Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Uncertainty-Guided Triple-Memory Fusion for Robust Multi-Object Tracking

Weichao Guo
Chinese Aeronautical Establishment
Leiming He
Chinese Aeronautical Establishment
Han Li
Chinese Aeronautical Establishment
Yi Sun
Chinese Aeronautical Establishment
Yuchen Zhang
Chinese Aeronautical Establishment
Chenguang Xing
knife1107@163.com
Chinese Aeronautical Establishment

Article

Keywords: Multi-object tracking, Triple memory collaborative modeling, Uncertainty-aware data association, Mamba-based probabilistic motion model

Posted Date: July 22nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10121615/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Uncertainty-Guided Triple-Memory Fusion for Robust Multi-Object Tracking

Weichao Guo, Leiming He, Han Li, Yi Sun, Yuchen Zhang, Chenguang Xing*

Chinese Aeronautical Establishment, Beijing 100086, China

*Correspondent author, E-mail address: knife1107@163.com

Abstract: Multi-object tracking in complex dynamic scenes remains challenging due to non-linear motion, frequent occlusion, and appearance variation. Traditional methods rely on fixed motion priors and fail to model uncertainty or fuse multi-modal cues adaptively. This work presents an uncertainty-guided triple-memory fusion framework for robust multi-object tracking. The method constructs motion memory via probabilistic Mamba to quantify prediction uncertainty, along with appearance memory and category memory to maintain stable visual and semantic representations. A dynamic fusion mechanism adjusts cue weights based on motion uncertainty and trajectory state to suppress error accumulation. Here we show that the proposed method achieves HOTA scores of 58.2%, 74.8%, and 48.5% on VisDrone, DanceTrack, and SportsMOT datasets, respectively, outperforming several state-of-the-art approaches. This framework improves identity preservation and tracking stability in dense and high-dynamic scenarios, suitable for real-world surveillance and autonomous perception applications.

Keywords: Multi-object tracking; Triple memory collaborative modeling; Uncertainty-aware data association; Mamba-based probabilistic motion model

1 Introduction

As the core task in the field of computer vision, the Multi-object Tracking (MOT)[1] aims to continuously locate and associate the targets of interest in video sequences[2], and has important applications in autonomous driving[3-6], human behavior analysis[7-9], intelligent monitoring[10] and other scenarios. Among the various approaches developed for MOT, the paradigm represented by Tracking-by-Detection[11, 12] has emerged as the predominant approach due to its modular architecture and ease of deployment. In this framework, an object detector first generates bounding boxes for targets in each frame, and these detections are subsequently linked into complete trajectories via a data association process. Within this pipeline, motion prediction and feature-based matching constitute the core of the association stage, directly governing both the temporal continuity and the identity consistency of the resulting trajectories.

Although the Kalman filter-based motion model[13] performs adequately in scenarios dominated by simple linear motion, its underlying assumption of uniform motion renders it ill-suited for the nonlinear and highly dynamic movement patterns commonly encountered in dance, sports, and similar scenarios. In recent years, a number of studies have explored the use of learning-based motion predictors, such as recurrent neural networks (RNNs), Transformers, and emerging state space models (SSMs) [14, 15], to model temporal dynamics more effectively. While these approaches substantially enhance the capacity to capture complex motion patterns, they remain largely confined to modeling motion cues in isolation. As a result, they fail to systematically incorporate complementary sources of information, including appearance and category priors. Consequently, they remain vulnerable to identity switches and trajectory fragmentation in challenging situations involving severe occlusion, high inter-target appearance similarity, or abrupt scale variations.

In fact, robust MOT requires a combination of multiple cues: temporal motion rules, appearance feature discrimination, and

category attributes such as the scale and shape of the target itself. However, the existing methods often only focus on one or use simple linear weighted fusion, lacking dynamic evaluation and adaptive integration of various cue reliability. In addition, most learning trackers fail to explicitly quantify the uncertainty of the prediction results[16], which prevents the tracker from properly down-weighting low-quality cues under unreliable observations (e.g., occlusion and blurring), thereby harming system robustness.

To address the aforementioned challenges, this paper introduces an uncertainty-guided triple-memory fusion framework. Based on this framework, a concrete tracker is instantiated upon the YOLO detector, termed the Triple-Memory Uncertainty-Aware Tracker (Tri-Mem UAT). Its core idea is to build a collaborative multi-memory system, which continuously models and updates the motion evolution, appearance representation and long-term category attributes of the target, and introduces a probabilistic uncertainty perception mechanism to dynamically evaluate the prediction reliability of various types of memory, so as to achieve adaptive multi-cue fusion and robust data association. Specifically, the main innovations of this paper are reflected in the following aspects:

(1) A triple-memory collaborative modeling framework for target representation. This paper designs a triple-memory architecture consisting of motion memory, appearance memory, and category memory. Among them, the motion memory extends Mamba with probabilistic output modeling for uncertainty estimation; for convenience, we refer to this variant as Probabilistic Mamba; appearance memory is characterized by ROI[17] in the detector feature map, and the appearance consistency is maintained through the momentum update mechanism. Category memory is based on scale stability and historical confidence to describe the long-term credible attributes of the target. The three kinds of memory complementarily model the target state from different information dimensions, providing multi-source support for subsequent association and update.
(2) Uncertainty-aware dynamic multi-cue fusion mechanism. In the process of motion modeling, this paper explicitly distinguishes and models cognitive uncertainty and accidental uncertainty, and introduces them into data association and memory fusion. Based on the uncertainty level of motion prediction, trajectory life cycle state and time attenuation factors, the system can adaptively adjust the fusion weight between motion, appearance and category memory: make full use of appearance and category cues when the observation is reliable and the trajectory is stable, and enhance the dependence on motion memory when the scale changes, occlusion or observation is blurred, thereby improving the robustness of association decision-making.
(3) Unified online uncertainty-aware tracking framework. In this paper, the triple memory structure and uncertainty perception mechanism are organically integrated into a unified online multi-object tracking framework without additional offline post-processing or multi-stage re-identification strategy. By jointly completing similarity calculation, global matching, memory update and life cycle management in each frame.

The proposed Tri-Mem UAT effectively suppresses identity switches and mitigates unstable trajectory propagation in complex scenarios, while maintaining a favorable balance between tracking accuracy and real-time performance. It achieves superior results on challenging benchmarks characterized by intricate motion patterns and severe occlusions, including VisDrone[18], SportsMOT[7] and DanceTrack[8]. Experimental evaluations demonstrate that the proposed method significantly outperforms both conventional motion models and single-cue trackers in terms of identity preservation metrics.

The organizational structure of this paper is as follows: The second section reviews the related work; the third section introduces the overall framework and module design of Tri-Mem UAT in detail. The fourth section gives the experimental setting, result analysis and ablation experiment. Finally, we summarize the full text and look forward to the future work.

2 Related works

2.1 Based on detection and tracking paradigm

As the mainstream paradigm of MOT, detection-based tracking has been extensively validated in a large number of studies. In this framework, an off-the-shelf detector is first applied to extract object locations from each video frame, and the resulting detections are subsequently associated to form complete target trajectories. Within this line of work, SORT[19] pioneered the combination of Kalman filter and Hungarian algorithm, which laid a solid foundation for online tracking. Subsequent DeepSORT[20] introduced deep appearance features, which significantly improved the re-identification ability. ByteTrack[12] performs well in occlusion scenarios by making full use of low-confidence detection boxes. Such methods are popular because of their modular design and easy integration with advanced detectors. However, their performance depends largely on the design of association algorithms. Especially in complex motion scenarios, traditional motion models often become performance bottlenecks.

2.2 Motion model modeling method

Motion modeling in MOT can be divided into two categories: heuristic motion models and learnable motion models. Heuristic approaches, exemplified by the Kalman filter, typically assume that target motion adheres to a constant velocity model. Such methods offer high computational efficiency and perform well in linear motion scenarios, particularly in pedestrian tracking. OC-SORT[11] partially mitigates the limitations of the Kalman filter in nonlinear motion contexts through an observation-centric design. EGMT[21] encodes graph structure information into node features through embedding graph matching and differentiable Sinkhorn assignment, thereby transforming the NP-hard graph matching problem into a linear assignment problem, significantly enhancing the robustness of association in scenarios with occlusion and appearance changes. ROMOT[22] further decouples detection from tracking by introducing a training-free association module that combines a reconstruction similarity metric (RSM) with observation-centric momentum (OCM), and leverages a vision-language model to support open-set tracking of novel categories as well as referring expression comprehension (REC). Despite these advances, models grounded in fixed motion priors remain ill-equipped to capture the complex motion patterns prevalent in dance, sports, and similar scenarios, and they often entail a considerable number of hyperparameters that require manual tuning.

