Towards Generalizable Multi-Object Tracking

Zheng Qin[1] Le Wang[1*] Sanping Zhou[1] Panpan Fu[2] Gang Hua[3] Wei Tang[4]
[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]School of Software Engineering, Xi'an Jiaotong University
[3]Wormpex AI Research [4]University of Illinois at Chicago

Abstract

Multi-Object Tracking (MOT) encompasses various tracking scenarios, each characterized by unique traits. Effective trackers should demonstrate a high degree of generalizability across diverse scenarios. However, existing trackers struggle to accommodate all aspects or necessitate hypothesis and experimentation to customize the association information (motion and/or appearance) for a given scenario, leading to narrowly tailored solutions with limited generalizability. In this paper, we investigate the factors that influence trackers' generalization to different scenarios and concretize them into a set of tracking scenario attributes to guide the design of more generalizable trackers. Furthermore, we propose a "point-wise to instance-wise relation" framework for MOT, i.e., GeneralTrack, which can generalize across diverse scenarios while eliminating the need to balance motion and appearance. Thanks to its superior generalizability, our proposed GeneralTrack achieves state-of-the-art performance on multiple benchmarks and demonstrates the potential for domain generalization.

1. Introduction

Multi-Object Tracking (MOT) aims to locate targets and recognize their identities from a streaming video. It is an essential task for many applications such as autonomous driving [7], robotics [31], and visual surveillance [33]. Despite great progress in the past few years, the MOT task remains challenging when the trackers are generalized to diverse application scenarios.

Prior MOT methods mostly follow the tracking-by-detection (TbD) [1, 2, 59] or tracking-by-regression (TbR) [3, 62] paradigm. TbD methods detect objects in each frame and then associates objects across frames. TbR methods also conduct frame-wise object detection, but replaces the data association with a continuous regression of each tracklet to its new position. With rapid advances in object detection, TbD has become the dominant paradigm in the field.

Real-world application scenarios are diverse and characterized by a wide spectrum of different attributes, such as varying motion complexities, target densities, and frame rate, as shown in Figure 1. Unfortunately, current MOT methods heavily depend on extensive prior knowledge or intricate engineering efforts to excel in specific scenarios, but they struggle to generalize effectively to different situations. This limitation significantly restricts their utility in real-world applications.

For TbD, motion-dominated methods [5, 36, 40] are brittle when encountering irregular motion and substantial variation in target shape or position; appearance-dominated methods [13, 38, 52] are prone to failure when facing occlusion, for example, caused by dense crowds or shelters, light change, and small targets. To overcome these difficulties, some TbD methods require a *manual* adjustment of which information to rely on more in a specific scenario. For example, ByteTrack [59] constructs an affinity matrix based on motion in MOT17 and MOT20, and based on appearance in BDD100K. Other works [1, 13, 43, 52] directly balance the two affinity matrices with a weighting factor and adjust it for different scenarios. The limitation in generalizability similarly applies to TbR. For example, Tracktor [3] cannot handle videos with low frame rates and targets with large shape or position variation; Centertrack [62] uses the center point to represent each target, which would become overwhelming in crowded scenarios. Therefore, there is an urgent need to develop trackers that effectively generalize to different scenarios.

In this paper, we first conduct an in-depth analysis of the tracking scenarios to gain insight into why a particular tracker's performance varies significantly in different scenarios. Referring to the performance of previous trackers on different datasets, we parsed out the tracking scenario attributes as follows: motion complexity, variation amplitude,

*Corresponding author.

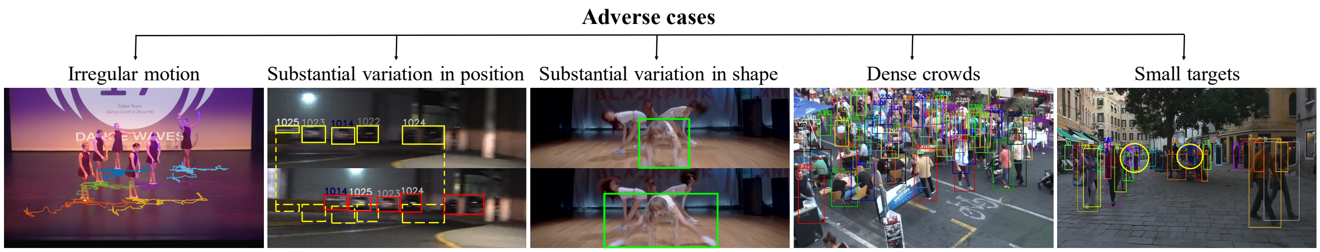

Figure 1. **Adverse cases for some attributes in tracking scenarios.** The line in 'irregular motion' is the trajectory of each target.

target density, small target, and frame rate. We analyze the datasets [8, 9, 30, 46, 55] commonly used in MOT based on these attributes, and find that these attributes vary drastically across different datasets. Both motion-dominated and appearance-dominated methods have their respective attributes which they do not excel at.

Based on the above analysis, we propose a "point-wise to instance-wise relation" framework for MOT, *i.e.*, GeneralTrack, which can generalize to diverse scenarios without manually balancing motion and appearance information. Specifically, instead of directly constructing relations between tracklets and detections at the instance level, we capture the point-wise relations and then translate them into instance-wise associations. The fine-grained features together with the fine-to-coarse translation can cope with dense targets and small targets. In contrast to searching in fixed local areas by previous motion-dominated methods, we contrust multi-scale point-region relation which implicitly contains a motion template guided by vision and geometry that does not flinch at irregular motion. The flexible scale of motion templates can be effectively adapted to various frame rates as well as the amplitude of position variation. Finally, we design a hierarchical relation aggregation paradigm to associate the tracklets and detections according to the point-part-instance hierarchy. The targets evolve from rigid bodies to flexible bodies and are suitable for scenarios with dramatic shape variation.

