TRANSFINER: A FULL-SCALE REFINEMENT APPROACH FOR MULTIPLE OBJECT TRACKING

Bin Sun[1]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

NOTE: Corresponding author: Bin Sun, binsun@hust.edu.cn

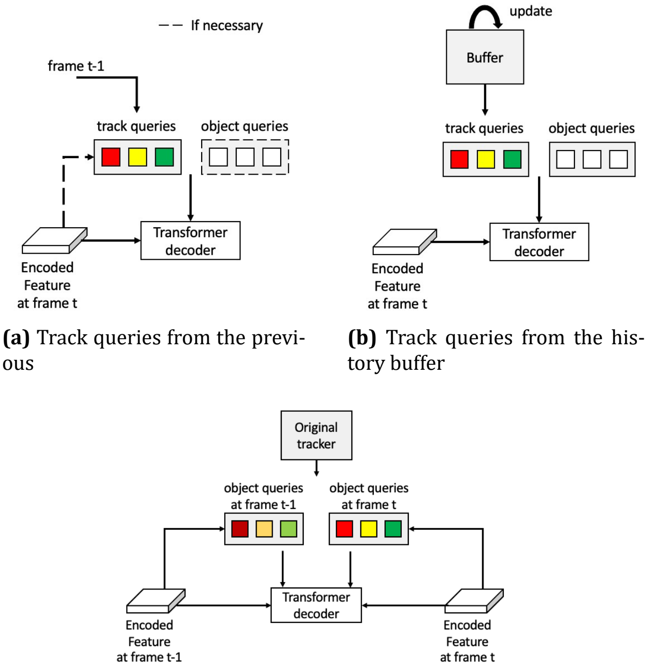
Abstract – *Multiple Object Tracking (MOT) is a task for containing detection and association. Plenty of trackers have achieved competitive performance. Unfortunately, for the lack of informative exchange on these subtasks, they are often biased toward one of the two and underperform in complex scenarios, such as the inevitable misses and mistaken trajectories of targets, when tracking individuals within a crowd. This paper proposes TransFiner, a transformer-based approach to post-refining MOT. It is a generic attachment framework that depends on query pairs, the bridge between an original tracker and TransFiner. Each query pair, through the fusion decoder, produces refined detection and motion clues for a specific object. Before that, they are feature-aligned and group-labeled under the guidance of tracking results (locations and class predictions) from the original tracker, finishing tracking refinement with focus and comprehensively. Experiments show that our design is effective, on the MOT17 benchmark, we elevate the CenterTrack from* 67.8% *MOTA and* 64.7% *IDF1 to* 71.5% *MOTA and* 66.8% *IDF1. The code is publicly available at https://github.com/BeenoSun/TransFiner.*

Keywords – Multi-object tracking, refinement, transformer

1. INTRODUCTION

Multiple Object Tracking (MOT) refers to linking identical detections across frames and primarily exists in the form of two mainstream paradigms, namely Tracking-By-Detection (TBD) and Joint Detection and Tracking (JDT). TBD approaches [1, 2, 3, 4, 5] split the MOT into two separate stages, including detection and association. JDT, alternatively, solves the MOT problem in unified ways via constructing a tracking-related structure [6, 7, 8, 9] within or adjusting the output objective of the particular branch [10] of the existing detectors. From an additionally emerging paradigm, transformer-based MOT formulations [11, 12, 13, 14, 15, 16] also finish tracking satisfactorily. Nevertheless, these methods still struggle with intricate scenarios, such as several objects passing each other and patches of crowded objects, which lead to either high false alarms (or a high miss rate) and degraded association simultaneously. On the other hand, with a Detection Transformer (DETR) [17], end-to-end object detection is realized through object queries and Hungarian loss, facilitating individual-separate detection.

In light of these, we show how to build a generic and targeted framework for refining MOT, referred to as TransFiner, a transformer-based refinement approach. Unlike most related work, Detection Refinement for Tracking (DRT) [18] refines MOT patch by patch, which indeed improves detection but hardly promotes association (even degrades it according to the IDF1 reported in experiments [18]). We, instead, take a full-scale approach by enriching query pairs guided by the original tracker (Fig. 1c), refinement then is a fine-tuning process for query pairs without scope restriction.



(a) Track queries from the previous

(b) Track queries from the history buffer

(c) Object query pairs enhanced by the original tracker (Ours)

Fig. 1 – Pipelines of preparing queries for the decoder. 1a Track queries from the previous frame, directly [12, 11] or enhanced by features from the current frame [13]. 1b History buffer is responsible for producing track queries [14, 15]. 1c Ours. As a post-refinement framework, we fill the object query pairs across frames with encoded features obtained under customized guidance from the original tracker.

As summarized in Fig. 1, the existing transformer-based MOT formulations [12, 11, 13, 14, 15, 16] primarily accomplish tracking via the tracklet record (e.g., track query). Instead, we use freshly initialized query pairs (i.e., separately for detection and association) for every shot.

With this design, we note that a competitive tracking refinement can be achieved while less affected by the formerly poor tracking predictions.

TransFiner takes originally estimated object locations, class predictions, and two successive frames as inputs, predicting detections (frame t) and association clues containing motions of center and box (mapping detections from frame t to frame $t-1$). These are achieved via TransFiner's *query pairs* plus *fusion decoder*. The latter consists of the fusion attention module and dual-decoder. Specifically, fusion attention is responsible for the interaction between query pairs, while the dual-decoder is assigned to take care of these two separately.

