# iKUN: Speak to Trackers without Retraining

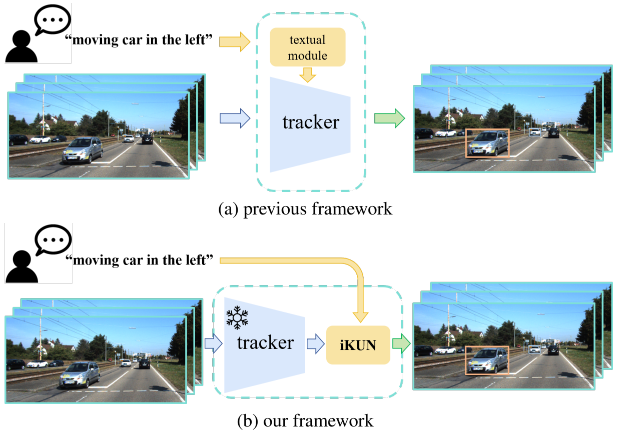
Yunhao Du[1], Cheng Lei[1], Zhicheng Zhao[1,2,3*], Fei Su[1,2,3]
[1]The school of Artificial Intelligence, Beijing University of Posts and Telecommunications
[2]Beijing Key Laboratory of Network System and Network Culture, China
[3]Key Laboratory of Interactive Technology and Experience System Ministry
of Culture and Tourism, Beijing, China
{dyh_bupt,mr.leicheng,zhaozc,sufei}@bupt.edu.cn

## Abstract

*Referring multi-object tracking (RMOT) aims to track multiple objects based on input textual descriptions. Previous works realize it by simply integrating an extra textual module into the multi-object tracker. However, they typically need to retrain the entire framework and have difficulties in optimization. In this work, we propose an **i**nsertable **K**nowledge **U**nification **N**etwork, termed **iKUN**, to enable communication with off-the-shelf trackers in a plug-and-play manner. Concretely, a knowledge unification module (KUM) is designed to adaptively extract visual features based on textual guidance. Meanwhile, to improve the localization accuracy, we present a neural version of Kalman filter (NKF) to dynamically adjust process noise and observation noise based on the current motion status. Moreover, to address the problem of open-set long-tail distribution of textual descriptions, a test-time similarity calibration method is proposed to refine the confidence score with pseudo frequency. Extensive experiments on Refer-KITTI dataset verify the effectiveness of our framework. Finally, to speed up the development of RMOT, we also contribute a more challenging dataset, Refer-Dance, by extending public DanceTrack dataset with motion and dressing descriptions. The codes and dataset are available at* https://github.com/dyhBUPT/iKUN.

## 1. Introduction

Traditional multi-object tracking (MOT) task aims to track all specific classes of objects frame-by-frame, which plays an essential role in video understanding. Although significant developments have been achieved, it suffers from poor flexibility and generalization. To address this problem, referring multi-object tracking (RMOT) task is recently proposed [37], whose core idea is to guide the multi-object tracking with a language description. For example, if we input "moving car in the left" as the query, the tracker will predict all trajectories corresponding to the description. However, as the cost of high flexibility, the model is required to perform detection, association and referring simultaneously. Therefore, balancing the optimization between subtasks becomes a critical issue.



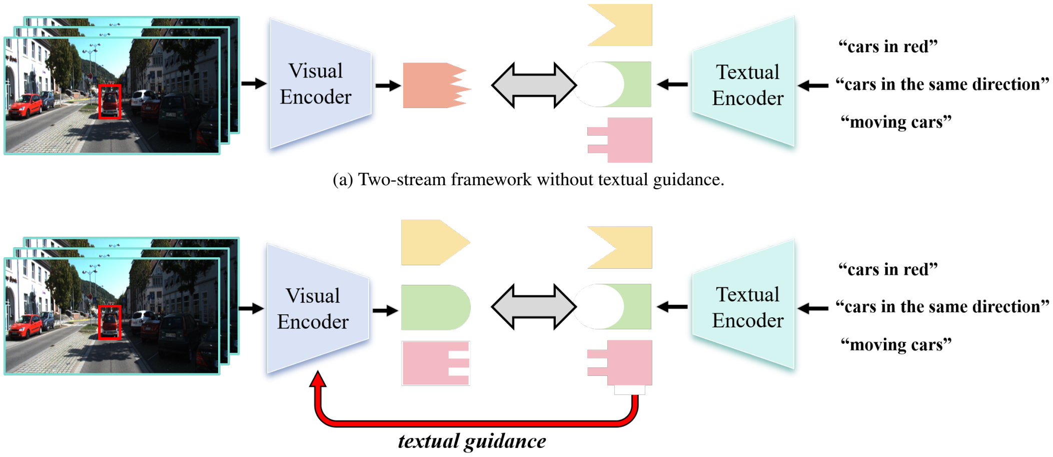
(a) previous framework

(b) our framework

Figure 1. **Comparison among previous RMOT frameworks and ours.** (a) Previous methods incorporate the referring module into the multi-object tracker, which need to retrain the overall framework. (b) Instead, our designed model iKUN can be directly plugged after an off-the-shelf tracker, in which the tracker is frozen while training.

To accomplish this task, existing methods, e.g., TransRMOT [37], simply integrate a textual module into existing trackers, as shown in Fig.1(a). However, this framework has several intrinsic drawbacks: **i)** Task competition. The optimization competition between detection and association has been revealed by some MOT methods [24, 48]. In RMOT, the added referring subtask will further exacerbate this problem. **ii)** Engineering cost. Whenever we want to replace the baseline tracker, we need to rewrite the codes and

[*]Corresponding author



(a) Two-stream framework without textual guidance.

(b) Two-stream framework with textual guidance.

Figure 2. **The motivation of KUM.** Given a tracklet and a set of descriptions, (a) without the guidance from textual stream, the visual encoder is asked to output a single feature to match multiple textual features; (b) with textual guidance, the visual encoder can predict adaptive features for each description.

retrain the entire framework. **iii)** Training cost. Training all subtasks jointly results in high computational costs.