Extensive experiments on multiple benchmark datasets show that our method achieves the state-of-the-art, demonstrating the superiority of generalizability over diverse scenarios. In particular, our GeneralTrack ranks 1st place on the BDD100K leaderboard (57.87 mTETA). In addition, we experimentally find that GeneralTrack has great potential for domain generalization with unseen data distributions (cross-dataset, cross-class). The main contribution of this work can be summarized as follows:

- We analyze the factors that hinder the generalizability of existing trackers and concretize them into tracking scenario attributes that can guide the design of trackers.
- We propose a "point-wise to instance-wise relation" framework for MOT. It first constructs point-wise relations through the multi-scale 4D correlation volume and then aggregates them into instance-wise associations through a novel "point-part-instance" hierarchy. Our new framework can address several fundamental challenges in MOT. Concretely, the point-wise correlation modeling deals with damage to instance-level representations by dense and small targets; the construction of multi-scale point-region relations handles severe motion complexity, and different position variations and frame rates; the hierarchical aggregation copes with shape variations.
- Extensive evaluation of the GeneralTrack shows that it achieves the state-of-the-art performance on multiple MOT datasets. In addition, GeneralTrack experimentally demonstrates strong domain generalization capabilities.

2. Related Work

Tracking-by-Detection. The dominant paradigm in the field of MOT has long been tracking-by-detection [5, 6, 13, 17, 26, 35, 48, 57, 59]. The core of TbD is to construct inter-frame relation (affinity matrix) between tracklets and detections, and then perform matching with the Hungarian Algorithm [24]. The affinity matrix for matching is often driven by motion information [16, 22, 36, 40] or appearance information [23, 34, 49, 52, 54]. As discussed in Sec. 1, both motion and appearance dominated methods have their respective scenarios in which they do not excel.

To address these issues, some methods work towards a better balance between motion and appearance [1, 13, 43, 52, 59]; some others, *i.e.*, TrackFlow [29], handle these by building an probabilistic formulation but requires virtual datasets for training. In contrast, we propose a new approach that achieves generalization and avoids balance between motion and appearance.

Dense Flow and Correspondences. Identifying correspondences between an image pair is a fundamental computer vision problem, encompassing optical flow and geometric correspondences. FlowNet [12] is the first end-to-end method for optical flow estimation. Then a series of works [11, 20, 21, 37, 44, 45] employ coarse-to-fine and iterative estimation methodology. To deal with missing small fast-motion objects in the coarse stage, RAFT [47] performs optical flow estimation in a coarse-and-fine and recurrent manner. Geometric correspondences [27, 39, 41] refer to correspondences between images captured from different views. MatchFlow [10] takes geometric correspondences as the prefixed task for optical flow.

Among these explorations, the 4D Correlation Volume is often used to capture the visual similarity of pixel pairs and as a core component supporting dense flow and correspondence estimation. In MOT, the visual relation between frames is significant. Inspired by these works, we address the tracking task from the perspective of constructing the relation from pixel to instance with low-level vision.

3. Methodology

3.1. Analysis of MOT Scenarios

There are countless real-world application scenarios. Current MOT methods heavily depend on extensive prior knowledge or intricate engineering efforts to excel in specific scenarios, but they hardly generalize to different situations. To gain insight into this phenomenon, we analyze the failure cases of previous trackers in different datasets and identify the following attributes that have substantial influence on a tracker's performance:

- **Motion Complexity** reflects the irregularity and unpredictability of target motion within the scenario. The more irregular and unpredictable the motion, the greater its complexity.
- **Variation Amplitude** reflects the target's variability, encompassing both shape and position variations.
- **Target Density** reflects the density of the crowds in the scenario, implicitly reflecting the degree of occlusion within the crowds.
- **Small Target** represents the average amount of small targets in the scenario.
- **Frame Rate** is the number of frames captured in one second of the input video stream.

We conduct thorough measurements of these attributes on five datasets [8, 9, 30, 46, 55] and form the tracking scenario attribute maps as shown in Figure 2. Note that frame rate takes the inverse in the maps. The detailed measurement metric is provided in the Supplementary Material.

Furthermore, we divide the attributes into two categories based on whether they damage motion or appearance, as shown by the background of the attribute map, *i.e.*, motion and appearance-dominated methods may not perform well in the white and blue areas, respectively. In particular, pedestrian tracking scenarios with regular motion, high frame rate, and small movement amplitude, *e.g.*, MOT17 and MOT20, are more motion-reliant; in tracking scenarios with highly complex motion patterns, *e.g.* DanceTrack and SportsMOT, appearance is more effective than motion; BDD100K cannot provide reliable motion information due to the low frame rate and large movement amplitude. Our observations are consistent with how previous methods balance motion and appearance information. For exmaple, GHOST [43] sets the optimal weight between appearance and motion (percentage of motion) as 0.6 for MOT17, 0.8 for MOT20, 0.4 for BDD100K, and 0.4 for DanceTrack. ByteTrack [59] utilizes motion to construct the affinity matrix in MOT17 and MOT20, and use appearance in BDD100K; it has very poor performance using motion on DanceTrack and SportsMOT in which the target motion is complex. For a tracker to have great generalizability, it is essential to take into account these attributes.

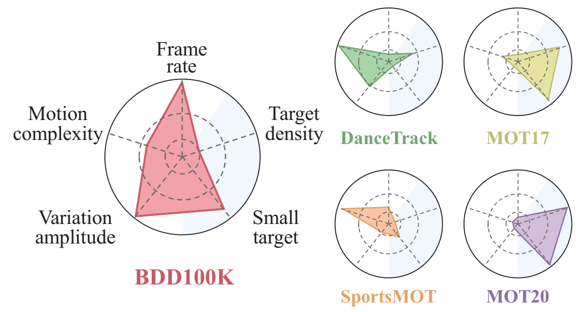

Figure 2. **Tracking scenario attribute maps**. Appearance performs poorly in the scenario with a large percentage in the blue area, as well as motion in the white area.