In order to better utilize information from the original tracker, predictions are categorized into qualified and poor ones in terms of their class scores. Together with learnable label embeddings, TransFiner finishes targeted refinement with different focuses of query embeddings on various estimations in parallel. During training, we additionally refer to ground-truth objects when preassigning refinement targets to original estimations with *close* distance, avoiding instability introduced in layerwise Hungarian matching when refining.

Experiments show that a tracker refined by TransFiner are robust enough to revisit compelling performance. With TransFiner's refinement, CenterTrack achieves 71.5% MOTA, and 66.8% IDF1 on the MOT17 benchmark.

2. RELATED WORK

2.1 Association in tracking

Motion and appearance are two crucial references when linking detections between frames. Several works rely solely on motions, guiding objects to the next frame [1, 2, 10, 8, 6] or moving them backward [7, 16] to search for associated ones. Some [19, 5] take advantage of appearance features to match interframe objects by computing similarity scores between feature embeddings. Naturally, combining both in association [3, 20, 21, 4, 22, 9, 23] is also widely explored.

Another recent popular trend builds on transformer [24], packaging the preceding information into high-level embeddings (e.g., track queries [12, 11, 13, 14, 15]). These embeddings are then processed together with the current information [12, 14, 13, 15], or they serve as the initialization in the latest detection [11], handling association problems via *another detection shot*. Our method extends this trend by injecting freshly aligned and grouped encoded features to query pairs focused on joint prediction (refinement) of detections and corresponding motions for association, which is completed in one run. Furthermore, we package information from centers and boxes into motions, facilitating precise association even among crowds.

2.2 DETR and its variants

DETR [17] handles object detection in an end-to-end manner. This benefits from the transformer's attention mechanism and the introduction of object query; however, unfortunately two dominating factors contribute to slow convergence of DETR. To be specific, several variants [25, 26, 27] improve the attention module by designing mechanisms to constrain the interaction fields (e.g., sampling points [25], additional spatial attention weight [26, 27]), easing the match burden in comparison to the inefficient global search from DETR. Different to this, object query alignment is studied in [28, 29], with the retrieval of the queries from encoded features showing effectiveness in accelerating convergence.

We build upon deformable DETR [25]. Specifically for tracking refinement, we construct a fusion decoder composed of fusion attention and a dual-decoder. Two decoders are connected through the fusion attention module, an additionally masked self-attention mechanism, ensuring effective intercommunication of query pairs. It is noteworthy that query pairs are iteratively aligned based on the inherent variable *reference locations*. Repetitive refinement is then realized through consecutive updates of the pairs via decoder layers.

2.3 Refinement

By exploring the joint space of inputs and outputs, refinement can generally be divided into multi and single-step approaches. The former involves iterative correction [30, 31] and cascaded rectification [32]. Contrastingly, the latter simply attaches an independent module to the original model [33, 34, 35, 36, 18, 37], yielding the refined results in a single pass.

MOT refinement focuses on optimizing detections and associations, and existing methods [37, 18] fall into the second category outlined above. ReMoT [37] enhances the tracklets of objects through a *split-then-merge* strategy, reducing the identity switches, which, however, are not primary causes of performance degradation. Alternatively, DRT [18] refines the detection results from ambiguous patches, resulting in decent improvements. Nevertheless, due to the patch-based nature, the scope of post-processing is limited to a predefined area, making it different from full-scale refinement and failing to strengthen association performance effortlessly, over the original tracker. These inspire the design of TransFiner, a full-scale and single-step approach to refine MOT on detection and association.

3. PRELIMINARIES

Original tracker. Generally taking a subset from frames $\{I^t\}_{t=1}^{T}$ (up to the last frame I^T of a video sequence) as

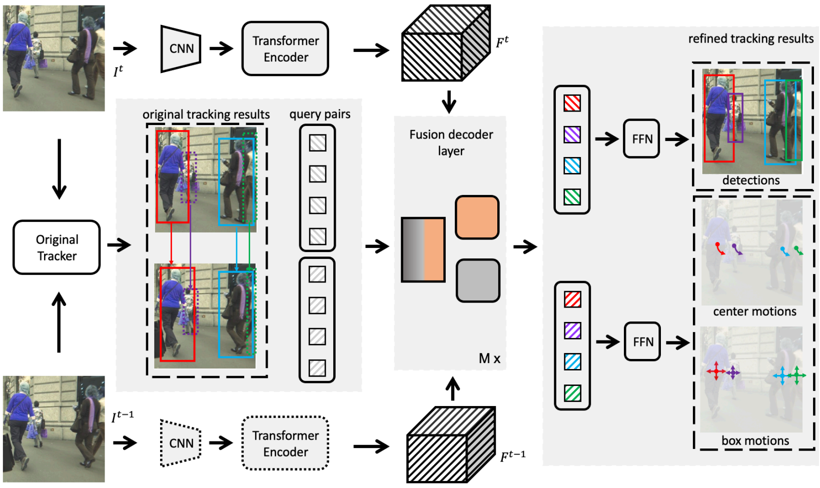


Fig. 2 – Refining tracker via TransFiner. Encoded features F^t and F^{t-1} produced by the CNN backbone and encoder, original tracking results, and plain query pairs serve as inputs for the fusion decoder (i.e., M fusion decoder layers). Following the FFN module, query pairs (Q^t, Q^{asso}) from the fusion decoder transform into detections and motions. For the original results, boxes ignored by post-processing are in dotted form and are partially picked for illustration. The dotted CNN and the encoder indicate that weights are shared with the solid ones.

input, *original tracker* refers to the tracker whose predictions are to be refined. Outputs from the origin are $\{\hat{o}_i^t\}_{i=0}^{K-1}$, where K predictions are extracted from the post-processing with *laxer* output settings (e.g., lower objectness score threshold). Let $\hat{o}_i^t = (\hat{c}_i^t, \hat{b}_i^t, \hat{a}_i^t)$, $\hat{c}_i^t$ and $\hat{b}_i^t$ respectively indicate the classification score, as well as bounding box of object i out of the K predictions in frame t. $\hat{a}^t$ represents the association clues (e.g., motions [1, 6, 10, 2, 8] or feature embeddings [19, 22, 5]) linking objects across frames.

