more stable long-range trajectories and fewer catastrophic failures under prolonged occlusions and large viewpoint changes.

5.3. Qualitative Results

Figure 4 presents a qualitative comparison between CoTracker3_Offline and PairFormer on a representative sequence from the APT-Bench benchmark. For each method, we first recover dense per-pixel trajectories by following the predicted correspondences starting from the reference frame, and then derive long-range flow maps from these trajectories. The visualized results are in Figure 4 Compared with CoTracker3_Offline, PairFormer produces visibly smoother and more coherent trajectories that adhere

more faithfully to object geometry and remain stable over time. The corresponding flow maps from PairFormer exhibit sharper motion boundaries and are better aligned with edges and structures in the starting frame, indicating stronger spatial consistency between the estimated motion and the underlying appearance. These qualitative observations are consistent with the quantitative gains in EPE and strided-EPE, and suggest that PairFormer can propagate information through long sequences without sacrificing local geometric fidelity.

5.4. Ablations

We conduct three ablation studies under a fixed training budget and report mean validation performance on a held-out split of APT-Bench. Unless otherwise noted, all variants adopt the same optimization settings, data schedule, and model width as the default PairFormer. We focus on two summary metrics: the TAP-style score δ_{avg} (higher is better) and endpoint error (EPE, lower is better), averaged over all validation videos.

Depth allocation. We first study how to allocate transformer depth between the ST-Patch Encoder and the Broadcast Motion Mixer (BMM). Concretely, we compare a configuration with deeper encoding and shallower mixing ($24/9$ blocks in ST-Patch Encoder/BMM) against a configuration that reverses this ratio ($9/24$), while keeping the total number of blocks comparable (Table 5, columns 1–2). Both variants use the same CorrBank and decoder.

The results show that shifting capacity from the encoder to BMM consistently improves both δ_{avg} and EPE. With a deeper encoder, the model can already extract strong local features, but additional encoder layers mainly refine per-frame descriptors. In contrast, allocating more depth to BMM increases the capacity for long-range spatio-temporal mixing over pairwise motion tokens, which directly benefits APT: the $9/24$ variant improves δ_{avg} and reduces EPE, indicating that extra computation is better used on trajectory- and sequence-level reasoning than on further deepening the initial cross-frame encoding.

With and without $\mathcal{L}_{\text{corr}}$. Next, we evaluate the effect of the correspondence regularization term $\mathcal{L}_{\text{corr}}$ from Section 3.3. This term encourages points that belong to the same physical trajectory to share a consistent all-pairs prediction by enforcing agreement between direct and multi-hop correspondences along the trajectory (beyond what is enforced by per-pair supervision alone).

Table 5, columns 3–4 compare models trained without and with $\mathcal{L}_{\text{corr}}$ under identical settings. Adding $\mathcal{L}_{\text{corr}}$ yields a moderate but consistent gain in δ_{avg} and a reduction in EPE. In practice, we observe that models trained with this constraint produce smoother trajectories over long temporal ranges and fewer small, inconsistent deviations when revisiting the same region multiple times. This suggests that explicitly tying together correspondences along the same trajectory helps the model resolve ambiguous matches and improves temporal coherence, even though the loss is applied only as a secondary regularizer.

Table 5. Joint ablation of ST-Patch Encoder / BMM depth, correspondence regularization, and CorrBank capacity.

	Depth Alloc.		$\mathcal{L}_{\text{corr}}$		# Corr-tokens		
Metric	24 / 9	9 / 24	w/o.	w/.	32	64	128
$\delta_{\text{avg}}\uparrow$	66.9	**67.6**	67.2	**67.8**	67.1	**67.6**	67.5
EPE $\downarrow$	4.4	**4.2**	4.3	**4.1**	4.3	**4.2**	4.2

CorrBank capacity. Finally, we study the capacity of CorrBank by varying the number of learnable correlation tokens from 32 to 128 (Table 5, columns 5–7). Each token participates in the construction of the pair-specific correlation memory, so increasing this number enlarges the space in which pairwise similarity patterns can be represented.

The ablation shows that using too few correlation tokens (32) slightly hurts both δ_{avg} and EPE, indicating that the correlation memory becomes under-parameterized and cannot fully capture complex appearance-motion relationships. Increasing to 64 tokens yields a noticeable improvement and is used as our default setting. Further expanding to 128 tokens does not bring additional gains and only increases memory and compute. Overall, these results suggest that a moderately sized CorrBank is sufficient to encode the relevant pairwise structure, and that most of the modeling benefit comes from how this memory is used (via CorrBank and BMM) rather than from unbounded capacity.