3.2. Overview of GeneralTrack

Notation. For online video streaming, we first process the current frame t with YOLOX [14] to obtain the detection results. The detections are denoted as $\mathcal{D}^t = \{\mathbf{d}_i^t\}_{i=1}^N$ containing N detections in frame t, where $\mathbf{d}_i^t$ represents the position and size of a detection bounding box. We denote the set of M tracklets by $\mathbb{T} = \{\mathcal{T}_j\}_{j=1}^M$. $\mathcal{T}_j$ is a tracklet with identity j and is defined as $\mathcal{T}_j = \{\mathbf{l}_j^{t_0}, \mathbf{l}_j^{t_0+1}, ..., \mathbf{l}_j^t\}$, where $\mathbf{l}_j^t$ is the location in frame t, and t_0 is the initialized moment.

Our GeneralTrack follows the well-known tracking-by-detection paradigm [59]. Given the current frame t, we obtain its detections $\mathcal{D}^t$ and the set of M tracklets $\mathbb{T}$ up to frame $t-1$. Then we associate existing tracklets $\mathbb{T}$ with current detections $\mathcal{D}^t$ by constructing the point-wise relations between frame $t-1$ and frame t, and transforming them to the instance-wise associations. As shown in Figure 3, this process consists of three stages: (i) We use **Feature Relation Extractor** (Sec. 3.3) to construct global dense relations with frame t for each point in frame $t-1$ by a 4D correlation volume. (ii) Then we transform the global relations into **Multi-scale Point-region Relations** (Sec. 3.4), and form a relation map for frame $t-1$ in which each point represents its movement trends. (iii) Finally, we progressively perform **Hierarchical Relational Aggregation** (Sec. 3.5) according to the *point-part-instance* hierarchy to associate the tracklets and detections. All stages are differentiable and com-

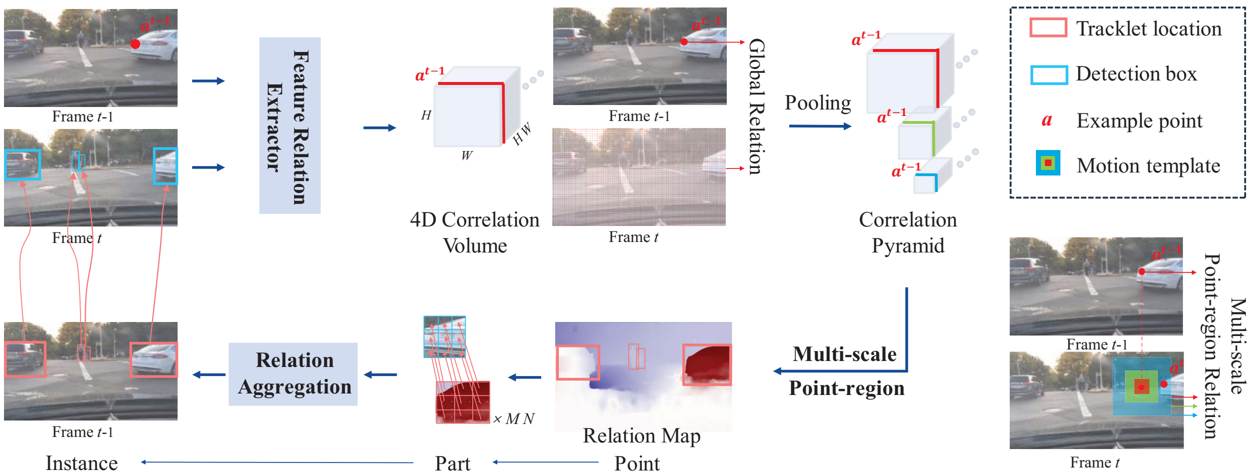

Figure 3. **Overview of our GeneralTrack.** The **Feature Relation Extractor** obtains global dense relations with frame t for each point in frame $t-1$ by a 4D correlation volume. Then by constructing a correlation pyramid, we transform the global relations into **Multi-scale Point-region Relations**, and form a relation map for frame $t-1$. Finally, We progressively perform **Relational Aggregation** to aggregate point-wise relation into instance-wise relation and achieve the association between tracklets and detections.

posed into an end-to-end trainable architecture.

3.3. Feature Relation Extractor

Considering that the target could be very ***small*** or ***occluded***, we exploit a extractor to capture the relationship at the point level. Given a pair of consecutive RGB images, $\mathbf{I}^{t-1}$ and $\mathbf{I}^{t}$, a convolutional neural network encodes them into two dense feature maps at a lower resolution, denoted as $\mathbf{F}^{t-1}, \mathbf{F}^{t} \in \mathbb{R}^{H\times W\times D}$, where H, W are respectively 1/8 of the image height and width, and D is the feature dimension.

After obtaining the pair of consecutive feature maps, $\mathbf{F}^{t-1}$ and $\mathbf{F}^{t}$, we compute the global dense relations by constructing a full correlation volume between them. The correlation volume, $\mathbf{C}^{\text{global}}$, is formed by taking the dot product between all pairs of feature vectors as follows:

$$\begin{aligned} &\mathbf{C}^{\text{global}}\left(\mathbf{F}^{t-1}, \mathbf{F}^{t}\right) \in \mathbb{R}^{H\times W\times H\times W}, \\ &c_{ijkl} = \sum_{d=1}^{D} f_{ijd}^{t-1} \cdot f_{kld}^{t}, \end{aligned} \tag{1}$$

The element c_{ijkl} in $\mathbf{C}^{\text{global}}$ represents the relation between the (i,j)-th feature point in frame $t-1$ and the (k,l)-th feature point in frame t.

Instance-level features would be damaged when the target is too small or occluded, whereas point-wise relations between adjacent frames are robust in this scenario.

3.4. Multi-scale Point-region Relation

This part is to adapt the tracker to ***various frame rates*** as well as ***the amplitude of position variation***. As demonstrated in Figure 4, there is a wide range of speeds among different classes, such as fast cars and slow people, and the same problem exists at different frame rates. Besides, $\mathbf{C}^{\text{global}}$ contains dense global relations, with significant invalid relations that would cause significant computational costs and slow convergence. For tracker's flexibility and computational simplicity, we transform global relations into multi-scale point-region relations.