Therefore, learnable motion models have attracted increasing scholarly attention. In contrast to heuristic approaches, learning-based motion models aim to infer motion patterns directly from data. Early attempts, such as Tracktor[23], leveraged the regression head of a detector to predict target displacement between consecutive frames; however, this approach was constrained by the limited temporal context of frame pairs and consequently yielded suboptimal performance. With recent advances in sequence modeling, deep learning-based motion predictors have garnered growing interest. MotionTrack[24] employs recurrent neural networks to encode trajectory history and forecast subsequent motion states. Transformer architectures, owing to their powerful sequence modeling capacity, have also been introduced to the tracking domain. For instance, TransTrack[25] represents the trajectory as a query vector for cross-frame association. However, the attention-based method faces the problem of quadratic growth of computational complexity when the number of targets increases, which limits its practical application[26]. Sawicki[27] employ an LSTM-based mixture density network (LSTM-MDN) as a learnable motion model to generate synthetic gait samples, which explicitly models the temporal dependencies of inertial signals and the parameters of their probability distribution. This generative approach enables effective data augmentation and improves cross-day person identification accuracy in VR scenarios. However, the quality of the generated samples heavily depends on the proper tuning of hyperparameters (e.g., variance gain and segment aggregation), and poorly trained instances may introduce low-quality samples that degrade system performance.

2.3 Application of state space model in tracking

As an emerging sequence modeling technology, SSMs offer near-linear computational complexity in long-sequence processing, thereby opening new avenues for motion modeling in tracking. The autoregressive process can be described by the following equation:

$$\begin{aligned} \dot{h}(t) &= Ah(t) + Bx(t), \\ \mathrm{y}(t) &= Ch(t) + Dx(t). \end{aligned} \tag{1}$$

MambaTrack[28] was the first to introduce a selective SSM, namely Mamba[29], into MOT. It incorporates a bidirectional Mamba encoding layer to capture complex motion patterns and proposes a trajectory repair module to handle short-term occlusions. TrackSSM[30] adopts an encoder-decoder architecture, progressively refining trajectory predictions through a stream decoder, and employs a stepwise linear training strategy to further enhance performance. While these works collectively demonstrate the potential of SSMs in modeling nonlinear motion, they remain primarily focused on isolated motion cues. Consequently, they fail to fully exploit the rich appearance and semantic information provided by modern detectors, nor do they explicitly model predictive uncertainty.

2.4 Multi-cue tracking method

Tracking methods that rely on a single cue exhibit inherent limitations in specific scenarios: motion cues become unreliable under nonlinear dynamics, while appearance cues tend to fail when targets share similar visual characteristics or undergo severe occlusion. the effective fusion of multiple complementary cues has emerged as a central challenge in enhancing tracking robustness. FairMOT[31] attempts to balance detection and re-identification tasks within a unified framework, yet the two objectives may conflict during optimization. QDTrack[32] employs dense similarity learning to obtain more reliable appearance associations. Nevertheless, these approaches generally adopt static weighting strategies and lack the capacity to dynamically modulate the contribution of each cue in response to scene variations.

In recent years, a growing body of research has turned to the role of uncertainty in tracking. Probabilistic formulations, such as Probabilistic Tracklet Scoring[33], model trajectory states as probability distributions, though they do not systematically investigate the origins and downstream effects of uncertainty. DiffMOT[34] treats trajectory regression as a denoising diffusion process, implicitly dealing with uncertainty, but the computational overhead is large. In contrast, the triple memory architecture proposed in this paper explicitly integrates uncertainty perception into the process of multi-cue fusion. It achieves more intelligent decision fusion by dynamically evaluating the credibility of each of the three memories of motion memory, appearance memory and category memory.

3 Methodology

3.1 Architecture overview

In the tracking process, the algorithm explicitly introduces three types of memory representations: motion memory, appearance memory, and category memory. Then, using uncertainty perception and dynamic weight modulation, these different information sources work together to compensate each other.

Memory Representation Layer. Given the trajectory set T_{t-1}, and detections B_t at frame *t*, we construct three types of memory representations for each trajectory: $\mathbf{M}_{mot}, \mathbf{M}_{app}, \mathbf{M}_{cat}$ denote motion, appearance, and category memories, respectively. Each memory provides a similarity measure between trajectory T_j and detection $\mathbf{b}_i^t$, denoted as: S_{ij}^{mot}, S_{ij}^{app} and S_{ij}^{cat}, which will be respectively defined in Eq. (6), Eq. (10), and Eq. (13) in the following sections.

Uncertainty-Aware Fusion Layer To account for the time-varying reliability of different cues, each memory is associated with an uncertainty estimate: motion uncertainty U_j^{mot}, appearance uncertainty U_j^{app} and category uncertainty U_j^{cat}, These uncertainty measures provide a unified representation of cue reliability, which will be used to guide adaptive fusion and the full mathematical description of these metrics is deferred to Section 3.3.

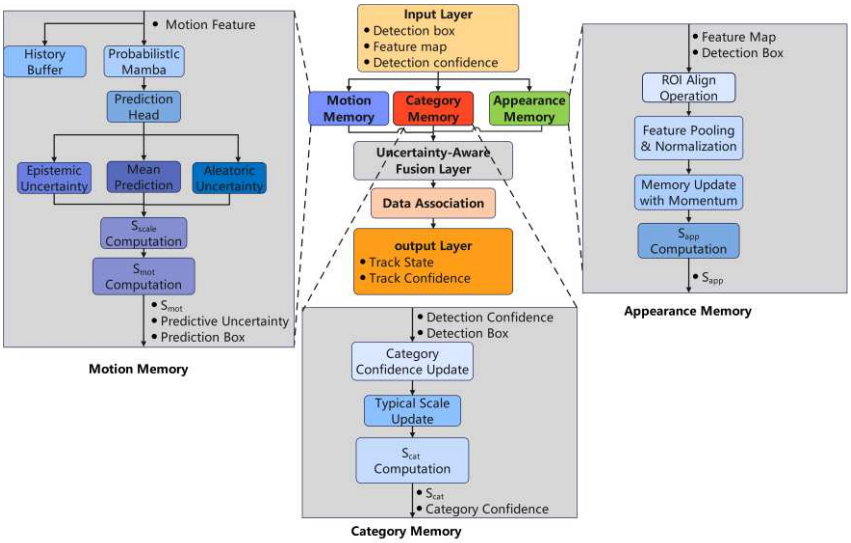


Fig.1 Structure block diagram of uncertainty-guided triple-memory fusion framework

Data Association Layer. The data association step in this framework requires a matching cost function: $\mathbf{C}(\mathrm{T}_{t-1}, \mathrm{B}_t) = \mathrm{f}(\mathbf{M}_{mot}, \mathbf{M}_{app}, \mathbf{M}_{cat}, \mathbf{U}_t)$.where $\mathrm{f}(\cdot)$ denotes the uncertainty-aware fusion function. $\mathbf{U}_t$ represents the uncertainty estimation used for dynamic fusion. The resulting cost matrix is then solved via the Hungarian algorithm to produce globally consistent track-detection assignments.

The specific situation of each component will be introduced in the following sections. The overall structure of uncertainty-guided triple-memory fusion framework is shown in Fig.1.

3.2 Memory Representation and Maintenance

3.2.1 Motion Memory with Probabilistic Mamba

Accurate motion modeling plays a critical role in MOT, especially under abrupt motion, occlusion, and viewpoint changes. The Kalman filter assumes linear motion and cannot cope with complex nonlinear modes; although learning predictors such as RNN and Transformer are more flexible, they are either computationally expensive or difficult to adapt to dynamic motion patterns.

