Foundation Model Driven Appearance Extraction for Robust Multiple Object Tracking

Teng Fu, Haiyang Yu, Ke Niu, Bin Li*, Xiangyang Xue*

Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University
{tfu23, kniu22}@m.fudan.edu.cn, {hyyu20, libin, xyxue}@fudan.edu.cn

Abstract

Multiple Object Tracking (MOT) is a fundamental task in computer vision. Existing methods utilize motion information or appearance information to perform object tracking. However, these algorithms still struggle with special circumstances, such as occlusion and blurring in complex scenes. Inspired by the fact that people can pinpoint objects through verbal descriptions, we explore performing long-term robust tracking using semantic features of objects. Motivated by the success of the multimodal foundation model in text-image alignment, we reconsider the appearance feature extraction module in MOT and propose a **F**oundation model **D**riven multi-object tracker (**FDTracker**). Specifically, we propose a two-stage trained appearance feature extractor. In the first stage, using a single image of the object as input, the model could capture the attributes of objects with the assistance of natural language instructions. In the second stage, using a sequence of images of objects as input, the model learns how to use these attributes to distinguish between different objects and connect the same object at different times. Finally, for coordinating appearance and motion information, we propose a reasonable combined strategy, which better facilitates trajectory assignment and reconnection. Extensive experiments on benchmarks demonstrate the robustness of FDTracker.

Introduction

Multiple Object Tracking (MOT) is one of the fundamental tasks in computer vision. Among many kinds of objects, pedestrian tracking is the most popular and valuable research field and has many downstream tasks like video surveillance and autonomous driving. MOT aims to localize each object in the input sequences and assign a unique ID to each object. There are two mainstream paradigms to address the task, Tracking-By-Detection (TBD) (Aharon, Orfaig, and Bobrovsky 2022; Yang et al. 2023; Zhou, Wang, and Krähenbühl 2019) and Transformer-based methods (Cai et al. 2022; Fu et al. 2023; Qin et al. 2023). The former calculates appearance similarity (Ma et al. 2022) and spatial location similarity (Zhang et al. 2022b) between existing active tracklets and new detection results and then performs the association using two distance metrics (or one of them). Due to the high-quality detection results from the existing detectors (Ge et al. 2021; Ren et al. 2015), those methods tend to achieve higher performance.

Figure 1: FDTracker provide accurate verbal descriptions of personal attributes. Thus, when a person disappears for a period of time and reappears, it can be reconnected to the original tracklet based on the appearance description.

Existing methods use motion or appearance information to associate objects in consecutive frames. The former typically uses the Kalman Filter for spatial location information updating, while the latter uses a ReID model to extract the appearance feature vector. However, existing methods still have some shortcomings. On the one hand, some methods use only a few frames or ignore the utilization of temporal information. On the other hand, they still struggle with special circumstances such as occlusion and blurring in complex scenes. Although motion information has been proven (Zhang et al. 2022b) to be good enough in existing benchmarks, we believe that appearance features extracted from long-time sequences are vital and promising solutions for those complex scenes.

In our quest to develop a more robust model for appearance feature extraction, we have observed that humans can quickly and clearly distinguish objects using natural language in complex scenarios. Inspired by the success of multimodal foundation models in text-image alignment (Radford et al. 2021; Li et al. 2022, 2023), we propose FDTracker, a **F**oundation model **D**riven tracker with robust appearance feature extraction capability, as shown in Figure 1. Specifically, FDTracker processes each frame sequentially and consists of three main stages: object detection, distance matrix calculation and tracklet management. Firstly, a pretrained detection model will detect objects in the frame. Then, the distance matrix calculation stage will calculate

*Corresponding authors
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

the appearance distance matrix and the IOU distance matrix and combine them in a reasonable way. For the IOU distance matrix, our method predicts the bounding boxes of the active tracklets in the current time step by Kalman Filter, and computes the IOU distance between those predictions and the new-detected bounding boxes in the first stage. For the appearance distance matrix, we propose a Temporal Appearance Extraction (TAE) module, which utilizes the robust language understanding and image encoding capabilities of foundation models. After two stages of training, the TAE module extracts robust appearance embeddings of each tracking object, and the appearance distance matrix can be formed by calculating the cosine similarities between those embeddings and the appearance embeddings of new-detected objects. Finally, the tracklet management stage will assign new-detected objects to existing tracklets according to the combined distance matrix. The interruption and initialization of the tracklets will also occur in this step. Besides, because we extract the detailed semantic features of the object, our method can still reconnect the object to its tracklet after it disappears for a long time and reappears, as shown in Figure 1.

We evaluate our FDTracker on MOT Challenge (Milan et al. 2016; Dendorfer et al. 2020) and DanceTrack (Sun et al. 2022) benchmarks for pedestrian tracking. Experimental results show that FDTracker achieves state-of-the-art or comparable performance. Extensive ablation studies further validate the motivation and effectiveness of FDTracker. The main contributions of the proposed method can be summarized as follows:

- Drawing inspiration from how humans describe and distinguish objects, driven by a multimodal foundation model, we propose a Tracking-By-Detection framework for MOT, which effectively tracks objects, especially in complex scenes.
- We propose a two-stage training Temporal Appearance Extraction module, which extracts the detailed semantic features of the objects. We propose an improved method of fusing two distance matrices, facilitating the association and reconnection of lost tracklets.
- The evaluation results on three benchmarks verify the effectiveness and generalization ability of the proposed FDTracker.