Inspired by the multi-scale 4D volume in [10, 47], we downsample the correlation volume and obtain the correlation pyramid $\{\mathbf{C}_s\}_{s=0}^{S}$ by pooling the last two dimensions as follows:

$$\mathbf{C}_s = \texttt{pooling}(\mathbf{C}_{s-1}), \mathbf{C}_s \in \mathbb{R}^{H\times W\times H/2^s\times W/2^s}, \tag{2}$$

where s is the layer of pyramid and $\mathbf{C}_0$ is initialized as $\mathbf{C}^{\text{global}}$. The set of volumes provides relations from points in frame $t-1$ to each area in frame t at different scales.

To reduce the invalid relations, we set a center-based relation searching region based on $\{\mathbf{C}_s\}_{s=0}^{S}$ as follows:

$$\mathcal{P}\left(\mathbf{x}\right)_s = \left\{\mathbf{x}/2^s + \mathbf{r} : \|\mathbf{r}\| \le R\right\}, \tag{3}$$

where $\mathbf{x} \in \mathbb{Z}^2$ is a coordinate in $\mathbf{F}^{t-1}$ and $\mathbf{r} \in \mathbb{Z}^2$ is the displacement from point $\mathbf{x}/2^s$ in the s-th layer in the correlation pyramid. The relation searching is restricted in the neighborhood of the coordinate $\mathbf{x}/2^s$, schematically as $\|\mathbf{r}\| \le R$, *i.e.* the maximum displacement in any direction is R. At each level of $\{\mathbf{C}_s\}_{s=0}^{S}$, the searching region in $\mathbf{F}^{t}$ gradually increases as the resolution decreases, as the red, green and blue boxes in Figure 4.

We employ $\mathcal{P}\left(\mathbf{x}\right)_s$ to search the relation on correlation pyramid $\{\mathbf{C}_s\}_{s=0}^{S}$ to form multi-scale point-region relation

Figure 4. **Multi-scale point-region Relation** on correlation pyramid. With downsampling, the searching region becomes progressively larger (red, green and blue box). Two examples points a and b are given in the figure, where the green dot and blue dot represent the target point in frame $t-1$ and frame t respectively. **(a)** Headlights of the car. The car moves very fast with large displacement, its relation point in frame t is not captured until the layer with the largest scale **(b)** Head of the man. Due to the relatively small movement of people, its relation can be obtained at the first level of the pyramid (the highest resolution). Such a relation searching paradigm can be flexibly adapted to both large and small displacements with low computational resources.

map $\mathbf{O} \in \mathbb{R}^{H\times W\times(S+1)(2R+1)^2}$ as follows:

$$\begin{aligned}\mathbf{O}_s &= \mathtt{search}(\mathbf{C}_s, \mathcal{P}\left(\mathbf{x}\right)_s),\\ \mathbf{O} &= \mathtt{concat}(\{\mathbf{O}_s\}_{s=0}^{S}),\end{aligned} \tag{4}$$

where the function search means searching operation on each layer with $\mathcal{P}\left(\mathbf{x}\right)_s$. $\{\mathbf{O}_s \in \mathbb{R}^{H\times W\times(2R+1)^2}\}_{s=0}^{S}$ is the point-region relations at each layer in $\{\mathbf{C}_s\}_{s=0}^{S}$, and they are concatenated at the last dimension. $\mathbf{O}$ contains the motion trends of the points in frame t-1 to frame t in all directions and at all ranges.

In essence, the multi-scale point-region relation implicitly contains a vision and geometry based motion template, with different layers representing different displacement scales, and different points in the same layer representing different directions and amplitudes, which can be adapted to various frame rates as well as the amplitude of position variation flexibly.

3.5. Hierarchical Relation Aggregation

In current frame t, the detection set $\mathcal{D}^t$ contains N detections and the tracklets set $\mathbb{T}$ contains M tracklets. The relation map $\mathbf{O}$ encodes frame $t-1$'s point-wise relations to multi-scale regions in frame t. Here we translate point-wise into instance-wise relations to construct tracklet-detection relation matrix $\mathbf{P}^{\text{rela}} \in \mathbb{R}^{N\times M}$ for data association.

To address the ***shape variation of instances***, we treat the instance as a flexible body. Instead of treating the instance as a rigid body as traditional motion or appearance methods do, we progressively aggregate relations through a point-part-instance hierarchy. For the i-th detection $\mathbf{d}_i^t$ in $\mathcal{D}^t$ and the j-th tracklet's location $\mathbf{l}_j^{t-1}$ in $\mathbb{T}$, we divide the instances $\mathbf{d}_i^t$ and $\mathbf{l}_j^{t-1}$ into $v\times v$ parts. We apply RoIAlign [18] to $\mathbf{l}_j^{t-1}$ on $\mathbf{O}$ and obtain the j-th tracklet's part-wise relation $\mathbf{O}_j \in \mathbb{R}^{v\times v\times D}$, where $v\times v$ is the RoIAlign size and D is the feature dimension. Then we encode the relative position of each part between $\mathbf{d}_i^t$ and $\mathbf{l}_j^{t-1}$, denotes as $\mathbf{E}_{ij} \in \mathbb{R}^{v\times v\times 2}$, where its element represents the displacement of the centroids of corresponding parts. Then we concatenate $\mathbf{O}_j$ and $\mathbf{E}_{ij}$ and apply the convolution operation to it. Finally, we obtain the prediction score with a multi-layer perception (MLP) as follows:

$$p_{ij}^{\text{rela}} = \mathtt{MLP}\left(\mathtt{conv}(\mathtt{concat}\left(\mathbf{O}_j, \mathbf{E}_{ij}\right))\right). \tag{5}$$

Eventually, we use the Hungarian Algorithm [24] for $\mathbf{P}^{\text{rela}}$ and complete the association. For the lost tracklets, we use Kalman filter to retrieve them.

