Representation Alignment Contrastive Regularization for Multi-Object Tracking

Zhonglin Liu[1,3], Shujie Chen[1,3*], Jianfeng Dong[1,3], Xun Wang[1,3], Di Zhou[2]

[1*]College of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou, 310018, China.
[2]Zhejiang Uniview Technologies Co.,Ltd., Hangzhou, 310051, China.
[3]Zhejiang Key Laboratory of Big Data and Future E-Commerce Technology, Hangzhou, 310018, China.

*Corresponding author(s). E-mail(s): chenshujie@zjgsu.edu.cn;
Contributing authors: zhonglinliu0@outlook.com; dongjf24@gmail.com;
wx@mail.zjgsu.edu.cn; zhoudi@uniview.com;

Abstract

Achieving high-performance in multi-object tracking algorithms heavily relies on modeling spatio-temporal relationships during the data association stage. Mainstream approaches encompass rule-based and deep learning-based methods for spatio-temporal relationship modeling. While the former relies on physical motion laws, offering wider applicability but yielding suboptimal results for complex object movements, the latter, though achieving high-performance, lacks interpretability and involves complex module designs. This work aims to simplify deep learning-based spatio-temporal relationship models and introduce interpretability into features for data association. Specifically, a lightweight single-layer transformer encoder is utilized to model spatio-temporal relationships. To make features more interpretative, two contrastive regularization losses based on representation alignment are proposed, derived from spatio-temporal consistency rules. By applying weighted summation to affinity matrices, the aligned features can seamlessly integrate into the data association stage of the original tracking workflow. Experimental results showcase that our model enhances the majority of existing tracking networks' performance without excessive complexity, with minimal increase in training overhead and nearly negligible computational and storage costs. Our code is available at https://github.com/liuzhonglincc/RATracker.

Keywords: Representation Alignment, Multi-Object Tracking, Contrastive Regularization, Spatio-Temporal Relationship

1 Introduction

Multi-Object Tracking (MOT) has been a long-standing challenge in the field of computer vision [1–3]. The main objective of MOT is to accurately determine the positions of various objects of interest within a video and to establish distinct trajectories for each of these objects. The potential applications of high-resolution MOT are widespread, encompassing areas such as autonomous driving [4], video analysis [5, 6], and scene comprehension [7].

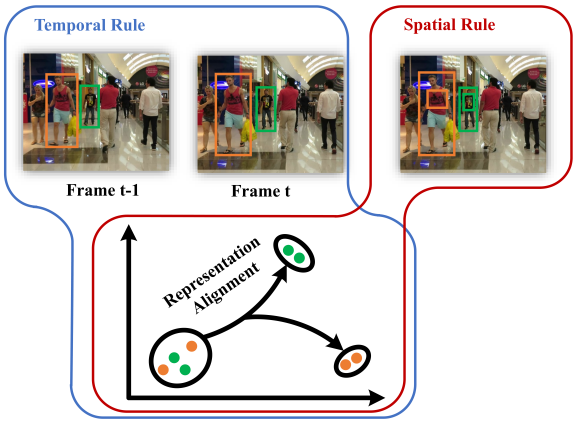


Fig. 1: Demonstration of representation alignment rules: Temporal rule reduces gap between consecutive frame's target representations, while spatial rule unites representations of the same object.

While many researchers [8–11] are increasingly inclined towards addressing the MOT problem by simultaneously tackling both object detection and tracking, the tracking-by-detection (TBD) approach remains a prominent paradigm in MOT due to its efficiency and cost-effectiveness [12–14]. In the tracking-by-detection approach, the MOT task is divided into two distinct tasks: object detection and association. The first task involves identifying and localizing target objects in each frame, while the second task revolves around solving the challenge of associating historical trajectories with presently detected objects.

To achieve high-performance tracking results, a diverse array of algorithms and models have been proposed, incorporating spatial and temporal clues. These include methodologies like the Kalman filter [15], optical flow [16], memory buffers [17, 18], Long Short-Term Memory (LSTM) networks [19], graph-based approaches [19, 20], and transformer models [21–23]. These algorithms and models aim to leverage spatial and temporal relationships through deep learning frameworks or manually devised association rules. While deep learning frameworks can yield remarkable tracking performance, their association components require meticulous design and can sometimes become intertwined with the foundational architecture. Conversely, rule-based algorithms offer greater flexibility and interpretability, but they lag in performance when faced with objects exhibiting irregular motion patterns.

Is there a simple decoupled module that can effectively model spatio-temporal relationships in principle to suit general tracking scenarios while maintaining excellent tracking performance? To answer this question, we first identify two straightforward yet highly effective rules that can be applied to complex object motions. These two rules are then formalized as contrastive regularization terms for training a lightweight module that doesn't rely on object detection. This detector-free module facilitates the provision of temporally and spatially aligned features, aiding in the improvement of data association.

The two proposed rules provide a broader and contrastive perspective on the alignment of representations, summarizing the spatial and temporal relationships among targets as:

1. Representations of same target in consecutive frames should be brought closer, while representations of different targets should be pushed farther apart.
2. Representations of regions originating from the same target should be brought closer, whereas they should be pushed apart otherwise.

The first rule ensures that same objects in consecutive frames are aligned to improve consistency over time. This is based on the idea that the appearance or position of an object doesn't change much between two successive frames. The second rule focuses on aligning object parts across different regions to enhance spatial consistency. This is guided by the principle that parts of the same category tend to have smaller differences between parts of different categories. When considering the distance between representations of different regions within the same object, it's akin to measuring differences within a category (intra-class difference). On the other hand, the distance between regions from different objects represents differences between categories (inter-class difference). As intra-class differences are typically smaller, regions from the same object should be brought closer in representation space, while regions from different objects should be pushed further apart. Please note that these two rules are not applicable to all scenarios. In certain specific situations, such as tracking rapidly

moving objects, the first rule is no longer applicable. Similarly, in cases involving tracking stage actors dressed similarly, the second rule ceases to be applicable. The above two rules are depicted in Figure 1.

We apply these two rules as contrastive regularization terms during the training of a module called *Representation Alignment Module* (RAM). The RAM is a versatile component due to its lack of dependency on detectors, allowing it to be seamlessly integrated into any tracker that follows the tracking-by-detection paradigm. It takes the detector outputs as inputs, enhances the features, and generates features that are aligned either spatially or temporally, which are then used for subsequent association steps. The RAM's efficiency lies in its simplicity, as it only requires a single-layer transformer for encoding the aligned features. Additionally, the training overhead and memory requirements are minimal, as the training solely relies on the target's bounding boxes in the video, rather than using complete video frames. We refer to trackers that incorporate RAMs as *RATrackers*.

The key to contrastive regularization lies in creating proper sets of triplets. Hermans *et al.* [24] confirmed that employing an appropriate triplet generation strategy can unleash the tremendous potential of triplets. Inspired by the successful approach of ByteTrack [14], which employs bounding boxes to achieve state-of-the-art performance, we also utilize bounding boxes as the primary clue for creating triplets. We reformulate the task of creating these sets as a problem of target association based on bounding boxes. In this setup, the target that corresponds to the anchor target is treated as the positive sample, while the ones that don't match are treated as negative samples. This target association problem has been extensively addressed in existing literature and resolved using conventional optimization techniques [25].

The contrastive regularization originates from the matching relationship between bounding boxes. Why can the new features improve the performance of data association compared to the original features? We explain this by looking at how the RAM training process resists noise. Due to limitations in traditional optimization methods like the greedy bipartite assignment algorithm [26] or the Hungarian algorithm [27], the solutions they provide for the target association problem are occasionally not optimal. This results in mismatched bounding boxes. Triplets made from these mismatches act as noisy samples and can harm the training of RAM. However, because RAM is trained using all triplets, it learns to disregard the noisy ones and produce better features. RAM's ability to filter out noise enhances the quality of aligned features compared to its baseline. To achieve optimal performance, we consider aligned features as complements to the original features. During the association phase, we integrate these aligned features by calculating a weighted sum of affinity matrices.

