RelationTrack: Relation-aware Multiple Object Tracking with Decoupled Representation

En Yu, Zhuoling Li, Shoudong Han and Hongwei Wang

***Abstract*—Existing online multiple object tracking (MOT) algorithms often consist of two subtasks, detection and re-identification (ReID). In order to enhance the inference speed and reduce the complexity, current methods commonly integrate these double subtasks into a unified framework. Nevertheless, detection and ReID demand diverse features. This issue results in an optimization contradiction during the training procedure. With the target of alleviating this contradiction, we devise a module named Global Context Disentangling (GCD) that decouples the learned representation into detection-specific and ReID-specific embeddings. As such, this module provides an implicit manner to balance the different requirements of these two subtasks. Moreover, we observe that preceding MOT methods typically leverage local information to associate the detected targets and neglect to consider the global semantic relation. To resolve this limitation, we develop a module, referred to as Guided Transformer Encoder (GTE), by combining the powerful reasoning ability of Transformer encoder and deformable attention. Unlike previous works, GTE avoids analyzing all the pixels and only attends to capture the relation between query nodes and a few self-adaptively selected key samples. Therefore, it is computationally efficient. Extensive experiments have been conducted on the MOT16, MOT17 and MOT20 benchmarks to demonstrate the superiority of the proposed MOT framework, namely RelationTrack. The experimental results indicate that RelationTrack has surpassed preceding methods significantly and established a new state-of-the-art performance, e.g., IDF1 of 70.5% and MOTA of 67.2% on MOT20.**

***Index Terms*—Multiple object tracking, optimization contradiction, decoupling representation, Transformer encoder, deformable attention.**

I. Introduction

As a fundamental vision task, multiple object tracking (MOT) aims to estimate the locations of several targets [1], [2] and identify which of them belong to the same object [3], [4]. Much attention has been drawn due to its numerous practical applications, such as video analysis [5], autonomous driving [6], robots [7], etc. Although prominent progress has been achieved, existing MOT systems still suffer from poor tracking precision and need improvements.

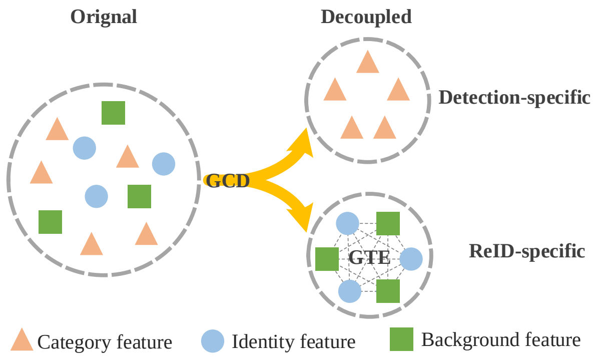


Fig. 1. Diagram that presents how the proposed two modules (GCD and GTE) affect the training procedure. GCD can decouple the learned features as detection-specific and ReID-specific embeddings. Instead of using dot-product attention of Transformer that would lead to huge computational cost, GTE combines deformable attention and Transformer encoder to capture the global semantic relation.

Former MOT frameworks mainly comprise two sub-models, a detection model to localize the targets and a re-identification (ReID) model for connecting the detected targets to the trajectories [8]. However, executing these two models separately results in slow inference speed and huge computational cost. A possible solution to this problem is building networks as the joint-detection-and-embedding architecture [9], which incorporates detection and ReID into a single network and conducts them simultaneously.

Nevertheless, directly merging the detection and ReID models into a single framework leads to a serious optimization contradiction [10]. For the detection part, the network wishes to strengthen the representation similarity of objects belonging to the same category. By contrary, the ReID part desires to maximize the feature discrepancy among various targets, even though they pertain to an identical category. Their inconsistent optimization objectives hinder current MOT frameworks from evolving towards more efficient forms.

In order to address this contradiction, we design a self-motivated feature decoupling module named Global Context Disentangling (GCD), which decouples the learned representation as the detection-specific and ReID-specific embeddings, as shown in Fig. 1. Verified by our experiments, this module contributes to alleviating the contradiction between detection and ReID, and it improves the tracking precision significantly (e.g., from 75.3% to 78.6% on MOT17 for the metric IDF1 as illustrated in Table IV).

Additionally, we observe that previous methods often track targets with only local information. However, a prior sense behind MOT is that the global relation among objects and background is important since the surrounding pixels are effi-

E. Yu and Z. Li contribute equally to this work (Corresponding author: Shoudong Han).

E. Yu, S. Han and H. Wang are with the National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Artificial Intelligence and Automation, Huazhong Univerisity of Science and Technology, 1037 Luoyu Road, Wuhan, China, PC 430074 (e-mail:{yuen, shoudonghan, hongweiwang}@hust.edu.cn)

Z. Li is with the Shenzhen International Graduate School, Tsinghua University, Ministry of Education, Shenzhen, China, PC 518000 (e-mail: lzl20@mails.tsinghua.edu.cn).

cient cues for tracking [11]. To capture the long-range relation, a possible solution is employing the global attention technique [12]. Nevertheless, global attention needs to compute the pairwise similarity of every query nodes with all the other pixels in the image to generate an attention map. This strategy brings a severe calculation burden.

We argue that not all pixels affect the semantic content of query nodes. Therefore, only considering the relation with a small handful of crucial key samples is a better alternative. Inspired by the deformable convolution [13], we propose a deformable attention network which considers the information of the global structural relation. Compared with global attention, deformable attention is quite lightweight and reduces the computational complexity from $O(n^2)$ to $O(n)$. Besides, unlike graph based methods that only gather information from restricted surrounding pixels [14], our developed strategy selects valuable key samples automatically across the whole image.

Furthermore, we resort to the powerful reasoning ability of Transformer encoder [15], [16] for better modeling the long-range dependency. By combining the deformable attention and Transformer encoder, the resulted module, Guided Transformer Encoder (GTE), allows the MOT framework (RelationTrack) to explore the rich content of pixel-to-pixel relation with a global receptive field.

To demonstrate the superiority of RelationTrack[1], extensive experiments have been conducted on three benchmark datasets, i.e., MOT16 [17], MOT17 [17] and MOT20 [18]. The results indicate that the proposed framework has outperformed preceding counterparts significantly. For instance, with respect to the metric IDF1, RelationTrack has surpassed the former state-of-the-art (SOTA) method FairMOT [8] by 3.0% on MOT16 and 2.4% on MOT17.

Comprehensively, our contributions are summarized as follows:

- We observe that the optimization contradiction between detection and ReID during the training procedure hinders the trained network evolving towards more efficient forms. To address this contradiction, we devise a self-motivated module named GCD that decouples the learned features as detection-specific and ReID-specific embeddings.
- We highlight the importance of the global relation among targets and background for the ReID process in MOT. By combining the advantages of deformable attention and Transformer encoder, we develop a lightweight module (GTE) for exploring the long-range dependency across the whole image.
- Incorporating the power of GCD and GTE, the proposed MOT framework, RelationTrack, surpasses its previous counterparts obviously. Evaluated with 5 groups of experiments on 3 benchmark datasets, RelationTrack establishes a new SOTA performance. For example, **we achieve IDF1 of 70.5% and MOTA of 67.2% on the MOT20 benchmark.**

II. Related Works

Influenced by recent great progresses of detection techniques [19], [20], [21], detection-based MOT algorithms have dominated the mainstream. These algorithms mostly comprise two parts, i.e., estimating the locations of targets and associating them to the trajectories. According to how the framework is organized, existing detection-based MOT methods can mainly be categorized into three classes, which are introduced as follows.