3.6. Training

We pick two consecutive frames in the video as a training sample. Two raw images are used as inputs $\mathbf{I}^{t-1}$ and $\mathbf{I}^t$ in the forward propagation process, and the grounding truth boxes are used as $\mathbb{T}$ and $\mathcal{D}^t$ respectively. The targets in two frames are combined in pairs to predict tracklet-detection relation $\mathbf{P}^{\text{rela}}$ and we label positive or negative by whether they have the same identity. Then, we supervise it with a weighted binary cross-entropy loss function (weighted BCE Loss):

$$\begin{aligned}\mathcal{L}^{\text{wBCE}} = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} -[w\cdot y_{ij}\log(p_{ij}^{\text{rela}})\\ + (1-y_{ij})\log(1-p_{ij}^{\text{rela}})],\end{aligned} \tag{6}$$

where p_{ij}^{rela} denotes the predicted correlation score, and y_{ij} indicates the ground truth correlation label, in which 1 and 0 represent the positive and negative correlation, respectively. There is a great difference between the amount of positive and negative samples, inspired by F³Net [51], we add weights to the positive samples, *i.e.*, the weighting factor w. To enhance the capability of Feature Relation Extractor, we use [15] for pre-training in the settings of dense flow and correspondence tasks.

3.7. Multi-class Object Tracking

FairMOT [58] regards each tracked object as a class and associates the detection results by feature similarity. Some

	Venue	mHOTA↑	mIDF1↑	mMOTA↑	HOTA↑	IDF1↑	MOTA↑	IDs↓	MT↑	ML↓
validation										
QDTrack [35]	CVPR'21	-	50.8	36.6	-	71.5	63.5	**6262**	9481	3034
Unicorn [53]	ECCV'22	-	54.0	41.2	-	71.3	66.6	10876	10296	2505
MOTR [56]	ECCV'22	-	44.8	32.3	-	65.8	56.2	-	-	-
TETer [25]	ECCV'22	-	53.3	39.1	-	-	-	-	-	-
ByteTrack [59]	ECCV'22	45.3	54.8	45.2	61.3	70.4	**69.1**	9140	9626	3005
MOTRv2 [60]	CVPR'23	-	**56.5**	43.6	-	**72.7**	65.6	-	-	-
GHOST [43]	CVPR'23	45.7	55.6	44.9	61.7	70.9	68.1	-	-	-
GeneralTrack(Ours)		**46.9**	56.2	**46.4**	**63.1**	**72.7**	68.8	8496	**11830**	**2035**
test										
DeepBlueAI	-	-	38.7	31.6	-	56.0	56.9	25186	10296	12266
madamada	-	-	43.0	33.6	-	55.7	59.8	42901	16774	5004
QDTrack [55]	CVPR'21	41.9	52.4	35.7	60.5	72.5	64.6	**10790**	17353	5167
ByteTrack [59]	ECCV'22	-	55.8	**40.1**	-	71.3	**69.6**	15466	18057	5107
GHOST [43]	CVPR'23	46.8	**57.0**	39.5	62.2	72.0	68.9	-	-	-
GeneralTrack(Ours)		**47.9**	56.9	39.9	**63.7**	**73.6**	69.1	14489	**21281**	**3715**

Table 1. Comparison with the state-of-the-art methods on BDD100K. The two best results for each metric are highlighted in bolded and blue. Note that some methods only provide the *validation set* result without *test set*.

	Venue	HOTA↑	MOTA↑	IDF1↑	AssA↑	DetA↑
GTR [63]	CVPR'22	54.5	67.9	55.8	45.9	64.8
ByteTrack [59]	ECCV'22	64.1	95.9	71.4	52.3	78.5
OC-SORT [6]	CVPR'23	73.7	96.5	74.0	61.5	88.5
MixSort-Byte* [8]	ICCV'23	65.7	96.2	74.1	54.8	78.8
MixSort-OC* [8]	ICCV'23	**74.1**	96.5	74.4	**62.0**	88.5
GeneralTrack(Ours)	-	**74.1**	**96.8**	**76.4**	61.7	**89.0**

Table 2. Comparison with the state-of-the-art methods on SportsMOT. The two best results for each metric are highlighted in bolded and blue. Note that $*$ denotes the methods use the validation set for training the association model.

	Venue	HOTA↑	MOTA↑	IDF1↑	AssA↑	DetA↑
Transformer based:						
MOTR [56]	ECCV'22	54.2	79.7	51.5	40.2	73.5
Hybird based:						
MOTRv2 [60]	CVPR'23	69.9	91.9	71.7	59.0	83.0
CNN based:						
ByteTrack [59]	ECCV'22	47.7	89.6	53.9	32.1	71.0
FineTrack [38]	CVPR'23	52.7	89.9	**59.8**	38.5	72.4
OC-SORT [6]	CVPR'23	55.1	**92.2**	54.9	40.4	80.4
GHOST [43]	CVPR'23	56.7	91.3	57.7	39.8	81.1
GeneralTrack (Ours)	-	**59.2**	91.8	59.7	**42.8**	**82.0**

Table 3. Comparison with the state-of-the-art methods on DanceTrack. The two best results are highlighted in bolded and blue. Note that MOTRv2 uses both YOLOX and MOTR with more than two hundred times training resource usage than ours.

	Venue	HOTA↑	MOTA↑	IDF1↑	AssA↑	DetA↑	IDs↓
MOT17							
MOTR [56]	ECCV'22	57.8	73.4	68.6	55.7	60.3	2439
ByteTrack [59]	ECCV'22	63.1	80.3	77.3	62.0	64.5	2196
OC-SORT [6]	CVPR'23	63.2	78.0	77.5	**63.2**	63.2	1950
MOTRv2 [60]	CVPR'23	62.0	78.6	75.0	60.6	63.8	-
GHOST [43]	CVPR'23	62.8	78.7	77.1	-	-	2325
GeneralTrack(Ours)	-	**64.0**	**80.6**	**78.3**	63.1	**65.1**	**1563**
MOT20							
ByteTrack [59]	ECCV'22	61.3	**77.8**	75.2	59.6	63.4	1223
OC-SORT [6]	CVPR'23	**62.1**	75.5	**75.9**	**62.0**	-	**913**
MOTRv2 [60]	CVPR'23	60.3	76.2	72.2	58.1	62.9	-
GHOST [43]	CVPR'23	61.2	73.7	75.2	-	-	1264
GeneralTrack(Ours)	-	61.4	77.2	74.0	59.5	**63.7**	1627

Table 4. Comparison with the state-of-the-art methods on MOT17, MOT20. The two best results are highlighted in bolded and blue.