The latest works such as QDTrack [28] and MTrack [18] also use contrastive regularization for improving association. QDTrack [28] adopts quasi-dense human features for conducting contrast learning while MTrack [18] aggregates the whole historical trajectory features. However, it is more likely to include noisy triplets when more candidates are employed, no matter in spatial view as QDTrack [28] did or in temporal view as MTrack [18] did. In comparison, we conduct contrast learning on sparse and clean spatial and temporal triplets so as to learn more reliable contrastive regularization.

Our RAMs demonstrated effectiveness in MOT dataset experiments and minimally impacted the speed of backbone trackers. Additionally, we evaluated RATracker's performance by training it with triplets from detected bounding boxes instead of annotated ones. The results indicated that in unsupervised scenarios, our method only marginally reduced the tracking performance gain, suggesting its capability to enhance tracker performance even without supervision.

The contributions of this paper are in three folds:

- Two simple yet effective rules based on representation alignment have been explored for characterizing the spatial and temporal consistency of targets in MOT. They can be formulated as contrastive regularization terms for training RAMs.
- A novel, detector-free and lightweight module has been introduced for data association. This module efficiently generates spatially and/or

temporally aligned features, seamlessly adaptable across multiple MOT tasks without substantial additional training or memory requirements.
- The results from experiments on MOT datasets have confirmed that our proposed rules and RAMs effectively improve the performance of different trackers.

2 Related Work

As our method focus on improving the performance in association stage using contrastive learning method, in what follows we elaborate on most related works of data association and contrastive learning.

2.1 Data Association

Data association plays a pivotal role in the field of tracking. The conventional approach to accomplish data association involves affinity computation and bipartite graph matching, as established by Munkres in his work [29].

During the affinity computation phase, three key factors are typically taken into account for linking trajectories and detections: motion [30], bounding box information [14], and appearance characteristics [8, 9]. The concept of motion as a clue for association was initially introduced by the SORT algorithm [30]. SORT utilized the Kalman Filter [15] to predict motion in the subsequent frame. Additionally, to capture complex and irregular motions, optical flow was integrated by Xiao et al. [31]. To address challenges posed by substantial camera or object movements, various deep learning-based techniques [10, 21, 32] were developed.To address more complex scenarios involving nonlinear motion and target occlusion, OC-SORT [33] use object observations to compute a virtual trajectory over the occlusion period to fix the error accumulation of filter parameters during the occlusion period. MotionTrack [34] utilizes the displacement of targets in the previous frame and employs attention mechanisms to explore the relationships of motion between targets. Bounding box information was employed in the affinity computation process by SORT [30]. ByteTrack [14] proposed a two-stage matching strategy exclusively based on bounding boxes to enhance association performance. Appearance-based clues were favored in the DeepSORT algorithm [1]. This approach utilized a Re-identification (Re-ID) model to extract appearance features and employed the cosine similarity metric for affinity computation. A recent advancement, TransMOT [35], harnessed a graph transformer to enhance Re-ID features and attain an improved affinity matrix. Some other notable methods like JDE [8], FairMOT [9], and CSTrack [36] achieved enhanced association results by utilizing appearance features and bounding boxes in separate association stages. STRN [13] leverage spatio-temporal relationships to enhance the original Re-ID features, aiming to maximize the dissimilarity between each target's features. However, these techniques focused on only one type of clue within each stage.

In the stage of bipartite graph matching, the matching is determined using either the greedy bipartite assignment algorithm as described in Breitenstein et al.'s work [26] or the optimal Hungarian algorithm as outlined in Xing et al.'s work [27].

A novel tracking-by-regression approach has been introduced in recent studies, including methods like CenterTrack [10], Chained-Tracker [11], TrackFormer [37], MOTR [22], MOTRv2[38] and others. In this approach, instead of explicitly associating the current matched bounding boxes with previous trajectories, the bounding boxes of the current frame are directly predicted based on regression, effectively accomplishing the data association implicitly.

2.2 Contrastive Learning

Due to its remarkable accomplishments in self-supervised representation learning, contrastive learning has gained widespread adoption across various domains, including classification and action recognition. Prominent examples include the works by He et al. [39], Henaff et al. [40], Tian et al. [41], and Wu et al. [42]. Furthermore, contrastive learning has recently found application in the field of MOT. The pioneering work of QDTrack [28] introduced contrastive learning to MOT, enhancing appearance features through quasi-dense similarity learning. Subsequently, MTrack [18] elevated trajectory representation quality by incorporating complete historical trajectory information and engaging in multi-view trajectory contrastive learning.

Despite these impressive achievements in tracking performance, these methods are susceptible to incorporating more instances of noisy triplets. Additionally, their effectiveness hinges on the availability of annotated matching relationships for facilitating contrastive learning.

3 Method

In this section, we first briefly introduce the overall architecture of RATrackers, then elaborate on the structure of RAMs that incorporate different representation alignment rules, and finally introduce the contrastive regularization for training RAMs.

3.1 Overview

The pipeline of RATrackers follows the tracking-by-detection paradigm. The detector and associator are the same as the backbone tracker, and the only difference lies in the RAM that assists associator, as shown in Figure 2. The backbone tracker can be *any* two-stage tracker that conforms to the TBD paradigm, such as FairMOT [9], ByteTrack [14], CSTrack [36] and so on. The RAM takes the target features as input, and outputs the aligned features. During the association stage, reliable association is achieved by utilizing the weighted sum of the affinity matrices derived from aligned features and the vanilla features. The rest association steps remain unaltered.

In what follows, letters with overbar indicates the aligned features. For example, the human features in frame t are characterized by set $\mathcal{H}^t = \{h_i^t\}_{i=1,2,\ldots,K}$ and the aligned human features are denoted as set $\bar{\mathcal{H}}^t$.

3.2 Representation Alignment Module

The RAM architecture is a simple single-layer transformer encoder that comprises fully connected layers (FCs), a multi-head attention layer (MHA), and a feed-forward network (FFN) as shown in Figure 2(b). The FCs transform inputs into higher-dimensional features, the MHA conducts self-attention on these features, and the FFN produces the resultant aligned features. According to rules used for contrastive regularization, RAMs can be divided into temporal RAM, spatial RAM, and spatial-temporal RAM.

Temporal RAM: The Temporal RAM (TRAM) utilizes two fully connected layers to embed inputs, incorporating current human features $\mathcal{H}^t$ and previous trajectory features $\mathcal{C}^{t-1}$ to generate temporally aligned features $\{\bar{\mathcal{H}}^t, \bar{\mathcal{C}}^t\}$ as outputs. The final affinity matrix A_T for bipartite graph matching is calculated by taking a weighted sum of two affinity matrices. One of these matrices is derived from the original features, while the other comes from temporally aligned features. Given coefficient $\alpha_T \in (0, 1)$, the final affinity matrix A_T is computed as

