MULTIPLE OBJECT TRACKING VIA PIXEL-LEVEL ASSOCIATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing tracking-by-query methods for Multi-Object Tracking rely heavily on the quality of the track query, which is typically generated based on detections. However, detections may contain background and parts of neighboring objects, contaminating the track query and reducing its consistency throughout the tracking propagation. This problem becomes particularly severe under long-term or severe occlusion. To address these issues, we propose PATrack, a pixel-level association framework that constructs the track query from high-quality object masks. Specifically, PATrack suppresses the effects of background features by constructing the track query using objects' masks and reduces contamination of the track query from neighboring objects through the calibration of the objects' masks. It further roughly estimates the occluded regions of object masks and suppresses unreliable trajectory generation for heavily occluded objects, thereby improving robustness to long-term and severe occlusion. PATrack achieves state-of-the-art performance on DanceTrack and SportsMOT, while delivering significant performance on MOT17, demonstrating its extremely high reliability and superiority.

1 INTRODUCTION

Multiple Object Tracking (MOT) is to accurately locating all the objects in the video stream while maintaining the identity of these objects consistently throughout the entire stream. As a fundamental task in computer vision, MOT is widely applied in various scenarios such as autonomous driving (Papaioannidis et al., 2022), intelligent surveillance (Wu et al., 2024), and industrial production (Wei et al., 2024). Early MOT followed the tracking-by-detection (TBD) paradigm (Li et al., 2024; Cao et al., 2023; Zhang et al., 2022; Bewley et al., 2016; Maggiolino et al., 2023). The TBD paradigm first detects objects within video frames through a detector, and then calculates the correlation between objects based on the spatial cues (Wang et al., 2025; Du et al., 2023) or appearance cues (Aharon et al., 2022; Yang et al., 2024; Wojke et al., 2017; Ren et al., 2023) to establish the association of objects across frames. These methods have benefited from the rapid development of object detection, but their performance is limited when dealing with specific scenarios such as occlusion, irregular movement of objects, camera perspective changes, and similar appearances of objects.

To address these issues, recent approaches (Zeng et al., 2022; Zhang et al., 2023; Gao et al., 2025a) have proposed a new paradigm based on the Transformer, called the tracking-by-query (TBQ) paradigm. These methods track objects by propagating the objects' track query features throughout the video stream. Thus, they are highly dependent on the reliability of the track query. However, these TBQ methods built on detection models still encounter some problems in scenarios with severe or long-term occlusion, as shown in Figure 1, which depicts the problems faced by the TBQ method based on the detection model in occlusion scenarios: The features generated by the detection model not only contain the features of the object but may also include the features of the background and other objects. When using these low-quality object features to update or initialize the track query, it will reduce the consistency level of its track query throughout the tracking propagation, thereby reducing the tracking performance of trackers. Although some approaches (Gao & Wang, 2023; Liu et al., 2023) have proposed methods such as long-term memory augmentation and refined the object track query to alleviate these issues, these approaches still encounter significant challenges.

In this paper, we propose a novel framework, PATrack, based on pixel-level association, which can effectively address these issues. The core idea of PATrack is to initialize and update the track query

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

using object features that exclude background pixels and regions belonging to neighboring objects, thereby improving the consistency of the objects' track query features throughout the tracking propagation, and thus addressing the severe challenges faced by existing TBQ methods when dealing with long-term occlusion and severe occlusion.



Figure 1: **Illustration of background and other objects in occlusion.** In crowded scenes, detections typically include the background region and other object parts.

To achieve this goal, we adopt SAM3 (Carion et al., 2025) as the base model to generate object masks and construct the track query from the corresponding mask features, effectively suppressing background interference. However, this also brings additional challenges because the object mask itself does not contain the data of the objects' occluded parts, resulting in the incompleteness of the object in tracking. Moreover, SAM3 generates object masks that may contain pixels from neighboring objects. To solve these problems, we first propose the Holistic Object Estimation (HOE) module, which employs an auxiliary detector to estimate the approximate complete spatial extent of each object, including its occluded regions, and establishes an association between the holistic object and its pixel-level mask. To further improve the quality of the pixel-level mask and enhance the consistency of the track query throughout the tracking propagation, we introduce the Pixel Mask Calibration (PMC) module, which can eliminate the pixels of the object in the mask that belong to other objects and improve the quality of the mask by calibrating the pixel-level mask of objects. To enable PATrack to better cope with the challenges brought by long-term occlusion and severe occlusion, we design the Occlusion-Aware Interaction (OAI) module, which can suppress the initialization and updates of the unreliable track query belonging to severely occluded objects to further improve the consistency of the objects' track query features throughout the tracking propagation. These designs work together to enhance the quality of the pixel-level masks, consequently improving the consistency level of the objects' track query features generated by these masks throughout the tracking propagation. This enables the tracker to maintain stable performance over long-term spans and effectively handle both severe and long-term occlusion scenarios. In this paper, our main contributions can be summarized as follows:

- We proposed a novel framework, PATrack, based on pixel-level association, which can effectively address the challenges posed by long-term and severe occlusion.
- We designed the Holistic Object Estimation module to estimate the objects' complete spatial extent corresponding to pixel-level masks and establish the correlation between objects and pixel-level masks.
- We designed the Pixel Mask Calibration module to enhance the quality of objects' pixel-level masks and the Occlusion-Aware Interaction module to mitigate the negative impact of severe occlusion on the objects' track query. Together, these modules enhance the tracker's robustness to both severe and long-term occlusion.
- Our PATrack achieves new state-of-the-art performance on the DanceTrack, SportsMOT, and MOT17 benchmarks, demonstrating strong improvements in both tracking accuracy and identity preservation.

2 Related Works

Tracking-by-Detection. The tracking-by-detection (TBD) paradigm remains the dominant pipeline in MOT, where objects are detected first and then associated across frames through data association. Many approaches (Cao et al., 2023; Zhang et al., 2022; Bewley et al., 2016) utilize motion cues as the basis for data association. They employ motion models, such as Kalman Filters (KF) (Kalman et al., 1960), to estimate the motion of objects during tracking. There are also some approaches (Wojke et al., 2017; Aharon et al., 2022; Gao et al., 2025b) that introduce the Re-identification (Re-ID) model (He et al., 2023; Gong et al., 2014) in the association stage, by combining the appearance