Related Work

In this section, we briefly introduce two mainstream paradigms in MOT. And then, we introduce the appearance feature extraction in those methods.

Tracking-By-Detection Methods. The methods using the Tracking-By-Detection paradigm usually consist of two stages. Firstly, a detection model (Ge et al. 2021; Ren et al. 2015) is used to detect objects in the frame. Then an association stage is performed on existing tracklets and detection results using spatial location or appearance information. SORT (Bewley et al. 2016) first uses the Kalman Filter to estimate the spatial location information of objects and uses Hungarian Matching to perform the association. Deep SORT (Wojke, Bewley, and Paulus 2017) then uses a simple CNN (He et al. 2016) to extract the appearance feature and uses the fused distance matrix for association. Subsequent approaches have mostly attempted to use better detectors, more accurate motion estimation, more robust appearance feature extraction and more efficient distance matrix fusion strategies. For example, OC SORT (Cao et al. 2023) iteratively updates the Kalman Filter through interpolation after the object reappears, while BoT SORT (Aharon, Orfaig, and Bobrovsky 2022) estimates the motion of the camera to correct the prediction of the Kalman Filter. ByteTrack (Zhang et al. 2022b) effectively utilizes both high-confidence and low-confidence detection results, proving that the best tracking accuracy can be achieved using only spatial location information. Our approach demonstrates that effective appearance information utilization still contributes to better tracking accuracy, which is more noticeable when the object moves fast and is not visible for long periods.

Transformer-based Methods. With the increasing popularity of the Transformer architecture (Dosovitskiy et al. 2020; Liu et al. 2021) in Natural Language Processing (Devlin et al. 2018), this structure is now being widely used in Computer Vision tasks. It achieves comparable and even state-of-the-art results in some fields (Carion et al. 2020; Zhang et al. 2022a; Zhu et al. 2020). In MOT, TransTrack (Sun et al. 2020) replaces the detection model and motion prediction model in the TBD paradigm with a Transformer architecture, while Trackformer (Meinhardt et al. 2022) uses detection query and track query to detect new tracklets and track existing tracklets respectively. DN-MOT (Fu et al. 2023) adopts the "Noising and Denoising" approach to help the model better handle objects in crowded scenarios. Since the queries in the decoder often need to discover new tracklets and process existing tracklets simultaneously, there is still a gap between those methods and the TBD methods.

Appearance Feature Extraction. Numerous approaches attempt to use appearance features for object tracking. Deep SORT (Wojke, Bewley, and Paulus 2017) first uses a convolutional neural network with a dozen layers to extract the appearance features. JDE (Wang et al. 2020) couples this process with an object detection network. FineTrack (Ren et al. 2023) presents a Multi-head Part Mask Generator to extract fine-grained appearance representation. At the same time, UTM (Ma et al. 2022) proposed Identity-Aware Boosting Attention and Identity-Aware Erasing Attention to focus more on the external features of the objects themselves. For the first time, our approach uses underlying semantic information for appearance feature extraction. Besides, these methods trivialize or ignore the role of temporal information in appearance feature extraction. Instead, by adding explicit temporal embeddings, our method obtains more robust semantic appearance features.

Methodology

We present FDTracker, a TBD paradigm approach that comprises an existing object detection model (Ge et al. 2021) to detect objects in the frames, a distance matrix calculation module to calculate the distance between detected objects and active tracklets and a tracklet management stage.