A. Tracking-by-detection

There exist numerous publications following the tracking-by-detection paradigm [8], [9], [22], [23]. Many of them concentrate on how to enhance the association ability of methods. Early works often address this challenge through designing algorithms based on kinematics, such as Kalman filtering [24]. These methods usually first take current states of concerned targets as input and predict their locations in the next frames. Afterwards, the Hungarian algorithm [25] is applied to adjust the predicted results.

Nevertheless, since the trajectories of moving objects (such as pedestrians) are highly diverse and hard to be predicted, kinematics based methods often fail to track the targets. To overcome this obstacle, appearance based strategies are introduced. For example, DeepSort [23] utilizes techniques of ReID to extract features and compute the similarity. FGAGT [26] considers both motion and appearance information through the graph neural network.

Although obtaining competitive performance, the aforementioned models still suffer from some restrictions. For instance, they usually implement detection and association as two independent sub-models. Both them affect the final tracking precision considerably. Therefore, if any sub-model fails to behave well, the final results become terrible. In addition, since the two sub-models do not share network layers, the resulted slow inference speed and heavy computation burden restrict the tracking-by-detection paradigm from further improvements.

B. Joint-detection-and-prediction

Many attempts have been conducted to overcome the disadvantages brought by implementing two separate sub-models [27], [28], [29], [30]. Among these attempts, incorporating the two parts, detecting targets and predicting trajectories, into a unified framework is a common practice.

With respect to this motivation, joint-detection-and-prediction methods employ a single network to localize and predict the positions of targets and then associate them. For example, Tracktor [30] adopts the bounding box regression module in Faster R-CNN [31] to correct the predicted results. CTracker [32] exploits rich content among adjacent frames to enhance the regression precision. MAT [33] regards the information of well-established kinematic models as extra cues to facilitate the estimation procedure.

Moreover, some works take advantage of Siamese network [34], [35], [36] to explore the feature similarity and estimate trajectories, such as DeepMOT [37], which decomposes MOT as several single object tracking (SOT) tasks. Likewise, Centertrack [38] produces bounding box offsets with the inspiration from CenterNet [39].

[1] Code is available at: https://github.com/Ahnsun/RelationTrack.

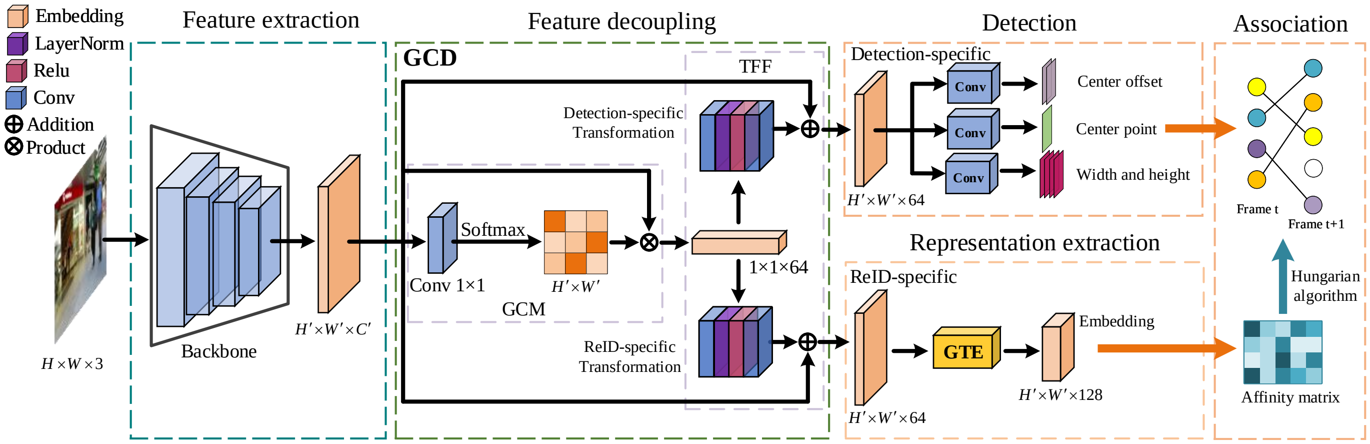


Fig. 2. The overall pipeline of RelationTrack (LayerNorm: layer normalization, Relu: rectified linear unit, Conv: convolution).

Generally, joint-detection-and-prediction models behave better than the tracking-by-detection paradigm mainly due to their trajectory prediction blocks [8], [9], [22], [23]. However, when they are applied to sophisticated application scenarios, further improvements are still demanded.

C. Joint-detection-and-embedding

Similar to the above joint-detection-and-prediction strategies, joint-detection-and-embedding methods often implement their two components, detection and identification, as a one-stage network. However, rather than directly estimating the moving offsets, they associate the concerned targets to trajectories based on extracted embeddings. Among these methods, the outstanding ones include JDE (the first real-time MOT system) [9], FairMOT (an anchor-free tracker) [8], etc.

As the double branches (detection and identification) of joint-detection-and-embedding models contribute to the performance of each other, the tracking precision of trained networks is often competitive. Nevertheless, we observe that there still exist some obstacles which restrict this paradigm. First of all, the contradiction between detection and identification hurts the optimization procedure. Meanwhile, previous frameworks primarily only utilize the local appearance information and ignore the global semantic relation among targets and background regions. In this paper, we propose several strategies to address these obstacles.

III. Method

This section explains how RelationTrack is organized. First of all, Subsection A presents the problem formulation. Then, Subsection B describes the overall framework of RelationTrack. Afterwards, Subsection C, D and E introduce the implementation details of modules (GCD, GTE, detection and association) that compose RelationTrack, respectively. Finally, Subsection F provides the detailed optimization objective settings during the training phase.

A. Problem Formulation

RelationTrack aims to detect the concerned objects (detection) and associate the ones with the same identity among various frames to form trajectories (ReID). It consists of three parts, i.e., a detector $\phi(\cdot)$ to localize the targets, a feature extractor $\psi(\cdot)$ for obtaining representative embeddings and an associator $\varphi(\cdot)$ to produce trajectories.

Formally, given an input image $I_t \in \mathbb{R}^{H\times W\times C}$, we denote $\phi(I_t)$ and $\psi(I_t)$ as b_t and e_t, where $b_t \in \mathbb{R}^{k\times 4}$ and $e_t \in \mathbb{R}^{k\times D}$. In these definitions, H, W and C represent the height, width and number of channels in the input image I_t, respectively. k, t and D are the number of detected targets, index of I_t and dimension of embedding vectors. b_t and e_t severally refer to the coordinates of bounding boxes and corresponding embedding vectors. After detecting the targets and extracting the corresponding embedding vectors, $\varphi(\cdot)$ link b_t in various frames based on e_t to generate the final trajectories.

Generally, in order to estimate the trajectories correctly, the following optimization objectives should be satisfied.

- The bounding boxes corresponding to b_t should contain accurate targets properly.
- e_t should represent the identity information of targets appropriately. Specifically, the extracted embeddings of targets in various frames with the same identity should be more alike to each other in contrast to the ones belonging to other identities.

Technically, when the two objectives are both fulfilled, the tracking performance is promising even with a simple $\varphi(\cdot)$, such as the Hungarian algorithm.

B. Overall Framework

As illustrated in Fig. 2, RelationTrack is composed of 5 parts, i.e., feature extraction, feature decoupling, detection, representation extraction and association. In the first part, given a video with N frames I_t ($t = 1, 2, ...N$), the backbone (DLA-34 [40]) transforms every frame to its corresponding feature maps, respectively. Then, in the feature decoupling part (GCD), the learned features are decomposed as detection-specific and ReID-specific information to address the aforementioned feature contradiction problem. Afterwards, the network of the detection branch (similar to Centernet [39]) localizes the concerned objects based on the detection-specific information. Meanwhile, GTE in the representation estimation part encodes the ReID-specific information as discriminative representation. With respect to the bounding boxes and obtained representation, we link the detected targets to the final trajectories using Hungarian algorithm in the association part.