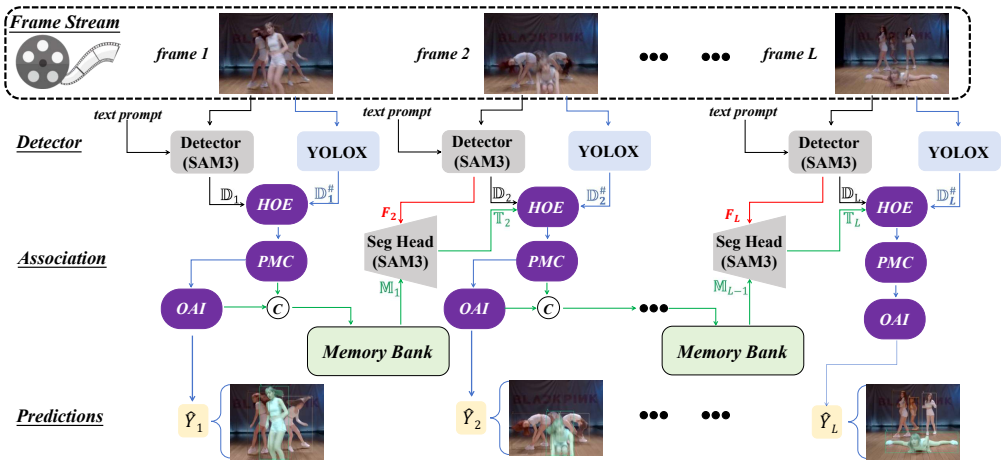


Figure 2: **Overview of our PATrack.** The **HOE** represents the Holistic Object Estimation module, the **PMC** represents the Pixel Mask Calibration module, and the **OAI** represents the Occlusion-Aware Interaction module. $\hat{Y}_L$ represents the tracking prediction results at frame L. The $\textcircled{C}$ represents the operation of set merging.

features and motion cues of the objects for tracking them. DeconfuseTrack (Huang et al., 2024) has adopted a multi-stage decision strategy for data association, thereby enhancing the accuracy of the tracker's association. SparseTrack (Liu et al., 2025) introduces the concept of pseudo-depth as a cue for data association. DiffMOT (Lv et al., 2024) abstracts the data association as a denoising diffusion process, which is used to estimate the motion of objects. Recently, Hybrid-SORT (Yang et al., 2024) uses weak cues such as object height and detection confidence as auxiliary bases for data association.

Tracking-by-Query. Recently, the tracking-by-query (TBQ) paradigm has become a promising direction in MOT, particularly for handling complex object motions. Early approaches such as MOTR (Zeng et al., 2022) and TrackFormer (Meinhardt et al., 2022) build upon the DETR (Carion et al., 2020) architecture and perform joint detection and tracking by propagating the track query across frames. MeMOT (Cai et al., 2022) builds short-term and long-term instance feature repositories to generate track queries. MOTRv2 (Zhang et al., 2023) improves detection performance by incorporating an external YOLOX (Ge et al., 2021) detector into MOTR. MeMOTR (Gao & Wang, 2023) introduces long-term memory to support long-term tracking. ColTrack (Liu et al., 2023) preserves identity consistency by applying self-attention based on the trajectory between the current query and its historical queries. MOTIP (Gao et al., 2025a) further decouples detection and association into two learnable components, providing a more interpretable and flexible query interaction framework. However, due to the risk that the track query is constructed on detections that may include background and other object features, these methods still encounter significant challenges when dealing with long-term occlusion and severe occlusion scenarios.

Although SAM2MOT (Jiang et al., 2026) incorporates the segmentation model SAM2 (Ravi et al., 2024), its track queries are still initialized and updated based on detections that may include background regions and neighboring object parts, thus failing to solve the above contamination issues. SAM3 (Carion et al., 2025) extends SAM2MOT by introducing a perception encoder (Bolya et al., 2025), enabling tracking without detection. However, the masks generated by SAM3 seriously lack the occlusion features of objects and may also contain pixels belonging to other objects. This reduces the consistency of the track query throughout the tracking propagation, which makes the tracker difficult to handle long-term occlusion and severe occlusion scenarios.

3 METHODOLOGY

3.1 OVERVIEW OF PATRACK

Figure 2 illustrates the overview of PATrack. The detections $\mathbb{D}_t$=$\{D_1, \cdots, D_s\}$ and the vision feature F_t at frame t are obtained after the image I_t and text prompt $*p$ are input to the detector of

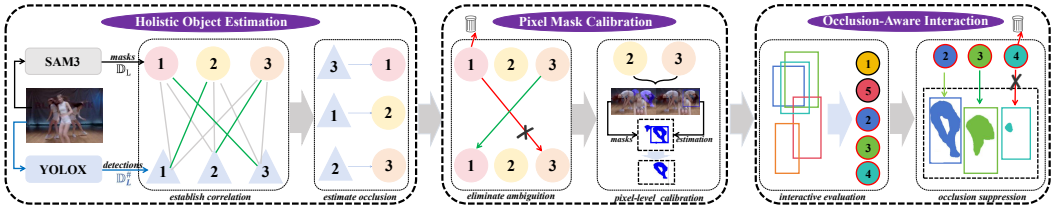


Figure 3: **Detailed illustration of proposed modules.** This figure illustrates the detailed working mechanisms of the three modules we have proposed.

SAM3. The memory bank is used to manage the memory features of trajectories. Assuming the memory bank has stored the memory features belongs to n trajectories till frame $t-1$, denoted as $\mathbb{M}_{t-1}$=$\{M_{em_1}^{t-1}, \cdots, M_{em_n}^{t-1}\}$, where $M_{em_n}^{t-1}$=$\{f_{born}^{n}, \cdots, f_{t-1}^{n}\}$ represents the selected memory features of object n (from born to now). Combining $\mathbb{M}_{t-1}$ and F_t and input them into the segmentation head of SAM3 to obtain the prediction of trajectories at frame t, denoted as $\mathbb{T}_t$=$\{T_1, \cdots, T_n\}$, where T_n=$\{M_n^T, x_n, y_n, w_n, h_n\}$.

Figure 3 illustrates the specific working processes of the HOE, PMC and OAI modules. The HOE module introduces the auxiliary detector YOLOX (Ge et al., 2021) to obtain the detection for image I_t, the results denotes as $\mathbb{D}_t^\sharp$=$\{D_1^\sharp, \cdots, D_m^\sharp\}$. $D_m^\sharp$ contains the bounding box of object m, denotes as $D_m^\sharp$=$\{x_m^\sharp, y_m^\sharp, w_m^\sharp, h_m^\sharp, c_m^\sharp\}$. Before completing the occlusion parts of objects, we eliminate the detections in $\mathbb{D}_t$ that have a high correlation with $\mathbb{T}_t$. Among these, the detections within $\mathbb{D}_t$ that have a low correlation with $\mathbb{T}_t$ are regarded as new trajectories and are added to the $\mathbb{T}_t$ set, resulting in a new set $\mathbb{T}_t'$. Next, we use the HOE module to complete the occlusion parts of the objects in $\mathbb{T}_t'$, obtaining the completed objects set $\mathbb{T}_t^H$. After that, we utilize the PMC module to calibrate the pixel-level masks of the objects in $\mathbb{T}_t^H$, eliminating the pixel points that belong to other objects and the background. The trajectories processed by the PMC module are denoted as $\mathbb{T}_t^C$. Ultimately, for each object in $\mathbb{T}_t^C$, we evaluate the occlusion degree of the object through the OAI module. For objects with severe occlusion, we block their updating of memory features in the memory bank to guarantee the consistency level of the track query throughout the tracking propagation, enhancing the stability in long-term tracking.