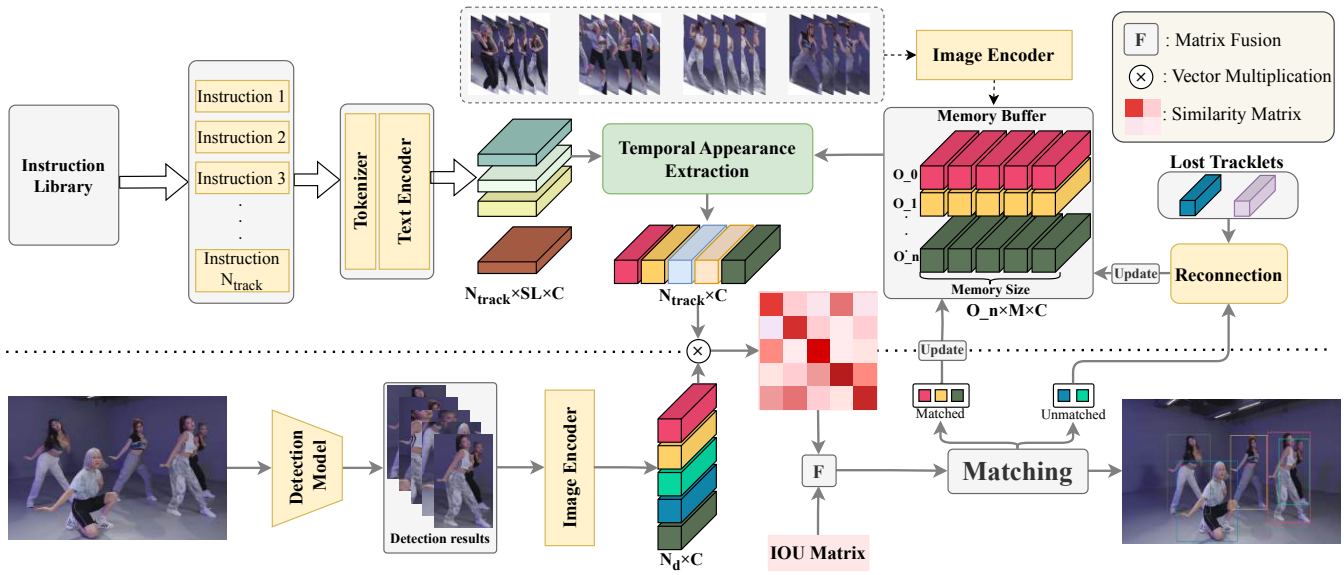

Figure 2: The overview of our proposed FDTracker. The algorithmic inference process is shown below the dotted line and the detailed appearance feature extraction process is shown above. The boxes in yellow are the frozen parts and different colors of rectangles represent different people.

This section will start with a general overview of our approach. Specifically, our method proposes a two-stage training Temporal Appearance Extraction (TAE) module, which will be presented next. Then, we introduce our tracklet management strategy. Finally, we introduce the loss functions of our method.

Overview

As shown in Figure 2, given a sequence of video frames $\boldsymbol{I} = \{I^0, I^1, \cdots, I^t\}$, the goal of online MOT is to localize a set of K objects $\boldsymbol{O} = \{O^0, O^1, \cdots, O^{k-1}\}$ over time. In each time step, the frames are first fed into an object detector like YOLOX (Ge et al. 2021) and output the detection results $\boldsymbol{D} \in \mathbf{R}^{N_d \times 5}$, which means N_d objects and each denoted as a quintuple $<x, y, w, h, conf>$. The boxed area is then passed through to the image encoder of the multimodal foundation model and output as an image embedding, while the image embeddings of the last M frames will be stored as a memory buffer for that tracklet.

Then, the N_{track} tracked objects will predict their current location by Kalman Filter, and those predictions will be compared with new-detected bounding boxes according to IOU distance and form an IOU matrix. At the same time, an instruction will be randomly sampled from a pre-constructed instruction library for each tracked object. Those instructions will interact with the image embeddings in the memory buffer and output appearance features. The appearance distance matrix could be formed by calculating the cosine similarities between appearance features and image embeddings. After fusing two matrices, the Hungarian Match algorithm will use the final distance matrix to process one-to-one matching. For those matched tracklets, we update both the location predictor using the matched detection results and the memory buffer using image embeddings following the "First-In-First-Out" rule. For those unmatched tracklets, they will convert to "unactive tracklets", keeping their appearance features. Once it's been "unactive tracklets" too long, they will be treated as "lost tracklets" and will not attend the normal matching process. Those unmatched detections, they will reactivate the "lost tracklets" or become a new tracklet after a trial period of several time steps.

Temporal Appearance Extraction

The proposed TAE module takes a text embedding and M image embeddings as input. The text embedding is obtained from an instruction processed by a tokenizer and text encoder, denoted as $\mathbf{T} \in \mathbf{R}^{SL \times C}$, SL is the sequence length of the instruction and C is the channel dimension. The image embeddings are from a memory buffer with the size of M, denoted as $\mathbf{I} \in \mathbf{R}^{M \times C}$. As shown in Figure 3, $\mathbf{I}$ is first added with the time position embeddings explicitly representing the different time steps, and then the information from the various time steps is made to interact with each other by a self-attention layer, which can be formulated as:

$$\mathbf{I_o} = \mathrm{SA}(\mathrm{Concat}(\mathbf{I} \oplus \mathbf{Time\_pos})) \tag{1}$$

where $\mathbf{Time\_pos}$ are learnable parameters with the same shape of $\mathbf{I}$, which explicitly represent the individual time steps. Then, $\mathbf{T}$ interacts with image embedding in $\mathbf{I}_o$ sequentially after an MLP. After passing another MLP, we obtain the final appearance vector, which can be formulated as:

$$\begin{aligned} \mathbf{T_0} &= \mathrm{MLP}(\mathbf{T}) \\ \mathbf{T}_i &= \mathrm{CA}(\mathbf{T}_{i-1}, \mathbf{I_o}[i]),\ i = 1, 2, \cdots, M \\ \mathbf{T_{out}} &= \mathrm{MLP}(\mathbf{T}_M) \end{aligned} \tag{2}$$

where CA denotes Cross Attention function module and $\mathbf{I_o}[i]$ denotes i_{th} embedding in $\mathbf{I_o}$.