C. Global Context Disentangling (GCD)

In this section, we introduce the details of GCD that decouples features extracted by the backbone as detection-specific and ReID-specific representations. GCD exactly comprises two phases, i.e., Global Context Modeling (GCM) and Task-specific Feature Transformation (TFF).

The GCM phase aims to aggregate features across all pixels for producing global context embedding. Inspired by Non-local Neural Networks [41], we design a global context block in GCM based on self-attention. With a more concise structure than Non-local Neural Networks, this block aggregates global information efficiently. The implementation details of GCM are as follows.

Denote $x = \{x_i\}_{i=1}^{N_p}$ as the input feature maps, where $N_p = H' \times W'$ (H' and W' are the height and width of input feature maps, respectively). The process of modeling global context z (the output of GCM) can be expressed as

$$z = \sum_{j=1}^{N_p} \frac{exp(W_k x_j)}{\sum\limits_{m=1}^{N_p} exp(W_k x_m)} x_j, \qquad (1)$$

where W_k represents a learnable linear projection and it is modeled as a 1×1 convolution layer in our implementation. The process described in Eq. (1) can be regarded as a weighted pooling operation.

Afterwards, in TFF phase, we desire to obtain the decoupled task-specific information from z. To realize this purpose, we utilize two bottleneck transformations to decouple the features from global content z. However, we observe that some low-level cues are discarded during the transformation procedure and they are valuable for accurate tracking. In order to restore the lost low-level features, we add features produced by bottleneck transformations to x through broadcast element-wise addition. In this way, we obtain the detection-specific embeddings $d = \{d_i\}_{i=1}^{N_p}$ and ReID-specific embeddings $r = \{r_i\}_{i=1}^{N_p}$, respectively. This procedure is formulated as follows:

$$d_i = x_i + W_{d2} ReLU(\Psi_{ln}(W_{d1} z)), \qquad (2)$$

$$r_i = x_i + W_{r2} ReLU(\Psi_{ln}(W_{r1} z)), \qquad (3)$$

where W_{d1}, W_{d2}, W_{e1} and W_{e2} denote four learnable matrices. $ReLU(\cdot)$ and $\Psi_{ln}(\cdot)$ represent the rectified linear unit and layer normalization operator [42], respectively. Given a data batch I with the shape of (B', H', W', C'), $\Psi_{ln}(\cdot)$ can be defined as

$$\mu_b = \frac{1}{H'W'C'} \sum_1^{H'} \sum_1^{W'} \sum_1^{C'} I_{bhwc}, \qquad (4)$$

$$\sigma_b^2 = \frac{1}{H'W'C'} \sum_1^{H'} \sum_1^{W'} \sum_1^{C'} (I_{bhwc} - \mu_b)^2, \qquad (5)$$

$$\tilde{I}_{bhwc} = \frac{I_{bhwc} - \mu_b}{\sqrt{\sigma_b^2 + \epsilon}}, \qquad (6)$$

where I_{bhwc} and $\tilde{I}_{bhwc}$ are the elements in input and output data batches with the index of (b, h, w, c), and ϵ denotes a tiny predefined value.

We can observe from Equation (1) that z is invariant to the selection of i in the process of aggregating the global context information. All the elements of d and r can be computed using the same z. Caused by this characteristic, the calculation complexity of GCD is only $O(C^2)$. In contrast to former global attention methods with complexity of $O(HWC^2)$ [11], GCD is quite computationally efficient. Moreover, according to the experiments in Subsection D and E of Section IV, GCD successfully decouples learned features and addresses the feature contradiction problem.

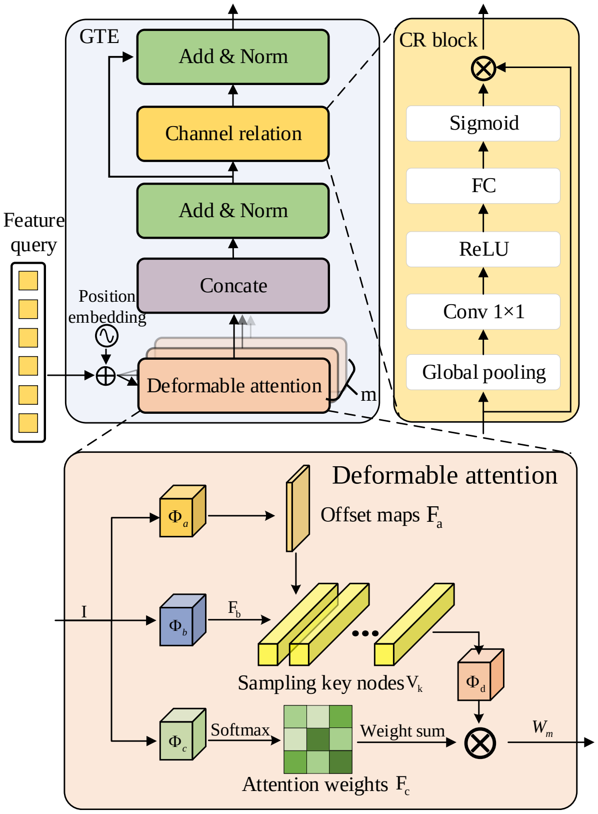


Fig. 3. Diagram of the proposed GTE Module (FC: fully connected layer , ⊕: addition, ⊗: multiplication).

D. Guided Transformer Encoder (GTE)

Attention is a widely adopted strategy to enhance the discriminability of learned features [11]. Most previous works generate attention maps by convolution layers, the receptive field of which is limited [43]. This kind of attention strategy fails to consider the long-range dependency among various targets and background regions.

To bridge this gap, we attempt to utilize the power of Transformer, which considers the interaction among global features and captures the relation of long ranges. Nevertheless, the original Transformer results in serious computing burden, which limits the depth of networks and resolution of input feature maps.

With the purpose of addressing this restriction, we resort to deformable attention to capture the structural content. Unlike the dot-product attention operation of Transfomer, deformable attention can self-adaptively detect valuable key samples and avoid calculating the similarity between query nodes and all values in feature maps. This strategy successfully reduces the computing complexity from $O(H^2W^2C)$ to $O(HWC)$.

Furthermore, we propose the GTE module through combining the advantages of deformable attention and Transformer encoder as illustrated in Fig. 3. Integrated with the outstanding inference capability of Transformer and self-adaptive global receptive field of deformable attention, GTE produces representative embeddings.

In the following, we elaborate the details of two components of GTE, Transformer encoder and deformable attention, respectively.

Transformer encoder. Transformer [16] is first proposed for natural language processing and then extensively applied to various computer vision tasks [44]. Standard Transformer mainly comprises two components, an encoder and a decoder. In GTE, we build a network with a structure similar to the Transformer encoder to obtain powerful embeddings for subsequent association operations.

As shown in Fig. 3, Transformer encoder typically consists of a multi-head attention block and one feed-forward network (FFN). Generally, given a query q and a set of key elements Ω_k as input, Transformer first produces the relation maps via dot-product between q and k ($k \in \Omega_k$). Then, the obtained relation maps are normalized and correlated with k again to generate representative embeddings. Afterwards, we design a channel relation block, consisting of a global pooling layer, a convolutional layer followed by ReLU activation, a fully connected layer followed by sigmoid activation and a broadcast element-wise addition operator. The block is utilized to further capture the channel-wise relation in generated embeddings.

