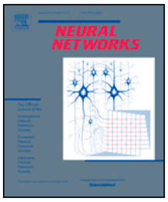
Neural Networks 168 (2023) 363–379

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



STMMOT: Advancing multi-object tracking through spatiotemporal memory networks and multi-scale attention pyramids

Hamza Mukhtar [*], Muhammad Usman Ghani Khan

Department of Computer Science, University of Engineering and Technology Lahore, G.T. Road, Lahore, 54890, Punjab, Pakistan
Intelligent Criminology Lab, National Center of Artificial Intelligence, AlKhawarizmi Institute of Computer Science, University of Engineering and Technology, GT, Road, Lahore, 54890, Punjab, Pakistan

ARTICLE INFO

Keywords:
Multi-object tracking
Transformer
Cross-attention

ABSTRACT

Multi-object Tracking (MOT) is very important in human surveillance, sports analytics, autonomous driving, and cooperative robots. Current MOT methods do not perform well in non-uniform movements, occlusion and appearance–reappearance scenarios. We introduce a comprehensive MOT method that seamlessly merges object detection and identity linkage within an end-to-end trainable framework, designed with the capability to maintain object links over a long period of time. Our proposed model, named STMMOT, is architectured around 4 key modules: (1) Candidate proposal creation network, generates object proposals via vision-Transformer encoder–decoder architecture; (2) Scale variant pyramid, progressive pyramid structure to learn the self-scale and cross-scale similarities in multi-scale feature maps; (3) Spatio-temporal memory encoder, extracting the essential information from the memory associated with each object under tracking; and (4) Spatio-temporal memory decoder, simultaneously resolving the tasks of object detection and identity association for MOT. Our system leverages a robust spatio-temporal memory module that retains extensive historical object state observations and effectively encodes them using an attention-based aggregator. The uniqueness of STMMOT resides in representing objects as dynamic query embeddings that are updated continuously, which enables the prediction of object states with an attention mechanism and eradicates the need for post-processing. Experimental results show that STMMOT archives scores of 79.8 and 78.4 for IDF1, 79.3 and 74.1 for MOTA, 73.2 and 69.0 for HOTA, 61.2 and 61.5 for AssA, and maintained an ID switch count of 1529 and 1264 on MOT17 and MOT20, respectively. When evaluated on MOT20, it scored 78.4 in IDF1, 74.1 in MOTA, 69.0 in HOTA, and 61.5 in AssA, and kept the ID switch count to 1264. Compared with the previous best TransMOT, STMMOT achieves around a 4.58% and 4.25% increase in IDF1, and ID switching reduction to 5.79% and 21.05% on MOT17 and MOT20, respectively.

1. Introduction

In computer vision, tracking multiple objects (Bergmann, Meinhardt, & Leal-Taixe, 2019; Du et al., 2023; Zhang, Wang, Wang, Zeng, & Liu, 2021) consistently ranks as a critical and demanding task in various fields such as self-driving cars (Shi, Chen, & Kim, 2023), sports analytics (Cioppa et al., 2022), and human activity recognition (Qiu et al., 2022). The primary objective is to continuously track the locations of multiple targets within a sequence of frames while preserving each target's identity and predicting their respective movement trajectories within the visual scene (Chen, Wang, Zhao, Lv, & Niu, 2022). MOT techniques can be categorized into two main types: online (Bergmann et al., 2019; Zhang et al., 2021) and offline technique (Li, Wang, & Gong, 2023). Offline MOT leverages global data from frame sequences, while Online MOT focuses on identifying a set of objects and tracing corresponding trajectories over frames to ensure that each object maintains consistent identification throughout the sequence. Online methods have increased interest due to their relevance and alignment with real-world use cases such as football analytics (Cioppa et al., 2022), table tennis (Voeikov, Falaleev, & Baikulov, 2020), and human activity recognition (Qiu et al., 2022).

Traditional online MOT approaches typically involve two distinct phases: (1) the object detection phase, which identifies and localizes each desired object instance in each frame (Kiefer, Quan, & Zell, 2023; Zhang et al., 2021); and (2) the object re-identification (ReID) as the identity association phase, which connects detected objects instances over time (Du et al., 2023). This process involves emulating the state changes of the objects being tracked and subsequently resolving the identity-matching problem between these tracked objects and

[*] Corresponding author at: Department of Computer Science, University of Engineering and Technology Lahore, G.T. Road, Lahore, 54890, Punjab, Pakistan
E-mail addresses: hamza.mukhtar@kics.edu.pk (H. Mukhtar), Usman.ghani@uet.edu.pk (M.U.G. Khan).

https://doi.org/10.1016/j.neunet.2023.09.047
Received 9 June 2023; Received in revised form 14 September 2023; Accepted 27 September 2023
Available online 29 September 2023

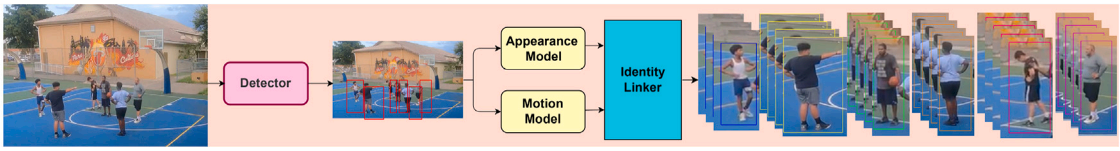


Fig. 1. Concept of traditional online MOT where objects are detected using an object detector and then appearance and motion models are used in post-detection association linker for the re-identification.

the results derived from the detection process. It essentially ensures continuity and coherence in the tracking of objects over time, thus contributing significantly to the overall effectiveness of the multi-object tracking system. The general concept of online MOT is shown in Fig. 1. Most online MOT approaches (Du et al., 2023; Xu, Cao, Zhang, & Hu, 2019; Yu et al., 2016) initially acquire bounding boxes for all objects in the current frame using a detector, then extract ReID features for each bounding box, and subsequently match candidate boxes to existing trajectories based on these ReID features. This process requires extracting ReID features for every box, resulting in considerable computational overhead. To address this issue, the Joint Detection and Embedding (JDE) methods (Bergmann et al., 2019; Feichtenhofer, Pinz, & Zisserman, 2017; Wang, Zheng, Liu, Li, & Wang, 2020; Zhang, Cheng, Zhu, Lin, & Dai, 2018; Zhang et al., 2021; Zhou, Koltun, & Krähenbühl, 2020) were introduced which combine the object detection and ReID feature extraction modules into a singular network capable of simultaneously predicting an object's location and extracting its ReID features. Merging object detection and ReID tasks directly within a single model can result in a competitive relationship between the two tasks, potentially decreasing tracking accuracy (Hyun, Kang, Wee, & Yeung, 2023; Yang, Jiang, Wen, & Fan, 2023). The detection task aims for objects within the same category to share similar semantics, allowing the network to minimize intra-class variation. However, the ReID task focuses on identifying disparities among distinct objects within the same class, expecting the network to accentuate these differences. The incongruity in optimization objectives for the two tasks obstructs the network's ability to optimize both tasks simultaneously (Sun et al., 2020). While recent research (Lu, Jiang, Liu, & Mu, 2023; Meinhardt, Kirillov, Leal-Taixe, & Feichtenhofer, 2022) indicates that merging these two stages could offer advantages, this integration often inadvertently simplifies the identity association module's ability to model the temporal evolution of objects.

The recent advancements in computer vision involving transformer (Vaswani et al., 2017) for tasks like object detection (Carion et al., 2020; Dosovitskiy et al., 2020; Lin et al., 2020; Zhu et al., 2020), person re-identification (PRe-ID) (Liu, Mu, Lu, Zhang, & Tian, 2023; Rao & Miao, 2023) semantic segmentation (Zheng et al., 2021) and image super-resolution (Lin et al., 2022) demonstrate the advantages of attention-based mechanisms. Transformers excel at concurrently modelling dependencies among various input components and making holistic decisions. These benefits align seamlessly with the inherent challenges of MOT, as current techniques often struggle with accurately modelling interactions between objects, particularly in densely populated scenes. In this study, we introduce a Transformer-based MOT model, STMMOT, which carries out object detection and identity association within a unified framework in an online fashion. At the core of STMMOT design is the creation of an extensive spatiotemporal memory to store previous state observations of tracked objects. This memory is dynamically encoded at every time step by referencing pertinent information, allowing for a more precise approximation of object states for the identity association. The comprehensive representation of tracked objects derived from the spatio-temporal memory enables the resolution of object detection and identity association tasks within a consolidated decoding module. This module directly outputs tracked object instances that reappear in the most recent frame, as well as new object instances encountered for the first time. The illustration STMMOT concept is depicted in Fig. 2.

At every time step, STMMOT operates 4 primary components: (1) a candidate proposal creation module that detects objects from the frame sequence and creates proposals from the input image to form object embedding vectors of feature maps; (2) a multi-scale attention pyramid module to inject the robustness in scale variance proposal embeddings that receives the proposal embedding vectors as input and process through a progressive pyramind to learn self-scale and cross-scale similarity at the object tracklet level, utilizing an attention mask; (3) a memory encoder that transforms the spatio-temporal memory corresponding to each object into a vector called the tracklet embedding; and (4) a memory decoding module that takes the object proposal and tracklet embeddings, simultaneously resolving object detection and identity association tasks for MOT. The candidate creation module to detect objects is realized through a Transformer detector based on the encoder–decoder network (Sun et al., 2020; Zhu et al., 2020), generating a collection of embedding vectors called the proposal embedding, with each vector against a desired object proposal. The memory encoder initially separates the spatio-temporal memory for each proposal object into short-term and long-term memories and then aggregates those memories into joint embedding vectors using a cross-attention interaction between short-term and long-term memories (Chen, Fan, & Panda, 2021; Vaswani et al., 2017). Then, both vectors subsequently interact through a self-attention mechanism, resulting in the tracklet embedding for the tracked object.

Short-term memory block takes previous state of each tracklet and assembles the embeddings of neighbouring frames to smooth out the noises, while long-term memory block utilizes the state history longer than short-term memory for extracting the relevant features in the temporal window. Short-term memory concentrates on proximate temporal sequences. Its primary objective is to capture immediate variations within the proposal embeddings, thereby refining the subsequent track embeddings ensuring that short-lived patterns are taken into account. The long-term memory focuses on an extensive temporal retrospective, accumulating data from more prolonged intervals. The Dynamic Memory Aggregation Tokens (DMAT) are integral in this phase, offering a continuously updating historical context to each sequence. The intention is to ensure the system's interpretation is a synthesis of both recent and historically proposal embeddings. Both short-term and long-term memory blocks are implemented through multi-head cross-attention, where history states are fed as key and value to the attention mechanism. The outputs of the short- and long-term memory blocks are then fused through a self-attention block which outputs the tracklet embedding against each proposal embedding. The proposal and tracklet embeddings, along with the original image features, are then input into the memory decoder that determines the spatial–temporal location and visibility of the tracked object in the current frame for each tracklet embedding. Against each proposal embedding vector, it predicts whether the desired object proposal represents a new object, a tracklet object, or just a background region. The whole STMMOT pipeline undergoes end-to-end training on frame sequences where each desired object is annotated with bounding boxes and identity. During the inference process, STMMOT predicts the localization and identity outputs in a single inference run of the model at each time step, eliminating the need for additional optimization (Chu, Wang, You, Ling, & Liu, 2023; Hosseini-Asl, McCann, Wu, Yavuz, & Socher, 2020) or post-processing (Sun et al., 2020; Zhang et al., 2021).

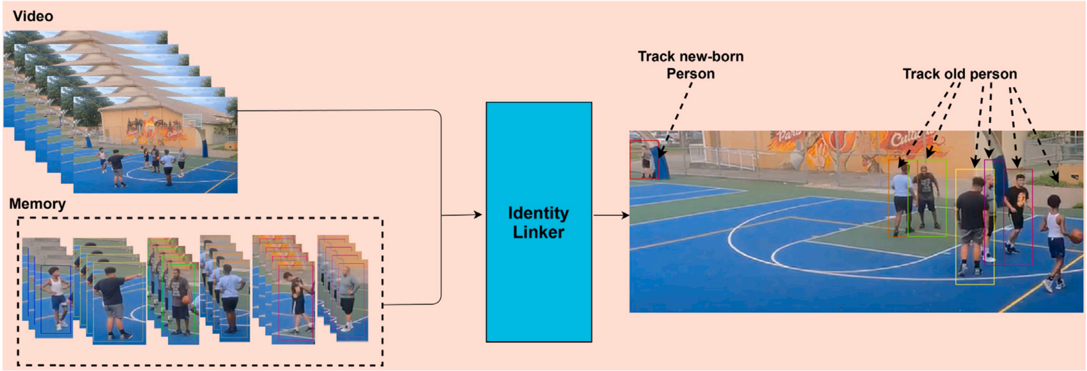


Fig. 2. The concept of STMMOT is depicted through the use of a dynamic spatio-temporal memory buffer that keeps the object states of all tracked objects and updates them over time.

We evaluate STMMOT on the MOT (Li, Wang et al., 2023; Xu et al., 2019) benchmark datasets for pedestrian tracking. The experimental outcomes indicate that STMMOT attains superior performance among other MOT algorithms featuring an in-network identity association linker and remains competitive with those employing a post-network identify association. Notably, STMMOT surpasses other vision Transformer-based MOT in the identity linkage task. In-depth ablation experiments provide additional validation of different components in STMMOT design.