Essentially, the tight bundling of tracking and referring subtasks is the main reason of these limitations. This raises a natural question: *"Is it possible to decouple these two subtasks?"* In this work, we present a *"tracking-to-referring"* framework with an insertable module **iKUN**, which first tracks all candidates, and then recognizes the queried objects based on language descriptions. As shown in Fig.1(b), the tracker is frozen while training, and the optimization procedure can focus on the referring subtask.

Therefore, the core problem lies in the design of an insertable referring module. An intuitive choice is the CLIP-style [30] module, which is pretrained on over 400 million image-text pairs for contrastive learning. Its main advantage is the excellent alignment of visual concepts and textual descriptions. For simplicity, the visual and textual streams of CLIP are independent. This means that for a given visual input, CLIP will extract a fixed visual feature regardless of the textual input. However, in the RMOT task, one trajectory often corresponds to multiple descriptions, including color, location, status, etc. It's hard to match a single feature with multiple various features. Motivated by this observation, we design a knowledge unification module (KUM) to adaptively extract visual features with the textual guidance. The illustration is shown in Fig.2. Moreover, to mitigate the effects of long-tail distribution of descriptions, we propose a test-time similarity calibration method to refine the referring results. The main idea is to estimate pseudo frequencies for descriptions in the open test set, and use them to revise the referring score.

For the tracking subtask, Kalman filter [21] is widely used for motion modelling. The process noise and observation noise are two vital variables, which affect the accuracy of prediction and update steps. However, as a hand-crafted module, these two variables are determined by preset parameters, and are difficult to adapt to changes of motion status. We circumvent this problem by designing a neural version of Kalman filter, dubbed **NKF**, which dynamically estimates process and observation noises.

We conduct extensive experiments on the recently released Refer-KITTI [37] dataset, and our methods show substantial superiority to existing solutions. Specifically, our solutions surpass previous SOTA method TransRMOT by **10.78% HOTA**, **3.17% MOTA** and **7.65% IDF1**. Experiments for the traditional MOT task are also conducted on KITTI [14] and DanceTrack [33], and the proposed NKF achieves noticeable improvement over baseline trackers [36, 47]. To further verify the effectiveness of our methods, we contribute a more challenging RMOT dataset, **Refer-Dance**, by extending DanceTrack with language descriptions. Our methods report a significant improvement than TransRMOT, i.e., 29.06% vs. 9.58% HOTA.

## 2. Related Work

### 2.1. Multi-object Tracking

Predominant approaches in MOT mainly follow the tracking-by-detection paradigm [4, 35, 42, 43, 46]. They typically first predict the bounding boxes of objects and extract their features. Then an association step is used to match these instances across different frames. SORT [3] applies Kalman filter for motion modelling and associates instances based on the intersection-over-union (IoU) of bounding boxes. DeepSORT [36] extends it by adding

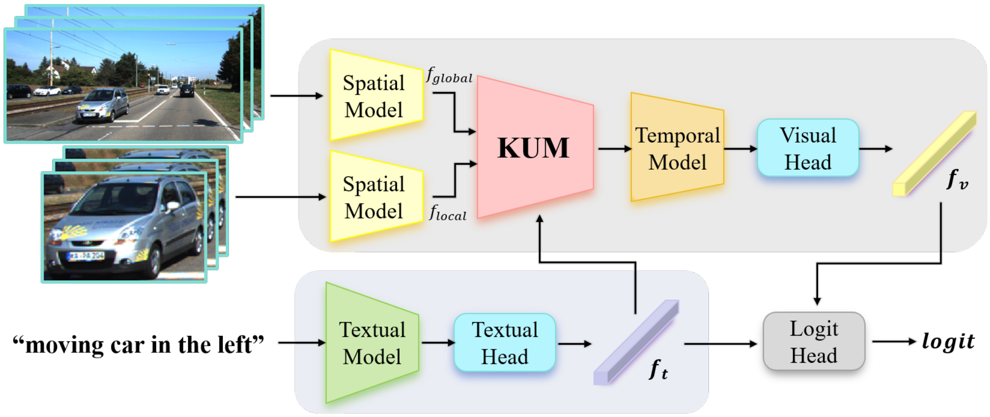


Figure 3. **The overall framework of iKUN.** The visual stream first embeds the local object feature $f_{local}$ and global scene feature $f_{global}$, and then aggregates them using the knowledge unification module (KUM). A temporal model and a visual head are followed to generate the final visual feature $f_v$. Meanwhile, the textual stream encodes the textual feature $f_t$. Finally, a logit head is utilized to predict the similarity score between $f_v$ and $f_t$

an extra embedding network to extract appearance features of instances. OMC [23] introduces an extra re-check network to restore missed targets. MAT [15] focuses on high-performance motion-based prediction with a plug-and-play solution. ByteTrack [47] improves the association algorithm with the idea of "associating every detection box". Some recent SOTA methods, e.g., StrongSORT [13], OC-SORT [6] and BoT-SORT [1], further design better association and post-processing solutions.

Among these methods, the hand-crafted algorithm, Kalman filter, is widely used for motion modelling. However, it depends on elaborate parameter setting. To crack this nut, we introduce neural networks into Kalman filter, which can dynamically update the noise variables and adapt to more challenging scenarios.

## 2.2. Referring Tracking

Referring single-object tracking (or segmentation) has been studied for several years. Benefiting from the flexibility of Transformer [34], recent SOTA solutions mainly follow the joint tracking paradigm. TransVLT [49] designs proxy tokens to bridge between the visual and textual modalities, which are followed by a Transformer-based cross-modal fusion module to estimate queried trajectories. JointNLT [52] directly inputs language, template and text image embeddings into the Transformer encoder for relation modelling. MMTrack [51] serializes language descriptions and bounding boxes into discrete tokens, and casts referring tracking as a token generation task. OVLM [45] proposes a one-stream network with memory token selection mechanism, which utilizes textual information to eliminate redundant tokens. MTTR [5] applies a DETR-like [7] multi-modal module to decode instance-level features into a set of multi-modal sequences. ReferFormer [39] inputs a set of object queries conditioned on language descriptions into Transformer to estimate the referred object. OnlineRefer [38] employs an elaborate query propagation mechanism to realize online referring tracking.