To address these limitations, we propose Probabilistic Mamba, a selective state space model based on the Mamba architecture. Different from the deterministic prediction given by the standard Mamba, our method regards the evolution of the motion state as a sequence and outputs a probability distribution about the future bounding box. It decomposes the prediction uncertainty into two parts: aleatoric uncertainty, which depends on data, and epistemic uncertainty, which depends on the model. The input-dependent selection mechanism allows the model to adaptively modulates the contribution of historical states based on the current motion pattern. At the same time, the probability output gives us calibrated uncertainty estimates, which can then be used to guide data association and memory fusion.

Implementation with Mamba. For each trajectory T_j, the motion memory maintains a motion feature sequence over a sliding window of the last *L* frames where the trajectory was active. The input motion feature at frame t is defined as follows: $x_{\mathrm{t}}=[\Delta x_t, \Delta y_t, \Delta w_t, \Delta h_t, s_t]$.

where $\Delta x_t, \Delta y_t$ denote inter-frame displacements of the bounding box center, $\Delta w_t, \Delta h$ represent changes in width and s_t is the detection confidence score.

The selective SSM maps this input sequence to a hidden state $\mathbf{h}_t \in \square^{d}$ via a continuous-time representation:

$$\begin{aligned} \dot{\mathbf{h}}(t) &= \mathbf{A}(\mathbf{x}(t))\mathbf{h}(t) + \mathbf{B}(\mathbf{x}(t))\mathbf{x}(t) \\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{x}(t))\mathbf{h}(t) \end{aligned} \tag{2}$$

We adopt a zero-order hold discretization scheme for numerical stability. The system is discretized for implementation as:

$$\begin{aligned} \mathbf{h}_t &= \exp(\Delta_t \mathbf{A}(\mathbf{x}_t))\mathbf{h}_{t-1} + \Delta_t \mathbf{B}(\mathbf{x}_t)\mathbf{x}_t, \\ \mathbf{y}_t &= \mathbf{C}(\mathbf{x}_t)\mathbf{h}_t. \end{aligned} \tag{3}$$

where $\mathbf{A}(\mathbf{x}_t)$, $\mathbf{B}(\mathbf{x}_t)$, and $\mathbf{C}(\mathbf{x}_t)$ are input-dependent parameters generated by lightweight neural networks. Δ_t is a learnable step size obtained via a linear projection followed by a Softplus activation, ensuring numerical stability. The matrix exponential $\exp(\cdot)$ provides a stable discretization of continuous-time dynamics.

This selective mechanism allows the model to dynamically adjust how it uses motion history: it retains long-term information during steady motion while quickly adapting to sudden changes. This flexibility is precisely what Kalman filters lack due to their fixed transition matrix.

We implement the selective SSM using the official Mamba library[29], which provides an efficient, hardware-aware implementation of the selective scan algorithm. For each trajectory, we take the motion feature sequence $\mathbf{X}=[\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t]$ and pass it through a causal Mamba encoder to capture long-range temporal dependencies. The output at the current time step is taken as the motion memory state $\mathbf{h}_t$: $\mathbf{H} = \mathrm{Mamba}(\mathbf{X}), \quad \mathbf{h}_t = \mathbf{H}_t$.

The encoder operates in a causal manner, where only historical observations are available at each time step. This design ensures that motion prediction strictly adheres to the online inference protocol, avoiding any information leakage from future frames.

Prediction and Uncertainty Modeling. From the motion memory state $\mathbf{h}_t$, we predict the bounding box at the next frame: $\mathbf{b}_t = \mathbf{W}_{pred}\mathbf{h}_t + \mathbf{b}_{pred}$ .where $\mathbf{b}_t = (\hat{x}_t, \hat{y}_t, \hat{w}_t, \hat{h}_t)$ represents the predicted center coordinates and dimensions.

To quantify prediction reliability, we adopt a probabilistic output that decomposes uncertainty into aleatoric (data-dependent) and epistemic (model-dependent) components:

Aleatoric uncertainty captures observation noise and is estimated by a second output head that predicts the log-variance of the bounding box coordinates:

$$\mathbf{\Sigma}_t^{ale} = \mathrm{diag}(\exp(\mathbf{W}_{ale}\mathbf{h}_t)) \tag{4}$$

Epistemic uncertainty reflects model ignorance and is approximated via Monte Carlo Dropout applied to the projection layers. During inference, we perform *K* stochastic forward passes and compute the variance of the predictions:

$$\mathbf{\Sigma}_t^{epi} = \frac{1}{K}\sum_{k=1}^{K}(\mathbf{b}_t^{(k)} - \bar{\mathbf{b}}_t)(\mathbf{b}_t^{(k)} - \bar{\mathbf{b}}_t)^{\cdot} \tag{5}$$

The total predictive covariance is $\mathbf{\Sigma}_t = \mathbf{\Sigma}_t^{ale} + \mathbf{\Sigma}_t^{epi}$. The final motion similarity score between a predicted box $\mathbf{b}_t$ and a detection $\mathbf{b}_t$ integrates both geometric consistency and uncertainty:

$$S_{ij}^{mot} = \mathrm{IoU}(\mathbf{b}_t, \hat{\mathbf{b}}_t) \cdot S_{scale} \cdot \exp(-\mathrm{tr}(\mathbf{\Sigma}_t)) \tag{6}$$

where $\mathrm{tr}(\mathbf{\Sigma}_t)$ penalizes high overall uncertainty. We adopt $\mathrm{tr}(\mathbf{\Sigma}_t)$ instead of other norms because the trace of $\mathbf{\Sigma}_t$ is equal to the sum of the variances of each coordinate, which can be clearly interpreted as the overall prediction uncertainty.

This design ensures that uncertain predictions are down-weighted during data association.

The S_{scale} measures scale consistency: given a prediction box $\mathbf{b}_t$ and a detection box $\mathbf{b}_t$, the scale consistency score is defined as:

$$S_{scale} = \exp\left(-\frac{\left|\log\frac{\hat{w}_t\hat{h}_t}{w_t h_t}\right|}{\tau_s}\right) \tag{7}$$

where $w_t h_t$ denotes the width and height of the detected bounding box, correspond to the predicted size, and $\tau_s > 0$ is a temperature parameter that controls the sensitivity to scale differences.

Training Loss of the Motion Memory. The motion memory module is trained jointly within the tracking framework. We use the official Mamba library implementation with hidden dimension $d = 128$. The training loss combines smooth L1 and uncertainty

calibration:

$$L = L_{bbox} + 0.1 \cdot \frac{1}{2} \sum_i \left(\frac{\left(\hat{\mathbf{b}}_i - \mathbf{b}_i \right)^2}{\sigma_i^2} + \log \sigma_i^2 \right) \tag{8}$$

where $\sigma_i^2 = \mathrm{tr}(\mathbf{\Sigma}_i)$ denotes the scalar uncertainty measure derived from the trace of the predictive covariance matrix $\mathbf{\Sigma}_t$.

In the Tri-Mem UAT framework, motion memory is not the only basis for decision-making, but the main temporal cue in the three memory parts. Its prediction, and the ensuing uncertainty estimation, is used to generate candidate matching regions and adjust the fusion weights of appearance memory and category memory in real time through an uncertainty-aware mechanism. When the motion prediction uncertainty is low, the system adopts a strong correlation constraint based on motion continuity. When the motion becomes unstable or the prediction becomes unreliable, that is, when the uncertainty is high, the system will automatically reduce the influence of motion cues. Consequently, the irreversible interference caused by the wrong motion assumption is avoided.

3.2.2 Appearance Memory

In MOT, appearance information constitutes a key cue for maintaining target identity consistency, particularly in scenarios characterized by frequent interactions, similar motion patterns, or temporary occlusions. Under such conditions, relying solely on motion continuity is often insufficient to prevent identity switches. Moreover, in real-world videos, appearance cues alone may prove unreliable due to viewpoint variations, scale changes, and illumination fluctuations. Direct feature matching from the current frame is prone to introducing noise and accumulating errors over time. To address these challenges, Tri-Mem UAT incorporates an appearance memory module with temporal smoothing, which captures the persistent appearance characteristics of each target across frames. Unlike approaches that maintain a fixed-size feature gallery or rely exclusively on raw detection embeddings, our appearance memory employs a momentum-updated prototype. This design enables Tri-Mem UAT to strike a favorable balance between preserving temporal coherence in memory and adapting to gradual appearance variations.