Mathematically, the aforementioned procedure is formulated as:

$$\Phi_T(q,k) = \Gamma(\sum_{i=1}^{N_{head}} W_i(\sum_{j\in\Omega_k} A_{ij}W_i^{'}k_j)), \tag{7}$$

$$A_{ij} \propto exp(\frac{q^T U_i^T V_i k_j}{\rho}), \tag{8}$$

where W_i, $W_i^{'}$, U_i and V_i are learnable weights. $\Phi_T(\cdot)$, $\Gamma(\cdot)$, N_{head} and ρ represent the Transformer, CR block, number of attention heads and normalization factor, respectively.

According to the above description, the computational complexity of Transformer encoder is $O(H^2W^2C)$ for input data with the shape of (H, W, C) [16], [41]. Therefore, the computing cost grows quadratically with respect to the expansion of image size, and the cost is mainly caused by the matrix multiplication operation in multi-head attention. In this work, we adopt a novel attention strategy to alleviate the enormous calculation burden.

Deformable attention. As mentioned before, the huge computational cost of attention based on matrix multiplication results in slow convergence and limited image resolution during the training phase. In order to overcome this problem, we employ deformable attention.

Fig. 4 presents the basic idea behind deformable attention. For any query node in the detected regions of interest (Fig 4(a)), deformable attention self-adaptively selects valuable key samples across the whole image (Fig 4(b)). Afterwards, discriminative representation is produced (Fig 4(c)) through interacting information between query nodes and the corresponding key samples.

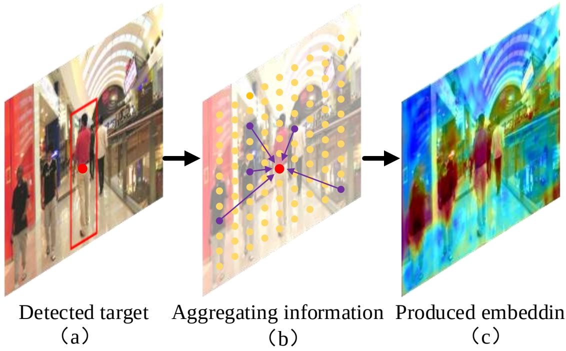


Fig. 4. Abstract diagram of deformable attention. The red point is the query node and purple points are the self-adaptively selected key samples. In contrast to the global encoder, deformable attention avoids the huge computational burden of considering the relation among all pixels.

The details of deformable attention are illustrated in Fig. 3. First of all, given input feature maps I, three independent encoders, $\Phi_a(\cdot)$, $\Phi_b(\cdot)$ and $\Phi_c(\cdot)$, severally encode the input as the offset maps F_a, key maps F_b and query attention maps F_c. Notably, if we select N_k key samples for every query node, F_a contains $2N_k$ channels, which are horizontal and vertical coordinate offsets of the N_k key samples relative to the corresponding query node. Hence, for every query node $q \in I$, we can know its coordinate Z_q and the offsets of key samples $\triangle Z_k = \{\triangle Z_k^i\}_{i=1}^{N_k}$ relative to Z_q based on F_a. Then, the coordinates of key samples $Z_k = \{Z_k^i\}_{i=1}^{N_k}$ can be computed as follows:

$$Z_k^i = Z_q + \triangle Z_k^i. \tag{9}$$

Afterwards, according to the coordinates of selected key samples $Z_k = \{Z_k^i\}_{i=1}^{N_k}$ and key maps F_b, we obtain the key sample vectors $V_k = \{V_k^i\}_{i=1}^{N_k}$. They are further transformed as $\tilde{V}_k$ by the encoder $\Phi_d(\cdot)$. Moreover, we crop the query attention vectors $V_q = \{V_q^i\}_{i=1}^{N_k}$ from F_c with respect to Z_k. The final output maps F_o can be calculated as:

$$F_o = W_m \sum_{i=1}^{N_k} V_q^i \bullet F_c^i, \tag{10}$$

where W_m denotes trainable parameters and $\bullet$ is the Hadamard multiplication [45]. According to the described procedure, the computational complexity of deformable attention is $O(HWC)$ compared with global attention, the complexity of which is $O(H^2W^2C)$.

E. Detection and Association

We devise a detection module $\Psi_d(\cdot)$ similar to Centernet to localize the targets. Given the decoupled detection-specific representation, $\Psi_d(\cdot)$ detects the center points of concerned objects, regress the corresponding center offsets and estimate the bounding box shapes simultaneously. Combing all the obtained outputs, the regions containing targets are determined.

Afterwards, based on the embeddings produced by GTE and estimated bounding boxes, we employ the Hungarian algorithm to match objects among various frames and generate the desired trajectories. The detailed process of associating objects and producing trajectories is similar to FairMOT [8].

First of all, we initialize trajectories based on the detected boxes in the first frame. Then, in the subsequent frames, we link the detected boxes to existing trajectories according to the cosine distances with the computed trajectory embedding vectors. If there exist unmatched detected boxes, they are connected to newly initialized trajectories. In addition, we adopt the trajectory filling strategy proposed in MAT [33] to balance the false positive and false negative scores.

F. Optimization objectives

Since RelationTrack comprises several subtasks, we leverage multiple optimization objectives to train its various parts. The details of these optimization objectives are introduced as follows.

Detection branch. To the end of localizing the concerned objects, the detection branch first estimates the center points of targets. Denote the i_{th} bounding box annotation in a frame as b^i, and its corresponding upper left and lower right coordinates are (l^i, t^i) and (r^i, b^i), respectively. The center point of this bounding box is expressed as (c_x^i, c_y^i), where $c_x^i = \frac{l^i+r^i}{2}$, $c_y^i = \frac{t^i+b^i}{2}$. Assuming there are totally N bounding box annotations in this frame, we can produce the heatmap groundtruth $\hat{R}$ as follows

$$\hat{R}_{xy} = \sum_{i=1}^{N} exp(-\frac{(x-c_x^i)^2+(y-p_y^i)^2}{2(\sigma_p)^2}), \tag{11}$$

where $\hat{R}_{xy}$ is the heatmap pixel with the coordinate of (x, y) and σ_p represents the standard deviation value that is self-adaptively adjusted according to the target scale. Denoting the estimated heatmap pixel at (x, y) as R_{xy}, the similarity value for regressing this pixel can be defined as

$$L_{xy}^h = \begin{cases} (1-R_{xy})^\alpha log R_{xy}, \hat{R}_{xy}=1 \\ (1-\hat{R}_{xy})^\beta (R_{xy})^\alpha log(1-R_{xy}), \hat{R}_{xy} \neq 1 \end{cases} \tag{12}$$

where α and β are hyper-parameters [46]. Afterwards, the loss function for estimating the center points of targets is formulated as follows.

$$L^h = -\frac{1}{N}\sum_{y=1}^{H}\sum_{x=1}^{W} L_{xy}^h \tag{13}$$

In order to determine the bounding box regions, we build another network branch to estimate the box shapes and offsets. In our implementation, the label of the i_{th} box shape is expressed as $\hat{s}^i = (r^i - l^i, b^i - t^i)$ and its corresponding offset label is $\hat{o}^i = (\frac{c_x^i}{4} - \lfloor\frac{c_x^i}{4}\rfloor, \frac{c_y^i}{4} - \lfloor\frac{c_y^i}{4}\rfloor)$, where $\lfloor\cdot\rfloor$ represents an operator that rounds down the input decimals. The optimization objective of this branch for predicting bounding boxes can be expressed as

$$L^b = \sum_{i=1}^{N} \|o^i - \hat{o}^i\|_1 + |s^i - \hat{s}^i\|_1, \tag{14}$$

where o^i and s^i are the output of networks, and $\|\cdot\|_1$ denotes measuring the l_1 distance.