Our contributions can be outlined as follows:

1. We introduce a fully end-to-end MOT network, STMMOT, which jointly learns object detection and identity association.
2. We introduce a spatiotemporal memory that concurrently models the temporal and spatial contexts of the target object, effectively achieving robust object association across time and target-specific discriminative capabilities for robust MOT.
3. We formulate a multi-scale attention pyramid module that efficiently leverages cross-scale similarities of the same object tracklets over time to ensure robustness in scale variation.
4. Comprehensive experiments conducted on MOT17 (Dendorfer et al., 2021), and MOT20 (Dendorfer et al., 2020) demonstrate that our approach significantly improves the identity association capabilities of ReID features, making it highly competitive compared to existing algorithms.

The rest of the article is organized as follows: Section 2 reviews the different types of previous MOT methods such as tracking-by-detection, joint-detection-tracking, transformer-tracking and memory-tracking, and identifies the research gap. Section 3 presents our proposed STMMOT, while training, validation and performance comparison on public benchmark datasets are carried out in Section 4, followed by Section 5, which summarizes the research and provides the future research direction.

2. Related work

This section discuss various MOT methods that are roughly classified into four broader categories, including tracking-by-detection, joint-detection-tracking, transformer-tracking and memory-tracking.

2.1. Tracking-by-detection

As object detection techniques (Kiefer et al., 2023; Shi et al., 2023) have advanced, numerous MOT approaches (Bewley, Ge, Ott, Ramos, & Upcroft, 2016; Du et al., 2023) have enhanced the tracking-by-detection paradigm. These methods primarily concentrate on tracking, that is, linking the same objects across different frames, as high-performance detectors provide the bounding boxes of the objects. Initial research employed motion data from objects to forecast their positions in subsequent frames using methods like the Kalman Filter (Zhang et al., 2022) and particle filters (Nenavath, Ashwini, Jatoth, & Mirjalili, 2022). Sort (Bewley et al., 2016) pioneered the use of the Kalman Filter for predicting the location of all potential bounding boxes in subsequent frames. It is then calculated the Intersection over Union (IOU) between the previous bounding box and predicted boxes in the next frame, ultimately matching them utilizing the Hungarian algorithm. StrongSort (Du et al., 2023) further enhances the development by incorporating appearance information of objects being tracked, extracting visual features of objects through passing the cropped object region-of-interest to a separate convolutional neural network (CNN), while Yuan et al. (2021) also integrates human poses. The LSTM (Babaee, Li, & Rigoll, 2018) is employed to predict the object location in the current frame using prior frame data. POI (Yu et al., 2016) determines the affinity value to assign the identity to a detected object by measuring the distance between the visual feature maps of the detected objects. STRN (Xu et al., 2019) learns to identify similarities between tracked objects by encoding various cues such as visual, motion, and location over long span through maintaining spatial–temporal relationships. While the tracking-by-detection pipeline delivers remarkable performance, its model complexity and computational demands are suboptimal. Ideal filters in these methods maintain tracking states with historical information for new frame predictions. Although optimal states can be determined in linear-Gaussian cases, estimating them in non-linear and non-Gaussian scenarios, such as occlusions in visual multi-object tracking, is challenging due to finite-dimensional state representation cons.

2.2. Joint-detection-tracking

The joint-detection-and-tracking pipeline strives to accomplish detection and tracking concurrently in a single stage. D&T (Feichtenhofer et al., 2017) an end-to-end and jointly trained spatio-temporal MOT, introduces a siamese architecture for current frame object detection and cross-frame co-occurrences capturing for object tracking. More recently, Tracktor (Bergmann et al., 2019) employs prior frame tracking boxes as regression. Integrated-Detection (Zhang et al., 2018) enhances detection by merging detection boxes in the region proposals, applying box offset regression to yield tracking coordinates in the current frame, thus replacing the identity association stage. Centertrack (Zhou et al., 2020) leverages prior frame feature embeddings as supplementary input for estimating the motion direction of the object's central point between the part and current frames. Despite the incorporation of detection and tracking details within a unified network, effectively dealing with intricate scenarios continues to pose a significant challenge. Consequently, the JDE (Wang et al., 2020) suggests incorporating an embedding layer to the detector for ReID feature extraction and appearance integration where detection and identity association share the same features to enable the model to produce both detection outcomes and associated ReID features jointly, thus enhancing MOT accuracy and speed. FairMOT (Zhang et al., 2021) a multi-task MOT network, detects objects and extract visual embedding from a shared backbone to improve association accuracy. While this joint end-to-end MOT tries an equilibrium between accuracy and efficiency, the competition between object detection and ReID persists, negatively impacting network performance.

The competition between ReID and object detection is a phenomenon that occurs when joint-detection-tracking methods try to optimize both tasks simultaneously (Wang et al., 2020). Competition arises because of the inherent differences and relations between detection and ReID, such as: Detection focuses on locating and classifying objects, while ReID focuses on distinguishing and matching objects. Detection requires high sensitivity and low specificity, while ReID requires low sensitivity and high specificity (Liang, Li, Wang, Tan, & Luo, 2022). Detection is affected by the scale, pose, occlusion, and illumination of objects, while ReID is affected by the appearance, style, and identity of objects (Zhang et al., 2021; Zhou et al., 2020). The impact mechanism of the competition is that it can degrade the performance of both tasks, leading to inferior results compared with existing two-stage methods (Liang, Zhang et al., 2022; Zhang et al., 2018). For example: If the detection is too sensitive, it may produce more false positives or negatives, which can increase the difficulty of ReID and cause detection errors or identity switches. If the ReID is too strict, it may fail to match the same object across different frames or views, which can reduce the recall of detection and cause identity switches or track fragmentation (He et al., 2022; Liang, Zhang et al., 2022). Moreover, joint detection and tracking models boost runtime but compromise tracking recovery after occlusion and are unable to re-establish connections with objects missing for long-term periods (Cai et al., 2022).

To overcome the above issues, STMMOT employs a large spatiotemporal memory to store previous observations of tracked objects. This memory is continuously updated with relevant data, enhancing object state approximation for association tasks. By extracting detailed representations from this memory, object detection and association are combined in a single decoding module. This module identifies both previously tracked objects in the latest frame and new objects appearing for the first time. By unifying object detection and association tasks within a singular decoding module, STMMOT streamlines the process, addressing the inherent competition between detection and ReID. Consequently, this design aids in accurately identifying recurring objects in new frames while also adeptly identifying new object instances.

2.3. Transformer-tracking

The transformer architecture (Dosovitskiy et al., 2020; Lin et al., 2022) has demonstrated its potential and impact on vision tasks. As a unique query-key mechanism, the transformer primarily depends on the attention mechanism to process the extracted deep features. Initially exhibiting exceptional efficiency in natural language processing (Vaswani et al., 2017), it later transitioned to visual perception tasks (Carion et al., 2020), achieving notable success. The transformer's elegant structure and impressive performance have attracted the vision community. It has exhibited significant potential in detection (Carion et al., 2020; Lin et al., 2020, 2022), segmentation (Zheng et al., 2021) and 3D data processing (Pang et al., 2022). Transformers have only recently been employed in MOT, before transformers, a few attention-based modules were introduced for MOT applications. Specifically, Guo, Wang, Wang, and Tao (2021) suggests a target-aware and distractor-aware attention mechanism generates more dependable visual embeddings, which also aids in suppressing old objects. Following the success of transformers in detection, two concurrent works, TrackFormer (Meinhardt et al., 2022) and MOTR (Zeng et al., 2022), apply vision-transformers based on the DETR framework (Zhu et al., 2020) for MOT task. These methods concurrently execute detection and identity association by concatenating the object and autoregressive track queries as inputs to the Transformer decoder in the subsequent step. Conversely, TransCenter (Xu et al., 2022) and TransTrack (Sun et al., 2020) utilize Transformers solely as feature extractors and recurrently pass track object features to learn each object's joint embedding aggressively.

For occluded Re-ID, dual-branch Transformer (Lu et al., 2023) has developed comprising a global branch for global feature extraction and a local branch featuring Selective Token Attention for local feature extraction using multi-headed self-attention. Another dual-branch MOT JDE (Tsai, Shen, & Nisar, 2023) includes a Patch-Expanding mechanism to boost object detection and identity association in crowded scenarios by enhancing feature maps spatially using CNN and Einops Notation rearrangement. These dual-branch design lacks end-to-end learning and the performance of identity association depends on the first branch. Conversely, an end-to-end transformer-based occluded person Re-ID model (Liu et al., 2023) leverages a multi-headed self-attention to learn the distribution of common non-occluded target person regions and generates accurate crops using the Minimized Character-box Proposal method. TranSG (Rao & Miao, 2023) utilizes a structure-trajectory mechanism to capture both relations and critical spatial–temporal semantics from skeleton graphs to get the fine-grained representations of body joints. The P3AFormer (Zhao, Wu, Zhuang, Li, & Jia, 2022) utilizes flow information to guide the propagation of pixel-wise features, generate multi-scale feature embeddings through a meta-architecture and pixel-wise identity assignment is used to associate connections between the same object appearances in a sequence of frames. Another approach for unsupervised transformer Re-ID (Li, Wang et al., 2023) is proposed where local tokens are divided into multiple parts to create the part-level feature embeddings, while global tokens are averaged to produce a global embedding. The DC-Former (Li, Zou et al., 2023) further improves the representation discrimination by dividing the embedding into multiple diverse and compact subspaces. TransMOT (Chu et al., 2023) employs a graphical transformer to model spatio-temporal interactions between objects by representing object trajectories as sparse weighted graphs. Although the aforementioned work focuses on representing object states using dynamic embeddings, it lacks adequate modelling of long-term spatio-temporal information and adaptive feature fusion methods.

2.4. Memory-tracking

Memory networks (Sukhbaatar, Weston, Fergus, et al., 2015; Weston, Chopra, & Bordes, 2014) have been widely applied in NLP for understanding the long-term dependencies of temporal reasoning tasks such as textual question-answering (Kumar et al., 2016) and task-oriented dialogue systems (Hosseini-Asl et al., 2020), and video analysis (Xu et al., 2021) for storing and accessing time-indexed features to better remember the historical scene information. Memory networks are also being experimented with in MOT which requires both spatio-temporal information, such as MemTrack (Yang & Chan, 2018) and STMTrack (Li, Liu, Yao, & Chang, 2021), however, these solely utilize the space–time memory network for acquiring object appearance information, neglecting memory frame relationships and embedded temporal context, thus not fully exploiting the network's potential. MOT requires a large spatiotemporal memory is required for robust object association across time. Recent work, such as MeMOT (Cai et al., 2022) and LGST (Xie et al., 2021), build a spatio-temporal memory bank for the transformer to encode the track queries and decodes for both detection and association, compared to TrackFormer (Meinhardt et al., 2022), these models have tried to enhance the performance by using track queries of multiple frames, while SAMN (Xie et al., 2021) performs object segmentation to extracts the spatio-temporal visual features and learns the identify association with spatial-appearance memory networks that utilize spatiotemporal non-local similarity to

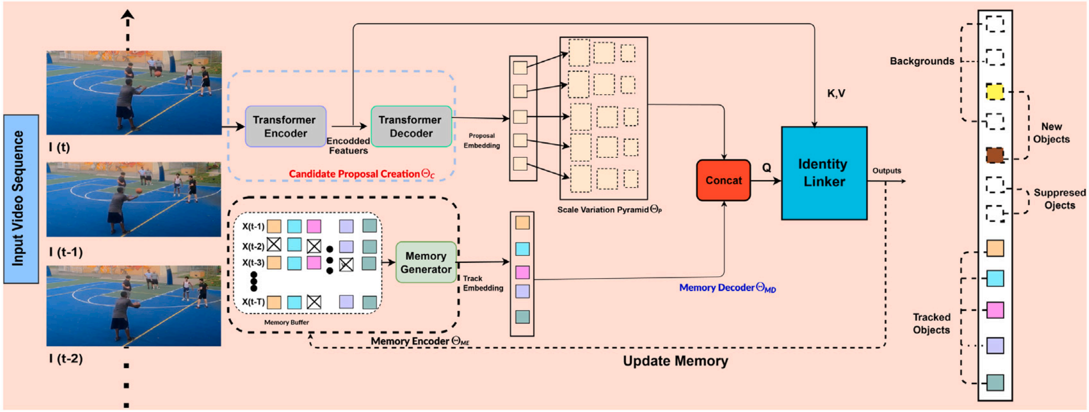


Fig. 3. STMMOT system operates with four primary components: (1) a candidate proposal generation module Θ_C that generates object proposals for the present frame, (2) Scale Variant Progressive Pyramid module Θ_P that constructs a hierarchical feature pyramid with different levels of resolution to enables the extraction of features from candidates proposal embeddings at self and cross scales, (3) a memory encoder Θ_{ME} that extracts essential representation associated with each object under tracking and (4) a memory decoder module Θ_{MD} that concurrently addresses the object localization and identity linkage. STMMOT preserves a memory buffer that holds long-term object states, with an encoding-decoding mechanism helping to associate objects over time. Each candidate proposal object is classified as new, tracked, or part of the background.

propagate segmentation masks for effectively capturing long-range appearance variations. Nonetheless, these studies are less effective in crowded scenes, highly non-linear motion and when occlusion time becomes longer. We suggest employing a large spatiotemporal memory to facilitate robust object associations over a long time and highly occluded scenes for MOT.