3.2 Holistic Object Estimation

Taking the frame t as an example, we first determine the correlation between each object in $\mathbb{D}_t$=$\{D_1, \cdots, D_s\}$ and $\mathbb{T}_t$=$\{T_1, \cdots, T_n\}$ for generating the new trajectory. Specifically, the first step is to calculate the correlation cost matrix $\mathbb{C} = \{C(D_i, T_j) | i \in [1, s], j \in [1, n]\}$ between each object in $\mathbb{D}_t$ and each object in $\mathbb{T}_t$. Taking D_s=$\{M_s^D, x_s, y_s, w_s, h_s, c_s\}$ and T_n=$\{M_n^T, x_n, y_n, w_n, h_n\}$ as an example, the correlation cost $C(D_s, T_n)$ between D_s and T_n is calculated as follows:

$$C(D_s, T_n) = \frac{\sum_{v=1}^{H}\sum_{u=1}^{W}\left(M_s^D \cap M_n^T\right)}{\sum_{v=1}^{H}\sum_{u=1}^{W}\left(M_s^D \cup M_n^T\right)} \tag{1}$$

where $M_n^T \in \mathbb{R}^{H \times W}$ represents the pixel-level mask of the object T_n, c_s means the detection confident score of the object D_s. v represents the first dimension coordinate in the pixel-level mask, and u represents the second dimension coordinate in the pixel-level mask.

After obtaining the correlation cost matrix $\mathbb{C}$ between $\mathbb{D}_t$ and $\mathbb{T}_t$, for the objects in $\mathbb{D}_t$ whose detection confidence score is greater than the trajectory generation threshold θ and whose correlation costs with each objects in $\mathbb{T}_t$ are lower than the association limit threshold ρ, we regard them as a new set $\mathbb{D}_t'$=$\{D_i | C(D_i, T_j) < \rho \wedge c_i > \theta, i \in [1, s], j \in [1, n]\}$. Suppose s' new trajectories are generated. Merge $\mathbb{D}_t'$ and $\mathbb{T}_t$ into a new trajectories set $\mathbb{T}_t'$=$\{T_1, \cdots, T_n, T_{n+1}, \cdots, T_{n+s'}\}$, D_s' is denoted as $T_{n+s'}$=$\{M_{n+s'}^T, x_{n+s'}, y_{n+s'}, w_{n+s'}, h_{n+s'}, c_{n+s'}\}$ at this moment. Subsequently, we established the association between the detections $\mathbb{D}_t^\sharp = \{D_1^\sharp, \cdots, D_m^\sharp\}$ obtained by YOLOX

and trajectories $\mathbb{T}'_t$, utilizing $\mathbb{D}_t^\sharp$ to estimate the occluded parts of objects and roughly complete the holistic completion of objects. Specifically, the HOE firstly calculate the correlation estimation cost matrix $\mathbb{E}$=$\{\mathcal{E}(T_i, D_j^\sharp)|i \in [1, n{+}s'], j \in [1, m]\}$ between the detection $\mathbb{D}_t^\sharp$ and the trajectory $\mathbb{T}'_t$. Take trajectory $T_{n+s'}$ and detection $D_m^\sharp$ as an example, the correlation estimation cost $\mathcal{E}(T_{n+s'}, D_m^\sharp)$ is calculated as follows:

$$\mathcal{E}(T_{n+s'}, D_m^\sharp) = (1 - \frac{\sqrt{(x_{n+s'} - x_m^\sharp)^2 + (y_{n+s'} - y_m^\sharp)^2}}{\sqrt{(w_{n+s'})^2 + (h_{n+s'})^2} + \sqrt{(w_m^\sharp)^2 + (h_m^\sharp)^2}}) \times \text{IoU}(T_{n+s'}, D_m^\sharp) \tag{2}$$

where $D_m^\sharp$=$\{x_m^\sharp, y_m^\sharp, w_m^\sharp, h_m^\sharp, c_m^\sharp\}$, and $(x_m^\sharp, y_m^\sharp)$ represent the center coordinates of the bounding box of $D_m^\sharp$. $w_m^\sharp$ and $h_m^\sharp$ respectively represent the width and height of $D_m^\sharp$, IoU$(T_{n+s'}, D_m^\sharp)$ represents the calculation of Intersection-over-Union (Everingham et al., 2010) between the bounding box of $T_{n+s'}$ and the bounding box of $D_m^\sharp$. Thereafter, based on the correlation estimation cost matrix $\mathbb{E}$, the occlusion parts of objects in $\mathbb{T}'_t$ are completed. Specifically, we employ the Hungarian (Kuhn, 1955) algorithm to obtain the optimal solution $\mathbb{S}$=$\{(T_i, D_j^\sharp)|T_i \in \mathbb{T}'_t, D_j^\sharp \in \mathbb{D}_t^\sharp\}$. The element $(T_i, D_j^\sharp)$ in $\mathbb{S}$ represents that the correlation between T_i and $D_j^\sharp$ is the highest. $D_j^\sharp$ contains more occlusion information of the object compared to T_i. $D_j^\sharp$ is regarded as the holistic estimation for T_i that includes occlusion parts. After the holistic estimation, T_i is denoted as T_i^H=$T_i \cup D_j^\sharp$=$\{M_i^T, x_i, y_i, w_i, h_i, x_j^\sharp, y_j^\sharp, w_j^\sharp, h_j^\sharp, c_j^\sharp\}$, For ease of understanding, T_i^H is further denoted as T_i^H=$\{M_i^T, x_i, y_i, w_i, h_i, x_i^H, y_i^H, w_i^H, h_i^H, c_i^H\}$. After all the trajectories complete the holistic estimation, they are recorded as new trajectories $\mathbb{T}_t^H$=$\{T_i^H|T_i \in \mathbb{S}\}$.