ReID branch. We regard the ReID task as a classification problem and the targets with an identical identity belong to the same category. Given multiple bounding boxes, the network of the ReID branch transforms the features in every bounding box to a class distribution vector $p = \{p_i\}_{i=1}^K$, where K denotes the total number of categories. Assuming the one-hot annotation for the ReID task is $q = \{q_j\}_{j=1}^K$, the loss function adopted by the ReID branch is formulated as

$$L^r = -\sum_{j=1}^{K}\sum_{i=1}^{K} q_j log(p_i). \tag{15}$$

Overall optimization objective. Combining the aforementioned loss functions with learnable coefficients ω_1 and ω_2, we can obtain the overall optimization objective L for RelationTrack, which is given as:

$$L^d = L^h + L^b, \tag{16}$$

$$L = \frac{1}{2}(\frac{1}{e^{\omega_1}}L^d + \frac{1}{e^{\omega_2}}L^r + \omega_1 + \omega_2). \tag{17}$$

IV. Experiment

This section presents the experimental results. Specifically, Subsection A introduces the adopted training and evaluation datasets as well as the evaluation metrics. Among the datasets, MOT16, MOT17 and MOT20 are used for validating the models. Subsection B presents the implementation details in the experiments. Afterwards, Subsection C demonstrates the superiority of RelationTrack by comparing it with existing SOTA counterparts. Subsection D proves the effectiveness of various components in RelationTrack through ablation experiments. Moreover, Subsection E and F indicate that the proposed GCD and GTE modules can enhance the tracking precision efficiently. Subsection G reveals the robustness of RelationTrack towards extreme cases. Finally, Subsection H analyzes the limitations of RelationTrack and how to improve it.

A. Datasets and evaluation metrics

MOT15. MOT15 [49] is the first release of MOTChallenge and it comprises 22 sequences, a half for training and the other half for testing. This dataset totally contains 996 seconds of videos, which include 11286 frames.

MOT16. MOT16 [17] is a commonly adopted benchmark in MOT. Composed of 14 sequences, it covers various scenarios, viewpoints, camera poses and weather conditions. Similar to MOT15, 7 sequences in MOT16 are for training and the others are for validation.

MOT17. MOT17 [17] is established through reconstructing MOT16. In contrast to MOT16, MOT17 provides more reliable groundtruth and more detection bounding boxes produced by various detectors, which include DPM [50], SDP [51], Faster RCNN [31]. The rest is the same as MOT16.

MOT20. Compared with aforementioned datasets, MOT20 [18] is more challenging. It consists of 8 video sequences captured in 3 very crowded scenes. In some frames, more than 220 pedestrians are included. Meanwhile, the data in MOT20 presents high diversity, which could be indoor or outdoor, at day or night.

Extended datasets. Following the settings of preceding works [8], besides the above benchmarks on MOT Challenge, we

TABLE I
COMPARISON WITH PRECEDING STATE-OF-THE-ART METHODS ON MOT16.

Model	IDF1↑	HOTA↑	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	IS↑
JDE [9]	55.8	-	64.4	-	35.4%	20.0%	-	-	1544	18.8
CTracker [32]	57.2	48.8	67.6	78.4	32.9%	23.1%	**8934**	48305	1117	**34.4**
TubeTK [47]	59.4	48.7	64.0	78.3	33.5%	19.4%	10962	53626	4137	1.0
DeepSortv2 [23]	62.2	50.1	61.4	79.1	32.8%	18.2%	12852	56668	2008	17.4
MAT [33]	63.8	54.4	70.5	80.4	**44.7%**	17.3%	11318	41592	928	9.1
CSTrack [10]	71.8	-	70.7	-	38.2%	17.8%	-	-	1071	15.8
FairMOTv2 [8]	72.8	59.8	74.9	**81.2**	44.7%	**15.9%**	10163	34484	1074	25.4
RelationTrack (ours)	**75.8**	**61.7**	**75.6**	80.9	43.1%	21.5%	9786	**34214**	**448**	9.9

TABLE II
COMPARISON WITH PRECEDING STATE-OF-THE-ART METHODS ON MOT17.

Model	IDF1↑	HOTA↑	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	IS↑
CTracker [32]	57.4	49.0	66.6	78.2	32.2%	24.2%	22284	160491	5529	6.8
TubeTK [47]	58.6	48.0	63.0	78.3	31.2%	19.9%	27060	177483	5529	6.8
MAT [33]	63.1	53.8	69.5	80.4	**43.8%**	18.9&	30660	138741	2844	9.0
CenterTrack [38]	64.7	52.2	67.8	78.4	34.6%	24.6%	**18489**	160332	3039	22.0
CSTrack [10]	71.6	-	70.6	-	37.5%	18.7%	-	-	3465	15.8
FairMOTv2 [8]	72.3	59.3	73.7	**81.3**	43.2%	**17.3%**	27507	**117477**	3303	**25.9**
RelationTrack (ours)	**74.7**	**61.0**	**73.8**	81.0	41.7%	23.2%	27999	118623	**1374**	9.8

TABLE III
COMPARISON WITH PRECEDING STATE-OF-THE-ART METHODS ON MOT20.

Model	IDF1↑	HOTA↑	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	IS↑
MLT [48]	54.6	43.2	48.9	78.0	30.9%	22.1%	45660	216803	**2187**	3.7
FairMOTv2 [8]	67.3	54.6	61.8	78.6	68.8%	7.6%	103440	**88901**	5243	**13.2**
CSTrack [10]	68.6	54.0	66.6	78.8	50.4%	15.5%	**25404**	144358	3196	4.5
RelationTrack (ours)	**70.5**	**56.5**	**67.2**	**79.2**	**62.2%**	**8.9%**	61134	104597	4243	4.4

adopt some additional datasets for training, which include ETH [52], CityPerson [53], CalTech [54], CUHK-SYSU [55], PRW [56] and CrowdHuman [57]. Meanwhile, the performance verification and analysis experiments are mainly performed on MOT16, MOT17 and MOT20.

Evaluation metrics. The verification of RelationTrack is carried out based on the CLEAR-MOT Metrics [58], which is a commonly adopted metric set. It is composed of ID F1 score (IDF1), higher order tracking accuracy (HOTA), multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP), mostly tracked rate (MT), mostly lost rate (ML), false positives (FP), false negatives (FN), identity switches (IDS) and inference speed (IS). Among them, HOTA [59] is a recently proposed metric. In contrast to former metrics, it reflects the capability of tracking, detection and association simultaneously. **IDF1, HOTA, MOTA are the most primary indexes for indicating the performance of evaluated models**.

B. Implementation Details

In our experiments, we employ DLA-34 pre-trained on the COCO dataset [60] and fine-tuned on the aforementioned datasets as the backbone of RelationTrack. Its parameters are updated using the Adam optimizer [61] with the initial learning rate of 10^{-4}. During the training procedure, the input batch size is set as 12 and the resolution of every image is 1088×608. The experiments are conducted on 2 NVIDIA GeForce RTX 2080Ti GPUs.

C. Comparison with preceding SOTAs

In this part, we compare the performance of RelationTrack with preceding SOTA methods on three widely adopted benchmarks, i.e., MOT16, MOT17 and MOT20. The results are reported in Table I, Table II and Table III, respectively. As shown in these three tables, RelationTrack has come out among the top in various metrics and surpassed the contrasted counterparts by large margins, especially on the IDF1, HOTA, MOTA and IDS metrics. Note that the classic Transformer consumes much memory and inference time. Due to our computing resource is limited, the classic Transformer without GTE is unable to be trained on our computing device.

MOT16/17. According to Table I and Table II, RelationTrack obtains the metric IDF1 of 75.8% on MOT16 and 74.7% on MOT17. It outperforms the recently proposed FairMOTv2 [8] by 3.0% $(75.8\% - 72.8\%)$ and 2.4% $(74.7\% - 72.3\%)$ on MOT16 and MOT17, respectively. Meanwhile, RelationTrack also behaves better on MOTA than most other trackers. The results indicate the outstanding tracking capability of RelationTrack, which is because that the GCD and GTE modules can produce discriminative features while maintaining competitive detection accuracies. Meanwhile, it can be observed from Table I and Table II that RelationTrack also behaves well on the metric IDS. This phenomenon reveals that the tracking trajectories produced by RelationTrack are still stable even in large-scale and complex scenes.