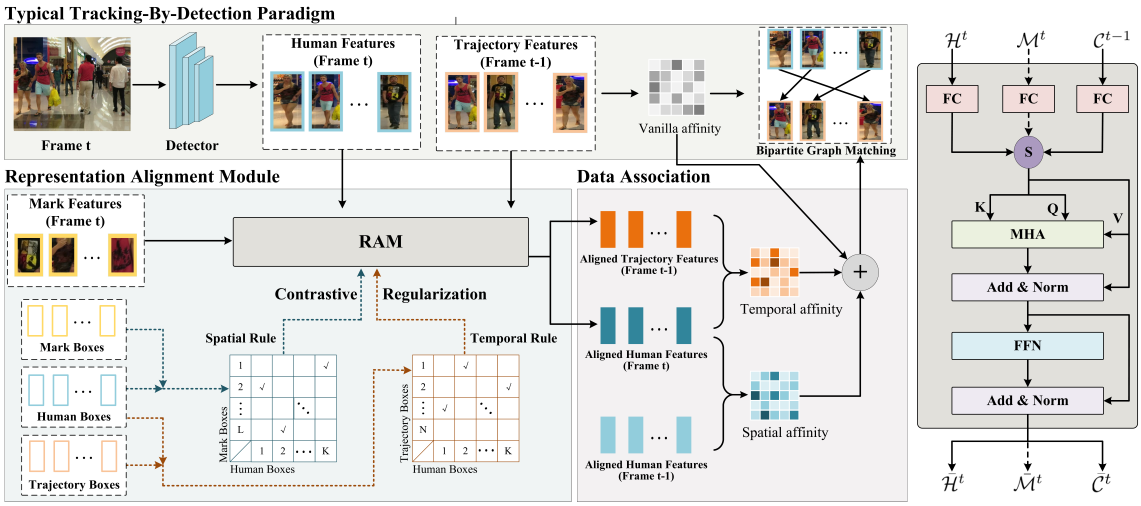
$$A_T = \alpha_T S(\mathcal{H}^t, \mathcal{C}^{t-1}) + (1 - \alpha_T) S(\bar{\mathcal{H}}^t, \bar{\mathcal{C}}^t), \quad (1)$$

where $S(\cdot, \cdot)$ denotes the similarity function that computes the similarities of two sets. Note that similarity function varies with respect to the type of input features. For example, when inputs are coordinates of bounding boxes, the intersection over union (IoU) metric is used. When inputs are encoded features, the clipped cosine distance over $\mathcal{L}_2$-normed features is preferred.

Spatial RAM: The spatial RAM (SRAM) operates by taking human features $\mathcal{H}^t$ and mark features $\mathcal{M}^t$ as inputs, producing spatially aligned features $\{\bar{\mathcal{H}}^t, \bar{\mathcal{M}}^t\}$ as outputs. Owing to variations in individual clothing, there exists a distinct region on each person's body that carries identifying information. The mark box serves the purpose of isolating this unique information, yielding characteristic features for establishing associations. They can be generated either by detecting specially designed marks or by following predefined guidelines, such as enclosing 60% of the area around the center of the detection box. As the aligned mark features $\bar{\mathcal{M}}^t$ are intricately linked to the methodology used for generating the original mark boxes $\mathcal{M}^t$, the aligned human features $\bar{\mathcal{H}}^t$ are more favorable for computing the affinity matrix, a key component in establishing associations. The final affinity matrix A_S for bipartite graph matching is obtained by taking a weighted sum of the original affinity matrix and the affinity matrix derived from spatially aligned features. Given the coefficient $\alpha_S \in (0, 1)$, it is computed as

$$A_S = \alpha_S S(\mathcal{H}^t, \mathcal{C}^{t-1}) + (1 - \alpha_S) S(\bar{\mathcal{H}}^t, \bar{\mathcal{H}}^{t-1}). \quad (2)$$

Spatial-Temporal RAM: The spatial-temporal RAM (STRAM) takes human features



(a) Temporal and spatial alignment (b) Structure of RAMs

Fig. 2: The general process of the RATracker and diagram of contrastive regularization derived from representation alignment rules. (a) Diagram of contrastive regularization terms guided by alignment rules, operator ⊕ means weighted sum. (b) Structure of RAMs, operator Ⓢ means sequence stack, letters with overbar represent aligned features.

$\mathcal{H}^t$, mark features $\mathcal{M}^t$ and previous trajectory features $\mathcal{C}^{t-1}$ as inputs, and outputs the spatially and temporally aligned features $\{\bar{\mathcal{H}}^t, \bar{\mathcal{M}}^t, \bar{\mathcal{C}}^t\}$. The final affinity matrix A_{ST} for bipartite graph matching is computed as the weighted sum of the spatial affinity matrix A_S and temporal affinity matrix A_T with given coefficient $\lambda \in (0, 1)$ as

$$A_{ST} = \lambda A_S + (1 - \lambda) A_T. \tag{3}$$

Note that the purpose of STRAM is to concurrently incorporate spatial and temporal regularization in association. There are multiple approaches to achieve this goal. We opted for a straightforward yet efficient method, which involves a weighted summation of SRAM and TRAM. It's worth considering that employing more intricate fusion techniques might lead to enhanced outcomes.

3.3 Contrastive Regularization for Training

In this section, we elaborate on the triplet generation and contrastive regularization for training RAMs under the guidance of the representation alignment rules.

Temporal Rule: The primary challenge in implementing the temporal rule is to establish consistent correspondences between the same targets across consecutive frames. This task involves solving an association problem, which can be addressed through affinity computation and bipartite graph matching. The entities to be associated are the detected or annotated human bounding boxes in the current frame and the previous frame. To measure their similarity, the Intersection over Union (IoU) metric is utilized.

To create triplets for training, the human boxes in the previous frame are treated as anchors. For a given anchor and its corresponding counterpart in the current frame, if their IoU surpasses a predefined threshold ϵ_{iou}, the matching outcome is considered reliable, and the counterpart is designated as the positive sample. All other human boxes in the current frame are treated as negative samples in this scenario. When the matching confidence is lower, the anchor itself is treated as the positive sample, and the remaining human boxes in the current frame serve as negative samples.

Furthermore, it is also possible to use human boxes in the current frame as anchors. The process of generating triplets follows a similar procedure as described above. In this case, we employ the InfoNCE (Noise Contrastive Estimation) loss [39] to establish contrast between samples. The contrastive loss derived from the initial set of triplets

constitutes the forward temporal loss, while the loss from the latter set of triplets forms the backward temporal loss. The overall temporal loss is the combination of both forward and backward temporal losses and can be calculated as outlined below.

$$\mathcal{L}_T = -\sum_{v_a \in \bar{\mathcal{H}}^t} \log \frac{\exp\left(v_a \cdot v_p^{t-1}/\tau\right)}{\exp\left(v_a \cdot v_p^{t-1}/\tau\right) + \sum_{v_n \in \bar{\mathcal{H}}_n^{t-1}} \exp\left(v_a \cdot v_n/\tau\right)} - \sum_{v_a \in \bar{\mathcal{H}}^{t-1}} \log \frac{\exp\left(v_a \cdot v_p^{t}/\tau\right)}{\exp\left(v_a \cdot v_p^{t}/\tau\right) + \sum_{v_n \in \bar{\mathcal{H}}_n^{t}} \exp\left(v_a \cdot v_n/\tau\right)}, \tag{4}$$

where τ is the temperature hyper-parameter, v_a, v_p, v_n are anchor feature, positive feature and negative feature respectively. The set $\bar{\mathcal{H}}_n$ is the negative feature set with respect to the anchor feature v_a.

Spatial Rule: The primary challenge in implementing spatial rule is to establish connections between targets that belong to the same object within a single frame. This challenge can be addressed using affinity computation and bipartite graph matching. In this context, the targets to be matched are represented by human boxes and mark boxes.

However, the conventional Intersection over Union (IoU) metric is not suitable for measuring similarity in this scenario. IoU struggles to differentiate between a mark and two occluded human bodies due to its heavy reliance on the size of the bodies. This can lead to erroneous matches, where the IoU of a foreground mark and a background human with a smaller body size might exceed the actual ground truth, resulting in inaccurate matches. To mitigate this issue, we have observed that using the size of the mark as a basis for normalization of the intersection provides greater reliability, as it remains invariant to human occlusion. Consequently, we've introduced a more robust metric called *Intersection Rate* (IR) for computing the affinity matrix. The IR metric quantifies the intersection of the mark box and the human box relative to the mark box itself.