Refinement. Let encoded image features from frame t and $t-1$ be the F^t and F^{t-1}, respectively. Performing refinement on $\{\hat{o}_i^t\}_{i=0}^{K-1}$ contributes to $\{\tilde{y}_i^t\}_{i=0}^{N-1}$, where $\tilde{y}_i^t = (\tilde{c}_i^t, \tilde{b}_i^t, \tilde{a}_i^t)$, and N is the number of queries in the decoder. $\tilde{a}_i^t$ denotes the motion of object i between frames. Additionally, TransFiner is built upon deformable DETR [25], whose decoder relies on the initial reference locations $init\_ref$ to make final predictions.

4. MOT REFINEMENT DRIVEN BY TRANSFORMER

4.1 Why transformers in refinement

Based on DETR [17] and its derivations [38, 29, 25, 28, 27], we show a transformer's superiorities over convolutional neural networks in post-refinement in the following ways: (1) In DETR-like methods, stacked decoder layers gradually rectify predictions, resembling the refinement process. (2) The object query is regarded as the *complex* of the corresponding target, the initialization of which, under the guidance of initial predictions, is finished with fetching specific image features, enabling targeted refinement. (3) Inspired by training with joint denoising and matching [38], refinement with a transformer can be cast into two parallel processes: denoising qualified predictions and rematching for the poor ones. In the following sections, we describe how TransFiner incorporates these characteristics.

4.2 Framework of TransFiner upon query pair

A core concept in TransFiner is query pairs (Q^t, Q^{asso}). Q^t detects, and Q^{asso} produces *related* motions (*related* means each pair should take a specific object). As shown in Fig. 2, query pairs propagate within the fusion decoder. Thus, framework of TransFiner can be divided into three parts: decoder's inputs, decoding, and decoder's predictions over query pairs. For inputs, we package encoded features F^t and F^{t-1}, original results $\hat{o}^t$, and plain query pairs. Decoding, after targeted initialization on query pairs under $\hat{o}^t$, focuses on fusing pairs and separately processing queries for detection (Q^t) and association (Q^{asso}), separately contributing to target estimations of frame t (i.e., $(\tilde{b}^t, \tilde{c}^t)$) and association clues $\tilde{a}^t$ as target motions of center and box relative to that of the previous frame.

4.3 TransFiner's fusion decoder layer

TransFiner's fusion decoder layer consists of two modules including *dual-decoder layer* and *fusion attention*, which are introduced below.

Dual-decoder layer. TransFiner provides association clues $\tilde{a}^t$ in the form of motions, i.e., offsets (if ideal) pointing from b^t to b^{t-1} (b for ground truth boxes). The *iterative bounding box refinement* [25] mode of a decoder works by iteratively correcting box predictions from the former decoder layer (through rectifications $\Delta\tilde{b}^t$ from Q^t). We, inside query pairs, extend the mode by simultaneously rectifying (predicting) the attached bounding boxes in frame