3. Methodology

The ideal outcome for MOT is a complete and accurately ordered set of objects for every frame within a video. Given a series of frame sequences denoted as $I = \{I_0, I_1, I_2, \dots .I_T\}$, MOT aims to identify and track N objects locations while maintaining their trajectories $T = \{T_0, T_1, T_2, \dots .T_N\}$ through real-time processing. we introduce STMMOT which simultaneously learns object detection and identity association. Unlike proposed MOT techniques (Du et al., 2023; Tsai et al., 2023; Xu et al., 2022; Zhao et al., 2022) that only transfer tracked object states between consecutive frames, our method incorporates a spatio-temporal memory block for storing long-term dependencies of tracking objects. Additionally, we employ a memory encoder–decoder mechanism that effectively extracts relevant representation for associating the same objects in a sequence of frames even after an extended time period.

As illustrated in Fig. 3, STMMOT is comprised of four primary modules: (1) a frame-level candidate proposal creation module Θ_C, which detects objects and creates the object proposals against each frame for the current time step I_t, (2) a scale variant progressive pyramid module Θ_P to learn the cross-scale similarities for handling the varying sizes and scales within a frame sequence, (3) a track-level memory encoding module Θ_{ME}, responsible for aggregating associated object embeddings, and (4) a memory decoder Θ_{MD}, which associates detected proposal object candidates with already tracked objects. At time step t, Θ_C creates N^t_{can} candidate object proposals, denoted as proposal embeddings $Q^t_{can} \in R^{N^t_{can}\times d}$, using a Transformer encoder–decoder network. Θ_{ME} forms the signal compact representation from the historical states of each tracked object dynamically, referred to as tracklet embeddings $Q^t_{traklet} \in R^t_{traklet} \times d$. Taking encoded feature map of the object as a query with $[Q^t_{can}, Q^t_{tracklet}]$, Θ_{MD} determines and measures the inter-object linkage, and updates the embeddings as $[\widehat{Q}^t_{can}, \widehat{Q}^t_{tracklet}]$. The bounding box coordinates $[B^t_{can}, B^t_{tracklet}]$ and confidence scores $[C^t_{can}, C^t_{tracklet}]$ of new and tracked objects are then estimated based on these resultant embeddings. The location vector represents the spatial position of each object in the frame sequence, while the confidence scores represent the model's certainty regarding the presence of those objects. Lastly, the locations and states of already tracked objects are utilized to update their trajectory and memory. Trajectory refers to the path that the object has followed over time, while the memory buffer encapsulates the object's historical states, including its past locations, sizes, and appearance features. The historical data assists in maintaining consistency in object tracking, especially in challenging scenarios like occlusions or the temporary disappearance of the object from the frame. Newly detected objects are added in T, and their initial state is stored in the memory buffer. This action prepares the system to monitor this object in subsequent frames, effectively expanding the model's tracking scope.

3.1. Candidate proposal generation

The candidate proposal network (CPN) Θ_c utilizes a vision Transformer encoder–decoder design (Zhu et al., 2020) and a CNN as feature extractor to create N^t_{can} object proposals. These proposals either initiate the tracking of new objects in the current frame or update already tracked objects by providing new spatial and identity information. The Θ_c transformer encoder processes a sequentiality feature map $f^t_0 \in R^{D\times HW}$, derived from the current step input frame I_t using a CNN backbone where $I_t \in R^{H\times W\times C}$ is fed to a CNN that produces an information-rich but low-resolution representation. This sequential processing enables the model to account for temporal changes in object positions, shapes, and appearance attributes, which is crucial for tracking objects over consecutive video frames. The Encoder network expects to receive a sequence as input, so, spatial dimensions of the feature maps f^t_0 are transposed into one dimension to form the D × HW shape feature map. The image features are encoded as $f^t_1 \in R^{d\times HW}$ using stacked Transformer encoder layers. The decoder Θ_c takes the encoded image feature map f^t_1 and empty object queries (portrayed as learnable embeddings) and generates the final proposal embeddings $Q^t_{can} \in R^{N^t\times d}_{tracklet}$. Each decoder layer consists of multi-head self-attention (MSA) and a feed-forward network (FFN). As the transformer is permutation-invariant, Each element in f^t_0 is enhanced with a distinct positional encoding that is included in each MSA layer to signify its spatial location. The objectness scores and bounding box coordinates for each proposal are predicted using Q^t_{can}. Our decoder follows the transformer (Zhu et al., 2020), transforming N embeddings using self-attention and cross-attention. Unlike the transformer (Zeng et al., 2022), our model takes a different approach by simultaneously decoding N objects at each decoder layer. To ensure unique outcomes, the N input embeddings, referred to as object queries, need to be distinct. These object queries are positional encodings that are learned and added to each attention layer's input, similar to the encoder. The decoder transforms the object queries into output embeddings,

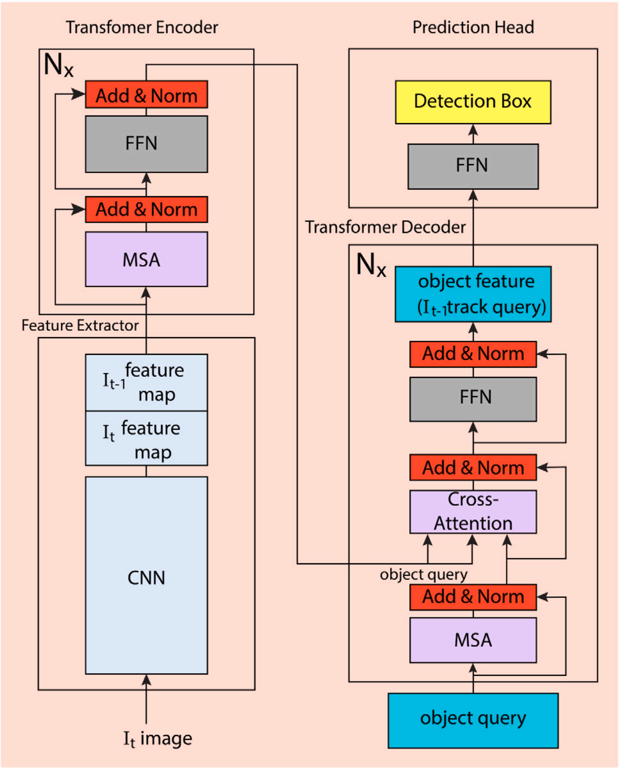


Fig. 4. The architecture of the candidate proposal network based on Vision-Transformer.

which are independently decoded into box coordinates and class labels using a feed-forward network (FFN). This process generates N's final predictions. By employing self-attention and cross-attention attention on these embeddings, our model reasons globally about all objects, considering pairwise relations between them. It can also utilize the entire frame as context. The architecture of the CPN is illustrated in Fig. 4.

3.2. Scale variant pyramid

The proposed Scale variant pyramid (SVP) detailed architecture is illustrated in Fig. 5. The SVP module accepts input features $X_{[0:N]}$ and requires $N+1$ executions. It uses a progressive structure to learn similarities between self-scale and multi-scale feature maps. In the pyramid structure, strided convolutions are used to downscale the feature maps from the upper layer by a factor of 2 to obtain feature maps at the current layer. Let M be the pyramid layer in the SVP module, and N be the number of progressive feature transfer layers (PFTLs) at each layer to progressively extract self-similarity, as shown in Fig. 6. At the mth layer, the nth PFTL takes the first feature X_0^m as input and the output of the previous block $(X_i^{n-1}, i \in [0, N])$. For the first PFTL, the inputs are X_0^m and X_i^m. Taking inspiration from TDAN (Tian, Zhang, Fu, & Xu, 2020) and DCN (Chan, Wang, Yu, Dong, & Loy, 2021), deformable convolution is applied in the PFTL. This process can be expressed as:

$$(X_i^D)^{m,n} = X_{Dconv}(X_0^m, X_i^{n-1})^m \tag{1}$$

where $X_{Dconv}(\cdot)$ denotes deformable convolution, and $(X_i^D)^{m,n}$ represents the output of the nth block's deformable convolution at the mth layer. Deformable convolution helps in capturing more complex patterns by introducing learnable offsets to the convolutional kernel, which adds an extra level of adaptability in handling geometric transformations. These learned offsets are predicted based on the input data, acting as an additional layer of complexity to the convolution operation. The advantage of this approach is that it allows for non-linear geometric transformations, providing more robustness to changes in object size, shape, and orientation. The learned offsets can adapt to the specificities of the input, focusing on areas of the image that contain the most relevant information, making the feature extraction process more efficient and accurate:

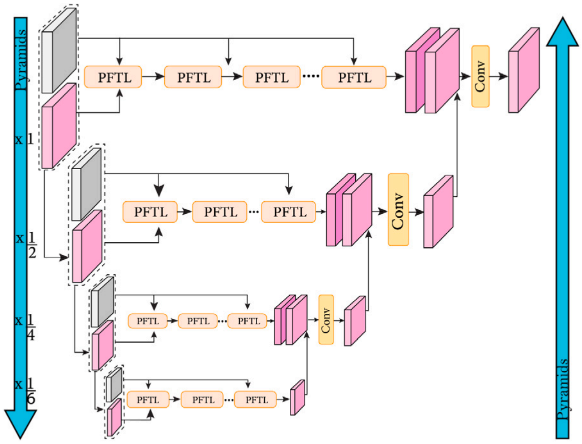


Fig. 5. Architecture of Scale variant pyramid.

$$(\triangle\ P_i)^{m,n} = X_C(X_0^m \parallel (X_i^{n-1})^m) \tag{2}$$

where $(\triangle\ P_i)^{m,n}$ denotes the learned offset of the nth block at the mth layer, $||$ represents channel-wise concatenation, and $X_C\ (\cdot)$ indicates the convolution operation. Learning the offset is a vital step as it allows for subtle adjustments of the feature map locations to better accommodate the object's characteristics. Channel-wise concatenation combines feature maps from different layers so that the subsequent layers can use the combined features. Next, we compute the feature-level mask of the nth block at the mth layer $(Mask_i)^{m,n}$ which forces the PFTL to concentrate on the most correlated feature information:

$$(Mask_i)^{m,n} = Softmax(X_C(X_0^m) - X_c((X_i^{n-1})^m)) \tag{3}$$

The motion attention mask is then element-wise multiplied with the output of the deformable convolution. This product is then sent through another convolutional layer, yielding a feature set that encapsulates the residual information within this block. The nth PFTL's output feature at the mth layer is extracted by adding the residual information to the first feature:

$$(X_i^n)^m = X_0^m + X_C(X_0^m \parallel (Mask_i)^{m,n} \otimes (X_i^D)^{m,n}) \tag{4}$$

where ⊗ denotes element-wise multiplication. Lastly, the SVP module's output feature at the mth layer $(X_i^{SVP})^m$ is defined as:

$$(X_i^{SVP})^m = F_C(Us(((X_i^{SVP})^{m+1})^{\xi} \parallel (X_i^N)^m)) \tag{5}$$

where $(X_i^N)^m$ denotes the feature generated after the N block of PFTL at the mth layer. US(·) is the scaling of the feature map by hyperparameters, and bilinear interpolation is used for the resizing. The proposed SVP builds a three-layer pyramidical architecture (M = 3) and learns multi-scale dependencies progressively coarse-to-fine fashion.

3.3. Spatio-temporal memory

MOT requires learning long-term spatio-temporal dependencies in occluded and non-uniform motion scenarios. This involves preserving the visual-motion features while capturing cross-frame similarities that co-occur both in the visual and spatial domains, which are crucial for effective tracking. Thus, we create a spatio-temporal memory buffer $B \in R^{N \times T \times d}$ to store all N tracked objects' historical states. The buffer is a three-dimensional structure that can hold N objects (each having a unique history) tracked over T time steps, each having a state in d-dimensions. It captures the spatial location and temporal progression of each object, i.e., where the object is and how it moves over time. This buffer keeps a maximum of N_{max} objects and up to T_{max} time steps for each tracked object. The structure of this memory buffer is built using a first-in-first-out (FIFO) data structure which ensures that the memory buffer does not exceed its capacity and always contains the most recent information. So, the oldest tracked object is first to be suppressed when the buffer reaches its maximum capacity. The maximum capacity is set by two parameters: N_{max} (the maximum number of tracked objects) and T_{max} (the maximum number of time steps for each tracked object). At time step t, the memory represents the states of $N^{t-1}_{tracklet}$ active objects in the previous T frames, denoted as:

$$X^{t-1-T:t-1} = \{X_n^{t-1-T:t-1}\}_{n=1:N^{t-1}_{tracklet}} \tag{6}$$

Here, $t-1-T : t-1$ denotes the time range where $t-1$ is the previous frame, $t-1-T$ indicates the frame T steps before the previous frame. $X_n^{t-1-T:t-1}$ represents the nth object's states spanning from T frames before the last frame to just the last frame and its state is padded with 0s if the object is not present in frame I_t. Hence, Eq. (6) expresses that the state information from the last T frames includes the states of each active object n from the first to the Once the $N^{t-1}_{tracklet}$ active object. T exceeds T_{max}, the initial state X_n^{t-1-T} of each tracklet is removed from the memory. The values of N_{max} and T_{max} are chosen based on the specifics of the application and hardware limitations. N_{max} should be large enough to handle the desired number of objects in a frame sequence, e.g., 200 or 400, while T_{max} needs to be large enough that be reasonable for handling the occlusion and encompass the length of time step (e.g. frame sequence) the objects need to be tracked, such as 25 or 50 frames. To remain within the hardware limitation, we set the T_{max} is 30 frames. Overall, the model keeps track of the states of active objects over a series of past frames. If an object is not present in a particular frame, its state for that frame is recorded as 0. If the memory goes back further than the system's maximum allowable time window, the earliest recorded state is discarded.