3.3 Pixel Mask Calibration

Occlusion and motion blur may cause objects' pixel-level masks to include pixels from other objects, leading to feature contamination and degrading the consistency of the track query throughout the tracking propagation. To address these issues, we employ the PMC module to calibrate the pixel-level mask of trajectories. Specifically, take two different completed trajectories $T_i^H \in \mathbb{T}_t^H$ and $T_j^H \in \mathbb{T}_t^H$ as examples. First, we calculate the mutual influence coefficient $\mathcal{K}_{i,j}$=IoU(T_i^H, T_j^H) for the two trajectories. For the two trajectories that satisfy the condition with $\mathcal{K}_{i,j} > \varepsilon_s$, we add them to the temporarily undetermined trajectories pair set, denoted as $\mathbb{U}$=$\{(T_i^H, T_j^H)|\mathcal{K}_{i,j} > \varepsilon_s, i \neq j\}$. The ε_s represents the overlapping suppression threshold, which is used to preliminarily assess the likelihood of mixing between the pixel-level masks of two trajectories. Subsequently, we calculate the mask comparison score matrix $\mathbb{C}_M$=$\{C_M(T_i^H, T_j^H)|T_i^H \in \mathbb{T}_t^H, T_j^H \in \mathbb{T}_t^H, i \neq j\}$ between all the different trajectories in $\mathbb{T}_t^H$. The mask comparison score $C_M(T_i^H, T_j^H)$ between trajectory T_i^H and T_j^H is calculated as follows:

$$C_M(T_i^H, T_j^H) = \frac{\sum_{v=1}^{H}\sum_{u=1}^{W} M_i^T(v,u)}{\sum_{v=1}^{H}\sum_{u=1}^{W} M_j^T(v,u)} \tag{3}$$

where $M_i^T \in \mathbb{R}^{H \times W}$ represents the pixel-level mask of trajectory T_i^H, and $M_i^T(v,u)$ represents the pixel of the mask M_i^T whose coordinates is (v,u). For trajectory T_i^H, if there exists $C_M(T_i^H, T_j^H) < \varepsilon_r$ and the element $(T_i^H, T_j^H) \in \mathbb{U}$, it indicates that trajectory T_i^H is highly likely to have been wrongly predicted by the tracker. We regard the trajectory T_i^H as an unreliable trajectory and add this trajectory to a new unreliable trajectory set $\mathcal{T}_t$ mentioned. Among them, ε_r represents the comparison ratio threshold, which combines with the overlapping suppression threshold ε_s to determine whether the trajectory is reliable. For trajectories not in the unreliable trajectories $\mathcal{T}_t$, we calculate the holistic ratio for each trajectory. Taking the trajectory T_i^H=$\{M_i^T, x_i, y_i, w_i, h_i, x_i^H, y_i^H, w_i^H, h_i^H, c_i^H\}$ as an example, the holistic ratio $H_R(T_i^H)$ is calculated as follows:

$$H_R(T_i^H) = \frac{w_i \times h_i}{w_i^H \times h_i^H} \tag{4}$$

where (x_i, y_i) represent the center coordinates of the minimum bounding box generated according to M_i^T, while w_i and h_i represent the width and height of the generated minimum bounding box, respectively. For the trajectory that satisfies the condition $H_R(T_i^H) > \gamma_c$, its predicted pixel-level mask M_i^T is regarded as being confused by the other trajectories. The pixel-level masks M_i^T of the trajectory need to be calibrated to suppress the pixel points from other trajectories or background, where γ_c represents the pixel calibration threshold. The pixel-level mask after calibration is denoted as M_i^C, and its calibration method is as follows:

$$M_i^C(v,u) = \begin{cases} M_i^T(v,u), & \text{if } \left|u - x_i^H\right| < \dfrac{w_i^H}{2} \text{ and } \left|v - y_i^H\right| < \dfrac{h_i^H}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where $M_i^C(v,u)$ represents the value of the pixel-level mask after calibration that belongs to trajectory T_i^H at the coordinate (v,u), After the calibration by the PMC module, the trajectory is updated and then denoted as T_i^H=$\{M_i^C, x_i, y_i, w_i, h_i, x_i^H, y_i^H, w_i^H, h_i^H, c_i^H\}$. Next, we utilize the calibrated pixel-level masks M_i^C to generate the memory features of trajectory T_i^H in the memory bank at frame t. Specifically, we utilize the memory encoder of SAM3 to generate the memory features f_t^i and add the memory features to the memory bank of trajectory i at frame t, denoted as $M_{em_i}^t$=$M_{em_i}^{t-1} \cup f_t^i$=$\{f_{born}^i, \cdots, f_t^i\}$.

3.4 Occlusion-Aware Interaction

Occlusion usually occurs during interactions between different objects and can easily cause the tracker to produce incorrect tracking results. This issue is particularly challenging for the tracker based on pixel-level association, as the masks of heavily occluded objects are often substantially incomplete relative to their full state. Updating the track query with such unreliable masks may degrade the consistency throughout the tracking propagation. To address this issue, we propose the OAI module, which perceives the degree of occlusion and suppresses the generation or update of the track query for heavily occluded objects, thereby improving tracking performance.

The trajectory set processed by the PMC module is then denoted as $\mathbb{T}_t^C$=$\{T_i^H | T_i^H \in \mathbb{T}_t^H, T_i^H \notin \mathcal{T}_t, H_R(T_i^H) \leq \gamma_c\}$. We first perceive the global correlation between each trajectory and the others in $\mathbb{T}_t^C$. Specifically, we first calculate the overall interaction ratio matrix of the trajectory set $\mathbb{T}_t^C$, denoted as $\mathbb{O}$=$\{O(T_i^H) | T_i^H \in \mathbb{T}_t^C\}$. Taking the trajectory T_i^H as an example, the calculation of the overall interaction ratio $O(T_i^H)$ is as follows:

$$O(T_i^H) = \frac{\displaystyle\sum_{v=1}^{H}\sum_{u=1}^{W}\left[S(T_i^H) \cap \left(\bigcup_{\substack{j=1 \\ j\neq i}}^{n} S(T_j^H)\right)\right]}{\displaystyle\sum_{v=1}^{H}\sum_{u=1}^{W} S(T_i^H)} \tag{6}$$

where $S(T_i^H) \in \mathbb{R}^{H\times W}$ represents the holistic estimation region of trajectory T_i^H, v represents the first dimension coordinate in the $S(T_i^H)$, and u represents the second dimension coordinate in the $S(T_i^H)$, and its region is defined as follows:

$$S_{(v,u)}(T_i^H) = \begin{cases} 1, & \text{if } \left|u - x_i^H\right| < \dfrac{w_i^H}{2} \text{ and } \left|v - y_i^H\right| < \dfrac{h_i^H}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where $S_{(v,u)}(T_i^H)$ represents whether the pixel point (v,u) is within the holistic estimation of T_i^H. For the trajectory T_i^H with its $O(T_i^H)$ higher than the overall interaction threshold δ, we further calculate their pixel completion ratio $PCR(T_i^H)$, which is calculated as follows:

$$PCR(T_i^H) = \frac{\displaystyle\sum_{v=1}^{H}\sum_{u=1}^{W} M_i^C(v,u)}{w_i^H \times h_i^H} \tag{8}$$

For trajectories with a pixel completion ratio lower than the completion suppression threshold λ, we set them to an unreliable state, still record these trajectories in the inactive state, and add them to the unreliable trajectory set $\mathcal{T}_t$. In the end, we measure the quality of the pixel-level mask belonging to the remaining trajectories $T_i^H \notin \mathcal{T}_t$ using Eq. 4 proposed in the PMC module. For the trajectory T_i^H whose $H_R(T_i^H)$ is less than the occlusion suppression threshold γ_o, we consider its pixel-level mask to be unreliable for updating the track query features. Therefore, we add T_i^H to the unreliable trajectory set $\mathcal{T}_t$, but at this time, we will record these trajectories as an active state at frame t.