MOT20. To further evaluate the proposed framework, we verify its performance on the MOT20 benchmark. As shown in Table III, RelationTrack behaves the best on most metrics. Particularly, it surpasses FairMOTv2 by 3.2% $(70.5\%-67.3\%)$ on IDF1, 1.9% $(56.5\%-54.6\%)$ on HOTA and 5.4% $(67.2\%-61.8\%)$ on MOTA.

Although RelationTrack obtains promising performance on the primary metrics (IDF1, HOTA, and MOTA), it does not behave the best on some other metrics, such as MOTP, FP, and IS. Among them, MOTP and FP are metrics that mainly reflect the detection capability of a model. However, all the proposed techniques in this work are about how to improve the ReID head, and the detection head is not the focus. Hence, RelationTrack does not achieve the best results on MOTP and FP.

For IS, the speed of RelationTrack is limited for mainly two reasons. First of all, the speed of RelationTrack is obtained from an RTX 2080Ti GPU, while many of the compared methods are tested on a Tesla V100 GPU, which is faster. Secondly, the attention operation in RelationTrack consumes much computing resource. To satisfy the demand in real-time tracking applications, we release a lite version of RelationTrack, which is introduced in Section IV-H.

D. Ablation Study

In this part, we analyze the effectiveness of GCD and GTE through ablation experiments on MOT17. To this end, we adopt the RelationTrack without GCD and GTE as the baseline model. The results are presented in Table IV.

According to Table IV, the proposed GCD module enhances the tracking performance of the baseline by 3.3% (78.6% – 75.3%) on IDF1 and 2.9% (78.4% – 75.5%) on MOTA. These improvements confirm the importance of alleviating the aforementioned optimization contradiction by decoupling features.

TABLE IV
THE EFFECTIVENESS OF PROPOSED BLOCKS.

Model	IDF1↑	MOTA↑	FP↓	FN↓	IDS↓
Baseline	75.3	75.5	**4025**	22893	580
Baseline + GCD	78.6	78.4	7447	16462	335
Baseline + GTE	76.8	77.4	10376	**14611**	422
RelationTrack	**79.0**	**79.3**	4568	18346	**305**

Meanwhile, it is observed that GTE also benefits the inference phase. As shown, the baseline with GTE outperforms the pure baseline by 1.5% (76.8% – 75.3%) on IDF1 and 1.9% (77.4% – 75.5%) on MOTA. However, only applying GTE alone without GCD does not improve the tracking performance significantly. The main reason for this phenomenon is that the optimization contradiction among various branches restricts the final tracking precision seriously. This issue further proves the importance of decoupling features. When GCD is adopted, GTE is able to enhance the performance of our model in terms of IDF1, MOTA and IDS with larger margins (Baseline + GCD vs. RelationTrack). The results show that GTE helps the network capture complex relation information.

Incorporating the baseline model with both GCD and GTE, the resulted RelationTrack achieves outstanding tracking precision. As presented in Table IV, RelationTrack surpasses the baseline by 3.7% (79.0% – 75.3%) on IDF1 and 3.8% (79.3% – 75.5%) on MOTA. Moreover, IDS is reduced by 47.4% (from 580 to 305). The results indicate the great power brought through combining GCD and GTE.

E. Visualization of GCD

This part aims to verify whether GCD really addresses the optimization contradiction between detection and ReID through decoupling features. To realize this target, we visualize and compare the original and decoupled representation, which is illustrated in Fig. 5.

As shown, in the original feature maps, the model ignores many small yet important targets. Besides, some irrelevant areas are concentrated on mistakenly. On the contrary, when the features are decoupled, the center parts of targets are highlighted in the detection-specific feature maps. Meanwhile, for the ReID-specific embeddings, only the regions covering pedestrians are focused on. This phenomenon demonstrates that GCD decouples the representation vectors as designed. Correspondingly, the aforementioned optimization contradiction between the detection and ReID branches is addressed successfully.

F. Impact of key sample numbers

As introduced in Section V, instead of utilizing global attention, we employ deformable attention to capture the long-range dependency. Rather than analyzing the relationship between query nodes and all the other pixels, deformable attention only considers a small amount of adaptively selected key samples. Therefore, the number of the key samples influences the final tracking precision of models. In this part, we aim to study this problem and the corresponding results are reported in Table V.

TABLE V
COMPARISON OF DIFFERENT NUMBER OF SAMPLING KEY NODES.

Module	Num	MOT17				
		IDF1↑	MOTA↑	FP↓	FN↓	IDS↓
GTE	6	77.9	78.5	4889	18820	378
	9	**79.0**	**79.3**	4568	**18346**	**305**
	12	78.8	79.1	4657	18380	320
	15	78.7	79.1	**4549**	18771	311

As shown in the 3_{th} column of Table V, when the numbers of key samples are 6, 9, 12 and 15, the corresponding IDF1 scores are 77.9%, 79.0%, 78.8% and 78.7%, respectively. It can be observed that the performance of GTE varies slightly with the increase of key node number from 9 to 15. Hence, we think that sampling 9 key nodes is enough for producing discriminative features. Although we can further incorporate more key samples, it results in more computing resource demand. Considering both the tracking performance and calculation burden, we decide to select 9 key samples for every query node.

G. Robustness analysis

In this part, we analyze the robustness of RelationTrack under extreme cases. A hard case is given in Fig. 6 as an example. In this figure, a person is partly occluded and hard to be detected. Both FairMOT and CSTrack fail to extract representative embeddings and associate it with the correct trajectory. Specifically, FairMOT labels this person with a false identity in Frame #514 and overlooks this region in Frame #517. Likewise, CSTrack produces wrong results in the two frames. On the contrary, RelationTrack identifies the target successfully. This example reveals the robustness

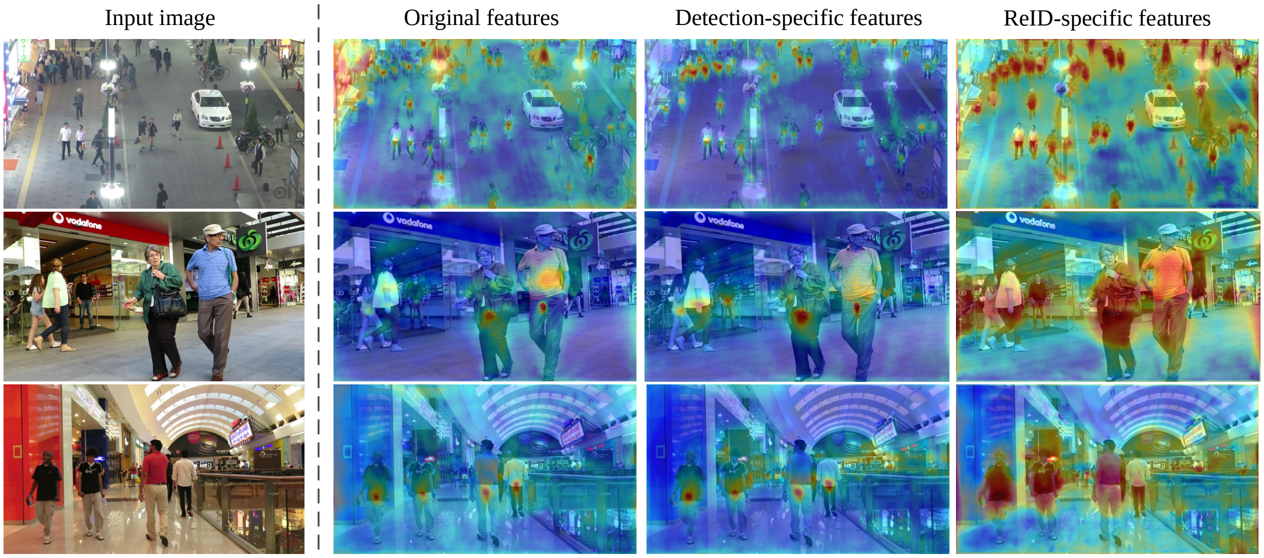


Fig. 5. Visualization of the original and decoupled representation (the detection-specific and ReID-specific features).