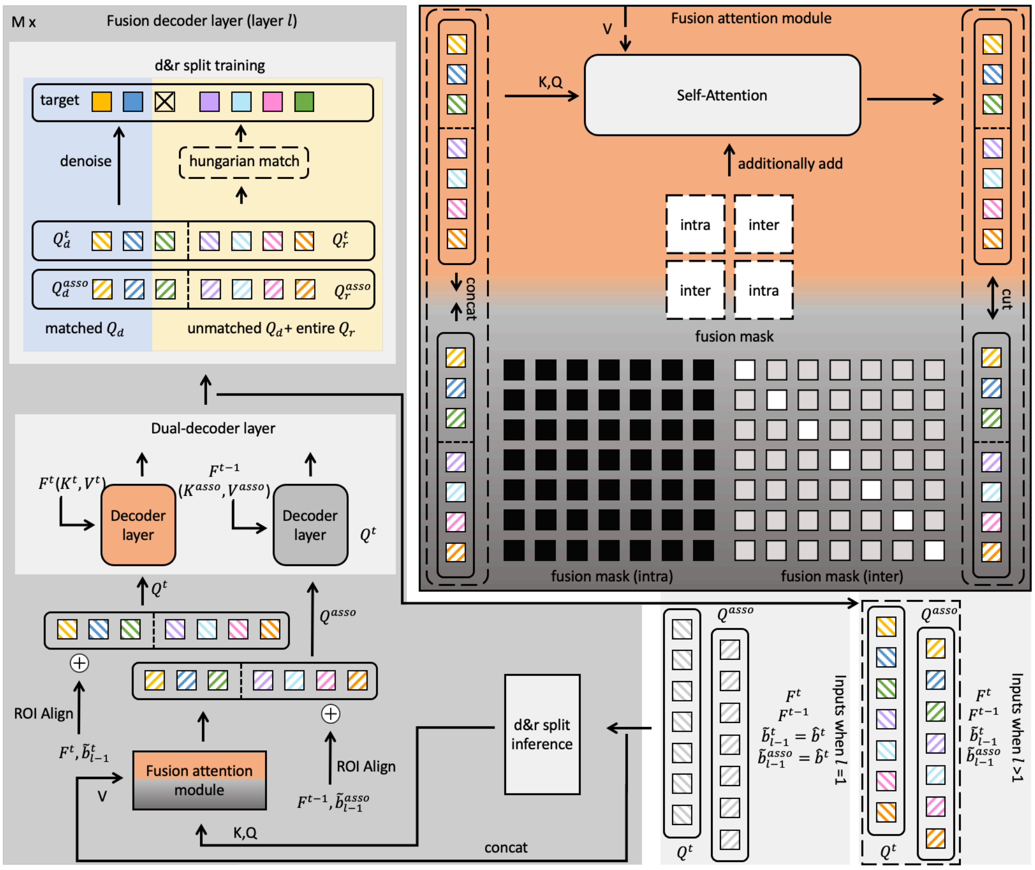


Fig. 3 – Roadmap of training fusion decoder layers. The fusion decoder layer starts by splitting queries into denoising and rematching groups (indicated by a dotted line inside subsequent queries). Fusion between query pairs is performed afterward. In detail, the fusion mask blocks intraframe exchange while selectively allowing communication between frames (white means no mask, gray for partial mask). We use ROIAlign to encode features guided by $\tilde b_{l-1}$ of layer $l-1$ (or $\hat b^t$) before decoding. The aligned features are added to query pairs as extra semantic information. Additionally, we train categorized query pairs in a targeted manner.

$t-1$ (through motions $\tilde a^t$ from Q^{asso}), that is

$$\tilde b_l^t = \Delta \tilde b_l^t + (\tilde b_{l-1}^t \text{ if } l>1 \text{ else } \hat b^t) \qquad (1)\text{a}$$

$$\tilde b_l^{asso} = \tilde a_l^t + (\tilde b_{l-1}^t \text{ if } l>1 \text{ else } \hat b^t) \qquad (1)\text{b}$$

where l ($1 \leq l \leq M$) is the layer index of the fusion decoder. In general, $\tilde b^t$ and $\tilde b^{asso}$, connected by the motions $\tilde a^t$ across frames, separately propagate through the dual-decoder structure of the fusion decoder.

Fusion attention is the self-attention mechanism with an additionally added fusion mask $Mask_{fusion}$. Depicted as the block with a color ramp from orange to gray in Fig. 3, fusion begins by concatenating the embeddings from query pairs (i.e., $Q^t, Q^{asso} \in \mathbb{R}^{N\times d} \overset{Concat}{\longrightarrow} Q^{fusion} \in \mathbb{R}^{2N\times d}$, d is the feature dimension). Self-attention is then performed on Q^{fusion} constrained by $Mask_{fusion} \in \mathbb{R}^{2N\times 2N}$ to focus on the exchange of cross-frame information. Thus $Mask_{fusion} = [m_{i,j}]_{2N\times 2N}$ satisfies

$$m_{i,j} = \begin{cases} 0, \text{ if } (i-\dfrac{2N-1)}{2}) \times (j-\dfrac{2N-1)}{2}) < 0 \\ \quad \text{and } i\%N = j\%N; \\ -\infty, \text{ if } (i-\dfrac{2N-1)}{2}) \times (j-\dfrac{2N-1)}{2}) > 0; \\ \beta, \text{ otherwise.} \end{cases} \qquad (2)$$

β is a hyperparameter introduced in the following.

Detailedly, sub-masks of $Mask_{fusion}$ can be categorized into two groups, namely $Mask_{intra} \in \mathbb{R}^{N\times N}$ serving as the mask of intra-frame (top-left and bottom-right of $Mask_{fusion}$), along with $Mask_{inter}$, in a similar way. $\left(\tilde b_l^t, \tilde b_l^{asso}\right)$, following Equation *(1)*a and *(1)*b, are moved from $\tilde b_{l-1}^t$ through query pairs, which require each pair to pinpoint a specific object. It is for this reason that elements along the main diagonal of $Mask_{inter}$ are emphasized more than others, offering more room for each pair to determine its target (β in Equation *(2)* shows this attention difference, -10 is our default setting, for more to refer to the discussion on Table 2). In a nutshell, a fusion mask is designed to improve the match between query pairs while reserving space for retrieving *extra* information.

4.4 Decoder initialization

Query pairs of the fusion decoder greatly contribute to the object predictions. Hence, it is straightforward to consider integrating the original predictions $\hat{o}^t$ into their initialization.

Reference locations. TransFiner fills initial reference locations of two recent frames $init\_ref^t$ and $init\_ref^{asso}$ with $\hat{b}^t$ ($init\_ref$ is the same as $\tilde{b}_{l=0}$).

Query pairs. Some [28, 29] inject the query embeddings with encoded features from the regions of interest. As shown in Fig. 3, we similarly ROIAlign [39] the encoded features within reference locations $\tilde{b}_{l-1}$ under layer l, resulting in $2N$ aligned feature maps. Afterward, extracting and combining the features from the sampling points of each feature map yields $2N$ distinct feature embeddings, which are then added to the corresponding query pairs.

4.5 Query denoising and query rematching

Prediction is often categorized as good or bad based on its accordance with the supposed ground truth. The former usually takes less effort than the latter under refinement. In other words, a query initialized from the former usually has a closely related target, which may suffer from the instability of the Hungarian matching (i.e., target shift as the disturbance introduced in refinement, a similar question discussed in [38]). Hence, we introduce *denoising and rematching split* (d&r split for short), including inference and training steps shown in Fig. 3.