Feature Extraction from Detector Backbone. In each frame, the appearance memory extracts the target region features from the intermediate feature map of the detector. Specifically, given a detection box $\mathbf{b}_t$, the ROI Align is used to align the target area on the feature map to obtain a fixed-dimensional appearance feature vector: $\mathbf{f}_i^t \in \square^{\,d}$ .where *d* denotes the feature dimension.

To ensure robust feature representation across scale variations, we select the feature map from the semantically strongest layer of the detector's, P5 level of the FPN backbone, which provides rich semantic information while maintaining spatial alignment accuracy. Compared with directly using the classification features output by the detector, this approach better preserves appearance discriminability under scale changes.

Momentum Update of Appearance Prototypes. For each trajectory T_j, the appearance memory maintains a historical representation $\mathbf{m}_j^{\mathrm{app}}$, that describes the appearance stability feature of the target in the time dimension. This memory is not just a simple stack of past features, but a momentum-updated prototype:

$$\mathbf{m}_j^{\mathrm{app}} \leftarrow (1-\alpha)\mathbf{m}_j^{\mathrm{app}} + \alpha \mathbf{f}_i^t \tag{9}$$

where $\alpha \in (0,1)$ is the update coefficient. The update is performed only when the trajectory is successfully matched with a detection.

For trajectories that are not matched, the appearance memory remains the original state. As a result, errors or missed detection will not contaminate the appearance representation of long-term terms. This momentum-based update enables appearance memory to resist the influence of noise and the rapid change of target appearance while obtaining stable visual features.

Appearance Similarity Computation. In the data association stage, appearance memory provides a matching score by computing the similarity between the current detection feature and the appearance memory of trajectory. We adopt cosine similarity as the appearance similarity measure:

$$S_{ij}^{app}\left(\mathrm{T}_j, \mathbf{b}_i^t\right) = \frac{\mathbf{m}_j^{\mathrm{app}} \cdot \mathbf{f}_i^t}{\left\|\mathbf{m}_j^{\mathrm{app}}\right\|_2 \cdot \left\|\mathbf{f}_i^t\right\|_2} \tag{10}$$

The similarity score is in the range of [-1,1], with higher values indicating stronger appearance consistency. Considering the change of target size in multi-object tracking scenarios, we adopt cosine similarity because it is less sensitive to the change of feature amplitude.

Unlike traditional appearance matching methods based on fixed weights, appearance memory in Tri-Mem UAT is not always dominant. Its contribution is regulated by an uncertainty-aware mechanism: when the motion memory prediction is stable and the uncertainty is low, the appearance memory participates in the association as an auxiliary cue; when the motion prediction becomes unreliable or the target is at risk of occlusion, appearance consistency gains higher weight in data association, thereby enhancing ID retention.

This design ensures that appearance information contributes to decision-making as a form of memory rather than instantaneous observation, effectively reducing the risk of false matches caused by single-frame appearance anomalies.

3.2.3 Category memory

At present, many researches on motion-appearance joint modeling have achieved good results, but their limitations are enhanced in scenarios where the target has similar appearance or abrupt motion. In this case, category-level information provides an additional source of long-term semantic stability. Specifically, objects belonging to the same category usually exhibit consistent geometric and scale patterns over time, even under significant appearance changes. For example, pedestrians maintain a relatively stable aspect ratio, while vehicles show limited scale changes under perspective transformation.

Therefore, incorporating category memory enables the tracker to exploit such category-specific priors, which serve as a complementary constraint when both motion and appearance cues are ambiguous. Unlike instantaneous classification outputs, the proposed category memory accumulates category confidence over time, resulting in a more robust and stable semantic representation. This design enhances discriminability under ambiguous appearance and motion conditions

Category Confidence Vector. For each trajectory T_j ,category memory maintains two core quantities: category confidence vector $\mathbf{c}_j \in \square^K, \sum_k \mathbf{c}_{j,K} = 1.$ where *K* is the number of categories and the typical scale representation $\mathbf{s}_j = (w_j, h_j)$ of the target is used to characterize the scale stability of the trajectory in historical observations.

Among them, the category confidence vector is used to describe the long-term probability distribution of the target belonging to each category, rather than the classification result of a single frame; the scale representation provides geometric support in the physical sense for category information.

Momentum Update of Category Priors. When a new trajectory is created from the first detection, the category confidence vector is initialized as the detection's class probability distribution $\mathbf{p}_i$, and the typical scale is initialized as the detection box size (w_i, h_i). When trajectory T_j successfully matches detection box $\mathbf{b}_i^t$ in the current frame, its category confidence vector is corrected by momentum update:

$$\mathbf{c}_j \leftarrow (1-\beta)\mathbf{c}_j + \beta \mathbf{p}_i^t \tag{11}$$

where $\mathbf{p}_i^t$ represents the class probability distribution of the detector output, and $\beta \in (0,1)$ is the update coefficient. This strategy avoids the decisive influence of single-frame classification error on trajectory category judgment, so that category memory can gradually converge to a stable distribution.

Unlike a generic scale prior, the category memory maintains category-specific typical scales. For trajectory T_j, its typical scale is smoothly updated by historical matching results, but only when the detection's dominant category aligns with the trajectory's dominant category. Specifically, let the dominant category of the trajectory be $\hat{c}_j = \operatorname{argmax} \mathbf{c}_j$ and that of the detection be $\hat{c}_i = \arg\max \mathbf{p}_i^t$. When $\hat{c}_j$ equals $\hat{c}_i$, the scale is updated as:

$$\mathbf{s}_j \leftarrow (1-\gamma)\mathbf{s}_j + \gamma\left(w_i^t h_i^t\right) \tag{12}$$

where (w_i^t, h_i^t) is the scale of the current matching detection box, and γ is the scale update coefficient. $\mathbf{s}_j$ is updated using only detections whose category confidence aligns with the trajectory's dominant category. This ensures that the scale prior reflects the physical characteristics of the target's category.

Category Similarity Computation. In the data association phase, by comparing the consistency between the current detection scale and the typical scale of the trajectory, the category-related scale matching score is calculated:

$$S_{ij}^{cat} = (\mathbf{c}_j \cdot \mathbf{p}_i^t) \cdot \exp\left(-\frac{\left| \log \frac{w_i^t h_i^t}{w_j h_j} \right|}{\tau_c} \right) \tag{13}$$

where $\mathbf{c}_j \cdot \mathbf{p}_i^t$ measures the cosine similarity between the trajectory's category distribution and the detection's category distribution, and $\tau_c > 0$ is a temperature parameter that controls the sensitivity to scale differences. This score measures the geometric consistency between the detection result and the trajectory's category-specific scale prior, thereby suppressing mismatches caused by misclassification or abnormal scale detections.

In Tri-Mem UAT, category memory does not directly determine the motion prediction or matching results of the target, but participates in the fusion decision as a long-term stable semantic and geometric prior. Category-level priors provide low-frequency constraints complementary to high-frequency motion and appearance cues. When the target appearance is similar or the motion cues are not clear, category memory can provide additional distinguishing ability for data association. When the category confidence fluctuates significantly or scale stability drops, the uncertainty-aware mechanism automatically reduces the influence of category memory, thereby preventing erroneous category information from interfering with trajectory evolution.

This design transforms the category information from static labels to memory cues that can be adaptively corrected in the tracking process, and realizes the implicit collaboration between category judgment and time series modeling.

3.3 Uncertainty-Aware Adaptive Fusion

In the proposed Tri-Mem UAT framework, motion, appearance, and category memories provide complementary descriptions of the target state from the perspectives of temporal dynamics, visual consistency, and semantic priors. However, in practical scenarios, the reliability of these cues is inherently time-varying due to factors such as abrupt motion, occlusion, and detection noise. Consequently, directly applying fixed-weight fusion may lead to suboptimal associations, especially when certain cues become unreliable. To address this issue, we introduce a formally defined uncertainty-aware fusion mechanism that dynamically adjusts the contribution of each memory based on its estimated uncertainty[35].