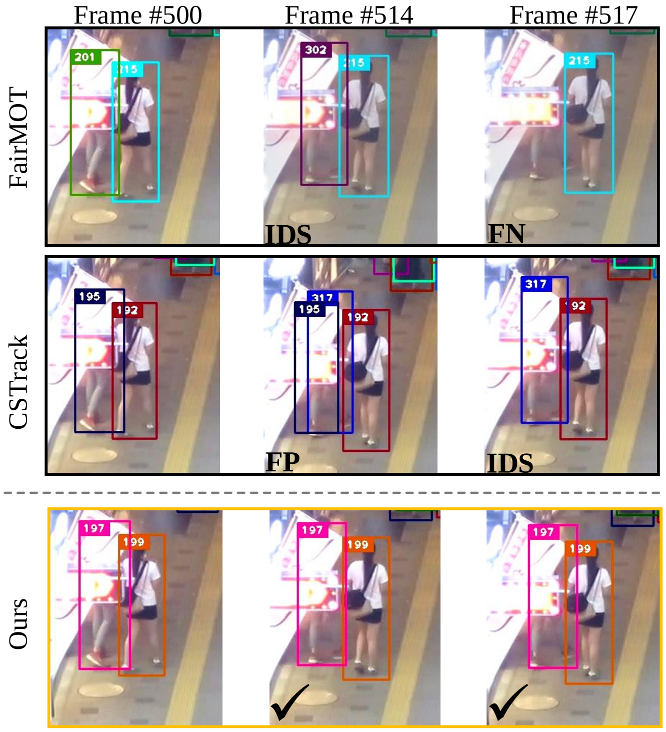


Fig. 6. Robustness analysis of RelationTrack compared with the FariMOT and CSTrack frameworks. The checkmark indicates that the results are correct. IDS, FP and FN denote different kinds of false estimations which include identity switch, false negative and false positive, respectively.

of RelationTrack due to its strong capability of extracting features.

In order to further confirm this issue, we illustrate more cases in Fig. 7. These cases have covered various practical tracking situations, which include indoor and outdoor scenarios, day and night periods, huge and small targets, etc. The results prove that RelationTrack can achieve robust and precise performance even under those challenging conditions.

H. Limitation on speed

According to the results in Table I, Table II, and Table III, RelationTrack is not very fast in the inference stage. This is because although we have simplified the original Transformer via GTE, it still consumes much computing resource and limits the inference speed. In order to satisfy the speed requirement of applications demanding real-time tracking, we further simplify the structure of RelationTrack and release its lite version, named RelationTrack-lite.

In RelationTrack-lite, the complexities of both GTE and GCD are reduced. For GTE, we retain its core components (deformable attention and CR block), and remove the position encoding, multi-head structure and some fully connected layers. In GCD, the length of the encoding vectors is reduced from 64 to 32. Meanwhile, the output dimension of the embedding head is decreased from 128 to 64. We compare the performances of RelationTrack and RelationTrack-lite in MOT17 under the same experiment setting. The results are reported in Table VI. As shown, RelationTrack-lite (23.58FPS) is much faster than RelationTrack (9.85FPS). In conclusion, RelationTrack-lite realizes real-time tracking while still maintaining tracking accuracy comparable with RelationTrack.

TABLE VI
COMPARISON BETWEEN RELATIONTRACK AND RELATIONTRACK-LITE.

Model	IDF1↑	MOTA↑	IS↑
RelationTrack	**79.0**	**79.3**	9.85
RelationTrack-lite	77.5	76.7	**23.58**

V. CONCLUSION

In this work, we observed that the optimization contradiction between the detection and ReID branches restricts current MOT methods from further improvements. Correspondingly, we have developed a module named GCD that alleviates this contradiction through decoupling the features as detection-specific and ReID-specific ones. Moreover, we noticed that preceding MOT frameworks mostly only utilize the local features and neglect to consider the global semantic relation

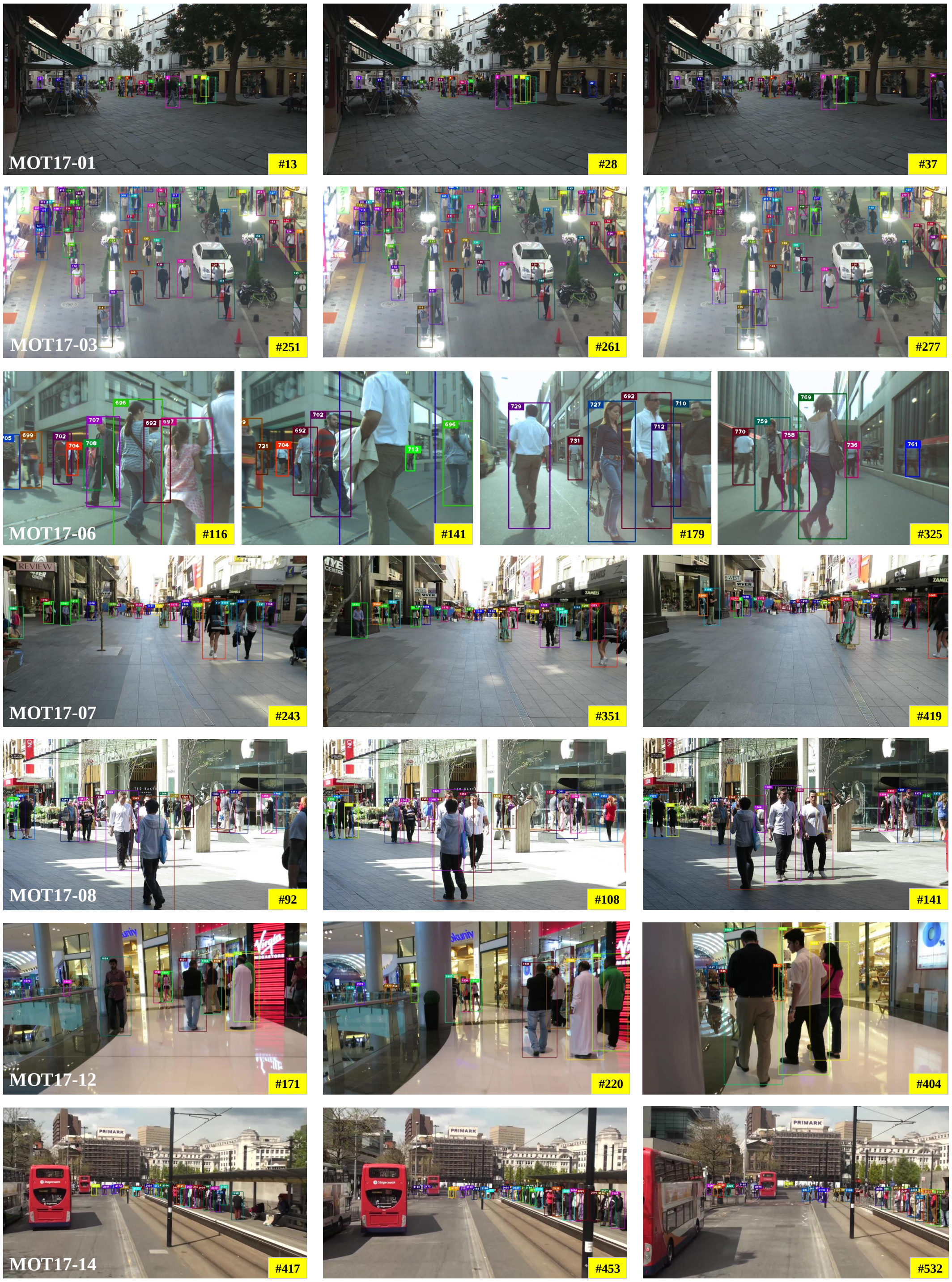


Fig. 7. Tracking examples of RelationTrack on the MOT17 dataset.