For the trajectory $T_i^H \notin \mathcal{T}_t$, we generate the memory features f_t^i based on its pixel-level mask M_i^C, and add it to the memory bank of these trajectories at the frame t. This is similar to the method mentioned in the PMC module. We use the memory encoder in SAM3 to generate the memory features f_t^i of trajectory i. For the trajectory T_j^H in the unreliable trajectory set $\mathcal{T}_t$, we do not generate new memory features of T_j^H at frame t. At this time, the memory bank of the trajectory T_j^H is updated to $M_{em_j}^t$=$M_{em_j}^{t-1} \cup \{\emptyset\}$=$\{f_{born}^j, \cdots, \emptyset\}$.

4 EXPERIMENTS

To comprehensively evaluate the performance of PATrack in different scenarios, we evaluate our method on three widely used MOT benchmarks: DanceTrack (Sun et al., 2022) is an MOT benchmark in the field of dance. Its characteristic is that the objects have similar unified appearances and diverse movements; SportsMOT (Cui et al., 2023) is a benchmark focused on athlete tracking, characterized by its diverse movements and frequent movement of the camera; MOT17 (Milan et al., 2016) is a widely used pedestrian tracking benchmark, characterized by a large number of objects and frequent mutual occlusion. We mainly use the HOTA (Luiten et al., 2021) to evaluate our method since it provides a balanced way to measure both DetA and AssA. We also report the MOTA (Bernardin & Stiefelhagen, 2008) and IDF1 (Ristani et al., 2016) metrics in our experiments.

4.1 IMPLEMENTATION DETAILS

All experiments are conducted on a single NVIDIA RTX A6000 GPU paired with an Intel(R) Xeon(R) Silver 4314 CPU (2.40GHz). Our PATrack is built on the SAM3 (Carion et al., 2025) and YOLOX (Ge et al., 2021). To ensure a fair comparison and demonstrate the superiority of our PATrack, we used the original weights of SAM3 directly and underwent no training. The weights of the auxiliary detector YOLOX were also directly taken from the weights trained by ByteTrack (Zhang et al., 2022). Due to the performance differences of the detector across benchmarks, the trajectory generation threshold θ in the HOE module is set to 0.8 for DanceTrack and SportsMOT, while 0.6 for MOT17. All the remaining parameters are fixed across the three benchmarks to ensure a fair evaluation of the proposed modules. Specifically, in the HOE module, the association limit threshold ρ is set to 0.1. In the PMC module, the overlapping suppression threshold ε_s, comparison ratio threshold ε_r, and pixel calibration threshold γ_c are set to 0.9, 0.02, and 1.55, respectively. In the OAI module, the overall interaction threshold δ, completion suppression threshold λ, and occlusion suppression threshold γ_o are set to 0.85, 0.05, and 0.15, respectively. The text prompts in DanceTrack, SportsMOT, and MOT17 are set to "dancer", "athlete", and "pedestrian", respectively.

4.2 COMPARISONS WITH STATE-OF-THE-ART METHODS

DanceTrack. Table 1 presents the results on the DanceTrack-test set. PATrack achieved the state-of-the-art performance in all evaluation metrics. Compared with the previous state-of-the-art performance methods, in terms of the three metrics of HOTA, IDF1, and AssA, we have respectively outperformed the performance of SAM2MOT (+4.3 HOTA, +2.6 IDF1, and +3.0 AssA). In terms of the MOTA metric, we have also surpassed the previous state-of-the-art performance of TrackTrack by 0.4. Compared to the baseline methods SAM3, under the same settings, we achieved significant improvements in all evaluation metrics.

SportsMOT. Table 2 presents the results on the SportsMOT-test set. PATrack achieved the state-of-the-art performance in HOTA, IDF1, and AssA. Both the HOTA and IDF1 exceed those of the previously state-of-the-art GTA (Sun et al., 2024) by 1.9, respectively, while the AssA score

Table 1: Comparison with the state-of-the-art methods on the DanceTrack-test set. The best results are **bolded**.

Method	HOTA	IDF1	AssA	MOTA
ByteTrack	47.7	53.9	32.1	89.6
MeMOTR	68.5	71.2	58.4	89.9
OC-SORT	55.1	54.6	38.3	92.0
Hybrid-SORT	65.7	67.4	52.2	91.8
DiffMOT	62.3	63.0	47.2	92.8
SPAM	64.0	63.4	-	89.2
DTMOT	74.1	78.6	65.6	92.5
LA-MOTR	71.1	71.8	59.0	93.5
MOTIP	69.6	74.7	60.4	90.6
TrackTrack	66.5	67.8	52.9	93.6
FDTracker	70.6	-	-	92.1
SAM2MOT	75.8	83.9	72.2	88.5
TCEI	70.6	75.6	62.3	-
SAM3	76.6	85.0	74.6	85.7
PATrack(ours)	**80.1**	**86.5**	**75.2**	**94.0**

Table 2: Comparison with the state-of-the-art methods on the SportsMOT-test set. The best results are **bolded**.

Method	HOTA	IDF1	AssA	MOTA
GTR	54.5	55.8	45.9	67.9
ByteTrack	62.8	69.8	51.2	94.1
TrackFormer	63.3	72.4	61.1	74.1
MeMOTR	68.8	69.9	57.8	90.2
OC-SORT	71.9	72.2	59.8	94.5
DiffMOT	72.1	72.8	60.5	94.5
Hybrid-SORT	73.0	74.1	62.3	94.4
GTA	81.0	86.5	74.5	**96.3**
DTMOT	73.9	78.7	66.6	91.5
LA-MOTR	72.4	73.3	61.8	95.6
MOTIP	72.6	77.1	63.2	92.4
OA-SORT	73.4	74.1	62.3	94.4
TCEI	73.0	77.5	64.0	-
SAM3	72.4	83.2	73.6	79.4
PATrack(ours)	**82.9**	**88.4**	**79.6**	93.5

surpasses GTA by 5.1. Although it did not surpass the performance of GTA on MOTA, its performance still reached an advanced level. Compared with the baseline, under the same settings, all metrics of PATrack significantly outperformed those of the baseline model.