Specifically, for a trajectory T_j and detection box $\mathbf{b}_i^t$, three similarity measures are computed, denoted as motion similarity S_{ij}^{mot}, appearance similarity S_{ij}^{app}, and category similarity S_{ij}^{cat}. To characterize the reliability of each cue, we explicitly model their uncertainties using statistically interpretable quantities.

For motion memory, the prediction module outputs both the estimated state and its associated covariance, from which a scalar uncertainty measure is derived as $U_j^{mot} = \sigma_j^2$, where σ_j^2 denotes the trace of the covariance matrix, reflecting the dispersion of the predicted state. For appearance memory, uncertainty is modeled as a function of temporal staleness, defined as $U_j^{app} = \Delta t_j$, where Δt_j represents the number of consecutive frames without successful association, capturing the degradation of feature reliability over time. For category memory, uncertainty is quantified using information entropy, $U_j^{cat} = -\sum_k c_{j,k} log c_{j,k}$, where $\mathbf{c}_j$ is the category probability distribution, and higher entropy indicates greater ambiguity in semantic prediction.

Based on the above uncertainty modeling, we formulate the fusion weights as exponentially decaying functions of uncertainty. Concretely, the unnormalized weights are defined as:

$$\tilde{\omega}_j^k = \exp(-\lambda_k U_j^k), k \in \{mot, app, cat\}. \tag{14}$$

where λ_k are scaling coefficients controlling the sensitivity to uncertainty. The final weights are obtained via normalization:

$$\omega_j^k = \frac{\tilde{\omega}_j^k}{\tilde{\omega}_j^{mot} + \tilde{\omega}_j^{app} + \tilde{\omega}_j^{cat}}, \quad k \in \{mot, app, cat\}. \tag{15}$$

This formulation establishes a monotonic relationship between uncertainty and contribution, such that cues with higher uncertainty are automatically down-weighted. From a probabilistic perspective, this mechanism can be interpreted as a soft reliability gating strategy that adaptively balances multiple information sources.

With the dynamically adjusted weights, the association cost between trajectory T_j and detection $\mathbf{b}_i^t$ is defined as:

$$\mathbf{C}_{ij} = \omega_j^{mot} \cdot \left(1 - S_{ij}^{mot}\right) + \omega_j^{app} \cdot \left(1 - S_{ij}^{app}\right) + \omega_j^{cat} \cdot \left(1 - S_{ij}^{cat}\right) \tag{16}$$

where S_{ij}^{mot} incorporates both IoU and scale consistency, S_{ij}^{app} is computed using cosine similarity in the embedding space, and S_{ij}^{cat} reflects the compatibility of semantic category priors. This unified formulation can be viewed as a weighted discrepancy measure, where each modality contributes proportionally to its reliability.

For each frame, a cost matrix is constructed based on $\mathbf{C}_{ij}$ and the global optimal assignment is obtained using the Hungarian algorithm. For matched trajectories, all memory modules are updated synchronously to incorporate new observations. Unmatched trajectories are propagated solely based on motion memory to prevent noisy updates, while high-confidence unmatched detections are used to initialize new trajectories. Trajectories that remain unmatched for a predefined number of frames are removed to maintain computational efficiency.

By introducing the dynamic fusion strategy of uncertainty perception, Tri-Mem UAT can adaptively adjust the contribution ratio of three memory modules in different scenarios, so that the system can make full use of multi-cue constraints when the observation is reliable, and maintain conservative decision-making when the information is uncertain or conflicting. This mechanism effectively alleviates the problem that the traditional fixed weight fusion method is susceptible to noise interference in complex scenarios, and lays a foundation for the stable improvement of performance in subsequent experiments.

3.4 Inference

In the inference phase, Tri-Mem UAT processes the input video sequence frame by frame online and continuously updates the target trajectory. Given the detection results of the current frame, the detector feature map, and the set of surviving trajectories in the previous frame, the system performs three core steps of prediction, correlation, and update in each frame. The overall process is summarized in Algorithm 1.

In the current frame *t*, the system first performs motion prediction on the trajectory set T_{t-1} that survives in the previous frame. The process is completed by the motion memory module, and the corresponding prediction bounding box is generated for each trajectory to characterize its potential position in the current frame. Then, the system builds a matching relationship between the

predicted trajectory and the current frame detection result.

Algorithm 1 Tri-Memory Uncertainty-Aware Tracking

Input: Detections $\mathbf{B}_t$, feature map $\mathbf{F}_t$, trajectory T_{t-1}

Output: Updated trajectory T_t

1: **for** each trajectory $\mathbf{T}_j \in \mathbf{T}_{t-1}$ **do**

2: Predict $\mathbf{b}_j^t$ and uncertainty σ_j^2 using Probabilistic Mamba

3: **end for**

4: **for** each detection $\mathbf{b}_i^t \in \mathrm{B}_t$ do

5: **for** each trajectory $\mathbf{T}_i \in \mathbf{T}_{t-1}$ **do**

6: Compute $S_{mot}, S_{app}, S_{cat}$

7: Fuse similarities with uncertainty-aware weights to obtain $\mathbf{S}_{total}$

8: **end for**

9: **end for**

10: Construct cost matrix $\mathbf{C} = 1 - \mathbf{S}_{total}$

11: Obtain matching set M fvia Hungarian algorithm

12: **for** each matched pair $(i, j) \in \mathrm{M}$ j**do**

13: Update motion, appearance, and category memories of T_j

14: **end for**

15: Initialize new trajectory from high-confidence unmatched detections

16: Remove trajectory exceeding miss threshold

17: **return** T_t

Specifically, for each pair of detection boxes and trajectories, the system calculates matching metrics such as motion consistency, appearance similarity, and category scale consistency, and on this basis, forms a comprehensive similarity matrix between trajectory-detection. After normalization, the similarity matrix is converted into a cost matrix for global data association.

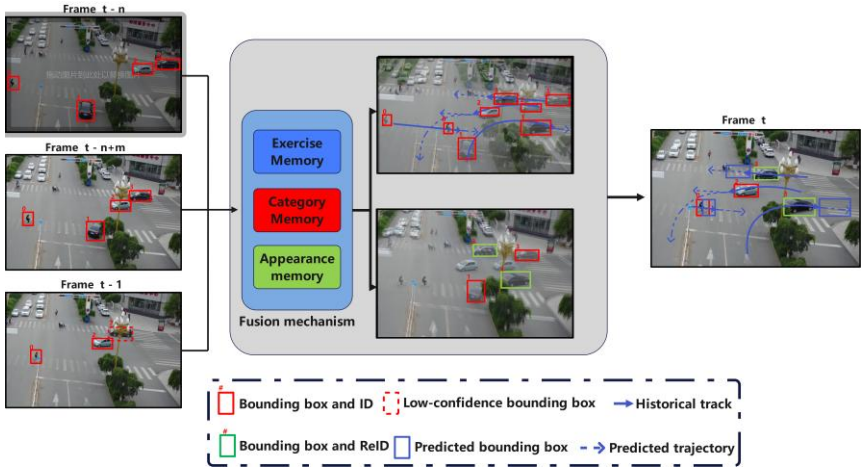


Fig.2 Overall inference pipeline of the Tri-Mem UAT framework. Legend information is located in the lower side box

The global matching is completed by the Hungarian algorithm to obtain a one-to-one correspondence between the trajectory and the detection in the current frame. For the successfully matched trajectory, the system synchronously updates its corresponding memory state and resets the lost count of the trajectory. For the detection results that are not matched but have high confidence, they are initialized as a new trajectory instance and assigned a unique identity. The trajectory that is not matched in the current frame will enter the prediction state, and the number of lost frames will accumulate over time. When it exceeds the preset threshold, it is

judged to be terminated. The overall inference pipeline of the proposed Tri-Mem UAT framework is shown in Fig.2.

4 Experiments

4.1 Setting

Datasets. In this paper, three multi-object tracking data sets with different scene features and challenges are selected for experiments, including DanceTrack, SportsMOT and VisDrone2019, to verify the generalization ability of the method in diverse application scenarios.

DanceTrack. The dataset is dominated by dance scenarios, which has the characteristics of dense number of targets, highly similar appearance, frequent interaction and occlusion. The dataset contains 40 training videos, 25 verification videos and 35 test videos. In these dance scenarios, the features of objects are easily identified, but they are similar in appearance, difficult to distinguish, and show complex and diverse action patterns.