To generate triplets for spatial contrast, we adopt the following approach: When the mark box serves as the anchor, a human box that shares an IR value surpassing a certain threshold ϵ_{ir} is designated as the positive sample, while all other human boxes become negative samples. Alternatively, if the IR threshold isn't met, the mark box itself is designated as the positive sample, and all human boxes are classified as negative samples. The same process is mirrored when the human box is utilized as the anchor.

The comprehensive spatial contrastive loss can be calculated in the ensuing manner:

$$\mathcal{L}_S = -\sum_{h_a \in \bar{\mathcal{H}}^t} \log \frac{\exp\left(h_a \cdot m_p/\tau\right)}{\exp\left(h_a \cdot m_p/\tau\right) + \sum_{m_n \in \bar{\mathcal{M}}_n^t} \exp\left(h_a \cdot m_n/\tau\right)} - \sum_{m_a \in \bar{\mathcal{M}}^t} \log \frac{\exp\left(m_a \cdot h_p/\tau\right)}{\exp\left(m_a \cdot h_p/\tau\right) + \sum_{h_n \in \bar{\mathcal{H}}_n^t} \exp\left(m_a \cdot h_n/\tau\right)}, \tag{5}$$

where h_a is the anchor human feature, m_n, m_p are the negative and positive mark features respectively, $\bar{\mathcal{M}}_n^t$ is the set of negative mark features w.r.t. the anchor feature h_a. Similarly, m_a is the anchor mark feature, h_n, h_p are the negative and positive human features respectively, $\bar{\mathcal{H}}_n^t$ is the set of negative human features w.r.t. the anchor feature m_a.

Spatial-Temporal Rule: The implementation of spatial-temporal rule simply replicates the implementation of spatial rule and temporal rule simultaneously. The overall contrastive loss can be computed as

$$\mathcal{L}_{ST} = \mathcal{L}_S + \mathcal{L}_T. \tag{6}$$

4 Experiments

We conduct extensive experiments over three publicly accessible datasets including MOT17 [43], MOT20 [44] and BDD100K [45]. We used the ID score [46] and CLEAR MOT metrics [47] to evaluate the performance of the proposed method. Throughout the experiments, we generated the mark boxes by boxing out 60% of the area around the center of the detection box. The input features can either be bounding boxes characterized by corner coordinates and box-size as (x, y, h, w), or the Re-ID features extracted from some pretrained modules like fastReID [48]. Unless specified otherwise, our input features are assumed to be bounding boxes.

Training Details: The experiments were carried out using PyTorch and an NVIDIA GeForce RTX 2080 Ti GPU. The training process involved running for 50 epochs with a batch size of 5. The chosen optimizer was AdamW [49], initialized with a learning rate of 2×10^{-3}, which

decreased by a factor of 10 every 10 epochs. To accommodate input sequences of varying lengths, a strategy inspired by DETR [50] was applied. Input sequences were standardized to a fixed length of 110 for MOT17, 260 for MOT20 and 100 for BDD100K. This was achieved by appending invalid bounding boxes with all zero coordinates. Notably, these added boxes were disregarded during loss calculations. The output dimension of the fully connected layers was configured to be 128, aligning with the approach.

Hyperparameters: During the course of the experiments, parameters $\lambda = 0.5, \tau = 0.1, \epsilon_{ir} = 0$ were consistently configured. The selection of parameters α_S, α_T, and ϵ_{iou}, however, varied based on the particular experiment being conducted. In cases where the ByteTrack [14] backbone tracker employed two association stages, parameters $\alpha_S = \alpha_T = 0.2$, $\epsilon_{iou} = 0.9$ were chosen for the initial stage, and parameters $\alpha_S = \alpha_T = 0.3$, $\epsilon_{iou} = 0.5$ were employed for the subsequent stage. In instances where only one association stage was utilized such as TransTrack[21], parameters $\alpha_S = \alpha_T = 0.3$, $\epsilon_{iou} = 0.9$ were employed.

4.1 Effectiveness of RAMs

4.1.1 On Different Trackers

This evaluation encompasses five state-of-the-art trackers employing RAMs: JDE [8], CSTrack [36], TransTrack [21], ByteTrack [14] and OC-SORT [33]. All of these trackers adhere to the tracking-by-detection paradigm and involve the computation of affinity matrices for the purpose of association. The experiment was carried out on the MOT17 validation dataset.

Table 1 illustrates that integrating RAMs consistently improves crucial performance metrics such as MOTA, IDF1, and IDS across various trackers. Among the trackers studied, CSTrack [36] and JDE [8] utilize CNNs for feature extraction in association, while TransTrack [21] uses the Transformer architecture for feature generation. ByteTrack [14] directly employs bounding boxes, and OC-SORT [33] refines them through a motion prediction model. These findings emphasize how RAMs enhance the performance of different trackers, irrespective of their specific backbone frameworks, showcasing their versatile applicability.

Method	IDF1 ↑	MOTA ↑	IDS ↓
JDE [8]	63.59	59.98	473
JDE+TRAM	67.30(**+3.71**)	60.31(**+0.33**)	383(**-80**)
JDE+SRAM	66.75(**+3.16**)	60.29(**+0.31**)	372(**-101**)
JDE+STRAM	67.20(**+3.61**)	60.47(**+0.49**)	374(**-99**)
CSTrack [36]	71.82	67.96	340
CSTrack+TRAM	73.56(**+1.74**)	68.52(**+0.56**)	260(**-50**)
CSTrack+SRAM	72.93(**+1.11**)	68.46(**+0.5**)	304(**-36**)
CSTrack+STRAM	73.70(**+1.88**)	68.63(**+0.67**)	291(**-49**)
TransTrack [21]	68.60	67.66	254
TransTrack+TRAM	71.64(**+3.04**)	67.86(**+0.2**)	245(**-9**)
TransTrack+SRAM	69.71(**+1.11**)	67.85(**+0.19**)	250(**-4**)
TransTrack+STRAM	71.14(**+2.54**)	67.98(**+0.32**)	238(**-16**)
ByteTrack [14]	79.07	76.49	165
ByteTrack+TRAM	79.92(**+0.85**)	76.82(**+0.33**)	145(**-18**)
ByteTrack+SRAM	79.90(**+0.83**)	76.82(**+0.33**)	139(**-26**)
ByteTrack+STRAM	80.87(**+1.8**)	76.90(**+0.41**)	155(**-10**)
OC-SORT [33]	77.85	74.12	195
OC-SORT+TRAM	78.09(**+0.24**)	74.35(**+0.23**)	169(**-23**)
OC-SORT+SRAM	78.07(**+0.22**)	74.21(**+0.09**)	192(**-3**)
OC-SORT+STRAM	78.72(**+0.87**)	74.38(**+0.26**)	164(**-31**)

Table 1: Results of applying RAMs to five popular trackers on the MOT17 validation set. ↑ means higher is better, ↓ means lower is better

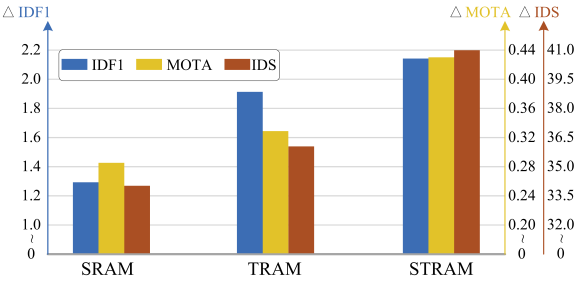


Fig. 3: The average performance of RAMs on various trackers in Table 1.