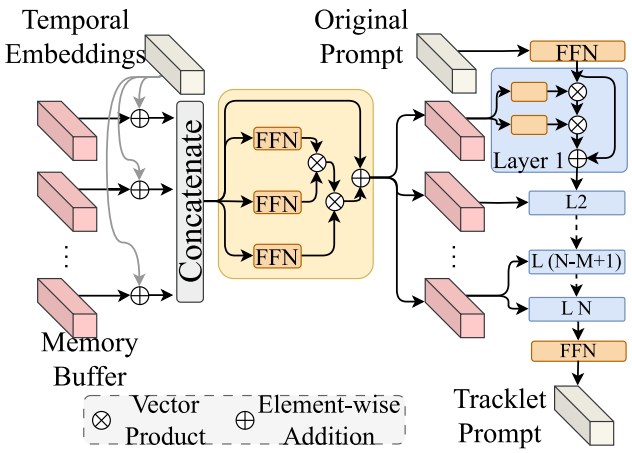

Figure 3: The proposed TAE module. the self-attention and cross-attention modules are represented in the blue and yellow boxes, respectively. The image embeddings and text embeddings are represented by pink and white rectangles, respectively.

Two-stage Training

As shown in Figure 4, we propose a two-stage training strategy to train our TAE module. We use frozen CLIP (Radford et al. 2021), and unless otherwise specified, the text encoder and image encoder in the following text are the corresponding text encoder and image encoder in CLIP. We focus on six attributes of the pedestrian: age and gender, hairstyle, top, bottom, shoes, and belongings. For each detail, we can add a description or a special $<unknown>$ term. Then we manually design dozens of sentences to string together these attributes and form an instruction library. Besides, like BERT (Devlin et al. 2018), we add $<cls>$ at the top of the sentence to summarize the statement.

In the first training stage, we train the TAE module to perceive these attributes. We use the SYNTH-PEDES (Zuo et al. 2023) as our training set, which has around 1.5 million pedestrians with attributes. The same number of instructions as batch size will be sampled from the instruction library for each training batch. After tokenizing and encoding, seven learnable embeddings will replace the $<cls>$ and six attributes in the instructions. The image embeddings will be copied times of memory buffer size and use only the time position embedding of the last time step in this stage. After the TAE module, seven learnable embeddings will calculate the similarities with the embeddings encoded from the original language description in the ground truth. These similarities will be involved in the loss calculation and parameter updates.

In the second stage, we train the model to extract some other appearance information from a sequence of images. These features include several underlying aspects in addition to the six attributes above, like action and gait, etc. By intercepting objects in MOT17 and DanceTrack and retaining their IDs, we construct a ReID dataset. As shown in Figure 4, we sample $M+1$ consecutive images of a pedestrian. The first M images will be sent to the TAE module for appearance feature extraction and the last image will be used as a positive sample for contrastive learning. Like CLIP, different objects in the same batch are negative samples of each other. Besides, we sample an additional random image for each object throughout its lifetime and we name them "weak positive sample". The appearance features generated from the TAE module will be contrasted with the image embeddings of different pedestrians and the image embeddings of the same pedestrian at different moments.

Tracklet Management

The matrices used to represent spatial location and appearance similarity have their characteristics. The IOU distance matrix depicts the distance between objects, but once the object moves too fast or the frame rate drops, the IOU score will quickly drop to zero. Instead, the appearance similarity matrix is smoother, and objects are often somewhat similar to each other rather than directly zero similarity; After all, they are all humans. This can also lead to situations where two people with very similar appearances are perceived as the same person even though they are far apart in a frame.

This makes it more meaningful to explore better ways of fusion. Two issues need to be addressed. First, we must address the fact that the IOU score decreases rapidly as movement speed increases. Second, we must solve the problem that there is still a certain degree of appearance similarity between different people.

We first normalize the appearance similarity, for an element d_{app} in the distance matrix. The process is as follows:

$$d_{\text{app}} = \begin{cases} (d_{\text{app}} + 1)/2, & d_{\text{app}} > \lambda_{\text{app}} \\ 0, & otherwise \end{cases} \tag{3}$$

where λ_{app} is a threshold. We then use a weighted summation to fuse the two similarity matrices:

$$M_{\text{fusion}} = \lambda_f * M_{\text{iou}} + (1 - \lambda_f) * M_{\text{app}} \tag{4}$$

the λ_f is the weighting factor, and the M_{fusion} will be involved in the final matching phase.

Once a tracklet has been determined to be in "Lost" status, the last generated appearance embedding for this tracklet will be retained. For every specific time step, detections that failed to get assigned will be matched with these lost tracklets considering only the appearance similarity. After passing through a higher confidence filter λ_{ltr}, matching pairs will be added to the active tracklet set and restarted to update their memory buffer.