SportsMOT. SportsMOT focuses on the sports competition scenario. The target moves fast, the direction changes violently, and there are significant nonlinear motion modes. The dataset focuses on sports scenarios such as basketball, football and volleyball. It contains 45 training videos, 45 validation videos, and 150 test video sequences collected from high-level sporting events. For athletes' fast and varied movements, SportsMOT requires a powerful tracking method.

VisDrone. The dataset consists of 261908 frames of images and 10209 static images taken by various drone-mounted cameras, covering a wide range of topics, including location, environment, objects, and density. In addition, some important attributes including scene visibility, object category and occlusion are also provided.

Metrics. We evaluate tracking performance using CLEAR metrics[36], HOTA[37], AssA, DetA and IDF1[38] to comprehensively evaluate the tracking performance. Among them, HOTA is used as the main indicator to comprehensively evaluate the overall performance of the tracker, IDF1 and AssA focus on the stability of identity association, and MOTA focuses more on the impact of detection quality. We will report the above indicators in the experimental results to comprehensively analyze the performance of the proposed method on different data sets. In addition, we also use the number of frames per second (FPS) to evaluate the efficiency of the tracker.

Parameter. During inference, the publicly available YOLOX-x[39] detector is used to obtain detection results. To ensure fair comparison with baselines, all hyperparameters in the tracking inference process are fixed. The confidence threshold for detector outputs is set to 0.6. For non-maximum suppression (NMS) [40] post-processing, the IoU threshold is fixed at 0.7 and the confidence threshold at 0.01. In the data association stage, the fusion weights of the triple memories are adaptively adjusted by the uncertainty-aware mechanism, where motion uncertainty dynamically modulates the contribution of motion, appearance, and category cues in the matching cost. Trajectory initialization and termination thresholds follow the default configuration in the implementation, with no additional tuning for specific datasets. Unless otherwise specified, the default historical trajectory length for the motion memory module is set to 5 frames, consistent with the configurations of MambaTrack and TrackSSM. For appearance feature extraction, the default feature source is the final backbone output (P5), corresponding to the strongest semantic level.

Device. All experiments are carried out in a unified hardware and software environment. The experimental platform is equipped with NVIDIA GeForce RTX 3090 GPU, and model inference and tracking evaluation are completed under GPU acceleration

conditions to ensure the stability and repeatability of the experimental results.

4.2 Benchmark results

We compare the proposed Tri-Mem UAT with the officially published state-of-the-art methods on the DanceTrack and SportsMOT test sets, as presented in Table 1 and Table 2, respectively. The results of the other methods in these tables are derived from the official benchmarks and corresponding papers. Additionally, trackers that perform well on DanceTrack and SportsMOT, namely MambaTrack, ByteSSM and MOTR are selected for evaluation on the VisDrone2019 dataset under the same experimental settings as Tri-Mem UAT, as these methods do not report VisDrone results in their original papers. The comparison results are presented in Table 3.

DanceTrack. The proposed method achieves the best overall performance among all the compared trackers, as shown in Table 1. The HOTA value of Tri-Mem UAT is 58.2%, which is 2.5 percentage points higher than that of SparseTrack, which performs best in the Kalman filter method. It has obvious advantages in overall tracking accuracy. Compared with the models that perform well in the methods based on learnable motion modeling, such as MambaTrack and ByteSSM, IDF1 and AssA have a stable improvement. It shows that the identity association under this method is more reliable. These results suggest that the proposed approach effectively improves the coordination between detection and association, leading to more stable tracking performance in scenarios with fast motion and frequent interactions.

Table 1: The performance of Tri-Mem UAT is compared with other popular trackers on the DanceTrack test set. The best results are shown in bold

Tracker	HOTA (%)	MOTA (%)	IDF1(%)	AssA (%)	DetA (%)
kalman filer:					
FairMOT[31]	39.7	82.2	40.8	23.8	66.7
ByteTrack[12]	47.7	89.6	53.9	32.1	71.0
OC-SORT[11]	55.1	92.0	54.6	38.3	80.3
SparseTrack[41]	55.7	91.3	**58.1**	39.3	79.2
learnable motion:					
CenterTrack[42]	41.8	86.8	35.7	22.6	78.1
TraDes[43]	43.3	86.2	41.2	25.4	74.5
TransTrack[25]	45.5	88.4	45.2	27.5	75.9
GTR[44]	48.0	84.7	50.3	31.9	72.5
QuasiDense[32]	54.2	87.7	50.4	36.8	80.1
MOTR[45]	54.2	79.7	51.5	40.2	73.5
MambaTrack[28]	56.8	90.1	57.8	39.8	80.1
ByteSSM[30]	57.7	92.2	57.5	41.0	81.5
Ours	**58.2**	**92.4**	58.0	**41.4**	**81.8**

SportsMOT. The proposed method achieves the best overall performance among all compared trackers, consistently outperforming both Kalman filter-based and learnable motion-based approaches, as shown in Table 2. The HOTA reaches 74.8%, surpassing the strong Kalman filter-based baseline OC-SORT by 1.1 percentage points, demonstrating clear advantages in overall tracking performance. For trajectory consistency, the proposed method achieves 74.3% in IDF1 and 63.0% in AssA, remaining at the leading level across all methods. When compared with recent advanced methods such as ByteSSM, the proposed method further improves AssA, reflecting a better balance between association and overall tracking accuracy. These results highlight the

effectiveness of the proposed method in handling dynamic and highly interactive sports scenarios.

VisDrone. Due to the lack of public reports on recent motion center trackers, all methods were evaluated under the same experimental settings, as shown in Table 3. Among the compared trackers, Tri-Mem UAT achieves the best overall performance. Specifically, it obtains the highest HOTA and AssA, with values of 48.5% and 34.5%, respectively. Compared with MambaTrack and ByteSSM, our method improves HOTA by 2.8 and 1.0 percentage points, respectively. These results indicate that in scenarios where target types are diverse and appearances change frequently, the proposed triple-memory design further improves trajectory consistency by smoothly adapting to appearance variations and the dynamic nature of category perception.

Table 2: The performance of Tri-Mem UAT is compared with other popular trackers on the SportsMOT test set. The best results are shown in bold

Tracker	HOTA (%)	MOTA (%)	IDF1(%)	AssA (%)	DetA (%)
kalman filer:					
FairMOT[31]	49.3	86.4	53.5	34.7	70.2
ByteTrack[12]	64.1	95.9	71.4	52.3	78.5
OC-SORT[11]	73.7	96.5	74.0	61.5	88.5
learnable motion:					
GTR[44]	54.5	67.9	55.8	45.9	64.8
QuasiDense[32]	60.4	90.1	62.3	47.2	77.5
CenterTrack[42]	62.7	90.8	60.0	48.0	82.1
TransTrack[25]	68.9	92.6	71.5	57.5	82.7
MeMOTR[46]	70.0	91.5	71.4	59.1	83.1
MambaTrack[28]	72.6	95.3	72.8	60.3	87.6
ByteSSM[30]	74.4	**96.8**	**74.5**	62.4	88.8
Ours	**74.8**	96.5	74.3	**63.0**	**88.9**

Table 3: The performance of Tri-Mem UAT is compared with other popular trackers on the VisDrone test set. The best results are shown in bold

Tracker	HOTA (%)	MOTA (%)	IDF1(%)	AssA (%)	DetA (%)
MeMOTR[46]	43.4	81.9	41.2	29.5	63.8
MambaTrack[28]	45.7	82.3	43.6	31.7	65.9
ByteSSM[30]	47.5	**85.4**	45.1	33.4	67.5
Ours	**48.5**	85.2	**46.8**	**34.5**	**68.3**

4.3 Ablations

The contribution of each component. Ablation studies are conducted by removing category information (w/o Cat), dynamic modeling (w/o Dyn), and appearance features (w/o App), as well as a motion-only baseline. All experiments were evaluated under the same detection settings on the DanceTrack dataset to ensure a fair comparison.