among targets and background. We attempted to bridge this gap by devising a network similar to the Transformer encoder. Nevertheless, this strategy suffers from heavy computation burden. To address this issue, we replaced the global attention operator in Transformer encoder as deformable attention and designed a novel module named GTE. This module can capture the global structural information while only consuming a limited amount of calculation resources. Combining the GCD and GTE modules, we proposed a competitive MOT framework, namely RelationTrack. Its performance has been validated through 5 groups of experiments on 3 classic MOT benchmark datasets, which include MOT16, MOT17 and MOT20. The results indicate that RelationTrack has outperformed the contrasted counterparts significantly and established new SOTA results. Finally, we have analyzed the limitations of RelationTrack on speed and released a lite version RelationTrack-lite to satisfy the demand from real-time tracking applications. In the future, we plan to explore how to exploit the temporal information in videos for learning more discriminative representation.

References

[1] W. Ruan, J. Chen, Y. Wu, J. Wang, C. Liang, R. Hu, and J. Jiang, "Multi-correlation filters with triangle-structure constraints for object tracking," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1122–1134, 2018.

[2] H. Yu, K. Zheng, J. Fang, H. Guo, and S. Wang, "A new method and benchmark for detecting co-saliency within a single image," *IEEE Transactions on Multimedia*, vol. 22, no. 12, pp. 3051–3063, 2020.

[3] S. Zhang, Q. Zhang, Y. Yang, X. Wei, P. Wang, B. Jiao, and Y. Zhang, "Person re-identification in aerial imagery," *IEEE Transactions on Multimedia*, vol. 23, pp. 281–291, 2020.

[4] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[5] N. Takahashi, M. Gygli, and L. Van Gool, "Aenet: Learning deep audio features for video analysis," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 513–524, 2017.

[6] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 3569–3577.

[7] A. Manglik, X. Weng, E. Ohn-Bar, and K. M. Kitani, "Forecasting time-to-collision from monocular video: Feasibility, dataset, and challenges," *arXiv preprint arXiv:1903.09102*, 2019.

[8] Y. Zhan, C. Wang, X. Wang, W. Zeng, and W. Liu, "A simple baseline for multi-object tracking," *arXiv preprint arXiv:2004.01888*, 2020.

[9] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, vol. 2, no. 3, p. 4, 2019.

[10] C. Liang, Z. Zhang, Y. Lu, X. Zhou, B. Li, X. Ye, and J. Zou, "Rethinking the competition between detection and reid in multi-object tracking," *arXiv preprint arXiv:2010.12138*, 2020.

[11] Z. Zhang, C. Lan, W. Zeng, X. Jin, and Z. Chen, "Relation-aware global attention for person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3186–3195.

[12] M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, and H. Hu, "Disentangled non-local neural networks," in *European Conference on Computer Vision*, 2020, pp. 191–207.

[13] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[14] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6499–6508.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Conference on Neural Information Processing Systems*, 2017.

[17] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

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

[19] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 039–13 048.

[20] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750.

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

[22] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *2018 IEEE International Conference on Multimedia and Expo*, 2018, pp. 1–6.

[23] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing*, 2017, pp. 3645–3649.

[24] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[25] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[26] C. Shan, C. Wei, B. Deng, J. Huang, X.-S. Hua, X. Cheng, and K. Liang, "Fgagt: Flow-guided adaptive graph tracking," *arXiv preprint arXiv:2010.09015*, 2020.

[27] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang *et al.*, "T-cnn: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2017.

[28] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 408–417.

[29] Z. Zhang, D. Cheng, X. Zhu, S. Lin, and J. Dai, "Integrated object detection and tracking with tracklet-conditioned detection," *arXiv preprint arXiv:1811.11167*, 2018.

[30] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 941–951.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[32] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *European Conference on Computer Vision*, 2020, pp. 145–161.

[33] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, X. Pan, and J. Zhao, "Mat: Motion-aware multi-object tracking," *arXiv preprint arXiv:2009.04794*, 2020.

[34] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision*, 2016, pp. 850–865.

[35] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, 2016, pp. 1420–1429.

[36] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.

[37] Y. Xu, Y. Ban, X. Alameda-Pineda, and R. Horaud, "Deepmot: a differentiable framework for training multiple object trackers," *arXiv preprint arXiv:1906.06618*, 2019.

[38] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European Conference on Computer Vision*, 2020, pp. 474–490.

[39] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[40] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," *arXiv preprint arXiv:2006.11275*, 2020.

[41] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[42] Z. Li, H. Wang, T. Swistek, W. Chen, Y. Li, and H. Wang, "Enabling the network to surf the internet," *arXiv preprint arXiv:2102.12205*, 2021.

[43] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 371–381.

[44] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," *arXiv preprint arXiv:2101.02702*, 2021.

[45] J. Müller, "The hadamard multiplication theorem and applications in summability theory," *Complex Variables and Elliptic Equations*, vol. 18, no. 3-4, pp. 155–166, 1992.

[46] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[47] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "Tubetk: Adopting tubes to track multi-object in a one-step training model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6308–6318.

[48] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke, and Z. Xiong, "Multiplex labeling graph for near-online tracking in crowded scenes," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7892–7902, 2020.

[49] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.

[50] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[51] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2129–2137.

[52] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[53] S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3213–3221.

[54] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 304–311.

[55] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3415–3424.

[56] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camstyle: A novel data augmentation method for person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1176–1190, 2018.

[57] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *arXiv preprint arXiv:1805.00123*, 2018.

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

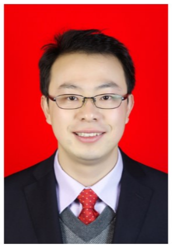
[59] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 548–578, 2021.

[60] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Leanring Representations*, 2015.

En Yu received the B.S. degree in Control Science and Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2020. He is currently pursuing a M.S. degree in Control Science and Engineering from HUST and will be a Ph.D. student in the WenBing Tao's team at the School of Artificial Intelligence and Automation (AIA). His research interests lie in multiple-object tracking applications and 3D object detection.

Zhuoling Li received the B.S. degree in automation from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2020. He is currently pursuing a M.S. degree in artificial intelligence from the Tsinghua University (THU), Beijing, China. His current research interests include deep learning and computer vision.



Shoudong Han was born in Wuhan, China, on March 28, 1983. He received the B.S. degree, M.S. degree and Ph.D. degree in Control Science and Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2005, 2007 and 2010, respectively. He is currently an Associate Professor in the School of Artificial Intelligence and Automation at HUST. His research interests focus on computer vision. He has published a number of innovative works in top-tier journals such as TIP, PR, TMM, Neurocomputing, SP, PRL, and JVCIR.

Hongwei Wang was born in Hubei, China, on August 10, 1998. He received the B.S. degree in Control Science and Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2020. He is now pursuing the M.S. degree in Control Science and Engineering at HUST. His research interests focus on multi-object tracking, motion prediction, object detection, and person re-identification.