Loss Function

In the first training stage, we simply used the MSE loss function to supervise the similarity between seven predictions in instruction and ground truth. In the second stage, there are positive samples, weak positive samples and negative samples for each anchor sample. Each appearance feature computes cosine similarities with all positive samples in this batch and forms a positive similarity matrix (named p), and computes cosine similarities with all weak positive samples in this batch to form a weak positive similarity matrix

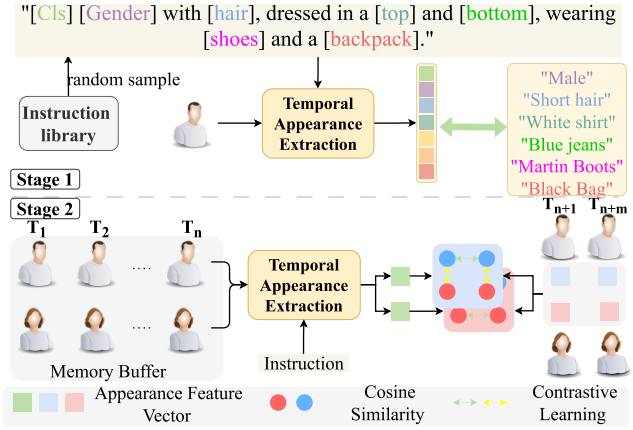

Figure 4: The two-stage training process for the TAE module. The batch size is 2 in stage 2 in the figure.

(named wp). To generate a more robust appearance vector, we adopt three single loss functions and combine them.

Cross Entropy Loss. Like CLIP (Radford et al. 2021), we compute the cross entropy loss for each matrix as follows:

$$\mathcal{L}_{ce}(D_*) = \frac{\mathrm{CE}(D_*/\tau) + \mathrm{CE}(D_*^{\mathrm{T}}/\tau)}{2} \tag{5}$$

where $\mathrm{CE}(\cdot)$ denotes Cross Entropy function, D_* denotes D_p or D_{wp}, τ denotes temperature parameter and $(\cdot)^{\mathrm{T}}$ denotes transpose operation. The final CE Loss is computed as follows:

$$\mathcal{L}_{CE} = \mathcal{L}_{ce}(D_p) + \mathcal{L}_{ce}(D_{wp}) \tag{6}$$

Triplet Loss. Cross Entropy Loss is not performing too well in subtle categorization tasks, so like some methods in Person Re-Identification (Chen et al. 2017), we use the combined optimization of Cross Entropy and Triplet Loss. For an anchor sample and its matching positive sample and negative sample, we compute the Triplet Loss as follows:

$$d_{tri}(d_+, d_-) = \mathrm{ReLU}(d_+ - d_- + margin) \tag{7}$$

$$\mathcal{L}_{tri}(D_*) = \frac{1}{n}\sum_{i}^{bs}\sum_{j,j\neq i}^{bs} d_{tri}(D_{*,i,i}, D_{*,i,j}) \tag{8}$$

where $D_{*,i,j}$ denotes the similarity in the location (i, j) of the matrix D_*, $n = bs \times (bs - 1)$ and the $margin$ is a constant greater than 0. We compute Triplet Loss in both similarity matrices:

$$\mathcal{L}_{TRI} = \mathcal{L}_{tri}(D_p) + \mathcal{L}_{tri}(D_{wp}) \tag{9}$$

Temporal Regularity Constraint. We want the model to be more similar to positive samples than to weak positive samples as a way to enhance the model's perception of time. So we add an extra regular loss:

$$\mathcal{L}_{RE} = \frac{1}{bs}\sum_{i}^{bs}(1 - (D_{p,i,i} - D_{wp,i,i})) \tag{10}$$

We add up the above three loss functions with weights:

$$\mathcal{L} = \lambda_{ce} * \mathcal{L}_{CE} + \lambda_{tri} * \mathcal{L}_{TRI} + \lambda_{re} * \mathcal{L}_{RE} \tag{11}$$

Where λ_{ce}, λ_{tri} and λ_{re} are used to adjust the proportion of different losses and we set $\lambda_{ce} = 1, \lambda_{tri} = 10, \lambda_{re} = 0.5$ respectively.

Experiments

In this section, we demonstrate the performance of our model on three public datasets, namely MOT17 (Milan et al. 2016), MOT20 (Dendorfer et al. 2020), and DanceTrack (Sun et al. 2022). In addition, we conduct ablation experiments to verify the effectiveness of our modules.

Datasets and Metrics

Datasets. We conduct all the experiments on three publicly available benchmarks: the private track of MOT17, MOT20 and DanceTrack to ensure a fair comparison. The MOT17 consists of 14 sequences, which are divided into 7 training sequences and 7 test sequences. Due to the restrictions on the number of submissions[1], we sample half of each training sequence as our validation set followed by ByteTrack (Zhang et al. 2022b). MOT20 includes 4 training and 4 test sequences with more objects in the scene compared to MOT17. DanceTrack is a recent multi-object tracking benchmark containing 100 sequences in dance, theater, and other similar scenes.

Metrics. We use the CLEAR MOT Metrics and IDF1 (Bernardin and Stiefelhagen 2008; Ristani et al. 2016) to evaluate the effectiveness of our method, which includes some basic items for quantitative evaluation, e.g., MOTA, ML, MT, and Identity Switches (ID.Sw). Besides, to mitigate the overdependence of the above metrics on detection capacity, we also use Higher Order Tracking Accuracy (HOTA) (Luiten et al. 2021) as our evaluation metric.