TbD methods, such as GHOST [43] and ByteTrack [59], restrict the tracking process in each class. Target tracking can easily be interrupted due to misjudgments of the class by object detection. In contrast, we first conduct class-agnostic association, allowing objects detected as different classes across frames to be associated. Second, we determine the "true class" as the most frequent class in the historical trajectory. Finally, we treat detections in the trajectory that differ from the "true class" as false positives and categorize them into the "true class".

4. Experiments

4.1. Setting

Datasets and Metrics. We evaluate GeneralTrack on BDD100K [55], SportsMOT [8], DanceTrack [46], MOT17 [30] and MOT20 [9] datasets. As standard protocols, CLEAR MOT Metrics [4] and HOTA [28] are used for evaluation. For multi-class tracking, average metrics across all classes are added as well as Tracking-Every-Thing Accuracy (TETA) [25] metric for ranking as they did in BDD100K MOT Challenge 2023.

Implementation Details. For fair comparisons, we directly apply the publicly available detector of YOLOX [14], trained by [43] for BDD100K, [8] for SportsMOT, [46] for DanceTrack, [59] for MOT17, MOT20. Note that all the detection results we used are the same as most of the SOTAs compared in the tables. The number of downsampling in correlation pyramid S is 3, the relation searching radius R is 4. The RoIAlign size $v * v$ is set to 3 * 3 for DanceTrack, SportsMOT and 2 * 2 for BDD100K, MOT17, MOT20.

Setting	**MR**	**HRA**	mHOTA↑	mIDF1↑	mMOTA↑	HOTA↑	IDF1↑	MOTA↑	IDs↓
#1	✓	✓	47.1	56.1	46.1	63.4	72.5	68.3	8503
#2	✓	✓	46.2	54.5	43.6	62.4	71.3	66.0	9070
#3	✓	✓	42.9	49.2	37.5	57.8	63.9	59.2	11584
#1		✓	46.9	55.7	45.6	63.1	72.2	67.9	9447
#2		✓	45.3	53.3	42.2	61.7	70.1	65.0	10673
#3		✓	41.7	47.8	36.3	55.8	61.2	56.7	14015
#1	✓		46.7	55.5	45.6	62.8	71.8	67.9	9070
#1	✓	✓							
	SR=1		46.9	55.7	46.0	63.1	72.1	68.2	9173
	SR=4		47.1	56.1	46.1	63.4	72.5	68.3	8503
	SR=7		46.9	55.7	46.0	63.5	72.7	68.3	8454

Table 5. Ablation study of Multi-scale Relation (MR), Searching Radius (SR), Hierarchical Relation Agregation (HRA). MR is assessed under three frame rate settings, where *#1*, *#2* and *#3* represent the original as well as double and quadruple downsampling.

Class	HOTA↑	IDF1↑	MOTA↑	IDs↓
Pedestrian	50.3(+0.1)	60.7(+0.1)	55.6(+0.2)	2236(↓ 1.2%)
Rider	43.7(**+3.6**)	57.9(**+2.8**)	46.3(**+2.6**)	52(↓**44.2%**)
Car	66.2(+0.0)	75.4(+0.1)	73.1(-0.1)	6018(↓ 1.6%)
Bus	60.0(**+2.1**)	69.1(**+1.8**)	56.5(**+1.9**)	70(↓**35.7%**)
Truck	54.2(+0.8)	61.7(**-1.1**)	48.7(**-1.5**)	219(↓**12.3%**)
Train	0.0 (+0.0)	0.0 (+0.0)	-0.6 (+0.0)	0.0(↓ 0.0%)
Motorcycle	46.6(**+1.0**)	58.6(+0.7)	39.2(**+3.2**)	11(↓**27.3%**)
Bycicle	47.8(+0.2)	60.1(+0.3)	43.1(+0.5)	144(↓ 0.7%)
Detect_average	63.3(+0.1)	72.5(+0.0)	68.4(-0.1)	8750(↓ 2.8%)
Class_average	46.1(**+1.0**)	55.5(+0.6)	45.2(+0.9)	8750(↓ 2.8%)

Table 6. Ablation study of Class Relaxation and Correction in multi-calss tracking, the performance changes they bring are given in parentheses, with the larger changes highlighted in bold.

4.2. Benchmark Evaluation

To validate generalizability, we conduct experiments on five public benchmarks with significantly different characteristics: multi-class tracking in autonomous driving scenario BDD100K, accompanied by low frame rates, large variations and the presence of small targets (in Table 1); ball game scenario SportsMOT with complex motion patterns (in Table 2); dancing scenario DanceTrack, comes with serious motion complexity and large variation amplitude (in Table 3); pedestrian tracking scenarios MOT17 and MOT20 with regular motion, high frame rate, along with higher target densities and the presence of a large number of small targets (in Table 4).

Generalizability Analysis. Rather than requiring strong hypotheses and huge training resources, our approach can be generalized to diverse scenarios with great performance. Even though ByteTrack selects different association information in different datasets and GHOST adjusts the weighting weights for association information in different datasets, we still outperform them on all datasets. For MOTRv2, which requires more than 200 times training resource usage than us, we perform better on all datasets except DanceTrack with very little training resources and time. More detailed analyses on benchmarks are provided in the supplementary material.

4.3. Ablation Studies

We conduct ablation experiments on the BDD100K validation set to evaluate our design of each component. Note that the ablation results are evaluated using the toolkit and the benchmark results are obtained by submitting to the evaluation server.