3.4. Memory encoder

Fig. 7 illustrates the encoding of candidate memory and extraction of track embedding via three attention blocks. Firstly, a short-term memory block, B_S, assembling embeddings from subsequent frames, aiming to reduce noise. The primary objective of B_S is to significantly mitigate the potential noise inherent in the data, ensuring the extraction of clearer, more concise embeddings. By focusing on recent temporal data, this block offers the ability to filter out short-lived, non-persistent fluctuations, thus emphasizing more consistent and repetitive patterns. Secondly, a long-term memory block, B_L, is utilized to derive pertinent features within the timespan encompassed by the memory. Its primary role is to mine salient features that lie within the temporal span that the memory covers. Rather than solely focusing on immediate or recent states, this block extends its attention to a broader temporal horizon, thereby capturing more extended dependencies and relationships inherent within the proposal embeddings. This facilitates a more holistic view of the memory, drawing connections between temporally distant data points. Lastly, a fusion block, B_f, integrates the embeddings generated from both the short and long-term branches. Its main function is to coherently aggregate the embeddings synthesized from both the short-term and long-term blocks. Through this fusion process, the model ensures that insights from both immediate and distant timeframes are considered, fostering a more comprehensive and nuanced understanding of the proposal embeddings.

For the short-term block, the most recent state is used as the query input for the cross-attention mechanism, ensuring the model responds quickly to recent changes in the data. For the long-term block, the input query is an updated embedding called Dynamic Memory Aggregation Tokens (DMAT), which represents an aggregated view of the memory, updated at every timestep. For each tracklet, the short-term block B_S processes its historical T_S states, while the long-term block B_L employs a comparatively long history length T_L (T_S<< T_L). This differentiation in temporal focus ensures that while the short-term module captures recent dynamics, the long-term module encapsulates overarching patterns and deeper relationships that might span across larger time intervals. Multi-head cross-attention is used in both short-term context B_S and long-term context B_L, with history states serving as key and value inputs, allowing the model to focus on various temporal parts of the states simultaneously. This dual-purpose usage of historical states, as key–value pair, aids in effectively mapping the recent state to its corresponding historical context, fostering a richer contextual understanding. The input query for B_S is the most recent state X^{t-1}, while a dynamically updated embedding, DMAT, $Q^{t-1}_{DMAT} = \{q_k^{t-1}\}k = 1 : N_{tracklet}$, is utilized for B_L. The DMAT tokens provide a consolidated view of the historical memory, ensuring that the long-term module does not just process isolated states instead incorporates a synthesized view which is continually updated to maintain relevance across temporal shifts. At the onset of every tracklet, which represents a short sequence of object states, DMAT is allocated. Initially, all tracklets have identical DMATs, symbolizing the same initial memory representation for each tracked object. Afterwards, at time step $t > 0$, the DMATs carry out iterative updates based on the states from the previous time step. This implies that at each step, the DMATs combine information from the previous state, updating their values to represent the evolution of the object states over time. This mechanism ensures that while all tracklets may start their journey from the same point, their subsequent trajectories are shaped by their unique histories and contextual influences. This design concept is further validated in Section 4.5.

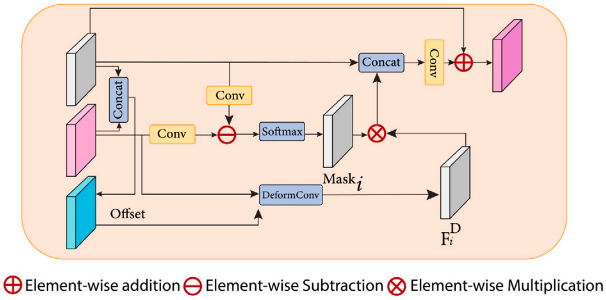

Concat
Conv
Conv
Softmax
Concat
Conv
Mask$_i$
DeformConv
Offset
F_t^D
⊕ Element-wise addition ⊖ Element-wise Subtraction ⊗ Element-wise Multiplication

Fig. 6. Architecture of Progressive feature transfer layer.

The outputs from both the short-term and long-term blocks, referred as Aggregated Short-term Context (ASC) Q^t_{ASC} and Aggregated Long-term Context (ALC) Q^t_{ALC}, respectively, are then merged together by a self-attention driven fusion block B_f. The fusion block combines the information from the short-term and long-term memory blocks into aggregated representation Q^t_{DMAT} and produces a consolidated tracklet embedding $Q^t_{tracklet}$, a holistic representation of the tracklet state, encompassing both immediate and historical insights. In addition to the tracklet embedding, the fusion block also outputs an updated Q^t_{DMAT} which carries forward the knowledge and insights from the current operation, ensuring continuity and progression in subsequent time steps. This updated token is saved and used in the subsequent time step, allowing the system to keep track of the ongoing evolution of object states over time. In this way, the model retains and exploits temporal context in tracking objects, enhancing the overall tracking performance.

3.5. Memory decoder

The memory decoder Θ_{MD} is the decision head of the STMMOT responsible for output object identity. It receives the encoded frame feature from the CPN encoder, proposal embedding from the CPN decoder and tracklet embedding from the memory aggregator to predict final tracking outcomes. Each of these carries specific nuances and insights pertinent to the tracklet. The proposal embedding provides a preliminary hypothesis about the object's presence, while the tracklet embedding captures the historical and current trajectory nuances. Meanwhile, the image feature serves as a rich visual representation of the image, providing contextual texture to the entire process. This is achieved through a stack of Transformer decoder modules, using the fused candidate proposal and tracklet embeddings $[Q^t_{can}, Q^t_{tracklet}]$ as queries. By adopting these embeddings as queries, the model is effectively seeking mappings that resonate with both the proposed hypothesis and the tracked trajectory of the object. Θ_{MD} utilizes the encoded frame feature x^t_i from Θ_c as key–value pair. By doing so, every query (reflecting proposal and tracklet information) is mapped to a rich visual representation encapsulated within the image feature. This design ensures that the resultant tracking predictions are not just based on historical or proposed data but are continually grounded and refined by the immediate visual context provided by the image feature.

In Θ_{MD} output $[\hat{Q}^t_{can}, \hat{Q}^t_{tracklet}]$, each entry q^t_i goes through a decoding process that transformer the fused embedding into three predictions: bounding box coordinates (offset to learned reference points), objectness, and uniqueness score. Through the use of a Transformer, decoder applied to these embeddings, the model reasons globally about all the candidate proposals via pair-wise relation, while concurrently incorporating the full frame as a contextual reference. The objectness score o^t_i and uniqueness score u^t_i for a query q^t_i ranges from 0 to 1, where $o^t_i = 1$ indicates that the model identifies a visible object and u^t_i =1 predicts that detected object is not just a mere repetition or redundancy but a unique object that needs to be included in the tracking results. Conversely, scores approaching 0 suggest potential overlaps or redundancies, hinting at the necessity for suppression and turning the object into a suppressed object when u^t_i is less than 1, implying that object is not novel. We assume $u^t_i = 1$ when $q^t_i \in \hat{Q}^t_{tractlet}$. When a proposal is determined to be unconnected to any identity currently being tracked, it is regarded to be novel and is assigned a uniqueness value u^t_i of 1. The decision for suppression revolves around the uniqueness score u^t_i. Specifically, the model mandates that $u^t_i = 1$ if q^t_i is part of the track embeddings, $\hat{Q}^t_{tractlet}$. This means that only objects that have been previously tracked (and therefore are part of the track embeddings) are considered as "novel". Consequently, a unified confidence score, which applies to proposal and tracklet entries, is established. In essence, the unified confidence score serves as a composite metric, echoing both the visibility and the distinctiveness of the object within an entry. This score is calculated as the product of objectiveness and uniqueness scores:

$$C^t_i = o^t_i.u^t_i \tag{7}$$

The model predicts two types of confidence score predictions for the tracking process, C^t_{can} for candidate proposal and $C^t_{tracklet}$ for tracklet queries. The confidence scores, calculated by the model, evaluate the probability of a proposal or tracklet query representing a valid object. For each entry q^i_t, the decoder predicts its bounding box coordinates b^t_i, containing the object's centre coordinates, width, and height. The centre coordinates denote the position of the object within the frame, while the width and height provide information about the object's size. This approach enables the simultaneous solution of object detection and identity association problems. During inference, entries with $s^t_i \geq \epsilon$ are retained by applying a threshold to each entry in $[\hat{Q}^t_{can}, \hat{Q}^t_{tracklet}]$. Such an approach ensures a calibrated sensitivity, retaining only those detections and tracks that the model ascertains with high confidence. The resulting entries will either inherit a track identity or initiate a new track based on whether they come from $\hat{Q}^t_{can}$ or $\hat{Q}^t_{tracklet}$. The final tracking results are obtained from merging track identities, either inherited or new, with the corresponding bounding box predictions, without the need for further post-processing (Du et al., 2023; Zhang et al., 2021).

Supervision signals for o^t_i, u^t_i, and b^t_i are generated for each frame by first assigning objectiveness scores and bounding boxes to entries in $\hat{Q}^t_{tracklet}$, depending on the presence of the tracked object in the frame. For each entry in $\hat{Q}^t_{can}$, ground truth bounding boxes, whether these are newly identified or already being tracked, are assigned through bipartite matching (Carion et al., 2020). Bipartite matching assigns ground truth bounding boxes to each entry in Q^t_{can} with the objective of finding a maximum matching in a bipartite graph (a graph whose vertices can be divided into two disjoint sets) that covers all vertices of one set. Subsequently, each proposal gets assigned a uniqueness score based on prior object encounters, as illustrated in Fig. 8.

Tracked objects are those that the model identifies as visible with an o^t_i of 1. They are distinguished or previously recognized in the current frame, giving them a Uniqueness Score of 1. As a result, by multiplying the o^t_i and u^t_i scores, their C^t_i is also 1. Similarly, New objects are visible and have an o^t_i score of 1. The difference is that they are newly identified and not previously recognized, hence they also have a u^t_i = 1 and $C^t_i = 1$. In contrast, Backgrounds are areas in the image without distinct objects, indicated by an o^t_i Score of 0. However, they are still considered unique with a u^t_i Score of 1. Their resulting C^t_i score is 0 due to their lack of visibility as distinct objects. Finally, Suppressed objects are visible and have an $o^t_i = 1$. But since they are not distinctive or might be similar to other objects, they get a u^t_i score of 0. This means their C^t_i score is 0, indicating the need to ignore or suppress them to avoid duplication in the results.

3.6. Loss function

In training STMMOT, we compute the tracking loss based on three components objectness score o^t_n, uniqueness score u^t_n, and bounding box transformation bb^t_n using the assignment procedure described earlier. The tracking loss L_track is defined as:

$$L_{track} = \lambda_{cls}(L_{obj} + L_{uni}) + \lambda_{L1}(L_{bbox} + L_{iou}) \tag{8}$$

where λ represents weight scaling factor that ensures appropriate balance among the different loss components, L_{obj} and L_{uni} are focal losses for objectness and uniqueness scores, L_{bbox} is the L1 loss for bounding box regression, and L_{iou} is the generalized Intersection over Union (IoU) loss. To enhance the object localization capabilities of STMMOT, a detection loss is applied to the candidate embedding, a concept similar to Zhu et al. (2020). An auxiliary linear decoder is connected to the candidate embedding to output bounding boxes for the purpose of object localization as well as to object classification scores. The assignment of object instances to these components is carried out as it typically is in standard object detection tasks, and the corresponding loss is computed as:

$$L_{det} = \lambda_{cls} L_{obj} + \lambda_{L1} L_{bbox} + \lambda_{iou} L_{iou} \tag{9}$$

Post-training, the auxiliary decoder is detached. Inspired by the MOTR approach (Zeng et al., 2022), the tracking loss for a series of frames is computed as the cumulative loss across individual track queries, with this sum being normalized by the total number of object instances. For a sequence that includes T frames, the total sequence loss, denoted as L_{seq}, is a fusion of the tracking loss and the auxiliary detection loss:

$$L_{seq} = \lambda_{truck} L_{seq-track} + \lambda_{det} L_{seq-det} \tag{10}$$

where $\lambda_{track} \in R$ and $\lambda_{det} \in R$ are the weights for balancing the tracking and auxiliary detection loss. Their specific role is to maintain an equilibrium between the tracking loss and auxiliary detection loss. λ_{track} specifically determines the relative importance of tracking loss in our overall cost function. The tracking loss considers the model's ability to consistently follow objects across successive frames, ensuring continuity in object tracking. If the model loses track of an object from one frame to the next, this would increase the tracking loss. Thus, a