Referring multi-object tracking is an emerging task, which can query an arbitrary number of instances [37]. Previous SOTA methods implement it by simply integrating the textual module to the tracking or detection model. Instead, we propose an insertable module to be plugged after any off-the-shelf trackers, which achieves much more flexibility and effectiveness.

# 3. Method

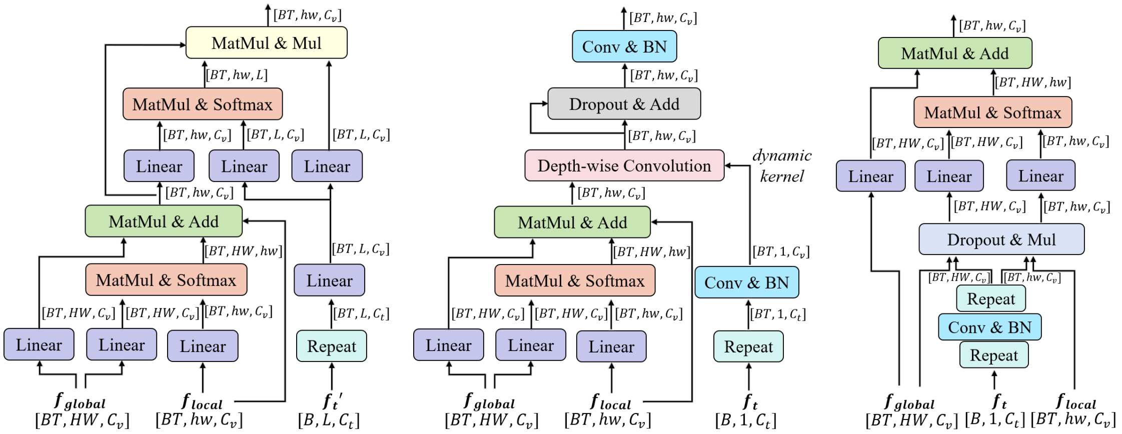
## 3.1. Method Overview

The input of RMOT consists of a sequence $\mathcal{I} = \{I_t\}_{k=1}^{K}$ with $K$ frames and a referring description $\mathcal{E} = \{e_l\}_{l=1}^{L}$ with $L$ words. Given an off-the-shelf tracker $\mathcal{F}_{trk}(\cdot)$, all candidate trajectories are predicted as $\mathcal{T} = \mathcal{F}_{trk}(\mathcal{I})$, where $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_N\}$ includes $N$ instance trajectories. Then, the referring module (i.e., iKUN) $\mathcal{F}_{ref}(\cdot)$ is applied to score candidates $\mathcal{T}$ by description as $\mathcal{S} = \mathcal{F}_{ref}(\mathcal{I}, \mathcal{T}, \mathcal{E})$, which is further refined by the similarity calibration method. Finally, candidates $\mathcal{T}$ are filtered by refined scores $S'$ and the final outputs $\mathcal{T}'$ with $M \leq N$ trajectories are generated.

In the following sections, we first detail the design of $\mathcal{F}_{ref}$ in Sec.3.2. Then the similarity calibration method is introduced in Sec.3.3. Finally, we present the neural Kalman filter in Sec.3.4 to improve the tracking performance of tracker $\mathcal{F}_{trk}$.

## 3.2. Insertable Knowledge Unification Network

iKUN is designed as a two-stream framework following CLIP [30] as shown in Fig.3. In the textual stream, descrip-



(a) cascade attention

(b) cross correlation

(c) text-first modulation

Figure 4. **Three designs of knowledge unification module.** The feature maps are shown as the shape of their tensors with batch size $B$. For clarity, the final spatial global average pooling operation is omitted here.

tion $\mathcal{E}$ with $L$ words is first encoded into $f_t' \in \mathbb{R}^{L \times C_t}$ via a textual model $E_{txt}$, where $C_t$ is the dimension of textual feature. Then a textual head is utilized to squeeze $f_t'$ into $f_t \in \mathbb{R}^{C}$, where $C$ is the dimension of final feature.

For the visual stream, full images $\mathcal{I}$ and cropped images $\mathcal{I}'$ are encoded into the global feature $f_{global} \in \mathbb{R}^{T \times HW \times C_v}$ and local feature $f_{local} \in \mathbb{R}^{T \times hw \times C_v}$ with spatial model $E_{vis}$, respectively. Here, $T$ is temporal window size, $H, W, h, w$ represent spatial size of feature maps, and $C_v$ is the dimension of visual feature. Then the knowledge unification module (KUM) $E_{kum}$ is applied to aggregate them:

$$f_{uni} = E_{kum}(f_{global}, f_{local}) \in \mathbb{R}^{T \times C_v}. \tag{1}$$

The temporal model and visual head are followed for temporal modelling and channel transformation, resulting in visual feature $f_v \in \mathbb{R}^{C}$. Finally, a logit head $H(\cdot)$ is utilized to compute the similarity score as $s = H(f_t, f_v)$.

Though the two-stream framework is widely used in recent cross-modal retrieval methods [9, 18, 22, 27, 40, 41], we claim that it is not suitable for RMOT task. Specifically, given one trajectory, there may be multiple positive descriptions, e.g., "cars in red", "cars in the same direction" and "moving cars". However, the design of two independent streams is not sufficiently powerful for such "one-to-many correspondence" problem.

To alleviate the above issues, we reformulate the knowledge unification module in Eq.1 as:

$$f_{uni} = E_{kum}(f_{global}, f_{local}, f_t) \in \mathbb{R}^{T \times C_v}. \tag{2}$$

That is, textual features are integrated into the unification process as guidance, which modulate visual feature extracting. In the following we will briefly introduce the three designs of KUM, as shown in Fig.4. The details are given in supplementary Sec.A.