Implementation Details

We choose ByteTrack as our baseline because it is a simple and efficient method that uses only a detector and a Kalman filter to perform the tracking. The enhancements achieved on this baseline illustrate the validity of our proposed module. The proposed method is implemented by Pytorch with 8 NVIDIA RTX 4090 GPUs. Our method uses the image and text encoders in CLIP (Radford et al. 2021) as corresponding modules. All the settings in FDTracker are the same as in ByteTrack (Zhang et al. 2022b) for a fair comparison. The TAE module is trained for 15 and 30 epochs, respectively, with 512 batch size and 336×336 input size in two training stages. The temperature parameter τ is set to 0.07. We adopt Adam (Kingma and Ba 2014) as our optimizer. The learning rate is initialized to 0.001 and decayed to 0.1 times its original value after 10 epochs. During inference, the λ_{app} is set to 0.4 and the λ_f is set to 0.8. The memory buffer size is set to 5. We use all the detection results whose confidence score is more than 0.1.

As ByteTrack has used the SOTA detection model YOLOX at the time, we use the brand new YOLOv8[2] as our

[1]https://motchallenge.net

[2]https://github.com/ultralytics/ultralytics

Method	Detector	MOTA↑	IDF1↑	HOTA↑	FP↓	FN↓	ID.Sw↓
MOT17							
MeMOT (Cai et al. 2022)		72.5	69.0	56.9	37221	115248	2724
MOTR (Zeng et al. 2022)		73.4	68.6	57.8	-	-	2439
FairMOT (Zhang et al. 2021)	✓	73.7	72.3	59.3	27507	117477	3303
TrackFormer (Meinhardt et al. 2022)		74.1	68.0	57.3	34602	108777	2829
GTR (Zhou et al. 2022)		75.3	71.5	59.1	26793	109854	2859
DNMOT (Fu et al. 2023)		75.6	68.1	58.0	24960	110064	2529
MOTRv2 (Zhang, Wang, and Zhang 2023)	✓	78.6	75.0	62.0	23409	94797	2619
BoT-SORT (Aharon, Orfaig, and Bobrovsky 2022)	✓	80.6	**79.5**	**64.6**	**22524**	85398	**1257**
UTM (Ma et al. 2022)	✓	**81.8**	78.7	64.0	25077	**76298**	1431
ByteTrack* (Zhang et al. 2022b)	✓	80.3	77.3	63.1	25491	83721	2196
FDTracker(Ours)	✓	**81.8**	79.2	64.2	23024	77402	2245
MOT20							
TransTrack (Sun et al. 2020)		64.5	59.2	48.9	28566	151377	3565
CorrTracker (Wang et al. 2021)	✓	65.2	69.1	-	79429	95855	5183
TransCenter (Xu et al. 2021)		67.7	58.7	43.5	56435	107163	3759
GHOST (Seidenschwarz et al. 2023)	✓	73.7	75.2	61.2	-	-	1264
BoT-SORT (Aharon, Orfaig, and Bobrovsky 2022)	✓	77.8	**77.5**	63.3	**24638**	88863	1257
UTM (Ma et al. 2022)	✓	78.2	76.9	62.5	29964	**81516**	1228
ByteTrack* (Zhang et al. 2022b)	✓	77.8	75.2	61.3	26249	87594	**1223**
FDTracker(Ours)	✓	**78.3**	76.9	**63.4**	26122	84924	1294

Table 1: Performance comparison between our methods and existing methods under the private detection protocols. The method marked by "*" represents our baseline and the bold and underlined numbers represent the best and second-best results, respectively. "Detector" represents whether existing detectors are used.

Method	Detector	MOTA ↑	HOTA ↑
OC-SORT (Cao et al. 2023)	✓	75.5	62.1
MOTR (Zeng et al. 2022)		79.7	54.2
TraDeS (Wu et al. 2021)	✓	86.2	43.3
DNMOT (Fu et al. 2023)		89.1	53.5
GHOST (Seidenschwarz et al. 2023)	✓	91.3	56.7
MOTRv2 (Zhang, Wang, and Zhang 2023)	✓	91.9	69.9
ByteTrack* (Zhang et al. 2022b)	✓	89.6	47.7
FDTracker(Ours)	✓	**92.1**	**70.6**

Table 2: Performance comparison between FDTracker and existing methods on the DanceTrack test set.

detector and name the version FDTracker+ to obtain models with higher application value.

Benchmark Evaluation

MOT Challenge. We compare our methods with some algorithms on the test set of MOT17 and MOT20 under private detection setting and summarize the related results in Table 1. The compared methods adopt either the Tracking-By-Detection paradigm or the Transformer-based paradigm. As shown in Table 1, our methods perform better than existing methods on most metrics. For the MOT17, Our methods achieve the best or comparable accuracy (i.e. 81.8 MOTA, 79.2 IDF1, and 64.2 HOTA). It is worth mentioning that our approach does not bother to design tricks for specific benchmarks, nor does it set different parameters for different sequences. Compared to our baseline, our method significantly improved (+1.5 MOTA and +1.1 HOTA).