Figure 3 provides the average performance of RAMs across various trackers. While individual metrics for TRAM might surpass STRAM in specific backbone trackers as indicated in Table 1, overall, STRAM consistently outperforms both SRAM and TRAM. This indicates that, in general, considering the performance of spatio-temporal alignment regularization is superior to solely focusing on single-branch regularization.

4.1.2 On Different Datasets

We validated the effectiveness of RAMs on three datasets: MOT17 [43], MOT20 [44], and

	MOT17-val			MOT20-val			BDD100K-val			Average Performance		
	IDF1↑	MOTA↑	IDS↓	IDF1↑	MOTA↑	IDS↓	IDF1↑	MOTA↑	IDS↓	IDF1↑	MOTA↑	IDS↓
Baseline	75.56	79.85	495	81.62	77.90	913	54.95	45.11	32963	70.71	67.62	11457
Baseline+TRAM	76.34	80.51	**478**	81.73	**78.08**	898	55.54	45.41	31372	71.20	68.00	10916
Baseline+SRAM	75.94	80.50	480	81.74	77.95	897	55.13	45.16	32104	70.94	67.87	11160
Baseline+STRAM	**77.14**	**81.14**	479	**81.86**	77.92	**896**	**55.90**	**45.55**	**30567**	**71.63**	**68.20**	**10647**

Table 2: The performance of RAMs on multiple datasets. The best results are marked in **bold**

Benchmark	**Method**	**MOTA↑**	**IDF1↑**	**HOTA↑**	**AssA↑**	**MT↑**	**ML↓**	**FP↓**	**FN↓**	**IDS↓**
MOT17	RelationTrack [51]	73.8	74.7	61.0	61.5	41.7	23.2	27999	118623	1374
	CenterTrack [10]	67.8	64.7	52.2	-	34.6	24.6	18489	160332	3039
	TraDeS [32]	69.1	63.9	52.7	50.8	36.4	21.5	20892	150060	3555
	CorrTracker [52]	76.5	73.6	60.7	58.9	47.6	12.7	29808	99510	3369
	CTracker [11]	66.6	57.4	49.0	45.2	32.2	24.2	22284	160491	5529
	QDTrack [28]	68.7	66.3	53.9	52.7	40.6	21.9	26589	146643	3378
	MTrack [18]	72.1	73.5	60.5	60.9	49.0	16.8	53361	101844	2028
	TransCenter [53]	73.2	62.2	54.5	49.7	40.8	18.5	23112	123738	4614
	MOTR [22]	78.6	75.0	62.0	60.6	50.3	13.1	23409	94797	2619
	TransMOT [35]	76.7	75.1	61.7	-	51.0	16.4	36231	93150	2346
	TransTrack [21]	75.2	63.5	54.1	47.9	55.3	**10.2**	50157	86442	3603
	TrackFormer [37]	74.1	68.0	57.3	54.1	47.3	10.4	34602	108777	2829
	MeMOT [17]	72.5	69.0	56.9	55.2	43.8	18.0	37221	115248	2724
	CSTrack [36]	74.9	72.6	59.3	57.9	41.5	17.5	23847	114303	3567
	FairMOT [9]	73.7	72.3	59.3	58.0	43.2	17.3	27507	117477	3303
	ByteTrack [14]	80.3	77.3	63.1	62.0	53.2	14.5	25491	83721	2196
	MOTRv2 [38]	78.6	75.0	62.0	60.6	-	-	-	-	-
	OC-SORT [33]	78.0	77.5	63.2	63.2	-	-	**15100**	108000	1950
	MotionTrack [34]	**81.1**	**80.1**	**65.1**	**65.1**	55.5	16.7	23802	81660	**1140**
	ByteTrack+STRAM(ours)	81.0	79.9	64.9	64.8	**56.2**	14.4	24459	**81198**	1383
MOT20	TransCener [53]	58.5	49.6	43.5	37.0	48.6	14.9	64217	146019	4695
	RelationTrack [51]	67.2	70.5	56.5	56.4	62.2	8.9	61134	104597	4243
	MeMOT [17]	63.7	66.1	54.1	55.0	57.5	14.3	47882	137983	1938
	MTrack [18]	63.5	69.2	55.3	55.7	68.8	**7.5**	96123	86964	6031
	TransTrack [21]	65.0	59.4	48.9	45.2	50.1	13.4	27191	150197	3608
	CSTrack [36]	66.6	68.6	54.0	54.0	50.4	15.5	25404	144358	3196
	FairMOT [9]	61.8	67.3	54.6	54.7	68.8	7.6	103440	88901	5243
	ByteTrack [14]	77.8	75.2	61.3	59.6	69.2	9.5	26249	87594	1223
	MOTRv2 [38]	76.2	73.1	61.0	59.3	-	-	-	-	-
	OC-SORT [33]	75.5	75.9	62.1	62.0	-	-	**18000**	108000	**913**
	MotionTrack [34]	**78.0**	76.5	62.8	61.8	**71.3**	9.5	28629	**84152**	1165
	ByteTrack+STRAM(ours)	77.9	**77.3**	**63.3**	**62.8**	70.3	9.6	24353	88867	1309

Table 3: Performance comparison with preceding SOTAs on the testing splits of the MOT17 and MOT20 benchmarks under the private detection protocols. The best results are marked in **bold** and the suboptimal results are annotated with underline

BDD100K [45]. MOT17 and MOT20 are popular pedestrian tracking datasets, with MOT20 having a higher density of pedestrians. BDD100K is a large dataset used for vehicle tracking, comprising 2000 training and testing scenes. To ensure a fair comparison, we introduced a baseline tracking method using YOLOVX [54] for object detection, Kalman filtering for trajectory prediction, and bounding boxes as the association features.

Table 2 illustrates that across various datasets, the use of RAMs consistently enhances the performance of the baseline method. Notably, there's a significant improvement in metrics for MOT17 and BDD100K datasets compared to a more marginal enhancement in MOT20. Additionally, TRAM outperforms STRAM in terms of MOTA in MOT20, likely due to MOT20 containing denser objects with higher chances of visual similarity, potentially causing spatial alignment regularization to be less effective. However, considering the overall results, STRAM still outperforms both single-branch TRAM and SRAM approaches.

4.1.3 On MOT Benchmarks

We use ByteTrack[14] as the backbone tracker and evaluate the performance of ByteTrack+STRAM in the MOT17 and MOT20 benchmarks using a private detection setup. We train STRAMs separately using the training sets from MOT17 and MOT20. To evaluate performance, we submit the tracking results from the test sets to the official MOT Challenge evaluation platform.

Table 3 demonstrates ByteTrack+STRAM's impressive performance. On the MOT17 dataset, it achieves significant scores of 81.0 MOTA and 79.9 IDF1. Even on the more complex MOT20 benchmark, it maintains strong results with 77.9 MOTA and 77.3 IDF1. Notably, both false positive (FP) and false negative (FN) metrics remain minimal for both MOT17 and MOT20. This indicates that STRAM effectively reduces incorrectly tracked boxes and successfully re-establishes tracking for previously overlooked targets through associations. The exceptional performance is credited to STRAM's integration of spatial and temporal rule-based contrastive regularization terms.

4.1.4 Computational Complexity

We performed an experiment to assess computational complexity using a single NVIDIA GeForce RTX 3090 Ti GPU. The outcomes are outlined in Table 4. Because the RAM employs only a single-layer transformer and the input sequence length is typically short (equivalent to the number of targets in each frame), the extra parameters and computational load are minimal. This has a negligible impact on the real-time performance of the original tracker.