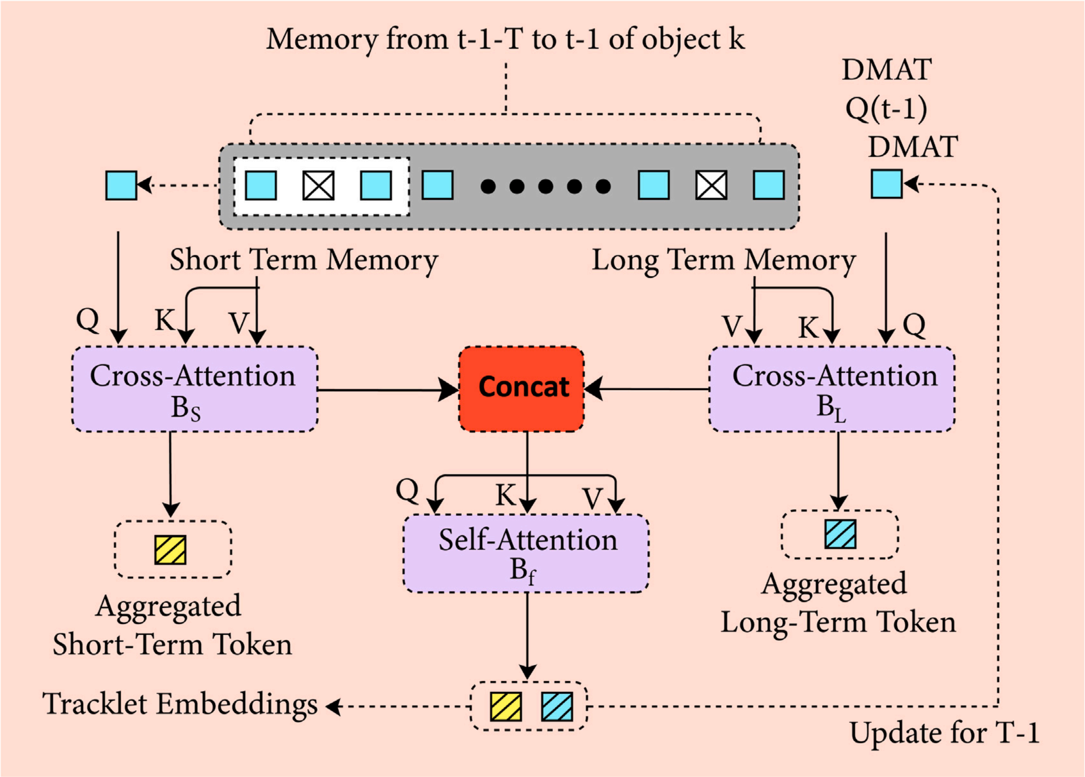


Fig. 7. The architecture of memory encoder consists of three parts: short-term memory block B_s that reduces recent frame noises, long-term B_l that pulls supportive features from a broad context, and fusion blocks B_f that merge short and long-term aspects. The merged embeddings become track embeddings, which update the DMAT for the next step.

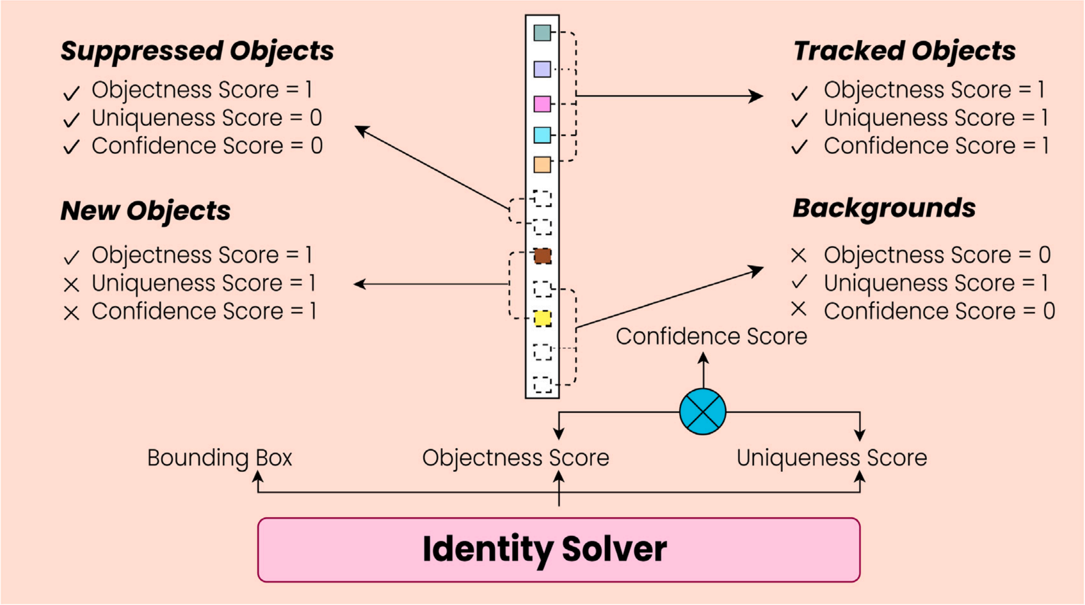


Fig. 8. Association assignment workflow.

higher λ_{track} gives more weightage to this aspect of the loss function, prioritizing consistency in object tracking across frames. Similarly, λ_{det} the weight assigned to the auxiliary detection loss. This loss component accounts for the model's ability to correctly identify and localize new objects in each frame. This involves accurately determining the bounding box dimensions and the class of each detected object. A larger λ_{det} implies more emphasis is being placed on the detection aspect of the model's performance. Both λ_{track} and λ_{det} tuned during the training process, optimizing the balance between the tracking and detection capabilities of our model. The selection of these weights is largely dependent on the specific application and the data at hand. Finally, N_t is a variable that signifies the visible objects present in a frame at a given time t. This variable is crucial because it influences the normalization of the tracking loss over a sequence of frames. Essentially, the cumulative tracking loss is divided by N_t, to calculate the average loss per visible object in the frame. This ensures that our model's performance evaluation is not unfairly penalized when more objects are present in a frame.

4. Experiment and result

In order to validate the efficiency of the technique, we perform extensive testing on two renowned benchmark pedestrian tracking datasets, namely, MOT17 (Yuan et al., 2021) and MOT20 (Zhu et al., 2020). In our ablation study, we follow the established procedure (Ioffe & Szegedy, 2015) by dividing the MOT17 training set into two sections, one for training and the other for validation. We utilize the widely accepted MOT metrics set (Kiefer et al., 2023) for quantitative assessment, with multiple object tracking accuracy (MOTA) serving as the primary metric for gauging overall performance.

4.1. Dataset

CrowdHuman (Shao et al., 2018). CrowdHuman is a large-scale, well-annotated dataset designed for enhancing crowd scenario detector evaluations. It contains 15,000 training, 4,370 validation, and 5,000 testing images, featuring a total of 470K human instances across the train and validation sets. Each image contains around 23 individuals and includes various occlusion types.

MOT17 (Dendorfer et al., 2021). The MOT17 dataset concentrates on tracking multiple individuals in crowded scenarios. It contains a total of 14 video sequences, with seven designated for testing. The MOT17 is widely utilized to benchmark MOT methodologies (Gao, Zhuang, Gu, Yang, & Nie, 2023; Yang et al., 2023). In accordance with previous studies (Cai et al., 2022; Chu et al., 2023; Yang & Chan, 2018), we divide the MOT17 into two subsets during validation. We employ the first subset for training and the second for validation purposes.

MOT20 (Dendorfer et al., 2020). The MOT20 dataset comprises 8 demanding video sequences set in uncontrolled environments and crowded scenes. It is divided into two parts: 4 sequences for the training set and 4 for the testing set. We train on the MOT20 training partition using identical hyper-parameters as those employed for the MOT17 dataset.

4.2. Evaluation metrics

The Multi-Object Tracking (MOT) benchmark (Dendorfer et al., 2021, 2020) employs Multi-Object Tracking Accuracy (MOTA) as its primary metric. MOTA is defined as:

$$MOTA = \frac{1-(\sum_n (FP_n + FN_n + IDsw_n))}{\sum_n (GT_n)} \tag{11}$$

where GT_n denotes the number of ground truth objects in nth frame, while FP_n, FN_n, and $IDsw_n$ represent the errors associated with false positive (FP), false negative (FN), and ID switches (IDsw), respectively. Additionally, following the MOT benchmark recommendations, we report performance on HOTA, a novel tracking metric (Xu et al., 2022). HOTA is the geometric mean of Detection Accuracy (DetA) and Association Accuracy (AssA) defined as:

$$HOTA = \sqrt{DetA \times AssA} \tag{12}$$

Here, DetA follows the formula $\frac{|TP|}{|TP|+|FN|+|FP|}$ where FP is the number of true positive detections, FN is the number of false negatives or missed detections and FP is the number of false positive detections or incorrect detections and $AssA = \sqrt{LocA \times MapA}$ where $LocA$ is the localization accuracy, which measures how well objects are tracked within individual frames and $MapA$ is the mapping accuracy, which measures how well objects are tracked across different frames.

The IDF1 score is a popular metric that quantifies the quality of the tracking. It is essentially the F1-score for identity, focusing on correct identity matching over time. Mathematically, IDF1 is defined as the harmonic mean of ID Recall (IDR) and ID Precision (IDP):

$$IDF1 = \frac{2 \times IDP \times IDR}{IDP + IDR} \tag{13}$$

Where, $IDP = \frac{TP}{TP+FP}$ and $IDR = \frac{TP}{TP+FN}$.

4.3. Experiment settings

Our proposed method is developed using PyTorch, with training and validation executed on a machine with a 16-core CPU @ 3.60 GHz, supported by an Nvidia GeForce RTX 4090 GPU having 128 GB RAM. Data augmentation techniques, such as random horizontal flips, random crops, and scale augmentation, are employed, resizing input images so their shorter side is 900 pixels and the longer side is no more than 1280 pixels. These augmentation techniques are only applied to the training and validation sets of the CrowdHuman and the training set of MOT17 datasets which are used in the training of our STMMOT. After augmentation, a training set of CrowdHuman and MOT17 increases 2 folds and become 38, 740 images and 14 sequence, respectively. Horizontal flips assist in incorporating variability in orientations and ensure the model remains orientation invariant, while random crops introduce random crops to simulate scenarios where only a fraction of the object might be visible, aiding the model in identifying partially visible objects. Resizing makes the image size standardized throughout the training set. This technique is employed to ensure that the model becomes robust to objects of different sizes and can recognize and track objects that appear larger or smaller due to their position in the field of view or their actual size. We utilize EfficientNet (Tan & Le, 2019) as the network backbone and DETR (Zhu et al., 2020) pre-trained on MSCOCO (Fleet, Pajdla, Schiele, & Tuytelaars, 2014) for candidate proposal creation. All Transformer units have their number of layers reduced to 4. Our memory buffer can hold up to 350 tracks for the MOT17 and 700 tracks for the MOT20. The maximum temporal length is 30 for MOT17 and 35 for MOT20, mainly limited by GPU memory constraints. This length refers to the maximum sequence of frames that STMMOT processes at once. The transformer has been assigned an initial learning rate of 0.03 (3e−2), while the backbone's learning rate has been set at a much lower value of 0.00002 (2e−5). The choice of these specific learning rates is determined by the different roles these modules play within the network architecture. In order to prevent overfitting and ensure a more stable training process, a weight decay parameter is incorporated, which is set to 0.01 (1e−2). This regularization technique helps in avoiding large weights, thus leading to a simpler and more generalizable model. As for weight initialization, the transformer weights are assigned initial values using Xavier initialization, and the backbone model is pre-trained with frozen batch-norm (Ioffe & Szegedy, 2015).

In line with previous work (Carion et al., 2020; Zhu et al., 2020), we choose the coefficients of the Hungarian loss with λ_{cls}, λ_{L1} and λ_{iou} as 3, 6, 3, respectively. We set $\lambda_{det} = \lambda_{track} = 2$ in Eq. (10) for the ablation study and performance comparison. The choice of the value for the weights determines the relative importance of the associated losses when training a model. Weight value 1 indicates that both detection and tracking losses are given equal importance in the optimization process. Different value of weights creates the competition between detector and ReID and the aim of STMMOT is to end this competition. The model is trained for 200 epochs, with the learning rate decreasing by a factor of 10 at the 100th epoch. We have employed a sequence-centric training approach. We utilize a sequence-oriented training scheme where the initial sequence length starts at 4 and grows by an increment of 4 for every 20 epochs. The frames within each sequence are chosen using a random interval that can vary between 1 and 10. A notable aspect of our approach, STMMOT, is that, unlike other leading-edge methods, it is trained on both the CrowdHuman training and validation sets and the MOT17 training set. In the case of the MOT20 benchmark, no supplementary data sources are leveraged.

4.4. Ablation study

We carry out tests with different Transformer architectures, configurations of memory, PFTL units, feature extractors and model structures. Unless specifically stated, our models are streamlined by decreasing

Table 1
Ablation study on different transformer based different candidate generator.

CPN architecture	IDF1 ↑	MOTA ↑	HOTA ↑	AssA↑	IDsw (%) ↓	FP (%) ↓	FN (%) ↓
DETR [60]	57.8	59.4	51.2	42.1	31.5	8.5	2.1
DETR-DC5 [60]	64.5	63.7	55.6	49.6	29.6	6.2	1.6
DETR-FPN	62.8	68.1	59.1	53.7	28.3	5.9	1.2
Deformable Transformer [27]	73.8	77.3	62.2	61.2	24.5	4.1	0.3

the layer count of all Transformer units from 4 to 2. The training is carried out on the CrowdHuman and MOT17 datasets, and we validate the models using the MOT17 validation dataset while ensuring that any overlapping videos from the training set are not included in the validation set.