The test set of MOT20 has fewer sequences but more objects, which is more difficult for the existing algorithms. Overcrowding leads to similar motion information and overlapping bounding boxes, so using any of the above distance metrics is not robust. Our method achieves 78.3 MOTA, 76.7 IDF1, and 63.4 HOTA, proving that the fusion of motion and appearance information is more effective under overcrowded scenes. Our method significantly improved over the best previous methods (i.e., +0.1 MOTA and +0.9 HOTA), all these indicate that FDTracker is a robust and strong tracker. Our method tries to reactivate tracklets that have ended due to complex scenes, so some incorrect attempts lead to an improvement in ID.Sw, but it leads to a more significant enhancement in other metrics like FN especially in more complex MOT20.

DanceTrack. We conduct experiments on the recently proposed DanceTrack dataset (Sun et al. 2022) and summarize the related results in Table 2. DanceTrack is characterized by two features, one of which is that the objects contain similar dress patterns, and the other is that the movement of the objects in it is strongly unpredictable. These simultaneously present challenges for both motion prediction and appearance feature extraction. Our methods finally achieved 92.1 MOTA and 70.6 HOTA, the best HOTA and MOTA scores among all the methods. Comparing our baseline, we get a significant improvement. This suggests that our approach can still extract robust appearance features even in cases where objects are similar in appearance.

BDD100k. We aim to extract semantic information for pedestrians for better robust tracking over long periods of time, and due to the powerful comprehension of the foundation model, we also extend our research to other object classes. Since there is no relevant dataset, we discarded the first stage of training and tailored the instructions in the second stage for objects other than pedestrians. We conducted experiments on BDD100k (Yu et al. 2020) and achieved the SOTA results of 45.6 mMOTA and 76.3 MOTA, which

Method	Image Encoder	Text Encoder	Input Image Size	MOTA↑	IDF1↑	ID.Sw↓
CLIP (Radford et al. 2021)	$ViT-L/14$ (Dosovitskiy et al. 2020)	$Transformer$ (Vaswani et al. 2017)	$(336, 336)$	90.2	86.1	178
BLIP (Li et al. 2022)	$ViT-L/16$ (Dosovitskiy et al. 2020)	$Transformer$ (Devlin et al. 2018)	$(384, 384)$	90.2	**86.4**	**172**
BLIP2 (Li et al. 2023)	$ViT-g/14$ (Dosovitskiy et al. 2020)	$FlanT5$ (Chung et al. 2022)	$(224, 224)$	**90.4**	83.1	263

Table 3: The effect of different foundation models. We have retained the original input sizes to ensure the model is fully functional.

Method	stage 1	stage 2	MOTA↑	IDF1↑	ID.Sw↓
Baseline (Zhang et al. 2022b)			87.0	80.1	477
	✓		77.6	71.2	1244
+ TAE Module		✓	89.6	81.6	277
	✓	✓	**90.2**	83.4	232
+ Tracklet Management	✓	✓	**90.2**	**86.1**	**178**

Table 4: The effect of our contributions. We adopt ByteTrack (Zhang et al. 2022b) with YOLOX-m (Ge et al. 2021) as our baseline. "stage 1" and "stage 2" denote two stages of training of TAE module.

proved that our appearance feature extractor has a strong generalization with the ability of a large model.

Ablation Study

FDTracker Components. Table 4 shows the impact of integrating different components. We adopt ByteTrack (Zhang et al. 2022b) equipped YOLOX-m as our baseline without the appearance feature extraction module. Integrating the proposed modules can gradually improve the overall performance. As shown in Table 4, the first training stage fails to achieve positive results. We infer that because there exists a domain gap between MOT datasets and the ReID datasets, and on the other hand, just using the first stage training doesn't allow the model to extract more information from the sequences. In addition to the six attributes we manually designed, this information includes the character's movements, gait, etc. Instead, using the second training can achieve positive results. Optimal MOTA scores can be achieved using two-stage training. Finally, the Tracklet Management module proposes a new approach to combine distance matrices and gives those lost tracklets a chance to reconnect, so we obtain 90.2 MOTA, 86.1 IDF1, and 178 ID Switches, which is a significant improvement (3.2 MOTA, 6.0 IDF1 improvement, and 63% ID Switch reduction).

Foundation Model. Our image and text encoders are frozen, so choosing a suitable foundation model is essential. CLIP (Radford et al. 2021) is a multi-modal model based on contrastive learning and achieves significant success in text-image alignment. BLIP (Li et al. 2022) and BLIP2 (Li et al. 2023) are both new Vision Language Pre-training frameworks that all performed well. We conducted experiments on their image and text encoder and put the results in Table 3. We ensure that all models in the comparison use the largest possible version of the model and the same input size as in the original paper to ensure maximum functionality. As the experimental results demonstrate, BLIP2 reaches the best results, obtaining 90.4 MOTA. However, the improvement in accuracy is very insignificant relative to the increase in model scales (MOTA accuracy from 90.2 to 90.4, while the parameters from 0.35B to 4.1B). So, there is no significant difference among existing foundation models in the capacity to extract the appearance features of objects in our task. Note that we do not involve the training of foundation models, so our method is essentially the same as previous methods in terms of the number of trainable parameters and computational efficiency.

Output	Time cost	MOTA↑	IDF1↑
Sentences	144h	90.3	85.5
Words	72h	90.2	86.1

Table 5: The experiment results of the different types of outputs in the first stage of training.