**Cascade attention.** The two visual features $f_{global}$ and $f_{local}$ are first aggregated via cross attention [34] with residual adding, where $f_{local}$ is query and $f_{global}$ is key / value. Then the resulting feature is fused with $f_t$ via another cross attention with residual multiplication.

**Cross correlation.** Similarly, two visual features are first aggregated. Then a description conditioned dynamic convolutional operation is designed. Specifically, dynamic kernels are estimated based on textual feature $f_t$, and then are used to perform cross correlation with the aggregated visual feature.

**Text-first Modulation.** The above two designs prioritize the visual aggregation. Instead, textual feature $f_t$ is introduced earlier here to modulate two visual features $f_{global}$ and $f_{local}$, which are later aggregated via cross attention and residual adding.

In summary, "cascade attention" and "cross correlation" utilize different mechanisms to model the cross-modal relation, while in "text-first modulation", textual feature is used to guide visual aggregation. All these designs show substantial superiority to the baseline method (Eq.1) in experiments, which identifies the effectiveness of our solutions.

### 3.3. Similarity Calibration

Learning under long-tail distribution has been widely studied in recent years [10–12, 17, 50]. The similar problem is also observed in RMOT task, that is, there is a huge difference in the number of positive instances of descriptions. However, it has the following two significant differences from the traditional long-tail distribution problem: **i) Non-uniform test set.** Most previous works assume the long-tail distribution on training set and uniform distribution on test

Table 1. **Comparison with state-of-the-art RMOT methods on Refer-KITTI.** †: the results are from official code base [*]. ⋆: the similarity calibration method is applied. "oracle": the RMOT localization results are corrected based on GT. The results of first six rows are reported from TransRMOT [37]. Previous best results are bolded in blue, and our best results are in red.

| Method | Detector | HOTA | DetA | AssA | DetRe | DetPr | AssRe | AssPr | MOTA | IDF1 |
|---|---|---|---|---|---|---|---|---|---|---|
| FairMOT† [46] | DLA-34 [44] | 23.46 | 14.84 | 40.15 | 17.40 | 43.58 | 53.35 | 73.15 | 0.80 | 26.18 |
| DeepSORT [36] | DLA-34 [44] | 25.59 | 19.76 | 34.31 | 26.38 | 36.93 | 39.55 | 61.05 | - | - |
| ByteTrack† [47] | DLA-34 [44] | 22.49 | 13.17 | 40.62 | 16.13 | 36.61 | 46.09 | 73.39 | -7.52 | 23.72 |
| CSTrack [24] | YOLOv5 [19] | 27.91 | 20.65 | 39.10 | 33.76 | 32.61 | 43.12 | 71.82 | - | - |
| TransTrack [32] | DeformableDETR [53] | 32.77 | 23.31 | 45.71 | 32.33 | 42.23 | 49.99 | 78.74 | - | - |
| TrackFormer [29] | DeformableDETR [53] | 33.26 | 25.44 | 45.87 | 35.21 | 42.19 | 50.26 | 78.92 | - | - |
| TransRMOT† [37] | DeformableDETR [53] | **38.06** | **29.28** | **50.83** | **40.20** | **47.36** | **55.43** | **81.36** | **9.03** | **46.40** |
| ByteTrack[47]+iKUN⋆ | YOLOv8 [20] | 41.25 | 29.59 | 57.83 | 44.23 | 43.39 | 63.77 | 74.96 | 5.27 | 51.82 |
| OC-SORT[6]+iKUN⋆ | YOLOv8 [20] | 42.08 | 29.76 | 60.01 | 42.39 | 46.30 | 64.76 | 81.26 | 11.02 | 53.16 |
| DeepSORT[36]+iKUN⋆ | YOLOv8 [20] | 42.46 | 31.64 | 57.56 | 46.03 | 46.32 | 63.48 | 77.66 | **12.50** | 52.57 |
| Deep OC-SORT[28]+iKUN⋆ | YOLOv8 [20] | 42.94 | 29.90 | 62.15 | 46.35 | 42.45 | 69.42 | 77.99 | 3.63 | 53.71 |
| StrongSORT[13]+iKUN⋆ | YOLOv8 [20] | 43.30 | 31.44 | 60.09 | 47.14 | 44.33 | 66.60 | 76.25 | 10.10 | 54.05 |
| **NeuralSORT+iKUN⋆ (ours)** | **YOLOv8 [20]** | **44.56** | **32.05** | **62.48** | **48.53** | **44.76** | **70.52** | **76.66** | **9.69** | **55.40** |
|  | **DeformableDETR [53]** | **48.84** | **35.74** | **66.80** | **51.97** | **52.25** | **72.95** | **87.09** | **12.26** | **54.05** |
| FairMOT† [46] | oracle | 33.02 | 26.39 | 41.33 | 28.13 | 81.03 | 42.54 | 88.58 | 18.72 | 33.46 |
| ByteTrack† [47] | oracle | 35.23 | 25.69 | 48.30 | 27.82 | 77.03 | 51.12 | 86.45 | 18.26 | 35.29 |
| TransRMOT† [37] | oracle | 54.50 | 48.74 | 60.95 | 56.67 | 77.70 | 63.23 | 93.82 | 39.44 | 58.57 |
| **NeuralSORT+iKUN⋆ (ours)** | **oracle** | **61.54** | **48.59** | **77.94** | **64.06** | **66.79** | **82.92** | **91.23** | **31.84** | **62.05** |

set. However, the test set of RMOT datasets follows non-uniform distribution, which makes the existing solutions ineffective. **ii) Open test set.** RMOT is an open-set task, i.e., there are unseen descriptions during the test time. That means previous training-time solutions may not work on the test set.