4.4.1. Comparison of transformer architecture

We conduct an evaluation to understand the impact of various Transformer architectures. Five such structures are examined. The first, referred to as Transformer, employs the design from DEtection TRansformer (DETR) (Carion et al., 2020) where the transformer is built atop the feature maps from the ResNet 5th stage (He, Zhang, Ren, & Sun, 2016). The second structure, DETR-DC5 (Carion et al., 2020), enhances the resolution of these feature maps through dilation convolution applied to the ResNet 5th stage, and removing a stride from the initial convolution of this stage. The third structure, DETR-FPN, implements FPN (Zheng et al., 2021) on the input feature maps. The encoder of the Transformer is entirely removed from the pipeline due to memory constraints, which in turn allows the learning rate of the backbone to be increased to match that of the transformers. Lastly, the Deformable Transformer (Zhu et al., 2020), a recently suggested architecture to address the limited resolution issue in transformers, is explored. It incorporates multiple-scale features into the entire encoder–decoder pipeline, which has proven effective on general object detection datasets within acceptable memory usage parameters.

Table 1 presents the quantitative results. The overall performance of baseline DETR is relatively low because of low feature resolution at IDF1 score of 57.8, MOTA of 59.4, and HOTA of 51.2. The DETR-DC5 which improves the feature resolution, shows an improvement across all metrics when compared to the baseline DETR model. The IDF1 score increases by 6.7 to 64.5, MOTA improves by 4.3 to 63.7, and HOTA and AssA show increases of 4.4 and 7.5 to 55.6 and 49.6, respectively. In terms of error metrics, IDsw decreases by 1.9 to 29.6, FP decreases by 2.3 to 6.2, and FN decreases by 0.5 to 1.6. However, it also has a drawback in terms of high memory usage due to dilation convolution. The DETR-FPN model exhibits a notable increase in performance compared to both the DETR and DETR-DC5 models. The IDF1 score decreases slightly by 1.7 to 62.8 compared to DETR-DC5, but MOTA increases significantly by 4.4 to 68.1. DETR-FPN's performance is comparable to DETR-DC5, indicating that a resolution higher than DC5 does not necessarily result in significant performance improvement, possibly due to the absence of encoder blocks. The Deformable Transformer, with its fusion of multiple-scale features into the full encoder–decoder pipeline, stands out with the most significant performance increase. The IDF1 score increases by 11.0 to 73.8 compared to the DETR-FPN model. MOTA shows an increase of 9.2 to 77.3, while HOTA and AssA improve to 62.2 and 61.2, respectively. Notably, error metrics decrease substantially, with IDsw decreasing by 4.1 to 24.2, FP decreasing by 1.8 to 4.1, and FN decreasing by 0.9 to 0.3. These comparisons demonstrate the effectiveness of the Deformable Transformer architecture in improving STMMOT performance across all metrics. Consequently, we select Deformable Transformer as our default architecture choice for STMMOT.

4.4.2. Comparison of backbone networks

We undertake this ablation to evaluate the impact of various backbone networks used as feature extractors for the Deformable Transformer in candidate generation. We replace EfficientNet-B4 (Tan & Le, 2019), the default feature extractor with various backbones including Res2Net (He et al., 2016), DenseNet-201 (Huang, Liu, Van Der Maaten, & Weinberger, 2017), Inception-v3 (Szegedy et al., 2015), and ResNet-50 (He et al., 2016). For each stream in these backbones, we discard the classifiers, comprised of two linear layers. As Table 2 shows, Res2Net has a total of 25.5 million parameters and its performance shows an IDF1 score of 65.3, MOTA of 66.1, HOTA of 56.1, and AssA of 57.8. Next, we employed DenseNet-201, which consists of 20 million parameters. Its performance exhibits an IDF1 score of 67.7, MOTA of 67.3, HOTA of 58.5, and AssA of 57.3. It demonstrates a slight increase in ID switches and false positives to 26.7% and 5.4% respectively, and a marginal increase in false negatives to 0.7%. The ResNet-50 backbone, having 25.6 million parameters, shows a significant improvement with an IDF1 score of 73.6, MOTA of 75.4, HOTA of 61.1, and AssA of 60.4. It further decreases ID switches to 24.3% and false positives to 4.9%, however, false negatives slightly increased to 0.8%. Finally, using EfficientNet-B4, with 19 million parameters, the IDF1 score is 73.8, MOTA is 77.3, HOTA is 62.2, and AssA is 61.2. It slightly increases ID switches to 24.5%, but decreases false positives to 4.1% and false negatives to 0.3%''. We select EfficientNet (Tan & Le, 2019) as the default feature extractor for STMMOT because of its high performance on the majority of metrics and low number of parameters.

4.4.3. Comparison of short-term memory length

STMMOT utilizes the short-term memory buffer to handle the nonlinear movement problem and mitigate the tracking problem due to short-term appearance–reappearance. This ablation study analyzes the impact of varying the short-term memory length in the Deformable Transformer, where EfficientNet-B4 is used as the backbone network and long-term memory TLTM is kept at 25. The experiment starts with a short-term memory length of 2, with a long-term memory length of 30, the metrics show an IDF1 score of 70.2, MOTA of 74.7, HOTA of 60.2, and AssA of 59.4. When the short-term memory length is increased to 3, there is a notable improvement in all metrics: IDF1 increases to 72.4, MOTA to 75.1, HOTA to 61.0, and AssA to 61.0. It also results in a decrease in ID switches to 25.9% and false positives to 4.9%, while significantly reducing false negatives to 0.4%. The optimal results are observed with a short-term memory length of 5: IDF1 reaches 73.8, MOTA hits 77.3, HOTA is 62.2, and AssA is 61.2. At this point, ID switches and false positives fall to their lowest at 24.2% and 4.1% respectively, while false negatives remain at 0.3%. However, when the short-term memory length is increased to 6 or 7, the metrics either stagnate or slightly degrade. For instance, the IDF1 score dips to 73.2 and then to 72.8, while MOTA maintains at 77.3 and then slightly drops to 77.0. HOTA slightly decreases to 62.0 and then to 61.6, and AssA drops to 60.7 and further to 60.1. Observation in Table 3 tells that there is a significant improvement in short-term memory length from 2 to 5, however, no significant performance improvement on increasing the length further, rather than decreasing. It can be established that higher memory length improves performance, but hardware limitations can be the hurdles to selecting the optimal length. Hence, length 5 is a trade-off between performance and computational efficiency, and it is selected as the default short-term memory length for the STMMOT.

4.4.4. Comparison of long-term memory length

Employing a long-term memory buffer, STMMOT effectively addresses the common occurrence of occlusion in real-world contexts. Table 4 compares the impact of different long-term memory lengths

Table 2
Ablation study on different backbone networks as a feature extractor for Deformable Transformer to generate candidates.

Backbone	Params (M)	IDF1 ↑	MOTA ↑	HOTA ↑	AssA↑	IDsw (%) ↓	FP (%) ↓	FN (%) ↓
Res2Net [65]	25.5	65.3	66.1	56.1	57.8.1	26.7	5.1	0.6
DenseNet-201 [66]	20	67.7	67.3	58.5	57.3	26.7	5.4	0.7
Inception-v3 [67]	23.9	71.4	71.8	58.2	58.0	26.4	5.0	0.7
ResNet-50 [64]	25.6	73.6	75.4	61.1	60.4	**24.3**	4.9	0.8
EfficientNet-B4 [63]	**19**	**73.8**	**77.3**	**62.2**	**61.2**	24.5	**4.1**	**0.3**

Table 3
Ablation study on varying short-term memory length in Deformable Transformer with EfficientNet-B4 as backbone network.

T_{STM}	T_{LTM}	IDF1 ↑	MOTA ↑	HOTA ↑	AssA↑	IDsw (%) ↓	FP (%) ↓	FN (%) ↓
2	25	70.2	74.7	60.2	59.4	26.7	5.1	0.7
3		72.4	75.1	61.0	61.0	25.9	4.9	0.4
4		73.1	75.4	61.7	**61.8**	25.3	4.5	0.4
5		**73.8**	**77.3**	**62.2**	61.2	**24.2**	**4 .1**	**0.3**
6		73.2	**77.3**	62.0	60.7	24.8	4.7	**0.3**
7		72.8	77.0	61.6	60.1	25.8	5.0	0.5

Table 4
Ablation study on varying long-term memory length in Deformable Transformer with EfficientNet-B4 as backbone network.

T_{STM}	T_{LTM}	IDF1 ↑	MOTA ↑	HOTA ↑	AssA↑	IDsw (%) ↓	FP (%) ↓	FN (%) ↓
5	5	71.1	74.4	60.0	59.4	27.1	5.7	0.6
	10	71.3	76.4	60.2	60.4	26.5	4.8	0.6
	15	71.8	77.1	61.4	60.1	26.3	4.9	0.4
	20	72.6	77.1	62.4	**61.3**	25.4	4.9	**0.3**
	25	**73.8**	**77.3**	**62.2**	61.1	**24.2**	**4.1**	**0.3**
	30	**73.9**	77.4	61.7	61.1	**23.9**	4.0	0.4

while keeping the length for short-term memory at 5. Increasing the memory from 5 to 25, there is a significant improvement in the performance which is quite obvious from the reduction in ID switching. The best performance occurs with a long-term memory length of 25: the IDF1 reaches 73.8, MOTA and HOTA peak at 77.3 and 62.2 respectively, and AssA hits 61.1. Furthermore, ID switches and false positives decrease to their lowest at 24.2% and 4.1%, respectively, while false negatives remain at 0.3%. However, when the long-term memory length is extended to 30, most metrics show a slight increase in performance. Based on these trends, it can be inferred that a long-term memory length of 25 provides the most optimal performance and provides the best trade-off accuracy–efficiency given our hardware limitation.

4.4.5. Comparison of memory aggregation designs

We experiment with the different structures of the memory aggregation module by initiating comparisons with heuristic pooling and two alternative attention-based aggregation structures into memory encoding. Given that the length of the tracklet can extend up to 25, we choose not to concatenate the embeddings, instead opting to evaluate pooling methodologies. The aggregation could be executed through either the calculation of the arithmetic mean or the maximum norm, encompassing the most recent T frames. As depicted in Table 5, it becomes clear that these basic pooling methods fail to capture the informative track features, leading to a significant reduction in IDF1 and MOTA performance as compared to the default aggregation method of STMMOT. The first attention-based aggregation structure uses only a cross-attention module, excluding the distinction between long and short-term memory. This strategy employs the most recent observation to query an object's past T embeddings. For the Single aggregation method, with T = 5, we observed significant improvements with an IDF1 score of 72.6, MOTA of 68.7 and HOTA of 58.9. With a memory length of 25 (T = 25), the performance was slightly decreased compared to T = 5, but it remained competitive. Taking inspiration from LSTR (Xu et al., 2021), the second method utilizes aggregated short-term embeddings to extract valuable information from long-term memory. The mixed aggregation method 'Long-after-short' showed overall worse performance compared to the Single method, but our proposed design outperformed all other methods. We propose that in the task of action detection, LSTR's main concentration, each frame's outcome is independent, and any shortcomings in short-term features have a minor effect on future predictions. However, when it comes to MOT, errors in associations can have a cumulative effect, which underscores the efficiency of using long-term features to make up for any weakness in short-term features.

4.4.6. Comparison of number of PFTL

In this subsection, We perform the ablation study to show the effectiveness of the number of PFTLs suitable that balance the performance and computational efficiency. We have compared the performance of base STMMOT without PFTL and with different numbers of PFTL in Table 6. Initially, the STMMOT model without PFTL achieved an IDF1 score of 67.2%, MOTA of 70.7%, and HOTA of 52.3%. The model also demonstrated an association accuracy (AssA) of 52.7%, with identity switches (IDsw) at 29.5%, false positives (FP) at 6.7%, and false negatives (FN) at 0.6%. When PFTL was introduced at Level 1, the model exhibited improvements across the board. IDF1 increased to 71.1% (a 5.8% improvement), MOTA to 72.4% (a 2.4% boost), and HOTA to 55.9% (a 6.9% uplift). AssA rose to 56.6% (a 7.4% increase), while the IDsw, FP, and FN percentages reduced to 28.1% (a decrease of 4.7%), 5.3% (down by 20.8%), and 0.6% (no change), respectively. On reaching PFTL 4, the STMMOT model's performance further improved. IDF1 rose to 73.8% (an increase of 3.8%), MOTA to 77.3% (up by 6.7%), and HOTA to 62.2% (up by 11.2%). Finally, at PFTL 7, the STMMOT model achieved its peak performance. IDF1 was marginally better at 74.2% (up by 0.5%), MOTA improved to 79.3% (up by 2.6%), and HOTA went up to 64.4% (an increment of 3.5%). It can be observed that the sequential addition of PFTLs in STMMOT from 1 to 7 progressively improves the model's performance. The model seems to have reached its peak performance at PFTL 7, albeit with only marginal improvements from 4 PFTLs. To better trade-off between performance and computational efficiency given the hardware limitations, we use the 4 PFTLs at each level of the SVP which is used in the rest of the experiment.

Table 5
Ablation study on different memory aggregation designs in Deformable Transformer with EfficientNet-B4 as backbone network.