MOT17. Table 3 presents the results on the MOT17-test set. Our PATrack achieves the best HOTA, AssA, and IDF1 performance among the tracking-by-query methods. This indicates that our method performs the best in terms of association accuracy and is capable of accurately and stably tracking objects over a long-term span. Compared with the baseline SAM3, our method can effectively handle the occlusion problem, and its performance is much better than that of the baseline model. Although PATrack lags behind the TBD method in the MOTA metric, which represents the detector's performance, and also falls behind a few advanced methods (MC-SORT(Li et al., 2024), SparseTrack(Liu et al., 2025), and SMILETrack(Wang et al., 2024)) in the HOTA metric, it still significantly outperforms others in the core metrics (IDF1 and AssA) that symbolize tracking accuracy.

Table 3: Comparison with the state-of-the-art methods on the MOT17-test set under private protocol. The best results are **bolded**.

Method	HOTA	IDF1	AssA	MOTA
TBD:				
FineTrack	64.3	79.5	-	80.0
OC-SORT	63.2	77.5	63.2	78.0
Hybrid-SORT	64.0	78.7	63.5	79.9
OA-SORT	64.2	79.1	64.0	79.6
OFTrack	64.1	78.8	63.3	80.1
PlugTrack	64.2	79.0	64.4	79.2
MC-SORT	65.2	**80.9**	**65.7**	80.6
SparseTrack	65.1	80.1	65.1	81.0
SMILETrack	**65.3**	80.5	-	**81.1**
TBQ:				
MeMOTR	58.8	71.5	58.4	72.8
ColTrack	61.0	73.9	-	78.8
MOTRv2	62.0	75.0	60.6	78.6
MOTIP	59.3	71.3	57.0	75.3
LA-MOTR	62.6	73.8	63.9	**80.7**
DTMOT	61.5	75.1	62.5	73.8
SAM3	47.5	58.6	56.0	19.2
PATrack(ours)	**64.7**	**81.7**	**68.3**	74.5

4.3 Ablation Studies

Components Ablation. As shown in Table 4, the results prove the effectiveness and high efficiency of the modules proposed in PATrack. The HOE module established the correlation between objects and pixel-level masks and estimated the occlusion parts of the objects. This significantly improved the perfor-

Table 4: Components ablation on DanceTrack-val. The best results are **bolded**.

HOE	PMC	OAI	HOTA	IDF1	AssA	MOTA
			73.38	82.11	72.37	79.41
✓			77.60	85.75	75.55	87.85
✓	✓		78.72	87.12	76.72	89.35
✓		✓	79.14	87.37	76.26	91.19
✓	✓	✓	**79.75**	**88.01**	**77.19**	**91.50**

mance of the baseline (+4.2 HOTA, +3.6 IDF1, +3.2 AssA, and +8.4 MOTA). The PMC module was used to improve the quality of pixel-level masks, effectively enhancing the stability of the long-term tracking and effectively improving the tracker's performance (+1.1 HOTA, +1.4 IDF1, +1.2 AssA, and +1.5 MOTA). The OAI module was used to suppress the negative impact of severe occlusion objects on the tracker, effectively improving the consistency level of tracking query features throughout propagation (+1.5 HOTA, +1.6 IDF1, +0.7 AssA, and +3.3 MOTA). The combination of our proposed modules achieves the best performance, which proves that there is a synergistic relationship between them.

Analysis of Correlation Estimation Methods. We proposed a function for measuring the correlation between objects in the HOE module. Considering the differences in shape between the mask and the detection, we introduce the Euclidean distance as an auxiliary measurement criterion based on the IoU function. To further investigate the influence of different cues on measuring the correlation between objects, we compared our methods with several different functions (IoU(Everingham et al., 2010), GIoU(Rezatofighi et al., 2019), HMIoU(Yang et al., 2024) and DIoU(Zheng et al., 2020)).

Table 5: Comparison of different computational methods in correlation estimation cost mentioned in the HOE module. The best results are **bolded**.

Method	HOTA	IDF1	AssA	MOTA
IoU	77.28	85.42	75.04	87.67
GIoU	77.31	85.44	75.06	87.71
HMIoU	77.55	85.47	75.17	**88.56**
DIoU	76.79	84.99	75.01	86.31
Ours (Eq. 2)	**77.60**	**85.75**	**75.55**	87.85

As shown in Table 5, our method is significantly superior to the other methods. Our analysis indicates that this is because we established the intrinsic correlation between the obscured and unobscured parts of the object through Euclidean distance. At this point, Other methods focus more on measuring the correlation between different object instances. In contrast, our method focuses more on the correlation between different parts of the same object.

Pixel-level Calibration. In this section, we have explored the impact of the pixel calibration threshold γ_c on the PMC module. As shown in Table 6, as γ_c gradually increased from 1.4 to 1.6, the tracking performance of the tracker followed a trend of first rising and then falling. A more detailed observation revealed that the tracking performance of the tracker began to incrementally slow down around γ_c=1.50, and reached its peak at γ_c=1.55. This phenomenon indicates that only an appropriate pixel calibration threshold is conducive to improving the quality of object pixel-level masks in the PMC module. An excessively high value may cause the PMC module to lose its calibration function, while a too-low value may frequently trigger the PMC module, increasing the possibility of some high-quality pixel-level masks being destroyed. To achieve the best performance, the pixel calibration threshold γ_c of PATrack is set by default to 1.55 in all experiments.

Table 6: Comparison of different pixel calibration threshold γ_c values mentioned in the PMC module. The best results are **bolded**.

γ_c	HOTA	IDF1	AssA	MOTA
1.40	78.08	86.24	75.49	89.31
1.45	78.63	86.98	76.67	89.17
1.50	78.70	87.05	76.67	89.32
1.55	**78.72**	**87.12**	**76.72**	**89.35**
1.60	78.40	86.74	76.09	89.30

5 CONCLUSIONS

We introduced a novel framework, PATrack, that follows the TBQ paradigm. Unlike previous approaches, PATrack entirely utilizes high-quality pixel-level masks to generate and update the track query, thereby ensuring the consistency level of the objects' tracking query features throughout the tracking propagation. Our proposed HOE module is used to estimate the occluded parts of objects and establish the correlation between objects and pixel-level masks. The PMC module is used to improve the quality of pixel-level masks, while the OAI module is employed to mitigate the negative impact of severe occlusion on the objects' track query features throughout the tracking propagation. These components together produce highly complete trajectories and achieve state-of-the-art performance on DanceTrack and SportsMOT, while delivering significant performance on MOT17, demonstrating the effectiveness and robustness of PATrack.