Inspired by above observations, we propose a test-time similarity calibration method to refine the predicted similarity score by iKUN. Specifically, the frequencies of all $N_{tr}$ training descriptions $\{\mathcal{E}_i^{tr}\}_{i=1}^{N_{tr}}$ are calculated, denoted as $\{p_i^{tr}\}_{i=1}^{N_{tr}}$. Given a test description $\mathcal{E}_j^{ts}$, the normalized similarity between it and $\mathcal{E}_i^{tr}$ is formulated as:

$$w_{ij} = \frac{exp(\tau \cdot x_{ij})}{\sum_k exp(\tau \cdot x_{ik})}, \tag{3}$$

where $\tau$ (set to 100) is the temperature parameter, and $x_{ij}$ is the similarity score estimated by any language model. Then, the *pseudo frequency* of $\mathcal{E}_j^{ts}$ can be estimated by $p_j^{ts} = \sum_i w_{ij} \cdot p_i^{tr}$, which is further utilized to refine the similarity score $s_j$ from iKUN as $s'_j = s_j + f(p_j^{ts})$. Here, $f(\cdot)$ is designed as a linear function $f(x) = a \cdot x + b$ for simplicity.

### 3.4. Neural Kalman Filter

Kalman filter [21] is widely used to estimate motion status of tracked objects in MOT. It operates in two distinct phases, i.e., state prediction and state update, in which Kalman gain is designed to balance the weights of estimates and observations. Concretely, Kalman gain $K_k$ at time step $k$ is calculated as:

$$K_k = P'_k H_k^T (H_k P'_k H_k^T + R_k)^{-1}, \tag{4}$$

where $H_k$ is observation model, $R_k$ is observation noise, and $P'_k$ is the predicted covariance by:

$$P'_k = F_k P_{k-1} F_k^T + Q_k, \tag{5}$$

where $F_k$ is state transition model, $Q_k$ is process noise, and $P_{k-1}$ is the state covariance at time step $k-1$.

It can be observed that $K_k$ is greatly influenced by $R_k$ and $Q_k$. However, in previous multi-object trackers, they are typically determined by preset parameters, which are hard to adapt to various scenarios. Inspired by it, we construct two neural networks, i.e., R-Net $F_R$ and Q-Net $F_Q$, to dynamically update $R_k$ and $Q_k$ based on the current motion status:

$$R_k = F_R(z_k),\ Q_k = F_Q(x_{k-1}), \tag{6}$$

where $z_k$ is the current observations, and $x_{k-1}$ is the state mean at time step $k-1$.

In implementation, $F_R$ and $F_Q$ are designed as fully connected layers. More complex structures (e.g., LSTM [16] and GRU [8]) are also tried, but don't show obvious improvements.

[*] https://github.com/wudongming97/RMOT

Table 2. **Comparison with state-of-the-art MOT methods on KITTI.** All trackers use the same detection results from YOLOv8.

| **Method** | **Car** | | | | | **Pedestrian** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **HOTA** | **DetA** | **AssA** | **MOTA** | **IDF1** | **HOTA** | **DetA** | **AssA** | **MOTA** | **IDF1** |
| ByteTrack [47] | 56.34 | 51.15 | 62.58 | 57.29 | 75.24 | 38.87 | 33.84 | 45.35 | 23.16 | 57.85 |
| OC-SORT [6] | 57.24 | 51.24 | 64.60 | 60.23 | 76.54 | 40.35 | 35.24 | 46.54 | 30.41 | 60.46 |
| Deep OC-SORT [28] | 58.78 | 52.85 | 66.08 | 60.01 | 78.12 | 45.65 | 38.22 | 54.77 | 30.60 | 66.61 |
| StrongSORT [13] | 59.78 | 55.69 | 64.96 | 65.36 | 77.78 | 48.89 | 43.22 | 55.61 | 51.60 | 73.58 |
| DeepSORT (baseline) [36] | 58.58 | 56.55 | 61.51 | 66.99 | 76.00 | 45.03 | 39.37 | 51.93 | 46.33 | 68.97 |
| **NeuralSORT (ours)** | **61.48** | **57.18** | **66.91** | **66.60** | **80.33** | **50.47** | **44.05** | **58.19** | **52.35** | **74.47** |

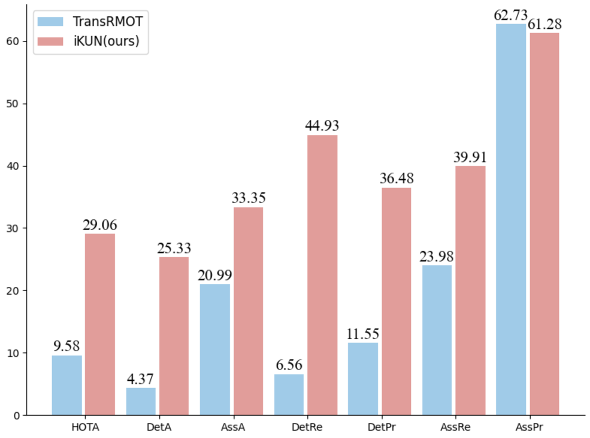


Figure 5. **The performance of TransRMOT[37] and iKUN on Refer-Dance.**

# 4. Experiment

## 4.1. Experimental setup

**Dataset and Metrics.** Refer-KITTI [37] is currently the only public dataset for referring multi-object tracking, which is extended from KITTI [14]. We follow the official data split protocols, in which 15 videos with 80 distinct descriptions are used for training, and 3 videos with 63 distinct descriptions are used for testing. HOTA series [26], MOTA [2], IDF1 [31] are used for evaluation.

We also construct a more challenging RMOT dataset Refer-Dance by extending DanceTrack [33]. It contains 40 videos with 39 distinct descriptions for training, and 25 videos with 17 distinct descriptions for testing. The description annotations focus on motion and dressing status, e.g., "dancing person with black T-shirt and green pants" and "standing person dressed all in black". Please refer to supplementary Sec.B for more examples.

To validate the effectiveness of proposed neural Klaman filter, we further conduct experiments on KITTI and DanceTrack. The same evaluation metrics as above are utilized.