Design	Memory length	IDF1 ↑	MOTA ↑	HOTA ↑	AssA↑	IDsw (%) ↓	FP (%) ↓	FN (%) ↓
Pooling	Avg T = 5	58.7	44.6	41.9	47.9	29.7	4.5	0.9
	Max T = 5	54.1	37.0	34.1	47.5	31.5	7.2	1.4
	Avg T = 25	38.4	28.6	30.0	34.5	27.0	8.8	2.1
	Max T = 25	47.0	33.4	31.7	42.8	34.4	7.3	2.3
Single	T = 5	72.6	68.7	58.9	59.5	28.4	**3.9**	**0.4**
	T = 25	**72.8**	66.25	**58.7**	**60.2**	**26.1**	5.2	0.8
Long-after-short	–	71.4	**67.4**	54.0	57.4	32.4	6.4	1.7
Ours	–	**73.8**	**77.3**	**62.2**	**61.2**	**24.2**	**4.1**	**0.3**

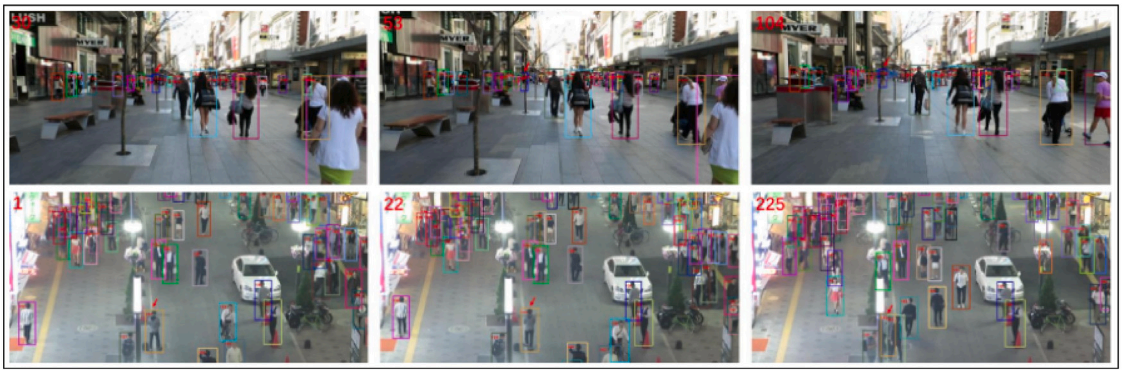


Fig. 9. Sample tracking results of STMMOT on the MOT17.

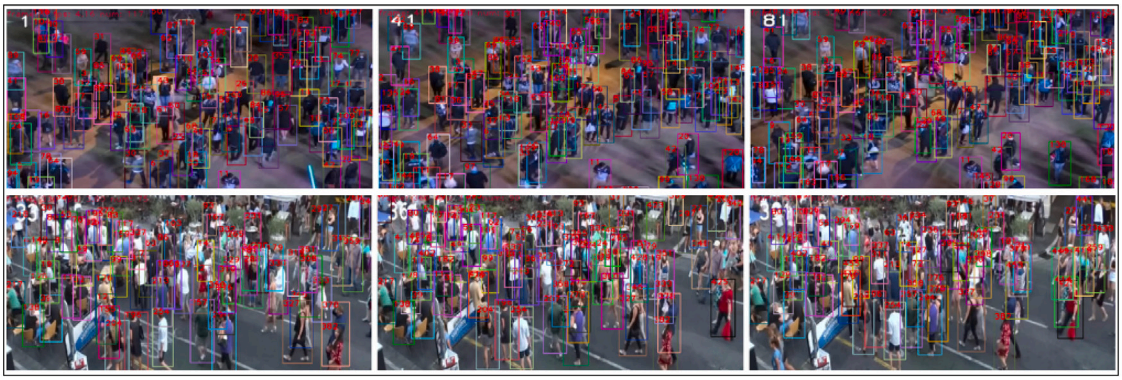


Fig. 10. The MOT example tracking results of the proposed STMOT on MOT20 sequences which shows that STMOT can track multiple objects in dense and occluded scenes while achieving excellent performance.

4.4.7. *Qualitative result visualization*

Figs. 9 and 10 visualize some tracking results of STMMOT on MOT17 and MOT20. The correct assignment of IDs, despite the challenges posed by occlusion or object reappearance. Moreover, STMOT shows robust detection of smaller objects, correctly identifying their respective IDs, which exhibits the strength of CPN. The experimental findings reveal that the proposed STMMOT is capable of unbroken tracking of the target, maintaining a steady identity label when the target re-emerges post-occlusion. This outcome underscores the robust tracking capabilities of the proposed method in overcoming scenarios characterized by the high density of tracking targets and temporal occlusion.

4.5. *Performance comparison*

For a fair comparison, we emphasize contrasting our approach, STMMOT, primarily with methodologies that implement an in-network identity association solver (IIAS). The IIAS provides a mechanism for predicting object identities directly within the network, effectively eliminating the necessity for any post-processing stage. This kind of arrangement offers certain advantages, such as increased efficiency and streamlined computation, since the entire operation can be embedded into the network and performed in an end-to-end manner. On the other hand, there are techniques that utilize a post-network identity association solver (PIAS). This approach applies various rule-based linking procedures to the detected results. Common methods include the application of Hungarian matching algorithms paired with Kalman Filters and re-ID features. The Kalman filter, on the other hand, is a recursive state estimation algorithm, typically utilized to predict the future state of a tracked object based on its past states. Despite their potential effectiveness, such heuristic linking strategies can impose limits on scalability and general applicability. This is primarily due to the reliance on post-processing and manually curated rules, which can become computationally intensive as the number of objects increases. Furthermore, these methods may not adapt well to variations in tracking scenarios, making them less flexible and scalable compared to methods employing an IIAS. For this comparison, we use the EfficientNet-B4 as feature extractor, Deformable Transformer as CPN, short-term memory length 5, long-term memory length 25 and 4 PFTLs.

Table 6
Ablation experiment on STMMOT without PFTL and different numbers of PFTLs.

Design	PFTL	IDF1 ↑	MOTA ↑	HOTA ↑	AssA↑	IDsw (%) ↓	FP (%) ↓	FN (%) ↓
STMMOT without PFTL	–	67.2	70.7	52.3	52.7	29.5	6.7	0.6
STMMOT With PFTL	1	71.1	72.4	55.9	56.6	28.1	5.3	0.6
	2	71.8	73.7	59.5	59.8	27.8	5.2	0.5
	3	72.4	75.9	51.7	60.1	25.7	4.7	0.4
	4	73.8	77.3	62.2	61.2	24.2	4.1	0.3
	5	73.9	78.1	63.1	61.7	24.1	4.0	0.3
	6	74.1	78.8	63.9	62.1	24.1	3.9	0.3
	7	74.2	79.3	64.4	62.4	23.9	3.9	0.3

Table 7
Performance comparison with state-of-the-art MOT models on MOT17 test set.

Method	Transformer	IAS	Memory network	Scale variant	IDF1 ↑	MOTA ↑	HOTA ↑	AssA↑	IDsw ↓
FairMOT Zhang et al. (2021)	–	–	–	–	72.3	73.7	59.3	58.0	3303
CenterTrack Zhou et al. (2020)	–	–	–	–	64.7	67.8			2583
MOTSynth Fabbri et al. (2021)	–	–	–	–	52.0	59.7	–	–	6035
StrongSort Du et al. (2023)	–	–	–	–	**78.5**	**78.3**	**63.5**	**63.7**	1446
TDT Yu, Wu, Gu, and Fathy (2022)	–	–	–	✓	60.9	63.8	–	–	4401
Transtrack Sun et al. (2020)	✓	–	–	✓	63.5	75.2	54.1	47.9	4614
TransCenter Xu et al. (2022)	✓	–	–	–	62.2	73.2	54.5	49.7	3663
MOTR Zeng et al. (2022)	✓	✓	–	–	67.0	67.4	–	–	1992
MeMOT (Cai et al., 2022)	✓	✓	✓	–	69.0	72.5	56.9	**55.2**	2724
TrackFormer Meinhardt et al. (2022)	✓	✓	–	–	63.9	65.0	–	–	3258
TransMOT Chu et al. (2023)	✓	✓	–	–	**76.3**	**76.4**	–	–	**1,623**
locality-enhanced Wu, Hadachi, Lu, and Vivet (2023)	✓	–	–	✓		72.1	–	–	2087
Swin-JDE Tsai et al. (2023)	✓	✓	–	–	70.7	72.3	**57.8**	–	2679
STMMOT (ours)	✓	✓	✓	✓	**79.8**	**79.3**	**73.2**	**61.2**	**1529**

4.5.1. MOT17 benchmark

Table 7 demonstrates that STMMOT attains better performance among IIAS MOTs, while comparative performance is compared to PIAS methods. Performance comparison of STMMOT against two powerful models, namely StrongSort and TransMOT, offers valuable insights. StrongSort demonstrates remarkable performance across several benchmarks. However, STMMOT still manages to outperform StrongSort on multiple measures. For instance, the IDF1 score sees an increase from 78.5% in StrongSort to 79.8% in STMMOT, suggesting enhanced identity preservation in the latter. Similarly, the MOTA score rises by 2.0 percentage points, from 78.3% in StrongSort to 79.3% in STMMOT, demonstrating the superior tracking accuracy of our proposed model. Particularly noteworthy is the significant increment in the HOTA score, which surges by 9.7 percentage points from 63.5% in StrongSort to a robust 73.2% in STMMOT, reflecting a substantial enhancement in both detection and association accuracy. Although AssA shows a slight decrement and the number of identity switches (IDsw) increases, STMMOT maintains a competitive edge in these areas. When compared with TransMOT, STMMOT maintains its performance edge. Specifically, the IDF1 score sees an uplift of 3.5 percentage points from 76.3% with TransMOT to 79.8% in STMMOT. Similarly, the MOTA score witnesses a modest increment of 2.9 percentage points, rising from 76.4% with TransMOT to 79.3% with STMMOT, reinforcing STMMOT tracking accuracy. The IDsw value shows a significant decrement from 1623 with TransMOT to 1529 with STMMOT, suggesting that STMMOT achieves more accurate data association by reducing identity switches.

4.5.2. MOT20 benchmark

We evaluated our proposed model, STMMOT, against a variety of state-of-the-art MOT models on the MOT20 test set. As Table 8 shows, the FairMOT model, with an IDF1 score of 67.3 and MOTA of 61.8, had a considerable number of IDsw at 5243. On the other hand, StrongSort showed improved performance with an IDF1 score of 73.8 and MOTA of 77.3, significantly reducing the IDsw count to 1729. ByteTrack and TransMOT achieved higher performance with IDF1 scores of 75.1 and 75.2 and MOTA scores of 76.5 and 77.4, respectively. However, the number of IDsw for ByteTrack was 1120, and for TransMOT, it was 1601. STMMOT outperforms all other models, achieving the highest IDF1 score of 78.4, MOTA score of 74.1, and keeping the IDsw count at a low of 1264. Additionally, it recorded high HOTA and AssA scores, 69.0 and 61.5, respectively. The superior performance of STMMOT can

Table 8
Performance comparison with state-of-the-art MOT models on MOT20 test set.

Method	Transformer	IAS	Memory network	Scale variant	IDF1 ↑	MOTA ↑	HOTA ↑	AssA↑	IDsw ↓
FairMOT Zhang et al. (2021)	–	–	–	–	67.3	61.8	54.6	54.7	5243
StrongSort Du et al. (2023)	–	–	–	–	**73.8**	77.3	**62.2**	**61.2**	1729
MOTSynth Fabbri et al. (2021)	–	–	–	–	39.7	43.7	–	–	3467
TDT Yu et al. (2022)	–	–	–	✓	46.0	47.9	–	–	5342
ByteTrack Zhang et al. (2022)	–	–	–	–	75.1	**76.5**	61.2	60.0	**1,120**
Transtrack Sun et al. (2020)	✓	–	–	✓	59.4	65.0	48.9	45.2	3608
TransCenter Xu et al. (2022)	✓	–	–	–	49.6	58.5	43.5	37.0	4695
MeMOT (Cai et al., 2022)	✓	✓	✓	–	66.1	63.7	54.1	**55.0**	1938
TrackFormer Meinhardt et al. (2022)	✓	✓	–	–	63.6				
TransMOT Chu et al. (2023)	✓	✓	–	–	**75.2**	77.4	–	–	**1,601**
Swin-JDE Tsai et al. (2023)	✓	✓	–	–	69.5	70.4	**55.7**	–	2026
STMMOT (ours)	✓	✓	✓	✓	**78.4**	**74.1**	**69.0**	**61.5‘**	**1264**

be attributed to the model's implementation of transformer architectures and memory networks, a combination seen in only a few of the compared models. The inclusion of scale variant designs in STMMOT might have also contributed to its dominant performance.

5. Conclusion

Our STMMOT proposition addresses online MOT through the integration of object detection and identity association. By integrating the strengths of transformer architectures and memory networks, our model successfully manages the complexity and uncertainties that come with tracking multiple objects in real-world environments. STMMOT maintains a broad spatio-temporal memory and translates previous observations using a memory aggregator that is attention-based. Through dynamically updating query embeddings that represent objects, STMMOT is capable of forecasting the states of these objects using an attention mechanism, eliminating the need for any post-processing. The results from our extensive experimentation and ablation studies validate the effectiveness and robustness of STMMOT under different settings and conditions. Key to the success of our model is the implementation of short- and medium-term memories, which allow the system to better manage object identities across frames and sequences. Our novel memory aggregation method also contributes to improving the model's performance by effectively consolidating different memory durations. Moreover, STMMOT showed excellent generalization and adaptability when dealing with different feature extraction backbone networks, proving its flexibility and wide applicability. Our tests on the MOT17 and MOT20 test sets confirmed the model's superior tracking accuracy, leading to a notable reduction in identity switches. While the results of STMMOT are encouraging, future work could explore more advanced fusion techniques or leverage more sophisticated transformer architectures to further enhance the model's tracking capabilities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Babaee, Maryam, Li, Zimu, & Rigoll, Gerhard (2018). Occlusion handling in tracking multiple people using RNN. In *2018 25th IEEE international conference on image processing* (pp. 2715–2719). IEEE.