Method	Params(M)	Flops(G)	FPS
FairMOT	16.5542	72.932	22.5
FairMOT+TRAM	16.6219	72.939	21.5
FairMOT+SRAM	16.6219	72.939	21.4
FairMOT+STRAM	16.6226	72.942	21.1
ByteTrack	98.9954	793.211	25.7
ByteTrack+TRAM	99.0677	793.218	25.0
ByteTrack+SRAM	99.0677	793.218	25.0
ByteTrack+STRAM	99.0684	793.221	23.8

Table 4: Computational complexity results on MOT17 test set

Type of RAMs	Input Feature	IDF1 ↑	MOTA ↑	IDS ↓
Without RAM	-	72.81	69.06	299
TRAM	Bounding Box	74.44(+**1.63**)	69.37(+**0.31**)	272(-**23**)
	Re-ID	73.93(+**1.12**)	69.18(+**0.12**)	250(-**49**)
SRAM	Bounding Box	74.02(+**1.21**)	69.13(+**0.07**)	290(-**9**)
	Re-ID	74.36(+**1.55**)	69.21(+**0.15**)	267(-**32**)
STRAM	Bounding Box	74.67(+**1.86**)	69.38(+**0.32**)	289(-**10**)
	Re-ID	73.92(+**1.11**)	69.32(+**0.26**)	240(-**59**)

Table 5: Results of RAMs to different input features with FairMOT[9] backbone on the MOT17 validation set. The best results are marked in blue

4.2 Ablation Study

4.2.1 Features for RAMs and Affinity Computation

The contrastive regularization triplets are exclusively derived from bounding boxes. However, when utilized in RAMs and computing affinity matrices, there exists flexibility in the types of features employed. Our experiments on MOT17 validation set, testing diverse input feature types in RAMs and affinity computation, showcase the resilience of our proposed method across these variations.

RAMs can intake appearance features such as Re-ID features or bounding boxes. To implement the RAM module with appearance features, we utilized FastReID [48] for extracting Re-ID features. Table 5 demonstrates the performance enhancements achieved by applying Re-ID features or bounding box features to FairMOT with RAMs. Notably, TRAM, SRAM, and STRAM consistently contributed to a stable improvement of at least 1 in IDF1, regardless of the input feature types.

Association Features	IDF1 ↑	MOTA ↑	IDS ↓
Bounding Box	79.55	77.65	333
Bounding Box + STRAM	81.01(+**1.46**)	77.94(+**0.29**)	267(**-66**)
Re-ID	70.43	73.27	447
Re-ID + STRAM	77.95(+**7.52**)	75.25(+**1.98**)	370(**-77**)
Bounding Box + Re-ID	79.07	77.73	223
Bounding Box + Re-ID + STRAM	81.82(+**2.75**)	78.21(+**0.48**)	203(**-20**)

Table 6: Results of ByteTrack [14] to various association features on the MOT17 validation set with and without STRAM

The robustness of our proposed method across diverse association features is delineated in Table 6. Utilizing ByteTrack[14] as the backbone tracker, we evaluated its tracking performance using various association feature combinations, with detection score thresholds set at $\tau_{high} = 0.6$ and $\tau_{low} = 0.1$. The results in Table 6, focusing on two association stages employing the same type of feature, consistently indicate that incorporating STRAM enhances tracking performance, irrespective of whether IoU, Re-ID, or a combination of both is considered. This underscores the adaptability of RAM in improving various association features.

4.2.2 Embedding Dimension

We conducted an experiment focusing on the output dimension of the FC layers within the STRAM. The backbone tracker used for this experiment was ByteTrack[14]. The results are summarized in Table 7. We achieved the highest values of 80.87 for IDF1 and 76.90 for MOTA when the output dimension was set to 128. Conversely, the lowest value of 154 for IDS was observed when the output dimension was set to 1024. There was only marginal improvement in tracking performance with increasing dimensions. Consequently, for consistency, we maintained an embedding dimension of 128 throughout the experiments detailed in the paper.

4.2.3 Hyperparameters

The objective of STRAM is to seamlessly integrate both spatial and temporal regularization within associations. There exist diverse methods for achieving this integration. In our study, we adopted a straightforward yet useful technique involving a weighted summation of SRAM and

Embedding Dimension	IDF1 ↑	MOTA ↑	IDS ↓
64	80.18	76.81	157
128	**80.87**	**76.90**	155
256	80.26	76.85	162
512	80.67	76.63	164
1024	80.16	76.77	**154**

Table 7: Exploring STRAM performance across various FC layer embedding dimensions on MOT17 validation set

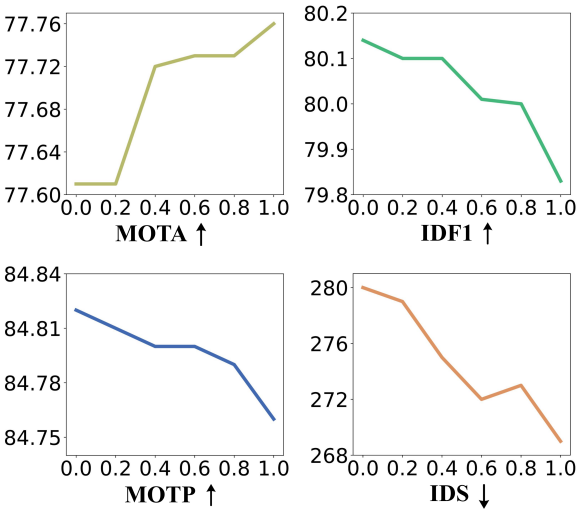


Fig. 4: The impact of hyperparameter λ on STRAM's performance.

TRAM. However, it's conceivable that employing more intricate fusion methods could yield even more promising results. Our experimentation extended to the exploration of various combination coefficient λ, as depicted in the Figure 4. In order to strike a balance among all performance indicators, a coefficient of 0.5 was ultimately determined as the optimal choice.

4.2.4 Association Stages

ByteTrack [14] employs a two-stage association process, comprising a primary stage dedicated to linking high-confidence tracks with detections and a subsequent stage focusing on pairing residual tracks with low-confidence detections. As outlined by [14], our configuration for ByteTrack establishes the detection score thresholds as $\tau_{high} = 0.6$ and $\tau_{low} = 0.1$.

Stage 1	Stage 2	IDF1 ↑	MOTA ↑	IDS ↓
IoU	IoU	79.55	77.65	333
IoU + STRAM	IoU	**81.01(+1.46)**	77.94(+0.29)	267(-66)
IoU	IoU + STRAM	79.70(+0.15)	77.77(+0.12)	328(-5)
IoU + STRAM	IoU + STRAM	80.18(+0.63)	**78.00(+0.35)**	**245(-88)**
Re-ID	Re-ID	70.43	73.27	447
Re-ID + STRAM	Re-ID	77.95(+7.52)	75.25(+1.98)	370(-77)
Re-ID	Re-ID + STRAM	71.73(+1.3)	74.01(+0.74)	409(-38)
Re-ID + STRAM	Re-ID + STRAM	**79.91(+9.48)**	**77.12(+3.85)**	**301(-146)**

Table 8: Results of ByteTrack+STRAM across the two association stages on the MOT17 validation set

Results obtained from the MOT17 dataset, employing two features across the two association stages, are presented in Table 8. During the initial association stage, the inclusion of IoU+STRAM yields a noteworthy 0.29 increase in MOTA, a substantial 1.46 enhancement in IDF1, and an impressive reduction of 66 IDS instances when contrasted with utilizing IoU alone. Correspondingly, when Re-ID+STRAM is applied, there is a substantial boost of 1.98 in MOTA, an impressive 7.52 surge in IDF1, and a notable decrease of 77 IDS instances compared to utilizing Re-ID exclusively. These findings unequivocally affirm the potency of STRAM in enhancing association performance, given that the bulk of detections are associated in the primary stage.

Moving on to the secondary association stage, despite STRAM receiving uncertain detections due to occlusion and motion blur, it adeptly generates appropriately aligned and complementary features. This is attested by the competitive performance of IoU+STRAM when juxtaposed with IoU alone and the superior performance of Re-ID+STRAM in comparison to Re-ID exclusive in this specific stage.