6. Conclusion

This work formalizes All-Pairs Tracking (APT), a paradigm that explicitly models dense displacement and visibility across all frame pairs in a video, extending beyond the query-conditioned focus of existing TAP methods. We propose PairFormer, a feed-forward transformer integrating spatio-temporal encoding, CorrBank, and global context aggregation, alongside the PAIRender platform for synthetic annotated data (supporting π-R10K and APT-Bench). Experiments confirm PairFormer's state-of-the-art performance on APT-Bench and competitive results on TAP benchmarks, advancing dense spatiotemporal tracking via a unified task, model, and data ecosystem. We believe that making all-pairs correspondence a first-class representation also opens up opportunities for downstream applications such as 4D reconstruction and motion editing.

Limitations. PairFormer faces quadratic computational costs for ultra-long videos, and its reliance on synthetic PAIRender data might introduces a domain gap with real-world scenarios.

References

[1] Arjun Balasingam, Joseph Chandler, Chenning Li, Zhoutong Zhang, and Hari Balakrishnan. Drivetrack: A benchmark for long-range point tracking in real-world videos. In *CVPR*, 2024. 6

[2] Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yitong Dong, Yijin Li, and Hongsheng Li. Context-pips: Persistent independent particles demands context features. *NeurIPS*, 2024. 2

[3] Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon, and Roberto Cipolla. Creatures great and SMAL: Recovering the shape and motion of animals from video. In *ACCV*, 2018. 6

[4] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *TPAMI*, 33(3):500–513, 2010. 2

[5] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. Flowtrack: Revisiting optical flow for long-range dense tracking. In *CVPR*, 2024. 1, 2

[6] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *ECCV*, 2025. 2, 6

[7] Pierre-Henri Conze, Philippe Robert, Tomas Crivelli, and Luce Morin. Dense long-term motion estimation via statistical multi-step flow. In *VISAPP*, 2014. 2

[8] Pierre-Henri Conze, Philippe Robert, Tomas Crivelli, and Luce Morin. Multi-reference combinatorial strategy towards longer long-term dense motion estimation. *Computer Vision and Image Understanding*, 150:66–80, 2016.

[9] Tomas Crivelli, Pierre-Henri Conze, Philippe Robert, and Patrick Pérez. From optical flow to dense long term correspondences. In *International Conference on Image Processing*, 2012. 2

[10] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024. 4

[11] Ahmad Darkhalil, Rhodri Guerrier, Adam W Harley, and Dima Damen. Egopoints: Advancing point tracking for egocentric videos. *arXiv:2412.04592*, 2024. 6

[12] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. TAP-Vid: A benchmark for tracking any point in a video. In *NeurIPS Datasets and Benchmarks*, 2022. 1, 2, 6

[13] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *ICCV*, 2023. 1, 2

[14] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *ACCV*, 2024. 6

[15] 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 *ICCV*, 2015. 2

[16] James J Gibson. The perception of visual surfaces. *The American journal of psychology*, 63(3):367–384, 1950. 2

[17] 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 *CVPR*, 2022. 2, 6

[18] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, 2022. 1, 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[20] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 1, 2

[21] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 2

[22] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv:2410.11831*, 2024. 1, 2, 6

[23] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *ECCV*, 2024. 2, 6

[24] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: connecting the dots. In *CVPR*, 2024. 2, 6, 7

[25] Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Feng Li, Tianhe Ren, Bohan Li, and Lei Zhang. TAPTRv2: Attention-based position update improves tracking any point. In *NeurIPS*, 2024. 2

[26] Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. TAPTR: Tracking any point with transformers as detection. In *ECCV*, 2024. 2

[27] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981. 1, 2

[28] Michal Neoral, Jonáš Šerỳch, and Jiří Matas. Mft: Long-term tracking of every pixel. In *WACV*, 2024. 1, 2

[29] Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. DELTA: Dense efficient long-range 3d tracking for any video. In *ICLR*, 2025. 6

[30] Diederick C Niehorster. Optic flow: a history. *i-Perception*, 12(6):20416695211055766, 2021. 2

[31] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 3, 4

[32] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. In *CVPR*, 2006. 2

[33] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan

Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 2

[34] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1, 2

[35] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *ECCV*, 2010. 2

[36] Ramana Sundararaman, Cedric De Almeida Braga, Eric Marchand, and Julien Pettre. Tracking pedestrian heads in dense crowd. In *CVPR*, 2021. 6

[37] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1, 2, 6

[38] Carlo Tomasi and Takeo Kanade. Detection and tracking of point. *IJCV*, 9:137–154, 1991. 2

[39] Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. RoboTAP: Tracking arbitrary points for few-shot visual imitation. In *ICRA*, 2024. 6

[40] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *ECCV*, 2024. 1, 2, 6

[41] 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 *ICCV*, 2023. 6, 7

[42] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *CVPR*, 2017. 2

[43] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023. 2, 6