The full model achieves the best performance across all metrics, demonstrating the effectiveness of the proposed Tri-Mem UAT framework, as shown in Table 4. The removal of the category memory module led to a decrease of more than 2 percentage points in both AssA and IDF1, verifying the significant role of category memory in identity association. When the appearance memory is removed, IDF1 drops from 58.0% to 54.2% and HOTA falls to 54.9%, because this memory module plays an indispensable role: it can smoothly adapt to appearance changes when a target has a similar static appearance but undergoes deformation. The removal of the dynamic fusion mechanism results in a drop of HOTA to 51.0%and a notable degradation in AssA,

indicating that the dynamic multi-cue fusion mechanism plays a critical role in maintaining consistent association. In the case where only motion information is retained, the performance degrades substantially, with HOTA decreasing to 46.2% and AssA to 27.9%, highlighting the limitations of motion-only tracking.

Table 4: Performance comparison of different function module removal in DanceTrack dataset

Tracker	HOTA (%)	MOTA (%)	IDF1(%)	AssA (%)	DetA (%)
Tri-Mem UAT	58.2	92.4	58.0	41.4	81.8
Tri-Mem UAT- (w/o Cat)	56.4	91.4	55.5	39.1	81.3
Tri-Mem UAT-(w/o App)	54.9	90.5	54.2	37.3	80.7
Tri-Mem UAT-(w/o Dyn)	51.0	90.3	48.3	32.9	79.1
Tri-Mem UAT-(Motion-Only)	46.2	89.0	44.2	27.9	76.8

The impact of historical trajectory length. In this experiment, the fixed model architecture and all other hyperparameters remain unchanged. Using the full model, only the length of the input sequence of the motion storage module is changed, and the trajectory segments of 5,10,15,20 and 30 frames are used for experiments. Training and reasoning use the same sequence length setting. The effect of different memory lengths on the performance of action modeling in dance scenarios was evaluated on the DanceTrack dataset.

Increasing the history length consistently improves tracking performance, particularly in HOTA, IDF1, and AssA, as shown in Table 5. HOTA rises from 58.2% at length 5 to 59.5% at length 30, while IDF1 and AssA also show steady gains, they reached 58.6% and 42.5% respectively, indicating that longer temporal context enhances identity consistency and association stability. Meanwhile, DetA exhibits a gradual improvement, suggesting that richer historical information also benefits detection quality to some extent. However, this performance gain comes at the cost of computational efficiency, as FPS decreases significantly from 25 to 10 with increasing sequence length. Notably, the improvement becomes marginal when the length exceeds 15, implying diminishing returns from further extending the temporal window These results indicate that a moderate history length achieves a better balance between performance and efficiency.

Table 5: Performance comparison of different historical frame numbers input

Lengths	HOTA (%)	MOTA (%)	IDF1(%)	AssA (%)	DetA (%)	FPS
5	58.2	92.4	58.0	41.4	81.8	25
10	58.9	91.7	58.3	41.9	82.9	21
15	59.2	92.1	58.4	42.2	83.2	15
20	59.4	92.3	58.6	42.3	83.5	12
30	59.5	92.2	58.6	42.5	83.6	10

The influence of feature semantic level on tracking performance. This experiment is conducted by extracting appearance features from different stages of the YOLOX backbone (CSPDarknet). Specifically, features are taken from three representative stages: dark3 (C3), dark4 (C4), and dark5 (C5), corresponding to low-level, mid-level, and high-level semantic representations, respectively. The C3 features provide higher spatial resolution with limited semantic abstraction, while C5 features encode stronger semantic information but with reduced spatial detail. For each setting, appearance embeddings are extracted using ROI Align from the corresponding feature maps. The evaluation is conducted on the DanceTrack dataset to analyze how feature semantics influence identity association and overall tracking performance.

As shown in Table 6. The results show that Ass A and IDF1 are steadily improved as the feature level increases from C3 to C5,

indicating that stronger semantic representation enhances identity discrimination and association robustness. Therefore, HOTA increased from 56.8% to 58.2%, reflecting the improvement of overall performance. Although the Det A of C5 is slightly lower than that of C4, it can be attributed to the lower spatial resolution of high-level features, which may have a slight impact on the positioning accuracy after ROI Align. Overall, the results show that high-level semantic features are more effective for appearance modeling, while middle-level features provide a favorable trade-off between spatial accuracy and semantic richness.

Table 6: Performance comparison of feature maps with different numbers of feature layers

Feature Layer	HOTA (%)	MOTA (%)	IDF1(%)	AssA (%)	DetA (%)
C3	56.8	91.5	55.3	39.7	81.3
C4	57.6	92.0	55.9	40.5	81.9
C5	58.2	92.4	58.0	41.4	81.8

5 Conclusion

This paper presented Tri-Mem UAT, a triple-memory uncertainty-aware tracking framework designed to address several limitations of existing online multi-object trackers, including insufficient motion modeling, limited robustness of appearance-based association, and inadequate exploitation of complementary cues in complex scenes. Unlike conventional tracking-by-detection approaches that rely primarily on predefined motion models or appearance similarity, the proposed framework explicitly maintains three complementary memories, namely motion memory, appearance memory, and category memory, and integrates them through an uncertainty-guided fusion strategy to achieve adaptive and reliable data association.

Specifically, a probabilistic state-space model is introduced as the motion memory to capture nonlinear temporal dynamics while simultaneously estimating prediction uncertainty, providing reliable motion priors for subsequent association. In addition, an appearance memory is constructed by continuously updating object representations extracted from detector feature maps, which improves identity preservation under frequent occlusions and target interactions. A category memory is further incorporated to model long-term scale priors, providing complementary semantic constraints for both motion prediction and appearance matching. Building upon these three memory components, an uncertainty-guided fusion strategy is developed to dynamically regulate the contribution of each memory according to the confidence of motion prediction and the current trajectory state, allowing the tracker to adaptively exploit the most reliable cues during data association. Extensive experiments on the DanceTrack, SportsMOT, and VisDrone2019-MOT benchmarks demonstrate that the proposed Tri-Mem UAT achieves competitive tracking performance across a variety of challenging scenarios.

Despite its effectiveness, several limitations remain. First, the proposed motion memory models each trajectory independently and does not explicitly consider interactions among neighboring targets, which may limit its ability to capture complex motion dependencies in highly crowded scenes. Second, the appearance memory relies on local visual representations extracted by the

detector, and its discriminative capability may degrade under long-term occlusions, severe pose variations, or extremely low-resolution observations. Moreover, the current category memory primarily exploits scale consistency as a category prior and does not incorporate richer semantic relationships across object categories. In addition, although uncertainty is explicitly modeled during data association, it has not yet been consistently integrated into the entire tracking pipeline. For example, the temporal smoothing factor used in appearance memory is still manually specified rather than being adaptively estimated. Therefore, a unified probabilistic formulation for motion prediction, memory updating, and trajectory management remains an important direction for future research.

Future work will focus on three aspects. First, we plan to incorporate target interaction modeling and scene context into the motion memory to better characterize collective motion patterns in crowded environments. Second, a unified probabilistic memory framework will be investigated to jointly model motion, appearance, and category information under a common uncertainty-aware inference paradigm. Finally, we intend to further optimize the network architecture and inference pipeline to improve deployment efficiency on edge computing platforms and onboard UAV systems, while exploring broader applications in low-altitude security, UAV surveillance, intelligent transportation, and autonomous perception for unmanned systems.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing interests

The authors have no financial or proprietary interests in any material discussed in this article.

Data Availability

The SportsMOT dataset is available at https://github.com/MCG-NJU/SportsMOT. The DanceTrack dataset is available at https://dancetrack.github.io/. The VisDrone-DET2019 dataset is available at http://aiskyeye.com/. The source code and related data generated in this study are available from the corresponding author upon reasonable request and with permission

Author Contributions

W.G. conceived the study, developed the methodology, wrote the software, performed the formal analysis, wrote the original draft, and prepared the visualizations. L.H. reviewed and edited the manuscript and conducted the investigation. H.L. provided resources and curated the data. Y.S. and Y.Z. performed validation. C.X. acquired funding, supervised the study, and administered the project. All authors reviewed the manuscript.

References

[1] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," IEEE transactions on pattern analysis and machine intelligence, vol. 44, no. 7, pp. 3614-3633, 2021. https://doi.org/10.1109/TPAMI.2021.3054719.