4.3 More Comparison

4.3.1 Qualitative Comparison

We conducted two qualitative experiments on the MOT17 dataset. One compared the results of ByteTrack [14] with and without RAM, while the other compared our RATracker with the typical rule-based method ByteTrack [14] and deep-learning based method DeepSORT [1].

The results of the first experiment conducted on three MOT17 scenarios, namely MOT17-05, MOT17-09, and MOT17-11, can be observed in Figure 6. The visualization showcases two sets

Fig. 5: Comparison of rule-based, deep learning, and our methods on MOT17 validation set. Notable tracking errors emphasized.

of results: the upper rows depict the tracking results using ByteTrack alone, while the lower rows exhibit the results obtained through ByteTrack+STRAM. It's worth noting that ambient bounding boxes are disabled to enhance the visibility of the targets.

Across all these scenarios, instances of occlusion are prevalent. In the tracking results generated by ByteTrack on its own, there are instances where the issue of identity switching arises. However, this problem is effectively addressed in the tracking results produced by ByteTrack+STRAM. This indicates that the integration of RAMs holds promising potential for ensuring stable tracking performance, particularly in scenarios with occlusions.

The findings from the second experiment are displayed in Figure 5. In scenarios where occlusion is notably prominent, both rule-based and deep learning-based methods face challenges in effectively establishing connections between targets before and after the occlusion. To ensure precise results, it becomes imperative to include additional temporal and spatial regularization on features, as showcased in our proposed approach.

4.3.2 Feature Comparison

We employ t-SNE as a visualization tool for Re-ID features extracted from targets within trajectories

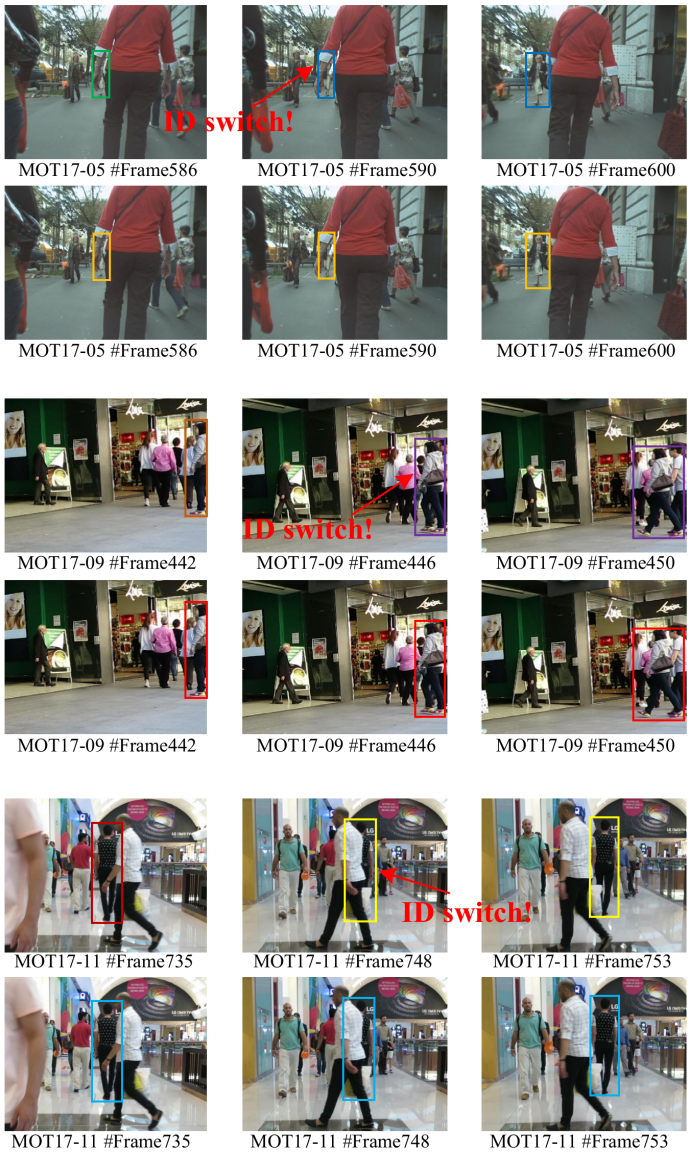


Fig. 6: ByteTrack vs. ByteTrack+STRAM. The upper rows feature ByteTrack's standalone tracking results, while the lower rows display the enhanced results from ByteTrack+STRAM. The ambient bounding boxes are deactivated for prioritizing target clarity.

generated by RATracker. Our experiment focuses on two randomly selected scenes from the MOT17 dataset, where we choose 20 trajectories at random for visualization. The backbone tracker we utilize is JDE [8]. To assess the quality of clustering, we employ the Davies–Bouldin index (DBI) [55], where a lower DBI value indicates more cohesive clusters.

The visualization outcomes are depicted in Figure 7. Each row in the figure represents results from a distinct random scene within MOT17. The colors represent individual trajectories, and points of the same color correspond to associated targets. Our observations reveal a notable distinction: points situated in the right column, generated by employing JDE+STRAM, exhibit greater clustering compared to those in the left column. This outcome is in alignment with the reduced DBI value attained by utilizing JDE+STRAM. Furthermore, the MOTA from association outcomes

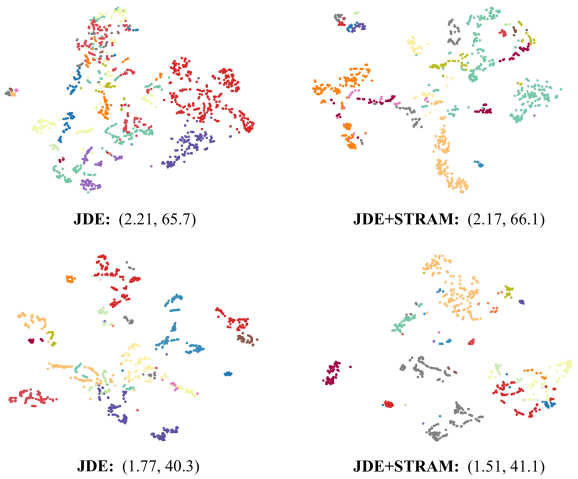


Fig. 7: Visualizing Re-ID features via t-SNE: results on MOT17 scenes. Rows represent random scenes, while colors depict trajectories. Associated targets share the same color. Metrics (DBI↓, MOTA↑) shown in brackets alongside each row.

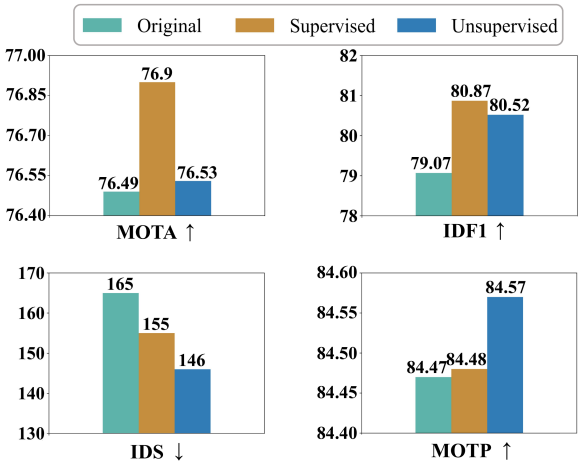


Fig. 8: Results of STRAM trained by annotated boxes (supervised) and by the outputs of pretrained ByteTrack [14] (unsupervised) on the MOT17 validation set.

using JDE+STRAM surpasses that of using JDE alone. This improvement indicates that trajectories with more closely clustered targets bear a stronger resemblance to ground truth data. Evidently, this insight validates our adherence to representation alignment rules, highlighting the likelihood of targets within ground truth trajectories sharing similarities, thus reinforcing the robustness of our approach.