Multi-scale Relation. Multi-scale relation (MR) is designed to enable the tracker to cope with different frame rates and variation amplitudes. We conducted experiments at three frame rate settings (original frame rate as well as double and quadruple downsampling). As shown in the first six rows of Table 5, all metrics got worse with ablation under all three settings, indicating the key role MR plays in tracking. As the frame rate drops, the performance variations intensify, and under setting3, MR brings a boost (+2.0 HOTA, +2.7 IDF1, +2.5 MOTA, +1.2 mHOTA, +1.4 mIDF1, +1.2 mMOTA). This suggests that the lower the frame rate, the more powerful MR is in the scene.

Searching Radius. As illustrated in Table 5, changing the searching radius has little effect on tracker's performance. Because the correlation pyramid is robust to this parameter, the correlation pyramid has the flexibility to cover multi-scale regions at any radius.

Hierarchical Relation Agregation. We ablate this part by setting the RoiAlign parameter $v*v$ to $1*1$ and treating it as a rigid body as in traditional methods. As Table 5 shows, all metrics worsened with ablation, suggesting that this component affects the tracking of all targets. This highlights the importance of transforming a rigid body into a flexible body during the matching procedure.

Class Relaxation and Correction. In order to observe the performance changes in each class, we list them separately in Table 6. The performance of rider, bus, and motorcycle improved substantially, reflecting that many targets are mistaken for other classes during the detection process, resulting in unsuccessful matches. Although two metrics of truck decreases, the average metrics still improve, suggesting that our method reduces a large number of false negatives while having a low false positive rate. This compensates for the weakness of the detector's classification ability.

4.4. Visualization

We focus on the red bus in Figure 5, where GHOST [43] experienced tracklet interruptions, id switch, and misclassification. We investigate why these phenomena occur. It is because low score detection cannot depend on a valid Intersection over Union (IoU) or a poor appearance to be tracked when there is a large movement amplitude; the misclassifi-

Figure 5. **Visualization of tracking results comparison.** Note that we use exactly the same detection results as for GHOST. The boxes of different colors represent the bounding boxes with different identities. The red bus shown in bold is the target of our comparison. It experienced tracklet interruptions, id switch, and misclassification in GHOST, in the meantime these were resolved in our approach.

Class	Car	Peds	Rider	Bus	Truck	Train	Motocy	Bycicle
Setting	**Source & Target**							
HOTA↑	66.2	50.4	47.3	62.1	55.0	0	47.6	48.0
IDF1↑	75.7	60.8	60.7	70.9	60.6	0	59.3	60.4
MOTA↑	73.0	55.8	48.9	58.4	47.2	-0.6	42.4	43.6
IDs↓	5917	2209	29	45	192	0	8	143
Setting	**Source**	**Target**						
HOTA↑	65.8	48.9	45.6	61.8	54.6	0	47.7	47.7
IDF1↑	74.9	58.6	57.7	70.4	60.4	0	60	59.8
MOTA↑	72.8	54.3	44.1	58.9	46.9	-0.6	41.7	43.3
IDs↓	6186	2790	23	44	140	0	8	152

Table 7. Domain generalization for data with different classes.

Training **(Source)**	Inference **(Target)**	HOTA↑	MOTA↑	IDF1↑	AssA↑	DetA↑
SportsMOT	SportsMOT	75.0	95.6	77.9	63.6	88.4
BDD100K	SportsMOT	73.8	95.7	76.7	61.6	88.4
DanceTrack	DanceTrack	56.9	90.1	57.5	41.1	79.1
BDD100K	DanceTrack	54.9	89.2	55.3	38.4	78.7

Table 8. Domain generalization for data in different datasets.

cation and id switch are due to the detector's class misjudgment. In contrast, we performed a complete track of it and successfully corrected the classification errors.

4.5. Domain Generalization

In this part, we take a further step to experiment and analyze whether our GeneralTrack can also perform well in domain generalization setting, which is a new and critical challenge in MOT field [42]. Domain generalization in MOT is organized into two phases, detection and association. Here we conduct domain generalization experiments for the association part, *i.e.*, training only on the source domain and inference on the unseen target domain without fine-tuning.

Cross-class and Cross-dataset Experiments. In Table 7, we set up domain generalization between different classes, *i.e.*, we train only with targets in car class and inference over all classes. It can be noted that there is still excellent tracking performance on the seven unseen classes. As shown in Table 8, we train on BDD100K and then generalize to SportsMOT and DanceTrack without fine-tuning. Comparison with results trained and inferred in the same domain demonstrates that our GeneralTrack has strong domain generalization capabilities.

Analysis. Focusing on the local key texture of the target is more generalizable than the global structural information [19, 32]. We accomplish tracking by constructing a point-wise relations between frames, which is based on low-level visual information such as textures, shapes, and corners points, *etc.*. These low-level visual information is shared by all targets and has greater flexibility and generalizability. So these factors enable our GeneralTrack the ability to domain generalization.

5. Conclusion

In this paper, we explore the difficulties in trackers' generalizability to diverse scenarios, and concretize them into a set of tracking scenario attributes that can guide the design of future trackers. Furthermore, guided by these attributes, we propose a "point-wise to instance-wise relation" framework for MOT, *i.e.*, GeneralTrack. We achieve excellent performance on multiple datasets while avoiding the need to balance motion and appearance and experimentally demonstrated great potential for domain generalization with unseen data distributions (cross-dataset, cross-class).

Limitation and Future Work. We focus more on modeling inter-frame relations and do not extend it to cross-frame relations. Inspired by [50, 61], in our next version, we will construct the relations between multiple frames based on video clips to achieve better tracking performance.