[2] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, "Deep learning in multi-object detection and tracking: state of the art," Applied Intelligence, vol. 51, no. 9, pp. 6400-6429, 2021. https://doi.org/10.1007/S10489-021-02293-7.

[3] H. Caesar et al., "nuscenes: A multimodal dataset for autonomous driving," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621-11631. https://doi.org/10.1109/CVPR42600.2020.01164.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in 2012 IEEE conference on computer vision and pattern recognition, 2012: IEEE, pp. 3354-3361. https://doi.org/10.1109/CVPR.2012.6248074.

[5] P. Voigtlaender et al., "Mots: Multi-object tracking and segmentation," in Proceedings of the ieee/cvf conference on computer vision and pattern recognition, 2019, pp. 7942-7951. https://doi.org/10.1109/CVPR.2019.00813.

[6] F. Yu et al., "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2636-2645. https://doi.org/10.48550/arXiv.1805.04687.

[7] Y. Cui, C. Zeng, X. Zhao, Y. Yang, G. Wu, and L. Wang, "Sportsmot: A large multi-object tracking dataset in multiple sports scenes," in Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 9921-9931. https://doi.org/10.48550/arXiv.2304.05170.

[8] P. Sun et al., "Dancetrack: Multi-object tracking in uniform appearance and diverse motion," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 20993-21002. https://doi.org/10.48550/arXiv.2111.14690.

[9] Y. Yuan, U. Iqbal, P. Molchanov, K. Kitani, and J. Kautz, "Glamr: Global occlusion-aware human mesh recovery with dynamic cameras," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 11038-11049. https://doi.org/10.1109/cvpr52688.2022.01076.

[10] W. Villegas-Ch, J. García-Ortiz, and S. Sánchez-Viteri, "Toward intelligent monitoring in IoT: AI applications for real-time analysis and prediction," IEEE Access, vol. 12, pp. 40368-40386, 2024. https://doi.org/10.1109/ACCESS.2024.3376707.

[11] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 9686-9696. https://doi.org/10.48550/arXiv.2203.14360.

[12] Y. Zhang et al., "Bytetrack: Multi-object tracking by associating every detection box," in European conference on computer vision, 2022: Springer, pp. 1-21. https://doi.org/10.48550/arXiv.2110.06864.

[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960. https://doi.org/10.1115/1.3662552.

[14] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," arXiv preprint arXiv:2111.00396, 2021. https://doi.org/10.48550/arXiv.2111.00396.

[15] J. T. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," arXiv preprint arXiv:2208.04933, 2022. https://doi.org/10.48550/arXiv.2208.04933.

[16] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?,"

Advances in neural information processing systems, vol. 30, 2017. https://doi.org/10.48550/arXiv.1703.04977.
[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961-2969. https://doi.org/10.1109/TPAMI.2018.2844175.
[18] D. Du et al., "VisDrone-DET2019: The vision meets drone object detection in image challenge results," in Proceedings of the IEEE/CVF international conference on computer vision workshops, 2019, pp. 0-0. https://doi.org/ 10.1007/978-3-030-66823-5_42.
[19] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in 2016 IEEE international conference on image processing (ICIP), 2016: Ieee, pp. 3464-3468. https://doi.org/10.1109/ICIP.2016.7533003.
[20] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in 2017 IEEE international conference on image processing (ICIP), 2017: IEEE, pp. 3645-3649. https://doi.org/10.1109/ICIP.2017.8296962.
[21] Y. Zhang, Y. Liang, J. Wang, H. Zhu, and Z. Wang, "Enhanced multi-object tracking via embedded graph matching and differentiable Sinkhorn assignment: addressing challenges in occlusion and varying object appearances," The Visual Computer, pp. 1-19, 2025. https://doi.org/10.1007/s00371-024-03772-x.
[22] W. Li, B. Li, J. Wang, W. Meng, J. Zhang, and X. Zhang, "ROMOT: Referring-expression-comprehension open-set multi-object tracking," The Visual Computer, vol. 41, no. 4, pp. 2425-2437, 2025. https://doi.org/10.1007/s00371-024-03544-7.
[23] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 941-951. https://doi.org/10.1109/ICCV.2019.00103.
[24] C. Xiao et al., "Motiontrack: Learning motion predictor for multiple object tracking," Neural Networks, vol. 179, p. 106539, 2024. https://doi.org/10.1016/j.neunet.2024.106539.
[25] P. Sun et al., "Transtrack: Multiple object tracking with transformer," arXiv preprint arXiv:2012.15460, 2020. https://doi.org/10.48550/arXiv.2012.15460.
[26] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," Neurocomputing, vol. 452, pp. 48-62, 2021. https://doi.org/10.1016/j.neucom.2021.03.091
[27] A. Sawicki, "Behavioral gait biometrics in VR: Is the use of synthetic samples able to increase person identification metrics?," Computer Animation and Virtual Worlds, vol. 36, no. 2, p. e70016, 2025. https://doi.org/10.1002/cav.70016.
[28] C. Xiao, Q. Cao, Z. Luo, and L. Lan, "MambaTrack: A Simple Baseline for Multiple Object Tracking with State Space Model," presented at the Proceedings of the 32nd ACM International Conference on Multimedia , 2024. https://doi.org/10.1145/3664647.3680944.
[29] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in First conference on language modeling, 2024. https://doi.org/10.48550/arXiv.2312.00752.
[30] B. Hu, R. Luo, Z. Liu, C. Wang, and W. Liu, "Trackssm: A general motion predictor by state-space model," arXiv preprint arXiv:2409.00487, 2024. https://doi.org/10.48550/arXiv.2409.00487
[31] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. L. Fairmot, "On the fairness of detection and re-identification in multiple object tracking., 2021, 129," DOI: https://doi. org/10.1007/s11263-021-01513-4, pp. 3069-3087. https://doi.org/10.1007/s11263-021-01513-4, 3069-3087.
[32] J. Pang et al., "Quasi-dense similarity learning for multiple object tracking," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 164-173. https://doi.org/ 10.48550/arXiv.2006.06664
[33] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould, "Probabilistic tracklet scoring and

inpainting for multiple object tracking," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 14329-14339. https://doi.org/10.1109/CVPR46437.2021.01410.

[34] W. Lv, Y. Huang, N. Zhang, R.-S. Lin, M. Han, and D. Zeng, "Diffmot: A real-time diffusion-based multiple object tracker with non-linear prediction," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2024, pp. 19321-19330. https://doi.org/ 10.1109/CVPR52733.2024.01828.

[35] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," Artificial intelligence, vol. 293, p. 103448, 2021. https://doi.org/10.1016/j.artint.2020.103448.

[36] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," EURASIP Journal on Image and Video Processing, vol. 2008, no. 1, p. 246309, 2008. https://doi.org doi:10.1155/2008/246309

[37] J. Luiten et al., "Hota: A higher order metric for evaluating multi-object tracking," International journal of computer vision, vol. 129, no. 2, pp. 548-578, 2021. https://doi.org/10.1007/s11263-020-01375-2.

[38] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in European conference on computer vision, 2016: Springer, pp. 17-35. https://doi.org/10.1007/978-3-319-48881-3_2.

[39] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," arXiv preprint arXiv:2107.08430, 2021. https://doi.org/10.48550/arXiv.2107.08430.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587. https://doi.org/10.1109/CVPR.2014.81.

[41] Z. Liu, X. Wang, C. Wang, W. Liu, and X. Bai, "Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth," IEEE Transactions on Circuits and Systems for Video Technology, 2025. https://doi.org/10.1109/tcsvt.2024.3524670.

[42] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in European conference on computer vision, 2020: Springer, pp. 474-490. https://doi.org/10.1007/978-3-030-58548-8_28.

[43] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 12352-12361. https://doi.org/10.48550/arXiv.2103.08 808.

[44] X. Zhou, T. Yin, V. Koltun, and P. Krähenbühl, "Global tracking transformers," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 8771-8780. https://doi.org/ 10.48550/arXiv.2203.13250.

[45] Y. Zhang, T. Wang, and X. Zhang, "Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 22056-22065.

[46] R. Gao and L. Wang, "MeMOTR: Long-term memory-augmented transformer for multi-object tracking," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 9901-9910.