Inference. We distinguish a query for denoising or rematching by comparing its objectness score $\tilde{c}_i^t$ from accordingly initialized original prediction $\hat{o}_i^t$ with thr_{out} (e.g., 0.4). Afterward, we label queries by assigning the denoising embedding q_d to those associated with decent predictions, i.e., (Q_d^t, Q_d^{asso}), and the rematching embedding q_r to those related to poor predictions, i.e., (Q_r^t, Q_r^{asso}). There is a reminder that decoder performs identification over denoising and rematching at the first layer.

Training. After conducting the *inference* step amid training, we further pre-determine the matched target-prediction pairs among Q_d following

$$\text{Target}\left[\left(Q_d^t\right)_m\right] = \begin{cases} b_{\sigma(m)}^t, & \text{if iou}\left(b_{\sigma(m)}^t, \hat{b}_m^t\right) > thr_{match}; \\ \emptyset, & \text{otherwise}. \end{cases} \tag{3}$$

σ is the optimal assignment from the Hungarian match between decent predictions and targets. thr_{match} is the threshold filtering denoising queries whose initialized locations intolerably deviate from targets even with high objectness scores.

In the subsequent layer-by-layer refinement, Hungarian matching is performed outside the matched Q_d, leaving unmatched Q_d and the entire Q_r to search for the best-associated targets in each layer.

5. EXPERIMENTS

5.1 Datasets and evaluation metrics

MOT. In multiple object tracking, MOT benchmarks are generally used to evaluate the performance of trackers. We conduct experiments on MOT16 and MOT17 [40], both including 7 training sequences and 7 test sequences. The final results reported in Section 5.3 are obtained through training on the entire training set (additionally with the validation set of CrowdHuman [41]) and evaluating on the test set officially under the private detection protocol. For the ablation study, we, following Centertrack [7], split the official training set into two halves. The first half is used for training, while the second is for validation.

CrowdHuman [41] is a detection dataset filled with collections of images of the crowd, containing 15000 training images and 4370 validation images, which is widely used as a pre-training dataset for the MOT trackers.

Metrics. We demonstrate our results using the popular MOT evaluation metrics set CLEAR [42], including Multiple-Object Tracking Accuracy (MOTA), Identity Switch (IDS), False Positive (FP), and False Negative (FN). Additionally, we report the Identification F1 score (IDF1) [43] and the Higher Order Tracking Accuracy (HOTA) [44], which is the geometric mean of two sub-metrics comprising Association Accuracy score (AssA) and Detection Accuracy score (DetA).

5.2 Implementation details

Model. We pick CenterTrack [7] as the original tracker in our experiments. For TransFiner, the backbone network is ResNet-50 [45] for its balance of speed and accuracy, coupled with the *twin* structure from a six-layer encoder and decoder of deformable DETR [25]. The number of query embeddings is set to 300. In MOT datasets, the bounding boxes fully cover the targets, which means parts of the objects have their box centers outside the images, making it suboptimal to directly predict the objects' centers, widths, and heights. Hence, following the solution in [7], we also formulate the box representation set $b = \left(x, y, ad_{tp}, ad_{lf}, ad_{bt}, ad_{rt}\right)$. The last four respectively show the non-negative distance from the center to the top, left, bottom, and right edge of the bounding box. This allows more precise estimations even when objects are heavily cropped.

Decoder initialization. Formalized in Section 3, the TransFiner's decoder outputs are $\{\tilde{y}_i^t\}_{i=0}^{N-1}$, while its initialization input is $\{\hat{o}_i^t\}_{i=0}^{K-1}$. Obviously, the mismatch be-

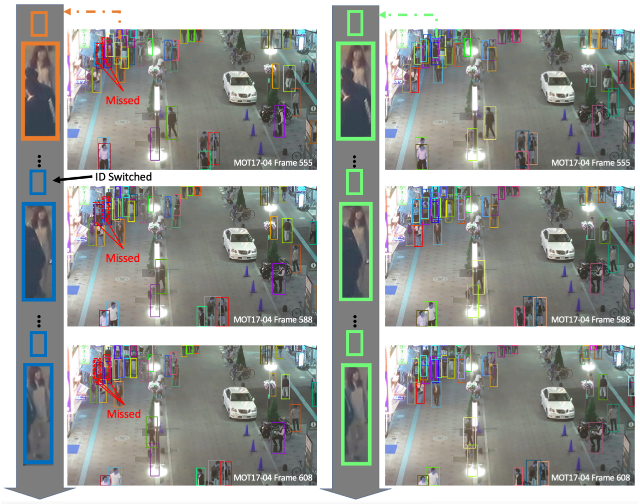


Fig. 4 – Case study. Examples of CenterTrack (left column) refined by TransFiner (right column). Tracks are marked by color. The big black arrow depicts a tracklet of identical objects across frames. Under CenterTrack, the pedestrian in the orange box at Frame 555 now appears in a blue box since Frame 588. TransFiner, on the other hand, handles the identity switch originally introduced by this target via continuous tracks with green boxes. Moreover, additional annotations in red denote objects that CenterTrack ignores, while TransFiner fixes them.

tween K and N raises the question of how to perform a one-to-one assignment at the beginning of the object querys' initialization. Here we provide a feasible solution. Following the categorizing standard in Section 4.5, we address this first by separating $\hat{o}^t$ into set $(\hat{o}^t_d, \hat{o}^t_r)$, and there are respectively K_d and K_r elements in $\hat{o}^t_d$ and $\hat{o}^t_r$. Next, we obtain the sequence by linking $\hat{o}^t_d$ with $\lceil \frac{N-K_d}{K_r} \rceil$ times repeated $\hat{o}^t_r$. The sequence is then clipped to that of length N.