**Implementation Details.** For iKUN, we borrow the visual and textual encoders from `CLIP-RN50` [30]. The feature dimensions are set as $C_v = 2048$, $C_t = C = 1024$. The window size $T$ is set to 8 with a stride of 4. The visual and textual heads are multi-layer perceptrons, temporal model is realized by temporal average pooling, and logit head is realized by cosine similarity. The model is trained on ground truth tracklets with focal loss [25] for 100 epochs. The initial learning rate is set to 1e-5 and decays according to the cosine annealing strategy. The textual model is frozen for training stability. For similarity calibration, pretrained CLIP is applied for textual encoding. The parameters in mapping function $f(\cdot)$ are set as $a = 8, b = -0.1$.

For neural Kalman filter (NKF), R-Net and Q-Net are jointly trained for 10 epochs with mean square error loss. The learning rate is set as the same as iKUN. For inference, we integrate NKF into DeepSORT [36], along with the following tricks: i) Extra exiting decision. Trajectories whose estimated positions exceed the image range are deleted. ii) Extra velocity cost. Extra association cost term based on velocity is introduced inspired by OC-SORT [6]. iii) Linear interpolation. Missing detections are restored by linear interpolation. The final tracker is termed as **NeuralSORT**.

## 4.2. Benchmark Experiments

**Refer-KITTI.** We compare our methods with previous solutions on Refer-KITTI in Tab.1. Current SOTA method, TransRMOT [37], obtains 38.06%, 29.28%, 50.83%, corresponding to HOTA, DetA, AssA, respectively. In comparison, we integrate our iKUN into various off-the-shelf trackers based on YOLOv8 [20], and achieve consistent improvements, i.e., 41.25%-44.56% HOTA. By switching to the same detector as TransRMOT, i.e., DeformableDETR [53], we achieve 48.84%, 35.74%, 66.80%, corresponding to HOTA, DetA, AssA, respectively. Importantly, benefiting from the flexibility of our framework, iKUN is only trained once for multiple trackers.

Moreover, to focus on the comparison of the association and referring ability, we conduct oracle experiments to eliminate the interference of localization accuracy. That is, the coordinates $(x, y, w, h)$ of final estimated trajecto-

Table 3. **Ablation study on different designs of knowledge unification module.** "YOLOv8+NeuralSORT" are used as multi-object tracker. The default setting is marked in gray .

| Method | KUM | HOTA | DetA | AssA | DetRe | DetPr | AssRe | AssPr |
|---|---|---|---|---|---|---|---|---|
| baseline | - | 39.09 | 26.13 | 58.90 | 33.12 | 50.33 | 66.93 | 75.51 |
| | cascade attention | 43.75 | 31.96 | 60.39 | 44.70 | 48.39 | 68.34 | 76.64 |
| | cross correlation | 40.14 | 28.30 | 57.65 | 40.18 | 45.10 | 65.32 | 76.92 |
| | text-first modulation | 40.90 | 26.58 | 63.62 | 53.93 | 32.50 | 72.66 | 76.73 |

Table 4. **Ablation study on different components of NeuralSORT on KITTI**. NKF: neural kalman filter; DEL: extra exiting decision; VEL: extra velocity cost; INT: linear interpolation. The default setting is marked in gray .

| Method | NKF | DEL | VEL | INT | Car | | | Pedestrian | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | HOTA | DetA | AssA | HOTA | DetA | AssA |
| DeepSORT [28] | - | - | - | - | 58.58 | 56.55 | 61.51 | 45.03 | 39.37 | 51.93 |
| | ✓ | | | | 59.90 | 56.85 | 63.91 | 48.53 | 42.40 | 55.88 |
| | | ✓ | | | 58.69 | 56.55 | 61.74 | 45.03 | 39.37 | 51.93 |
| | | | ✓ | | 58.81 | 56.65 | 61.90 | 47.25 | 42.48 | 52.94 |
| | | | | ✓ | 57.64 | 54.63 | 61.68 | 45.86 | 39.89 | 53.15 |
| | ✓ | ✓ | | | 60.61 | 56.87 | 65.40 | 48.53 | 42.40 | 55.88 |
| | ✓ | ✓ | ✓ | | 61.19 | 57.00 | 66.49 | 49.15 | 43.79 | 55.51 |
| NeuralSORT | ✓ | ✓ | ✓ | ✓ | 61.48 | 57.18 | 66.91 | 50.47 | 44.05 | 58.19 |

Table 5. **Effect of hyper-parameters of similarity calibration.** The second line "cascade attention" in Tab.3 are taken as baseline. The selected parameters are marked in gray .

| a | b | HOTA | DetA | AssA | DetRe | AssRe |
|---|---|---|---|---|---|---|
| 0 | 0 | 43.75 | 31.96 | 60.39 | 44.70 | 68.34 |
| 0 | -0.1 | 43.06 | 31.21 | 59.82 | 41.43 | 67.67 |
| 0 | -0.2 | 41.80 | 29.82 | 59.00 | 37.88 | 66.87 |
| 2 | 0 | 44.20 | 32.18 | 61.26 | 46.49 | 69.25 |
| 2 | -0.1 | 43.66 | 31.92 | 60.19 | 43.63 | 68.09 |
| 2 | -0.2 | 42.80 | 30.92 | 59.65 | 40.42 | 67.45 |
| 4 | 0 | 44.36 | 32.08 | 61.89 | 48.06 | 69.95 |
| 4 | -0.1 | 44.07 | 32.14 | 60.99 | 45.39 | 68.94 |
| 4 | -0.2 | 43.55 | 31.72 | 60.24 | 42.58 | 68.08 |
| 8 | 0 | 44.10 | 31.10 | 63.11 | 50.41 | 71.35 |
| 8 | -0.1 | 44.56 | 32.05 | 62.48 | 48.53 | 70.52 |
| 8 | -0.2 | 44.22 | 32.09 | 61.47 | 45.91 | 69.42 |

ries are revised based on ground truth. Please note that no bounding boxes are added or deleted, and no IDs are modified. In this setting, our methods are also performant compared with TransRMOT, i.e., 61.54% vs. 54.50% HOTA.