AI USE STATEMENT

In this work, we used generative AI tools for **assist with translation**. We have not used generative AI tools for "**help develop theoretical models or conceptual frameworks**, **formulate mathematical claims**, **provide critical ingredients for proving mathematical claims**, **assist in the writing of proofs**, **propose or refine hypotheses**, **design or provide feedback on research methodology or experiments**, **implement methods**, **support qualitative and thematic data analysis**, **interpret results**" , and "**generate synthetic data sets** and **clean and reformat dataset**" are not applicable to this work. Additionally, we used generative AI tools for "**edit a research paper to improve readability** and **create or edit software code**". We have reviewed all AI-assisted work. The usage of generative AI in this paper is only used to assist us in debugging code and polishing the writing. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv preprint arXiv:2206.14651*, 2022.

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

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468. Ieee, 2016.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8090–8100, 2022.

Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9686–9696, 2023.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. SportsMOT: A Large Multi-Object Tracking Dataset in Multiple Sports Scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9921–9931, 2023.

Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. StrongSORT: Make Deepsort Great Again. *IEEE Transactions on Multimedia*, 25:8725–8737, 2023.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

Ruopeng Gao and Limin Wang. MeMOTR: Long-Term Memory-Augmented Transformer for Multi-Object Tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9901–9910, 2023.

Ruopeng Gao, Ji Qi, and Limin Wang. Multiple object tracking as id prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27883–27893, June 2025a.

Ruopeng Gao, Yuyao Wang, Chunxu Liu, and Limin Wang. History-aware transformation of reid features for multiple object tracking. *arXiv preprint arXiv:2503.12562*, 2025b.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

Shaogang Gong, Marco Cristani, Chen Change Loy, and Timothy M Hospedales. The re-identification challenge. In *Person re-identification*, pp. 1–20. Springer, 2014.

Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 9664–9667, 2023.

Cheng Huang, Shoudong Han, Mengyu He, Wenbo Zheng, and Yuhao Wei. Deconfusetrack: Dealing with confusion for multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19290–19299, 2024.

Junjie Jiang, Zelin Wang, Manqi Zhao, Yin Li, and DongSheng Jiang. Sam2mot: A novel paradigm of multi-object tracking by segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5388–5396, 2026.

Rudolf Emil Kalman et al. Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2):102–119, 1960.

Harold W Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Xiangyu Li, Yunchuan Qin, Ruihui Li, Guanghua Tan, Zhuo Tang, and Kenli Li. MC-SORT: A Motion Correction-Based Framework for Long-Term Multiple Object Tracking. In *ECAI 2024*, pp. 634–641. IOS Press, 2024.

Yiheng Liu, Junta Wu, and Yi Fu. Collaborative tracking learning for frame-rate-insensitive multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9964–9973, October 2023.

Zelin Liu, Xinggang Wang, Cheng Wang, Wenyu Liu, and Xiang Bai. SparseTrack: Multi-Object Tracking by Performing Scene Decomposition based on Pseudo-Depth. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking. *International Journal of Computer Vision*, 129:548–578, 2021.

Weiyi Lv, Yuhang Huang, Ning Zhang, Ruei-Sung Lin, Mei Han, and Dan 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*, pp. 19321–19330, 2024.

Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. In *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 3025–3029. IEEE, 2023.

Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-Object Tracking with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8844–8854, 2022.

Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A Benchmark for Multi-Object Tracking. *arXiv preprint arXiv:1603.00831*, 2016.

Christos Papaioannidis, Ioannis Mademlis, and Ioannis Pitas. Fast CNN-based Single-Person 2D Human Pose Estimation for Autonomous Systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1262–1275, 2022.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Hao Ren, Shoudong Han, Huilin Ding, Ziwen Zhang, Hongwei Wang, and Faquan Wang. Focus on details: Online multi-object tracking with diverse fine-grained representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11289–11298, 2023.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pp. 17–35. Springer, 2016.

Jiacheng Sun, Hsiang-Wei Huang, Cheng-Yen Yang, Zhongyu Jiang, and Jenq-Neng Hwang. Gta: Global tracklet association for multi-object tracking in sports. In *Proceedings of the Asian Conference on Computer Vision*, pp. 421–434, 2024.

Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. DanceTrack: Multi-Object Tracking in Uniform Appearance and Diverse Motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20993–21002, 2022.

Yanchao Wang, Dawei Zhang, Run Li, Zhonglong Zheng, and Minglu Li. PD-SORT: Occlusion-Robust Multi-Object Tracking Using Pseudo-Depth Cues. *IEEE Transactions on Consumer Electronics*, 71(1):165–177, 2025. doi: 10.1109/TCE.2025.3541839.

Yu-Hsiang Wang, Jun-Wei Hsieh, Ping-Yang Chen, Ming-Ching Chang, Hung-Hin So, and Xin Li. Smiletrack: Similarity learning for occlusion-aware multiple object tracking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 5740–5748, 2024.

Xuewen Wei, Pengcheng Li, Wei Tian, Delan Wei, Haosong Zhang, Wenhe Liao, and Yansheng Cao. A fast dynamic pose estimation method for vision-based trajectory tracking control of industrial robots. *Measurement*, 231:114506, 2024.

Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649. IEEE, 2017.

Hongxin Wu, Qinghou Zeng, Chen Guo, Tiesong Zhao, and Chang Wen Chen. Target-Aware Camera Placement for Large-Scale Video Surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

Mingzhan Yang, Guangxin Han, Bin Yan, Wenhua Zhang, Jinqing Qi, Huchuan Lu, and Dong Wang. Hybrid-SORT: Weak Cues Matter for Online Multi-Object Tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 6504–6512, 2024.

Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-End Multiple-Object Tracking with Transformer. In *European Conference on Computer Vision*, pp. 659–675. Springer, 2022.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In *European Conference on Computer Vision*, pp. 1–21. Springer, 2022.

Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22056–22065, 2023.

Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12993–13000, 2020.

A APPENDIX

Table 1: Comparison of different overall interaction threshold δ values mentioned in the OAI module. ↑ indicates that the higher values denote better performance. The best results for each metric are **bolded**.

δ	**HOTA**↑	**IDF1**↑	**AssA**↑	**MOTA**↑	**DetA**↑
0.75	78.75	86.87	75.53	91.25	82.24
0.80	78.72	86.88	75.46	**91.28**	82.25
0.85	**79.14**	**87.37**	**76.26**	91.19	**82.26**
0.90	78.96	87.17	75.92	91.16	82.24

Table 2: Comparison of different completion suppression threshold λ values mentioned in the OAI module. ↑ indicates that the higher values denote better performance. The best results for each metric are **bolded**.