Training settings. Images are resized to 672×1184 as inputs. Training loss consists of generally two part: detection and association losses. Specifically, we adopt Hungarian loss [17] as loss for detection boxes $\tilde{b}^t$ in Equation *(1)*a, which has three sub-losses with coefficients λ_{cls}(=2), λ_{L_1}(=5), and λ_{iou}(=2), respectively. For association loss, we calculate Hungarian loss for association boxes $\tilde{b}^{asso}$ (from Equation *(1)*b) under the same setting as detection loss with coefficients being divided by 5. Due to GPU memory limitation, the batch size is set to 8, with gradient accumulation amid every two iterations and simulating a 16-batch setup. Overall, we use 2 NVIDIA RTX 3090 GPUs with batch size 8, optimizer AdamW [46], and the initial learning rate $2e-4$. TransFiner is first pre-trained on the CrowdHuman training set [41] for 95 epochs, with the learning rate dropping to $2e-5$ after 50 epochs. We then train the TransFiner on both MOT [40] and the CrowdHuman validation set [41] for another 130 epochs with the learning rate decreasing by 10 at the 100-th epoch.

5.3 Benchmark results

As a post-refinement model, we first discuss the improvement made by applying TransFiner after the original tracker (CenterTrack[7] in our experiments). Then we compare the refined tracker with recent MOT trackers on MOT16 and MOT17 [40].

Improvement under TransFiner. CenterTrack officially reports results on the MOT17 benchmark, where we have a detailed look. As shown in Table 1, refinement by TransFiner shows a comprehensive improvement (+2.1% IDF1 and +3.7% MOTA). This benefits from distinct focuses of query pairs over targets, contributing to apparent refinements on FN (decreasing by 31667), while IDsw virtually stays intact (from 3039 to 3056). An example is depicted in Fig. 4.

MOT16 & MOT17. Table 1 demonstrates the results reported on MOT16 and MOT17 test datasets. In MOT16, we chiefly compare enhanced CenterTrack with two other transformer-based trackers, namely PatchTrack [13] and MeMOT [15], which respectively obtain state-of-the-art performance in detection and association. Improved CenterTrack achieves comparative detection performance (73.0% MOTA and 58.6% DetA), with 0.3% less MOTA and 1.0% fewer DetA than PatchTracks. Alternatively, we better associate objects than PatchTrack, relying on the informative motions from query pairs, but still underperform MeMOT on IDF1 (67.6 vs. 69.7) and AssA (52.2 vs. 55.7), possibly due to our local linkage (performing on two continuous frames). In MOT17, CenterTrack powered by TransFiner embraces second-to-best tracking ability, surpassing most transformer-based approaches like TransTrack [11], TransCenter [16], Trackformer [12] and PatchTrack [13]. In addition, CenterTrack with TransFiner detects well (57.5% DetA) but is inferior to several SOTA transformer-based trackers. It is probably because query pairs restrict the prediction of objects on the current frame if they are out of scope on the previous frame.

5.4 Ablation study

We test our design choices with the same model combination (CenterTrack and TransFiner) in Section 5.3 on the train-val split of the MOT17 training dataset.

Decoder structure. The fusion attention module and dual-decoder are layered repeatedly to form the *fusion decoder*. Additionally, we receive the *single* version by throwing fusion attention and the decoder focusing on Q^{asso}. Straightforwardly, refining with TransFiner built on *single* merely redetects the objects of the current frame with specific decoder initialization. The results shown in the blue block of Table 2 suggest that the information fusion, as well as motion estimations, play a crucial role in MOT refinement. We observe that the fusion decoder el-

Table 1 – Evaluation results on MOT challenge datasets (private detection). The **TF** stands for TransFiner. The best result in each column is marked in red and in blue for the second best.

Method	**IDF1** ↑	**MOTA** ↑	**HOTA** ↑	**DetA** ↑	**AssA** ↑	**IDsw** ↓	**FP** ↓	**FN** ↓
				MOT16				
TubeTK [47]	62.2	66.9	50.8	55.0	47.3	1236	11544	47502
Chain-Tracker [48]	57.2	67.6	48.8	55.0	43.7	1897	8934	48350
TraDeS [9]	64.7	70.1	53.2	56.2	50.9	1144	8091	45210
QuasiDense[5]	67.1	69.8	54.5	56.6	52.8	1097	9861	44050
MeMOT [15]	69.7	72.6	57.4	-	55.7	845	14595	34595
PatchTrack [13]	65.8	73.3	54.2	59.6	49.7	1179	10660	36824
CenterTrack+**TF** (ours)	67.6	73.0	55.1	58.6	52.2	976	10463	37723
				MOT17				
TraDeS [9]	63.9	69.1	52.7	55.2	50.8	3555	20892	150060
QuasiDense[5]	66.3	68.7	53.9	55.6	52.7	3378	26589	146643
TransTrack [11]	63.9	74.5	53.9	60.5	48.3	3663	28323	112137
TransCenter [16]	62.2	73.2	54.5	60.1	49.7	4614	23112	123738
TubeTK [47]	58.6	63.0	48.0	51.4	45.1	4137	27060	177483
Chain-Tracker [48]	57.4	66.6	49.0	53.6	45.2	5529	22284	160491
TrackFormer [12]	63.9	65.0	-	-	-	3258	70443	123552
MeMOT [15]	69.0	72.5	56.9	-	55.2	2724	37221	115248
PatchTrack [13]	65.2	73.6	53.9	59.4	49.3	3795	23976	121230
CenterTrack [7]	64.7	67.8	52.2	53.8	51.0	3039	18498	160332
CenterTrack+**TF** (ours)	66.8	71.5	54.5	57.5	52.0	3056	29283	128665

Table 2 – Ablation studies on the MOT17 validation set. * means our default settings. An experimental attempt *back refer* in Section 5.4 is indicated by ⋆. **Baseline** is the tracking performance of CenterTrack [7] under the same experiment settings. We explore design options on **decoder structure** (single-decoder structure fails), **refinement tactic** (d&r split boosts refinement, and back refer drags it down), fusion mask **hyperparameter** β ($\beta = -10$ balances detection and association), and **motion** (box motion is critical in association). Color blocks with the best results are bolded.