Discussion

The SYNTH-PEDES (Zuo et al. 2023) dataset provides pedestrian captions according to the six attributes. So we could treat the first stage of training as training an image caption model using the descriptions they provided as supervision. As Table 5 shows, using those descriptions consumed more time and failed to achieve more excellent accuracy, and we infer that because those descriptions are also from the six attributes and lack variegation of language like other image caption datasets (Ng et al. 2020). We think of increasing linguistic diversity to generate more detailed pedestrian language descriptions as a future task of this work.

Conclusion

In this work, motivated by the success of the multimodal foundation model in text-image alignment, we propose a Foundation model Driven multi-object tracker, called FDTracker, which mainly introduces a two-stage training Temporal Appearance Extraction module. Specifically, we train the model to describe six attributes of a pedestrian according to an image in the first stage, and we train the model to capture other information like action from a sequence of pedestrian images in the second stage. Besides, we propose a Tracklet Management that consists of a distance combination strategy and a lost tracklet reconnection strategy. Extensive experiments are conducted on publicly available benchmarks, and the SOTA results validate the effectiveness of FDTracker in the multiple pedestrian tracking task. We also hope our work can foster future work toward Visual Object Verbalization.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants No. 62176061 and No. 62176060, the Shanghai Platform for Neuromorphic and AI Chip under Grant 17DZ2260900 (NeuHelium), and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

References

Aharon, N.; Orfaig, R.; and Bobrovsky, B.-Z. 2022. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*.

Bernardin, K.; and Stiefelhagen, R. 2008. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008: 1–10.

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. 2016. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, 3464–3468. IEEE.

Cai, J.; Xu, M.; Li, W.; Xiong, Y.; Xia, W.; Tu, Z.; and Soatto, S. 2022. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8090–8100.

Cao, J.; Pang, J.; Weng, X.; Khirodkar, R.; and Kitani, K. 2023. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9686–9696.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 213–229. Springer.

Chen, W.; Chen, X.; Zhang, J.; and Huang, K. 2017. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 403–412.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; and Leal-Taixé, L. 2020. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fu, T.; Wang, X.; Yu, H.; Niu, K.; Li, B.; and Xue, X. 2023. DeNoising-MOT: Towards Multiple Object Tracking with Severe Occlusions. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2734–2743.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 12888–12900. PMLR.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.; Geiger, A.; Leal-Taixé, L.; and Leibe, B. 2021. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129: 548–578.

Ma, F.; Shou, M. Z.; Zhu, L.; Fan, H.; Xu, Y.; Yang, Y.; and Yan, Z. 2022. Unified transformer tracker for object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8781–8790.

Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; and Feichtenhofer, C. 2022. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8844–8854.

Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; and Schindler, K. 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.

Ng, E. G.; Pang, B.; Sharma, P.; and Soricut, R. 2020. Understanding Guided Image Captioning Performance across Domains. *arXiv preprint arXiv:2012.02339*.

Qin, Z.; Zhou, S.; Wang, L.; Duan, J.; Hua, G.; and Tang, W. 2023. MotionTrack: Learning Robust Short-term and Long-term Motions for Multi-Object Tracking. *arXiv preprint arXiv:2303.10404*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Ren, H.; Han, S.; Ding, H.; Zhang, Z.; Wang, H.; and Wang, F. 2023. Focus On Details: Online Multi-object Tracking with Diverse Fine-grained Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11289–11298.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, 17–35. Springer.

Seidenschwarz, J.; Brasó, G.; Serrano, V. C.; Elezi, I.; and Leal-Taixé, L. 2023. Simple Cues Lead to a Strong Multi-Object Tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13813–13823.

Sun, P.; Cao, J.; Jiang, Y.; Yuan, Z.; Bai, S.; Kitani, K.; and Luo, P. 2022. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20993–21002.

Sun, P.; Cao, J.; Jiang, Y.; Zhang, R.; Xie, E.; Yuan, Z.; Wang, C.; and Luo, P. 2020. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Q.; Zheng, Y.; Pan, P.; and Xu, Y. 2021. Multiple object tracking with correlation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3876–3886.

Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; and Wang, S. 2020. Towards real-time multi-object tracking. In *European Conference on Computer Vision*, 107–122. Springer.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, 3645–3649. IEEE.

Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; and Yuan, J. 2021. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12352–12361.

Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; and Alameda-Pineda, X. 2021. Transcenter: Transformers with dense queries for multiple-object tracking.

Yang, F.; Odashima, S.; Masui, S.; and Jiang, S. 2023. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4799–4808.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; and Wei, Y. 2022. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*, 659–675. Springer.

Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L. M.; and Shum, H.-Y. 2022a. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*.

Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; and Wang, X. 2022b. Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, 1–21. Springer.

Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; and Liu, W. 2021. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129: 3069–3087.

Zhang, Y.; Wang, T.; and Zhang, X. 2023. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22056–22065.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.

Zhou, X.; Yin, T.; Koltun, V.; and Krähenbühl, P. 2022. Global tracking transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8771–8780.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

Zuo, J.; Yu, C.; Sang, N.; and Gao, C. 2023. PLIP: Language-Image Pre-training for Person Representation Learning. arXiv:2305.08386.