Acknowledgement

This work was supported in part by NSFC under Grants 62088102 and 62106192, Natural Science Foundation of Shaanxi Province under Grant 2022JC-41, and Fundamental Research Funds for the Central Universities under Grant XTR042021005.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprinbytetrack arXiv:2206.14651*, 2022. 1, 2

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, pages 1–8, 2008. 1

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019. 1

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *JIVP*, 2008:1–10, 2008. 6

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468, 2016. 1, 2

[6] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *CVPR*, pages 9686–9696, 2023. 2, 6

[7] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. DeepDriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, pages 2722–2730, 2015. 1

[8] Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. Sportsmot: A large multi-object tracking dataset in multiple sports scenes. *arXiv preprint arXiv:2304.05170*, 2023. 2, 3, 6

[9] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 2, 3, 6

[10] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Rethinking optical flow from geometric matching consistent perspective. In *CVPR*, pages 1337–1347, 2023. 3, 4

[11] Yonghao Dong, Le Wang, Sanping Zhou, and Gang Hua. Sparse instance conditioned multimodal trajectory prediction. In *ICCV*, pages 9763–9772, 2023. 2

[12] 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*, pages 2758–2766, 2015. 2

[13] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. StrongSORT: Make deepsort great again. *IEEE T-MM*, 2023. 1, 2

[14] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3, 6

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32 (11):1231–1237, 2013. 5

[16] Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, and Xiaofeng Pan. MAT: Motion-aware multi-object tracking. *Neurocomputing*, 476:75–86, 2022. 2

[17] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In *CVPR*, pages 5299–5309, 2021. 2

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pages 2961–2969, 2017. 5

[19] Andrew Hou, Ze Zhang, Michel Sarkis, Ning Bi, Yiying Tong, and Xiaoming Liu. Towards high fidelity face relighting with realistic shadows. In *CVPR*, pages 14719–14728, 2021. 8

[20] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Lite-flownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, pages 8981–8989, 2018. 2

[21] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow cnn—revisiting data fidelity and regularization. *IEEE T-PAMI*, 43(8):2555–2569, 2020. 2

[22] Tarasha Khurana, Achal Dave, and Deva Ramanan. Detecting invisible people. In *ICCV*, pages 3174–3184, 2021. 2

[23] Chanho Kim, Li Fuxin, Mazen Alotaibi, and James M Rehg. Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking. In *CVPR*, 2021. 2

[24] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, pages 83–97, 1955. 2, 5

[25] Siyuan Li, Martin Danelljan, Henghui Ding, Thomas E Huang, and Fisher Yu. Tracking every thing in the wild. In *ECCV*, pages 498–515, 2022. 6

[26] Yizhe Li, Sanping Zhou, Zheng Qin, Le Wang, Jinjun Wang, and Nanning Zheng. Single-shot and multi-shot feature learning for multi-object tracking. *arXiv preprint arXiv:2311.10382*, 2023. 2

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 3

[28] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *IJCV*, 129:548–578, 2021. 6

[29] Gianluca Mancusi, Aniello Panariello, Angelo Porrello, Matteo Fabbri, Simone Calderara, and Rita Cucchiara. DARTHTrackFlow: Multi-object tracking with normalizing flows. In *ICCV*, pages 9531–9543, 2023. 2

[30] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 2, 3, 6

[31] Venkatraman Narayanan, Bala Murali Manoghar, Rama Prashanth RV, Phu Pham, and Aniket Bera. Seeknet: Improved human instance segmentation and tracking via reinforcement learning based optimized robot relocation. *arXiv e-prints*, 2020. 1

[32] Pauline C Ng and Steven Henikoff. SIFT: Predicting amino acid changes that affect protein function. *Nucleic acids research*, 31(13):3812–3814, 2003. 8

[33] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, pages 3153–3160, 2011. 1

[34] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, pages 164–173, 2021. 2

[35] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, pages 164–173, 2021. 2, 6

[36] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. MotionTrack: Learning robust short-term and long-term motions for multi-object tracking. In *CVPR*, pages 17939–17948, 2023. 1, 2

[37] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, pages 4161–4170, 2017. 2

[38] Hao Ren, Shoudong Han, Huilin Ding, Ziwen Zhang, Hongwei Wang, and Faquan Wang. Focus on details: Online multi-object tracking with diverse fine-grained representation. In *CVPR*, pages 11289–11298, 2023. 1, 6

[39] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019. 3

[40] Fatemeh Saleh, Sadegh Aliakbarian, Hamid Rezatofighi, Mathieu Salzmann, and Stephen Gould. Probabilistic tracklet scoring and inpainting for multiple object tracking. In *CVPR*, pages 14329–14339, 2021. 1, 2

[41] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 3

[42] Mattia Segu, Bernt Schiele, and Fisher Yu. DARTH: Holistic test-time adaptation for multiple object tracking. In *ICCV*, pages 9717–9727, 2023. 8

[43] Jenny Seidenschwarz, Guillem Brasó, Víctor Castro Serrano, Ismail Elezi, and Laura Leal-Taixé. Simple cues lead to a strong multi-object tracker. In *CVPR*, pages 13813–13823, 2023. 1, 2, 3, 6, 7

[44] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 2

[45] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. *IEEE T-PAMI*, 42(6):1408–1423, 2019. 2

[46] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. DanceTrack: Multi-object tracking in uniform appearance and diverse motion. In *CVPR*, pages 20993–21002, 2022. 2, 3, 6

[47] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 2, 4

[48] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *ICCV*, pages 10840–10849, 2021. 2

[49] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *CVPR*, pages 3876–3886, 2021. 2

[50] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *ICCV*, 2023. 8

[51] Jun Wei, Shuhui Wang, and Qingming Huang. F^3Net: fusion, feedback and focus for salient object detection. In *AAAI*, pages 12321–12328, 2020. 5

[52] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649, 2017. 1, 2

[53] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, pages 733–751, 2022. 6

[54] En Yu, Zhuoling Li, and Shoudong Han. Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking. In *CVPR*, pages 8834–8843, 2022. 2

[55] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2633–2642, 2020. 2, 3, 6

[56] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *ECCV*, pages 659–675, 2022. 6

[57] Jimuyang Zhang, Sanping Zhou, Xin Chang, Fangbin Wan, Jinjun Wang, Yang Wu, and Dong Huang. Multiple object tracking by flowing and fusing. *arXiv preprint arXiv:2001.11180*, 2020. 2

[58] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 129: 3069–3087, 2021. 5

[59] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *ECCV*, pages 1–21, 2022. 1, 2, 3, 6

[60] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *CVPR*, pages 22056–22065, 2023. 6

[61] 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*, pages 19855–19865, 2023. 8

[62] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, pages 474–490, 2020. 1

[63] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, pages 8771–8780, 2022. 6