Ablation	Choice	MOTA	IDF1	HOTA	AssA
Decoder structure	Single	62.3	59.0	48.6	44.8
	*Fusion	**70.1**	**74.0**	**60.6**	**63.0**
Refinement tactic	w/ ⋆back refer	69.8	71.5	59.2	60.0
	w/o d&r split	69.0	72.6	59.8	61.6
	w/o d&r embeddings	68.9	72.8	59.3	60.5
	*Vanilla	**70.1**	**74.0**	**60.6**	**63.0**
Hyperparameter β	0	69.5	71.8	58.8	59.5
	-5	**70.5**	73.0	60.0	61.1
	*-10	70.1	**74.0**	**60.6**	**63.0**
	-∞	69.5	73.7	60.1	62.0
Motion	*Center+Box	**70.1**	**74.0**	**60.6**	**63.0**
	Center	69.0	67.5	56.6	55.4
	✗	67.9	65.7	55.5	53.7
-	Baseline	66.2	69.4	-	-

evates association significantly (15.0% improvements on IDF1 and ∼20.0% increases on AssA compared with single decoders), indicating motions from query pairs of the fusion decoder are robust in linking objects across frames.

Refinement tactic. We begin by exploring the initialization with back referring. Next, we discuss the ablations on the d&r split of queries.

To further leverage $\hat{o}^t$ during initialization of the decoder, we attempt to extend the locations assignment in Section 4.4 by back referring $init\_ref^{asso}$ through $\hat{a}^t$ instead of putting $init\_ref^{asso}$ identical to $init\_ref^t$. Specifically, back referring derives the reference locations of the previous frame through $\hat{b}^t, \hat{a}^t$. Here we consider $\hat{a}^t$ as backward motions. In this case, back referring is achieved via $init\_ref^{asso} = init\_ref^t + \hat{a}^t$. The effectiveness of *back refer* can be seen in the gray block of Table 2, which shows overall performance degradation. We conclude two reasons for this: (1) Motions $\hat{a}^t$ from objects whose objectness scores are uncertain usually have a significant bias, deteriorating refinement by acting as *unhealthy noises*; (2) Query pairs and the fusion mask allow for gradual adjustment of position pairs, discouraging excessive locations assignment beforehand.

For ablation studies on the d&r split, we drop it from the vanilla. The second row of the gray block in Table 2 shows that this lowers the model performance for, probably, pushing TransFiner to treat original predictions equally, without special attention to tough ones. In addition, we trial d&r split lacking embedding labeling denoising and rematching queries (i.e., without q_d and q_r). This, however, further degrades TransFiner. Part of the reason is that little information hints at the queries with different refinement purposes when functioning.

Hyperparameter β. The green rows of Table 2 show optimization performances under various choices of β. $\beta = 0$ leads to an obvious decline in association (reducing IDF1 by 2.2% and 3.5% for AssA from the default setting). In contrast, detection and association suffer slightly when $\beta = -\infty$, dropping from the vanilla by 0.6% MOTA and 0.3% IDF1. Moreover, we observe mild overall improvement when placing β to a moderate value (e.g., -10). An intuitive illustration is that a suitable value of β properly weighs interactions between queries outside and inside their *in-couples*, where queries are dynamically and controllably fitted.

Motion. Transfiner evaluates motions in the form of centers and boxes of objects from the present to the last frame. According to the yellow chunk of Table 2, we observe a considerable gap with and without box motions in the association (74.0% IDF1 vs. 67.5% IDF1 and 63.0% AssA vs. 55.4% AssA), considering box motions are more distinctive in crowded scenarios.

5.5 Limitations and future work

TransFiner performs on local tracking (within adjacent frames), limiting refinement when the targets are under long-term occlusions. To address these, the design of a prediction error buffer (e.g., contains the TransFiner's predictions crossing the border of d&r split), along with a stronger query interaction mechanism, may help improve this defect. Still, TransFiner leverages initial tracking non-trivially, and how to better *semantically* joint inputs (e.g., frames) and outputs (i.e., original predictions) space requires exploration.

6. CONCLUSION

TransFiner is a generic MOT post-refinement framework. We consider predicted locations and objectness scores from the original tracker for refinement. TransFiner fully exploits initial predictions, locations guide the extraction of image features for query pairs and scores are used to group pairs for targeted rectification. Labeled query pairs, highly representing original predictions, combine input and output space for refinement via the fusion decoder, which achieves impressive refinement outcomes on MOT16 and MOT17 benchmarks.

REFERENCES

[1] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. "Simple online and realtime tracking". In: *2016 IEEE international conference on image processing (ICIP)*. IEEE. 2016, pp. 3464–3468.

[2] Caleb Vatral, Gautam Biswas, and Benjamin Goldberg. "Online Multi-Object Motion Tracking by Fusion of Head and Body Detections". In: (2021). DOI: 10.13140/RG.2.2.34852.60800.

[3] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. "Learning by tracking: Siamese CNN for robust target association". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 33–40.

[4] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. "Recurrent autoregressive networks for online multi-object tracking". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 466–475.

[5] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. "Quasi-dense similarity learning for multiple object tracking". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 164–173.