λ	**HOTA**↑	**IDF1**↑	**AssA**↑	**MOTA**↑	**DetA**↑
0.03	78.90	87.35	76.15	90.79	81.88
0.04	78.81	87.01	75.82	91.06	82.07
0.05	**79.14**	**87.37**	**76.26**	91.19	82.26
0.06	78.40	86.05	74.73	**91.44**	**82.39**

A.1 OCCLUSION SUPPRESSION

In this section, we explored the impact of the overall interaction threshold δ and the completion suppression threshold λ in the OAI module. Table 1 and Table 2 show the tracking performance of different values of the δ and λ. The OAI module achieves the best performance with δ=0.85 and λ=0.05. The δ and λ are combined to determine the occlusion state of the object. If δ is set too high, the OAI module will lose its ability to handle occlusion, while a too low value will affect the OAI module's judgment of the objects' occlusion degree. If λ is set too high, some high-quality pixel-level masks cannot be applied to update objects' track query features, and if it is set too low, it reduces the ability of the OAI module to handle occlusion. To achieve the best results and in accordance with the principle of fairness, we have selected $\delta = 0.85$ and $\lambda = 0.05$, which achieved the best performance in DanceTrack-val, as the fixed values for PATrack across all benchmarks.

A.2 QUALITATIVE ANALYSIS

In this section, we perform a case analysis by comparing the tracking results of PATrack with its baseline model SAM3 on DanceTrack-val set and SportsMOT-val set.

On DanceTrack, as shown in Figure 1, we present some of the results for "dancetrack0026" and "dancetrack0081". SAM3 lost two objects (ID: 10 and ID: 19) from frame 89 to frame 92 in the case of "dancetrack0026". As mentioned in Sec. 1, the track query in SAM3 might contain pixels of other objects, and using these low-quality track queries to track the object will reduce the tracking robustness, leading to the situation shown in the figure, and ultimately resulting in the degradation of the tracker's performance. In contrast to this, our proposed PATrack calibrates the pixel-level masks of objects, estimates the occlusion of objects, and suppresses the severe occlusion features to update the track query, thereby enabling our PATrack to stably track the object in cases of severe occlusion. SAM3 maintained the incorrect updating of the track query (ID: 7) from frame 181 to frame 184 in the case of "dancetrack0081", leading to a decrease in subsequent tracking performance. Our PATrack effectively suppressed the updating of the incorrect track query by low-quality masks, thus having stronger robustness and enabling stable tracking in such scenarios.



Figure 1: **Qualitative comparison on the DanceTrack.**



Figure 2: **Qualitative comparison on the SportsMOT.**

On SportsMOT, as shown in Figure 2, we present some of the results for ”v_00HRwkvvjtQ_c001” and ”v_ITo3sCnpw_k_c012”. Similar to the situation in ”dancetrack0081” earlier, the original object (ID: 11) has been lost by SAM3 from frame 231 to frame 216 in the case of ””v_00HRwkvvjtQ_c001”, but the tracker is still maintaining the update of the incorrect track query, ultimately resulting in a decline in the tracker’s performance. From frame 695 to frame 698 in the ”v_ITo3sCnpw_k_c012”, two issues occurred. The first one was similar to the previous example.

Table 3: Evaluation of inference time about different components on the DanceTrack-val set.

Component	Infer Time	Ratio	$\frac{\text{Infer Time}}{\text{Frame}}$
Baseline	9342.18 s	91.51 %	366.25 ms
+Auxiliary Detector	+617.67 s	+6.05 %	+24.21 ms
+HOE	+89.49 s	+0.88 %	+3.51 ms
+PMC	+7.54 s	+0.07 %	+0.30 ms
+OAI	+151.62 s	+1.49 %	+5.94 ms
PATrack(ours)	10208.50 s	100 %	400.21 ms

The incorrect track query for the objects (ID: 8 and ID: 20) still kept updating in SAM3. The second case occurred at frame 695. The mask of the object (ID: 16) contains some pixels of the object (ID: 11), which causes errors in the tracker's tracking results. Moreover, using such a low-quality mask to update the track query of the object will reduce the consistency level of the track query throughout the tracking propagation, potentially leading to tracking failure. In this situation, our PATrack effectively avoided this error by correcting the pixel-level mask of objects. Furthermore, because we use high-quality masks to update the track query of each object, this makes PATrack less likely to produce low-quality masks throughout the tracking propagation, thereby addressing such issues.

A.3 Computational Overhead

In this section, we discuss the computational overhead of different components in PATrack.

Settings. All experiments were conducted using an Intel(R) Xeon(R) Silver 4314 and a single NVIDIA RTX A6000. As mentioned in the main text, the baseline model is SAM3 and the auxiliary detector is YOLOX. All the parameters in our proposed modules are consistent with the main text.

Metrics. As shown in Table 3, "Infer time" represents the total inference time of each component on the DanceTrack-val set, where the DanceTrack-val set contains 25,508 frames. "Ratio" indicates the proportion of the inference time for each component based on the inference time of the PATrack. "$\frac{\text{Infer Time}}{\text{Frame}}$" reports the average inference time per frame for each component.

Analysis. The results in Table 3 indicate that the computational overhead of PATrack is mainly concentrated in two components: the baseline model and the auxiliary detector. The additional overhead introduced by our three proposed modules is negligible, and their total overhead is less than 2.5% of the total inference time. Even including the additional auxiliary detector we introduced, the total cost of the calculation accounts for less than 9.5% of the overall inference time. In terms of the FPS, considering the tracker as a whole, the FPS calculated with only the baseline model is approximately 2.73, while the FPS of the full PATrack is approximately 2.50, representing a decrease of only 0.23. These results, combined with results in Sec. 4, demonstrate that our three proposed components introduce minimal computational overhead but yield substantial improvements in overall tracking performance, thereby further demonstrating the effectiveness and superiority of PATrack.

A.4 Limitation

In this section, we discussed the limitations of our PATrack. Despite the strong performance of PATrack, like other methods following the TBQ paradigms that are limited by their large computational requirements, its efficiency is constrained by computing resources. Our PATrack is built based on the SAM3 model. As illustrated in the appendix A.3, the SAM3 itself requires a significant amount of computing resources. Fortunately, the various modules we proposed consume as little computational overhead as possible. But computational overhead still remains a major constraint for the application of our method in practical scenarios. To solve this problem and make our method better adapted to actual application scenarios, we may consider distilling the SAM3 into a relatively lightweight model in the future, which will ensure the tracking accuracy of our method while also meeting the real-time requirements. Another limitation is that our method introduces an additional auxiliary detector, which brings additional computational burden to the entire system. In future work, we may build additional detection heads instead of additional detectors to reduce the computational overhead of our method.