4.3.3 Supervised vs Unsupervised

Our RAM can undergo training using not only annotated data but also utilizing the real-time tracker's output to further enhance the tracker's performance during operation. Essentially, our method involves introducing constraints that ensure both temporal and spatial consistency during the tracking association phase. This is achieved through the contrastive regularization that benefits from the noise-resistant characteristics of the encoder training process, as discussed in the introduction section. This regularization incorporates a certain level of uncertainty, specifically derived from the output of the running tracker.

We validated this concept through an experiment. Initially, we executed the pre-trained ByteTrack once on the MOT17 validation dataset to obtain the initial tracking results. Subsequently, we trained the STRAM using triplets generated from these tracking results. We then evaluated the performance of ByteTrack combined with STRAM (ByteTrack+STRAM) on the same MOT17 validation dataset. For comparative purposes, we also trained another STRAM using triplets based on annotated bounding boxes.

Figure 8 illustrates the performance comparison between STRAMs trained with annotated boxes (supervised) and those trained using the post refinement configuration (unsupervised). Although the unsupervised STRAM exhibits a slight decrease in performance compared to its supervised counterpart, it still surpasses the original tracker across all significant evaluation metrics.

5 Conclusion

In this work, we have investigated two simple yet effective rules aimed at enhancing the MOT performance. These two rules encapsulate the concepts of spatial and temporal consistency among targets, acting as a form of contrasting regularization. Leveraging these rules, we have developed a streamlined encoding module termed RAM. This module serves to produce supplementary association features that complement the existing ones. Experiments conducted on the MOT17,

MOT20 and BDD100K datasets have demonstrated that our proposed RAM is able to enhance the performance of various state-of-the-art trackers. Remarkably, this improvement persists even in scenarios where annotated data is not readily accessible.

Acknowledgments. This research was supported by the National Natural Science Foundation of China under grant number 62002323.

References

[1] Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: IEEE International Conference on Image Processing (ICIP), pp. 3645–3649 (2017). IEEE

[2] Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to track and track to detect. In: IEEE International Conference on Computer Vision (ICCV), pp. 3038–3046 (2017)

[3] Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: IEEE International Conference on Computer Vision (ICCV), pp. 941–951 (2019)

[4] Fremont, D.J., Kim, E., Pant, Y.V., Seshia, S.A., Acharya, A., Bruso, X., Wells, P., Lemke, S., Lu, Q., Mehta, S.: Formal scenario-based testing of autonomous vehicles: From simulation to the real world. In: IEEE International Conference on Intelligent Transportation Systems (ITSC), pp. 1–8 (2020). IEEE

[5] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: IEEE International Conference on Computer Vision (ICCV), pp. 6836–6846 (2021)

[6] Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3202–3211 (2022)

[7] Sharir, G., Noy, A., Zelnik-Manor, L.: An image is worth 16x16 words, what is a video worth? arXiv preprint arXiv:2103.13915 (2021)

[8] Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 107–122 (2020). Springer

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

[10] Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 474–490 (2020). Springer

[11] Peng, J., Wang, C., Wan, F., Wu, Y., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 145–161 (2020). Springer

[12] Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person re-identification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3539–3548 (2017)

[13] Xu, J., Cao, Y., Zhang, Z., Hu, H.: Spatial-temporal relation networks for multi-object tracking. In: IEEE International Conference on Computer Vision (ICCV), pp. 3988–3998 (2019)

[14] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 1–21 (2022). Springer

[15] Welch, G., Bishop, G., et al.: An introduction to the kalman filter (1995)

[16] Baker, S., Matthews, I.: Lucas-kanade 20

years on: A unifying framework. International Journal of Computer Vision **56**, 221–255 (2004)

[17] Cai, J., Xu, M., Li, W., Xiong, Y., Xia, W., Tu, Z., Soatto, S.: Memot: multi-object tracking with memory. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8090–8100 (2022)

[18] Yu, E., Li, Z., Han, S.: Towards discriminative representation: multi-view trajectory contrastive learning for online multi-object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8834–8843 (2022)

[19] Jiang, X., Li, P., Li, Y., Zhen, X.: Graph neural based end-to-end data association framework for online multiple-object tracking. arXiv preprint arXiv:1907.05315 (2019)

[20] Wang, Y., Kitani, K., Weng, X.: Joint object detection and multi-object tracking with graph neural networks. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 13708–13715 (2021). IEEE

[21] Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P.: Transtrack: Multiple object tracking with transformer. arXiv preprint arXiv:2012.15460 (2020)

[22] Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: Motr: End-to-end multiple-object tracking with transformer. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 659–675 (2022). Springer

[23] Dai, P., Feng, Y., Weng, R., Zhang, C.: Joint spatial-temporal and appearance modeling with transformer for multiple object tracking. arXiv preprint arXiv:2205.15495 (2022)

[24] Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)

[25] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Kim, T.-K.: Multiple object tracking: A literature review. Artificial Intelligence **293**, 103448 (2021)

[26] Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter. In: International Conference on Computer Vision (ICCV), pp. 1515–1522 (2009). IEEE

[27] Xing, J., Ai, H., Lao, S.: Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1200–1207 (2009). IEEE

[28] Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 164–173 (2021)

[29] Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics **5**(1), 32–38 (1957)

[30] Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: IEEE International Conference on Image Processing (ICIP), pp. 3464–3468 (2016). IEEE

[31] Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 466–481 (2018)

[32] Wu, J., Cao, J., Song, L., Wang, Y., Yang, M., Yuan, J.: Track to detect and segment: An online multi-object tracker. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12352–12361 (2021)

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

pp. 9686–9696 (2023)

[34] Qin, Z., Zhou, S., Wang, L., Duan, J., Hua, G., Tang, W.: Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17939–17948 (2023)

[35] Chu, P., Wang, J., You, Q., Ling, H., Liu, Z.: Transmot: Spatial-temporal graph transformer for multiple object tracking. In: IEEE Winter Conference on Applications of Computer Vision, pp. 4870–4880 (2023)

[36] Liang, C., Zhang, Z., Zhou, X., Li, B., Zhu, S., Hu, W.: Rethinking the competition between detection and reid in multiobject tracking. IEEE Transactions on Image Processing **31**, 3182–3196 (2022)

[37] Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8844–8854 (2022)

[38] Zhang, Y., Wang, T., Zhang, X.: 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, pp. 22056–22065 (2023)

[39] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9729–9738 (2020)

[40] Henaff, O.: Data-efficient image recognition with contrastive predictive coding. In: International Conference on Machine Learning (ICML), pp. 4182–4192 (2020). PMLR

[41] Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 776–794 (2020). Springer

[42] Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3733–3742 (2018)

[43] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)

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

[45] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2636–2645 (2020)

[46] Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: ECCV Workshops, pp. 17–35 (2016). Springer

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

[48] He, L., Liao, X., Liu, W., Liu, X., Cheng, P., Mei, T.: Fastreid: A pytorch toolbox for general instance re-identification. arXiv preprint arXiv:2006.02631 (2020)

[49] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

[50] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 213–229 (2020). Springer

[51] Yu, E., Li, Z., Han, S., Wang, H.: Relationtrack: Relation-aware multiple object tracking with decoupled representation. IEEE Transactions on Multimedia (2022)

[52] Wang, Q., Zheng, Y., Pan, P., Xu, Y.: Multiple object tracking with correlation learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3876–3886 (2021)

[53] Xu, Y., Ban, Y., Delorme, G., Gan, C., Rus, D., Alameda-Pineda, X.: Transcenter: Transformers with dense queries for multiple-object tracking. arXiv e-prints, 2103 (2021)

[54] Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)

[55] Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence (2), 224–227 (1979)