[6] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. "Detect to track and track to detect". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3038–3046.

[7] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. "Tracking objects as points". In: *European Conference on Computer Vision*. Springer. 2020, pp. 474–490.

[8] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. "SiamMOT: Siamese Multi-Object Tracking". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12372–12382.

[9] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. "Track to Detect and Segment: An Online Multi-Object Tracker". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12352–12361.

[10] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. "Tracking without bells and whistles". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 941–951.

[11] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. "Transtrack: Multiple object tracking with transformer". In: *arXiv preprint arXiv:2012.15460* (2020).

[12] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. "Trackformer: Multi-object tracking with transformers". In: *arXiv preprint arXiv:2101.02702* (2021).

[13] Xiaotong Chen, Seyed Mehdi Iranmanesh, and Kuo-Chin Lien. "PatchTrack: Multiple Object Tracking Using Frame Patches". In: *arXiv preprint arXiv:2201.00080* (2022).

[14] Tianyu Zhu, Markus Hiller, Mahsa Ehsanpour, Rongkai Ma, Tom Drummond, and Hamid Rezatofighi. "Looking Beyond Two Frames: End-to-End Multi-Object Tracking Using Spatial and Temporal Transformers". In: *arXiv preprint arXiv:2103.14829* (2021).

[15] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. "MeMOT: Multi-Object Tracking with Memory". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8090–8100.

[16] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. "Transcenter: Transformers with dense queries for multiple-object tracking". In: *arXiv preprint arXiv:2103.15145* (2021).

[17] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end object detection with transformers". In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[18] Bisheng Wang, Christian Fruhwirth-Reisinger, Horst Possegger, Horst Bischof, Guo Cao, and Embedded Machine Learning. "DRT: Detection Refinement for Multiple Object Tracking". In: *32nd British Machine Vision Conference: BMVC 2021*. The British Machine Vision Association. 2021.

[19] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. "Retinatrack: Online single stage joint detection and tracking". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 14668–14678.

[20] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric". In: *2017 IEEE international conference on image processing (ICIP)*. IEEE. 2017, pp. 3645–3649.

[21] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. "Multi-object tracking with quadruplet convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5620–5629.

[22] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. "Fairmot: On the fairness of detection and re-identification in multiple object tracking". In: *arXiv preprint arXiv:2004.01888* (2020).

[23] Yunhao Du, Yang Song, Bo Yang, and Yanyun Zhao. "StrongSORT: Make DeepSORT Great Again". In: *arXiv preprint arXiv:2202.13514* (2022).

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[25] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. "Deformable DETR: Deformable Transformers for End-to-End Object Detection". In: *International Conference on Learning Representations*. 2020.

[26] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. "Fast convergence of detr with spatially modulated co-attention". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3621–3630.

[27] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. "Conditional detr for fast training convergence". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3651–3660.

[28] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. "Efficient detr: improving end-to-end object detector with dense prior". In: *arXiv preprint arXiv:2104.01318* (2021).

[29] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. "Accelerating DETR Convergence via Semantic-Aligned Matching". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 949–958.

[30] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. "Human pose estimation with iterative error feedback". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4733–4742.

[31] Hao Tang, Xingwei Liu, Shanlin Sun, Xiangyi Yan, and Xiaohui Xie. "Recurrent mask refinement for few-shot medical image segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3918–3928.

[32] Spyros Gidaris and Nikos Komodakis. "Detect, replace, refine: Deep structured prediction for pixel wise labeling". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5248–5257.

[33] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. "Cascaded pyramid network for multi-person pose estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7103–7112.

[34] Mihai Fieraru, Anna Khoreva, Leonid Pishchulin, and Bernt Schiele. "Learning to refine human pose estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 205–214.

[35] Chufeng Tang, Hang Chen, Xiao Li, Jianmin Li, Zhaoxiang Zhang, and Xiaolin Hu. "Look Closer to Segment Better: Boundary Patch Refinement for Instance Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13926–13935.

[36] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. "Posefix: Model-agnostic general human pose refinement network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7773–7781.

[37] Fan Yang, Xin Chang, Sakriani Sakti, Yang Wu, and Satoshi Nakamura. "ReMOT: A model-agnostic refinement for multiple object tracking". In: *Image and Vision Computing* 106 (2021), p. 104091.

[38] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. "Dn-detr: Accelerate detr training by introducing query denoising". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13619–13627.

[39] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[40] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. “MOT16: A benchmark for multi-object tracking”. In: *arXiv preprint arXiv:1603.00831* (2016).

[41] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. “Crowdhuman: A benchmark for detecting human in a crowd”. In: *arXiv preprint arXiv:1805.00123* (2018).

[42] Keni Bernardin and Rainer Stiefelhagen. “Evaluating multiple object tracking performance: the clear mot metrics”. In: *EURASIP Journal on Image and Video Processing* 2008 (2008), pp. 1–10.

[43] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. “Performance measures and a data set for multi-target, multi-camera tracking”. In: *European conference on computer vision*. Springer. 2016, pp. 17–35.

[44] 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”. In: *International journal of computer vision* 129.2 (2021), pp. 548–578.

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[46] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*. 2018.

[47] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. “Tubetk: Adopting tubes to track multi-object in a one-step training model”. In: *CVPR*. 2020.

[48] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. “Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking”. In: *ECCV*. 2020.

AUTHOR

Bin Sun received a BS degree in communication engineering from the China University of Geoscience, Wuhan, China, in 2022. He is currently working toward a master’s degree at the Huazhong University of Science and Technology (HUST), Wuhan, China. His current research interests include artificial intelligence, password security, and privacy.