**Refer-Dance.** We further compare our methods with TransRMOT on our constructed Refer-Dance dataset. ByteTrack [47] with NKF is taken as our baseline tracker. The comparison results are shown in Fig.5, It can be observed that our methods achieve much better results among main metrics, e.g., 29.06% vs. 9.58% HOTA, 25.33% vs. 4.37% DetA and 33.35% vs. 20.99% AssA.

**KITTI.** We compare the designed NeuralSORT with current SOTA trackers on KITTI in Tab.2. All trackers utilize the same detections from YOLOv8. For simplicity, we use the same data split protocol as in Refer-KITTI. It is shown that our NeuralSORT achieves the best results for both car and pedestrian classes.

### 4.3. Ablation Experiments

**Knowledge unification module.** The three designs of KUM are compared in Tab.3. Unification without textual guidance in Eq.1 is taken as baseline. It is shown that all these strategies can bring remarkable improvements over baseline method, which demonstrates the effectiveness of the textual guidance mechanism. In detail, "text-first modulation" achieves best association performance (AssA), but is poor in detection (DetA). "Cross correlation" obtains higher DetA but lower AssA. "Cascade attention" achieves the best results for HOTA and DetA metrics, and is comparable for AssA metrics. Finally, we choose "cascade attention" as the default design of KUM.

**Similarity calibration.** We investigate the effect of hyper-parameters $a, b$ in the mapping function $f(\cdot)$ in Tab.5. As reported, the performance is robust to the varying values. In this work, we choose $a = 8$ and $b = -0.1$ as default, which achieves performance gains of 0.81% HOTA and 2.09% AssA.

**Neural Kalman Filter.** We first take DeepSORT as baseline and study different components of NeuralSORT on KITTI in Tab.4. Most importantly, NKF improves HOTA

Table 6. **The comparison between vanilla Kalman filter and neural Kalman filter for multiple pedestrian tracking.** We use ByteTrack[47] as the baseline tracker and experiment on the KITTI and DanceTrack dataset.

| Dataset | Kalman | HOTA | DetA | AssA | MOTA | IDF1 |
|---|---|---|---|---|---|---|
| KITTI | vanilla | 38.87 | 33.84 | 45.35 | 23.16 | 57.85 |
| | **neural** | **44.89** | **37.54** | **54.08** | **32.02** | **64.46** |
| DanceTrack | vanilla | 46.88 | 70.18 | 31.44 | 87.56 | 52.32 |
| | **neural** | **54.52** | **78.14** | **38.19** | **89.37** | **54.67** |

Table 7. **Comparison of training and inference time between TransRMOT and iKUN.** Both models are trained for 100 epochs. All experiments are conducted on the same machine with multiple Tesla T4 GPUs on dataset Refer-KITTI.

| Method | mode | GPU number | time cost |
|---|---|---|---|
| TransRMOT | training | 4 | 44 hours 53 minutes |
| | inference | 1 | 2 hours 34 minutes |
| iKUN | training | 2 | 2 hours 25 minutes |
| | inference | 1 | 1 hour 38 minutes |

by 1.32% for car and 3.50% for pedestrian. Other tricks further bring gains of 1.58% and 1.94% for car and pedestrian respectively. Then, we take ByteTrack as baseline and further investigate the effect of NKF on KITTI and DanceTrack in Tab.6. Significant improvements can be observed on both two datasets for all evaluation metrics, which shows the superiority of our method.

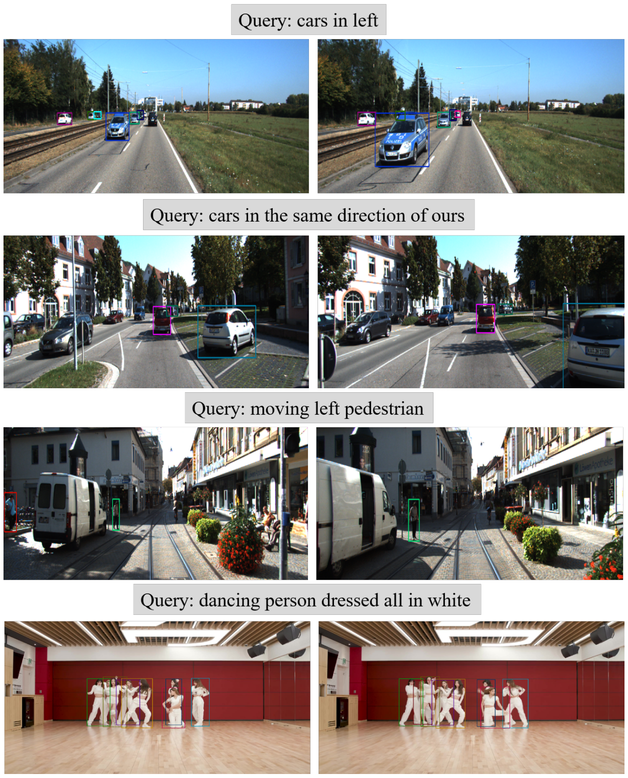
**Training and inference time.** One concern is the computational cost of our multi-stage framework. We conduct experiments on Refer-KITTI with multiple Tesla T4 GPUs and compare the training and inference time between TransRMOT and iKUN in Tab.7. It can be observed that our method achieves much lower time cost. Note that, for fair comparison, the tracking process is also included for the inference time.

### 4.4. Qualitative Results

We visualize several typical referring results in Fig.6. The first query focuses on the location of cars, the second one describes the orientation of cars, the third query includes the motion and location of persons, and the fourth one attends to the motion and dressing status of persons. Our methods can successfully track targets based on various queries.

### 4.5. Limitations

The main limitation of our multi-stage framework is the miscellaneous engineering details. For example, there exist several nontrivial hyperparameters that need to be tuned in the tracking, referring and post-processing components. Moreover, the temporal modelling capability of our model is limited, which constrains the performance for motion-related queries.