Bergmann, Philipp, Meinhardt, Tim, & Leal-Taixe, Laura (2019). Tracking without bells and whistles. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 941–951).

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

Cai, Jiarui, Xu, Mingze, Li, Wei, Xiong, Yuanjun, Xia, Wei, Tu, Zhuowen, et al. (2022). Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8090–8100).

Carion, Nicolas, Massa, Francisco, Synnaeve, Gabriel, Usunier, Nicolas, Kirillov, Alexander, & Zagoruyko, Sergey (2020). End-to-end object detection with transformers. In *Computer vision–ECCV 2020: 16th european conference, glasgow, UK, august 23–28, 2020, proceedings, part I 16* (pp. 213–229). Springer.

Chan, Kelvin CK, Wang, Xintao, Yu, Ke, Dong, Chao, & Loy, Chen Change (2021). Understanding deformable alignment in video super-resolution. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35, no. 2* (pp. 973–981).

Chen, Chun-Fu Richard, Fan, Quanfu, & Panda, Rameswar (2021). Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 357–366).

Chen, Fei, Wang, Xiaodong, Zhao, Yunxiang, Lv, Shaohe, & Niu, Xin (2022). Visual object tracking: A survey. *Computer Vision and Image Understanding*, *222*, Article 103508.

Chu, Peng, Wang, Jiang, You, Quanzeng, Ling, Haibin, & Liu, Zicheng (2023). Transmot: Spatial-temporal graph transformer for multiple object tracking. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 4870–4880).

Cioppa, Anthony, Giancola, Silvio, Deliege, Adrien, Kang, Le, Zhou, Xin, Cheng, Zhiyu, et al. (2022). Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3491–3502).

Dendorfer, Patrick, Osep, Aljosa, Milan, Anton, Schindler, Konrad, Cremers, Daniel, Reid, Ian, et al. (2021). Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision*, *129*, 845–881.

Dendorfer, Patrick, Rezatofighi, Hamid, Milan, Anton, Shi, Javen, Cremers, Daniel, Reid, Ian, et al. (2020). Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003.

Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Du, Yunhao, Zhao, Zhicheng, Song, Yang, Zhao, Yanyun, Su, Fei, Gong, Tao, et al. (2023). Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*.

Fabbri, Matteo, Brasó, Guillem, Maugeri, Gianluca, Cetintas, Orcun, Gasparini, Riccardo, Ošep, Aljoša, et al. (2021). Motsynth: How can synthetic data help pedestrian detection and tracking? In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10849–10859).

Feichtenhofer, Christoph, Pinz, Axel, & Zisserman, Andrew (2017). Detect to track and track to detect. In *Proceedings of the IEEE international conference on computer vision* (pp. 3038–3046).

Fleet, David, Pajdla, Tomas, Schiele, Bernt, & Tuytelaars, Tinne (2014). *Computer vision–ECCV 2014: 13th european conference, zurich, switzerland, september 6-12, 2014, proceedings, part I, Vol. 8689*. Springer.

Gao, Zeng, Zhuang, Yi, Gu, Jingjing, Yang, Bo, & Nie, Zhicheng (2023). A joint local–global search mechanism for long-term tracking with dynamic memory network. *Expert Systems with Applications*, *223*, Article 119890.

Guo, Song, Wang, Jingya, Wang, Xinchao, & Tao, Dacheng (2021). Online multiple object tracking with cross-task synergy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8136–8145).

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

He, Jian, Zhong, Xian, Yuan, Jingling, Tan, Ming, Zhao, Shilei, & Zhong, Luo (2022). Joint re-detection and re-identification for multi-object tracking. In *International conference on multimedia modeling* (pp. 364–376). Springer.

Hosseini-Asl, Ehsan, McCann, Bryan, Wu, Chien-Sheng, Yavuz, Semih, & Socher, Richard (2020). A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, *33*, 20179–20191.

Huang, Gao, Liu, Zhuang, Van Der Maaten, Laurens, & Weinberger, Kilian Q (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Hyun, Jeongseok, Kang, Myunggu, Wee, Dongyoon, & Yeung, Dit-Yan (2023). Detection recovery in online multi-object tracking with sparse graph tracker. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 4850–4859).

Ioffe, Sergey, & Szegedy, Christian (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). pmlr.

Kiefer, Benjamin, Quan, Yitong, & Zell, Andreas (2023). Memory maps for video object detection and tracking on UAVs. arXiv preprint arXiv:2303.03508.

Kumar, Ankit, Irsoy, Ozan, Ondruska, Peter, Iyyer, Mohit, Bradbury, James, Gulrajani, Ishaan, et al. (2016). Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning* (pp. 1378–1387). PMLR.

Li, Yun, Liu, Zhe, Yao, Lina, & Chang, Xiaojun (2021). Attribute-modulated generative meta learning for zero-shot learning. *IEEE Transactions on Multimedia*.

Li, Jiachen, Wang, Menglin, & Gong, Xiaojin (2023). Transformer based multi-grained features for unsupervised person re-identification. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 42–50).

Li, Wen, Zou, Cheng, Wang, Meng, Xu, Furong, Zhao, Jianan, Zheng, Ruobing, et al. (2023). DC-former: Diverse and compact transformer for person re-identification. arXiv preprint arXiv:2302.14335.

Liang, Tianyi, Li, Baopu, Wang, Mengzhu, Tan, Huibin, & Luo, Zhigang (2022). A closer look at the joint training of object detection and re-identification in multi-object tracking. *IEEE Transactions on Image Processing*, *32*, 267–280.

Liang, Chao, Zhang, Zhipeng, Zhou, Xue, Li, Bing, Zhu, Shuyuan, & Hu, Weiming (2022). Rethinking the competition between detection and reid in multiobject tracking. *IEEE Transactions on Image Processing*, *31*, 3182–3196.

Lin, Matthieu, Li, Chuming, Bu, Xingyuan, Sun, Ming, Lin, Chen, Yan, Junjie, et al. (2020). Detr for crowd pedestrian detection. arXiv preprint arXiv:2012.06785.

Lin, Xingtao, Zhou, Xiaogen, Tong, Tong, Nie, Xingqing, Wang, Luoyan, Zheng, Haonan, et al. (2022). A super-resolution guided network for improving automated thyroid nodule segmentation. *Computer Methods and Programs in Biomedicine*, *227*, Article 107186.

Liu, Zhi, Mu, Xingyu, Lu, Yunhua, Zhang, Tingting, & Tian, Yingli (2023). Learning transformer-based attention region with multiple scales for occluded person re-identification. *Computer Vision and Image Understanding*, *229*, Article 103652.

Lu, Yunhua, Jiang, Mingzi, Liu, Zhi, & Mu, Xinyu (2023). Dual-branch adaptive attention transformer for occluded person re-identification. *Image and Vision Computing*, *131*, Article 104633.

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

Nenavath, Hathiram, Ashwini, K, Jatoth, Ravi Kumar, & Mirjalili, Seyedali (2022). Intelligent trigonometric particle filter for visual tracking. *ISA Transactions*, *128*, 460–476.

Pang, Yatian, Wang, Wenxiao, Tay, Francis EH, Liu, Wei, Tian, Yonghong, & Yuan, Li (2022). Masked autoencoders for point cloud self-supervised learning. In *Computer vision–ECCV 2022: 17th european conference, tel aviv, israel, october 23–27, 2022, proceedings, part II* (pp. 604–621). Springer.

Qiu, Sen, Zhao, Hongkai, Jiang, Nan, Wang, Zhelong, Liu, Long, An, Yi, et al. (2022). Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. *Information Fusion*, *80*, 241–265.

Rao, Haocong, & Miao, Chunyan (2023). TranSG: Transformer-based skeleton graph prototype contrastive learning with structure-trajectory prompted reconstruction for person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 22118–22128).

Shao, Shuai, Zhao, Zijian, Li, Boxun, Xiao, Tete, Yu, Gang, Zhang, Xiangyu, et al. (2018). Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123.

Shi, Xuepeng, Chen, Zhixiang, & Kim, Tae-Kyun (2023). Multivariate probabilistic monocular 3D object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 4281–4290).

Sukhbaatar, Sainbayar, Weston, Jason, Fergus, Rob, et al. (2015). End-to-end memory networks. *Advances in Neural Information Processing Systems*, *28*.

Sun, Peize, Cao, Jinkun, Jiang, Yi, Zhang, Rufeng, Xie, Enze, Yuan, Zehuan, et al. (2020). Transtrack: Multiple object tracking with transformer. arXiv preprint arXiv:2012.15460.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tan, Mingxing, & Le, Quoc (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.

Tian, Yapeng, Zhang, Yulun, Fu, Yun, & Xu, Chenliang (2020). Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3360–3369).

Tsai, Chi-Yi, Shen, Guan-Yu, & Nisar, Humaira (2023). Swin-JDE: Joint detection and embedding multi-object tracking in crowded scenes based on swin-transformer. *Engineering Applications of Artificial Intelligence*, *119*, Article 105770.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Voeikov, Roman, Falaleev, Nikolay, & Baikulov, Ruslan (2020). TTNet: Real-time temporal and spatial video analysis of table tennis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 884–885).

Wang, Zhongdao, Zheng, Liang, Liu, Yixuan, Li, Yali, & Wang, Shengjin (2020). Towards real-time multi-object tracking. In *European conference on computer vision* (pp. 107–122). Springer.

Weston, Jason, Chopra, Sumit, & Bordes, Antoine (2014). Memory networks. arXiv preprint arXiv:1410.3916.

Wu, Shan, Hadachi, Amnir, Lu, Chaoru, & Vivet, Damien (2023). Transformer for multiple object tracking: Exploring locality to vision. *Pattern Recognition Letters*, *170*, 70–76.

Xie, Fei, Yang, Wankou, Zhang, Kaihua, Liu, Bo, Wang, Guangting, & Zuo, Wangmeng (2021). Learning spatio-appearance memory network for high-performance visual tracking. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2678–2687).

Xu, Yihong, Ban, Yutong, Delorme, Guillaume, Gan, Chuang, Rus, Daniela, & Alameda-Pineda, Xavier (2022). TransCenter: Transformers with dense representations for multiple-object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xu, Jiarui, Cao, Yue, Zhang, Zheng, & Hu, Han (2019). Spatial-temporal relation networks for multi-object tracking. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3988–3998).

Xu, Mingze, Xiong, Yuanjun, Chen, Hao, Li, Xinyu, Xia, Wei, Tu, Zhuowen, et al. (2021). Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems*, *34*, 1086–1099.

Yang, Tianyu, & Chan, Antoni B. (2018). Learning dynamic memory networks for object tracking. In *Proceedings of the european conference on computer vision* (pp. 152–167).

Yang, Wenyu, Jiang, Yong, Wen, Shuai, & Fan, Yong (2023). Online multiple object tracking with enhanced re-identification. *IET Computer Vision*.

Yu, Fengwei, Li, Wenbo, Li, Quanquan, Liu, Yu, Shi, Xiaohua, & Yan, Junjie (2016). Poi: Multiple object tracking with high performance detection and appearance feature. In *Computer vision–ECCV 2016 workshops: amsterdam, the netherlands, october 8-10 and 15-16, 2016, proceedings, part II 14* (pp. 36–42). Springer.

Yu, Shuzhi, Wu, Guanghang, Gu, Chunhui, & Fathy, Mohammed E. (2022). TDT: Teaching detectors to track without fully annotated videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3940–3950).

Yuan, Yuhui, Fu, Rao, Huang, Lang, Lin, Weihong, Zhang, Chao, Chen, Xilin, et al. (2021). Hrformer: High-resolution transformer for dense prediction. arXiv preprint arXiv:2110.09408.

Zeng, Fangao, Dong, Bin, Zhang, Yuang, Wang, Tiancai, Zhang, Xiangyu, & Wei, Yichen (2022). Motr: End-to-end multiple-object tracking with transformer. In *Computer vision–ECCV 2022: 17th european conference, tel aviv, israel, october 23–27, 2022, proceedings, part XXVII* (pp. 659–675). Springer.

Zhang, Zheng, Cheng, Dazhi, Zhu, Xizhou, Lin, Stephen, & Dai, Jifeng (2018). Integrated object detection and tracking with tracklet-conditioned detection. arXiv preprint arXiv:1811.11167.

Zhang, Yifu, Sun, Peize, Jiang, Yi, Yu, Dongdong, Weng, Fucheng, Yuan, Zehuan, et al. (2022). Bytetrack: Multi-object tracking by associating every detection box. In *Computer vision–ECCV 2022: 17th european conference, tel aviv, israel, october 23–27, 2022, proceedings, part XXII* (pp. 1–21). Springer.

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

Zhao, Zelin, Wu, Ze, Zhuang, Yueqing, Li, Boxun, & Jia, Jiaya (2022). Tracking objects as pixel-wise distributions. In *Computer vision–ECCV 2022: 17th european conference, tel aviv, israel, october 23–27, 2022, proceedings, part XXII* (pp. 76–94). Springer.

Zheng, Sixiao, Lu, Jiachen, Zhao, Hengshuang, Zhu, Xiatian, Luo, Zekun, Wang, Yabiao, et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6881–6890).

Zhou, Xingyi, Koltun, Vladlen, & Krähenbühl, Philipp (2020). Tracking objects as points. In *Computer vision–ECCV 2020: 16th european conference, glasgow, UK, august 23–28, 2020, proceedings, part IV* (pp. 474–490). Springer.

Zhu, Xizhou, Su, Weijie, Lu, Lewei, Li, Bin, Wang, Xiaogang, & Dai, Jifeng (2020). Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159.