Query: cars in left

Query: cars in the same direction of ours

Query: moving left pedestrian

Query: dancing person dressed all in white

Figure 6. **Qualitative results of our method on Refer-KITTI (row 1 to 3) and Refer-Dance (row 4).**

## 5. Conclusion

In this work, we present a novel module, iKUN, which can be plugged after any multi-object trackers to realize referring tracking. To address the one-to-many correspondence problem, knowledge unification module is designed to modulate visual embeddings based on textual descriptions. The similarity calibration method is further proposed to refine predicted scores with estimated pseudo frequencies in the open test set. Moreover, two light-weight neural networks are introduced into Kalman filter to dynamically update the process and observation noise variables. The effectiveness of our methods is demonstrated by experiments on the public dataset Refer-KITTI and our newly constructed dataset Refer-Dance.

## Acknowledgments

This work is supported by Chinese National Natural Science Foundation under Grants (62076033) and BUPT Excellent Ph.D. Students Foundation (CX2022145).

## References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 3

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 6

[3] 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)*, pages 3464–3468. IEEE, 2016. 2

[4] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2017. 2

[5] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4985–4995, 2022. 3

[6] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9686–9696, 2023. 3, 5, 6

[7] 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*, pages 213–229. Springer, 2020. 3

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 5

[9] Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8415–8424, 2021. 4

[10] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 4

[11] Yingxiao Du and Jianxin Wu. No one left behind: Improving the worst categories in long-tailed learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15804–15813, 2023.

[12] Yingjun Du, Jiayi Shen, Xiantong Zhen, and Cees GM Snoek. Superdisco: Super-class discovery improves visual recognition for the long-tail. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19944–19954, 2023. 4

[13] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deep-sort great again. *IEEE Transactions on Multimedia*, 2023. 3, 5, 6

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2, 6

[15] Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, and Xiaofeng Pan. Mat: Motion-aware multi-object tracking. *Neurocomputing*, 476:75–86, 2022. 3

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5

[17] Jang Hyun Cho and Philipp Krähenbühl. Long-tail detection with effective class-margins. In *European Conference on Computer Vision*, pages 698–714. Springer, 2022. 4

[18] Ding Jiang and Mang Ye. Cross-modal implicit relation reasoning and aligning for text-to-image person retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2787–2797, 2023. 4

[19] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, Yonghye Kwon, Kalen Michael, Jiacong Fang, Zeng Yifu, Colin Wong, Diego Montes, et al. ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation. *Zenodo*, 2022. 5

[20] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, 2023. 5, 6

[21] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 2, 5

[22] Shiping Li, Min Cao, and Min Zhang. Learning semantic-aligned feature representation for text-based person search. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2724–2728. IEEE, 2022. 4

[23] Chao Liang, Zhipeng Zhang, Xue Zhou, Bing Li, and Weiming Hu. One more check: making "fake background" be tracked again. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1546–1554, 2022. 3

[24] Chao Liang, Zhipeng Zhang, Xue Zhou, Bing Li, Shuyuan Zhu, and Weiming Hu. Rethinking the competition between detection and reid in multiobject tracking. *IEEE Transactions on Image Processing*, 31:3182–3196, 2022. 1, 5

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6

[26] 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. *International journal of computer vision*, 129:548–578, 2021. 6

[27] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647, 2022. 4

[28] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. *arXiv preprint arXiv:2302.11813*, 2023. 5, 6, 7

[29] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8844–8854, 2022. 5

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 6

[31] 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*, pages 17–35. Springer, 2016. 6

[32] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 5

[33] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20993–21002, 2022. 2, 6

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 4

[35] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conference on Computer Vision*, pages 107–122. Springer, 2020. 2

[36] 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)*, pages 3645–3649. IEEE, 2017. 2, 5, 6

[37] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14633–14642, 2023. 1, 2, 3, 5, 6

[38] Dongming Wu, Tiancai Wang, Yuang Zhang, Xiangyu Zhang, and Jianbing Shen. Onlinerefer: A simple online baseline for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2761–2770, 2023. 3

[39] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4984, 2022. 3

[40] Shuanglin Yan, Neng Dong, Liyan Zhang, and Jinhui Tang. Clip-driven fine-grained text-image person re-identification. *arXiv preprint arXiv:2210.10276*, 2022. 4

[41] Shuanglin Yan, Hao Tang, Liyan Zhang, and Jinhui Tang. Image-specific information suppression and implicit local alignment for text-based person search. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 4

[42] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. 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*, pages 4799–4808, 2023. 2

[43] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pages 36–42. Springer, 2016. 2

[44] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018. 5

[45] Huanlong Zhang, Jingchao Wang, Jianwei Zhang, Tianzhu Zhang, and Bineng Zhong. One-stream vision-language memory network for object tracking. *IEEE Transactions on Multimedia*, 2023. 3

[46] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021. 2, 5

[47] 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 *European Conference on Computer Vision*, pages 1–21. Springer, 2022. 2, 3, 5, 6, 7, 8

[48] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. 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*, pages 22056–22065, 2023. 1

[49] Haojie Zhao, Xiao Wang, Dong Wang, Huchuan Lu, and Xiang Ruan. Transformer vision-language tracking via proxy token guided cross-modal fusion. *Pattern Recognition Letters*, 168:10–16, 2023. 3

[50] Qihao Zhao, Chen Jiang, Wei Hu, Fan Zhang, and Jun Liu. Mdcs: More diverse experts with consistency self-distillation for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11597–11608, 2023. 4

[51] Yaozong Zheng, Bineng Zhong, Qihua Liang, Guorong Li, Rongrong Ji, and Xianxian Li. Towards unified token learning for vision-language tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 3

[52] Li Zhou, Zikun Zhou, Kaige Mao, and Zhenyu He. Joint visual grounding and tracking with natural language specification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23151–23160, 2023. 3

[53] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 5, 6