

Segmentation-Guided Association Refinement in Multiple Object Tracking

Manuel Bendaña^{a,*}, Victor M. Brea^a, Manuel Mucientes^a

^aCentro Singular de Investigación en Tecnoloxías Intelixentes, Universidade de Santiago de Compostela, Santiago de Compostela, Spain

Abstract

Multiple Object Tracking (MOT) aims to detect all objects in a video sequence and maintain consistent identities across frames. While Tracking-by-Detection (TbD) remains the dominant paradigm due to its effectiveness, its reliance on object detectors makes it vulnerable to failures, particularly in occlusion scenarios. These failures often result in fragmented trajectories and degraded tracking performance. In this work, we propose SGAR-MOT—which stands for Segmentation-Guided Association Refinement in Multiple Object Tracking—a novel framework that enhances TbD pipelines by recovering object tracks that a conventional pipeline would consider lost due to detection errors. SGAR-MOT introduces the Object Recovery Protocol (ORP), which leverages segmentation masks and integrates a video segmentation method with a Vision Transformer-based module, the Mask Instance Resolver (MIR), to assess track continuity. In this way, SGAR-MOT incorporates mask-level evidence and a learned identity-consistency verification step

*Corresponding author

Email addresses: `manuel.bendana.gomez@usc.es` (Manuel Bendaña), `victor.brea@usc.es` (Victor M. Brea), `manuel.mucientes@usc.es` (Manuel Mucientes)

through MIR to determine whether a track should be recovered. Experimental results across multiple benchmarks, including MOT20, SportsMOT and VisDrone, demonstrate that SGAR-MOT consistently outperforms its baseline tracker and achieves competitive results against state-of-the-art methods. These improvements highlight the effectiveness of integrating segmentation-guided reasoning into a conventional TbD pipeline, providing a hybrid tracking paradigm that is more robust to detector failures. The code is available at <https://github.com/ManuelBGomez/SGAR-MOT>.

Keywords: Multiple Object Tracking, Tracking-by-Detection, Segmentation-Guided Object Recovery

1. Introduction

The task of Multiple Object Tracking (MOT) involves detecting all objects present in each frame of a video sequence and maintaining consistent identities over time. Contemporary solutions predominantly rely on deep learning techniques, including Convolutional Neural Networks (CNNs) [30], and more recently, Vision Transformers (ViT) [9, 32, 39]. MOT is applied in domains like traffic monitoring [12], video surveillance [29] and multi-camera environments [1].

Tracking-by-Detection (TbD) remains the most widely adopted and effective paradigm for addressing the MOT problem, compared to other holistic alternatives. In this approach, an object detector is employed to identify instances in each video frame, followed by a temporal association mechanism that links detections across frames to maintain object identities. Despite its effectiveness, the detection component in TbD approaches is prone to failure,

particularly in scenarios involving object occlusions. Such detector failures can lead to fragmented trajectories that ultimately degrade performance, as reflected by standard MOT evaluation metrics.

TbD approaches commonly rely on bounding boxes, which can also be employed to re-identify lost objects [31]. However, we argue that bounding boxes are suboptimal for handling such detection failures, as illustrated in Fig. 1, where objects 1 and 2 are in an overlapping situation. Bounding boxes often encompass regions that include parts of other objects—in Fig. 1a, the bounding box of object 1 contains pixels from object 2, and vice versa—, making the recovery process more difficult. As a more accurate alternative, we propose the use of segmentation masks, which isolate the visual characteristics of each individual object (Fig. 1b), by providing pixel-level localization. Moreover, mask-based representations offer richer information from which more discriminative features can be extracted for object recovery. In addition to their spatial precision, segmentation masks offer stable and characteristic shape cues that remain informative even in occlusion scenarios. This makes them suitable for reasoning about the identity of the objects when detections temporarily fail. Unlike bounding boxes—whose overlap often leads to ambiguous or noisy appearance cues—mask-level representations retain distinctive information that can support a more robust verification of whether an object is still present in the scene. Although segmentation masks are used internally for recovery, we still perform tracking with bounding boxes.

To this end, we introduce SGAR-MOT—Segmentation-Guided Association Refinement in Multiple Object Tracking—, a new MOT framework that

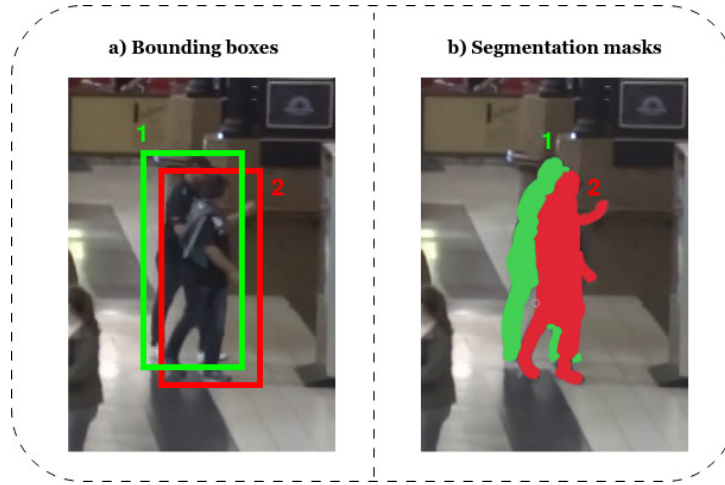


Figure 1: Example in which two objects occlude each other, extracted from MOT20 dataset: (a) bounding boxes contain information from other objects inside, (b) while segmentation masks provide more accurate and characteristic information about the target object.

extends a conventional TbD pipeline to recover object tracks that otherwise would be lost due to detector failures. A key component of SGAR-MOT is the Object Recovery Protocol (ORP), which augments the standard TbD association stage with a learned reasoning mechanism. Instead of relying on heuristics for persistence, ORP evaluates the temporal consistency of each track left unmatched by the association step through segmentation-based cues, to decide whether a track should be recovered or considered lost. The main contributions of this work are summarized as follows.

- **Object Recovery Protocol (ORP).** We introduce a module that manages tracks typically considered lost in standard TbD pipelines due to detector failures. ORP relies on segmentation masks and integrates a video segmentation method alongside a novel Vision Transformer-

based network, the Mask Instance Resolver (MIR), which determines whether a track is recovered or lost. Importantly, MIR performs identity verification through a learned spatiotemporal representation, reasoning jointly over multiple mask embeddings rather than relying on geometric thresholds or handcrafted similarity metrics.

- **SGAR-MOT Tracker.** We propose SGAR-MOT, a new tracking framework that integrates the ORP module into a conventional bounding box-based TbD architecture. This integration enables the recovery of tracks that would otherwise be lost in each frame due to detector failures, which improves robustness in occlusion scenarios through the use of segmentation masks.
- **Empirical validation.** We demonstrate that SGAR-MOT outperforms the baseline TbD tracker and achieves competitive results across multiple datasets, including widely used MOT benchmarks such as MOT20 [7], as well as more challenging scenarios like SportsMOT [6] and VisDrone [40]. Notably, SGAR-MOT achieves the best average ranking among all evaluated methods, confirming the effectiveness of segmentation masks in improving tracking performance under challenging conditions.

The remainder of this paper is organized as follows. Section 2 reviews related work and recent advances in MOT and video segmentation. Section 3 presents the SGAR-MOT framework in detail. Section 4 describes the experimental setup and reports quantitative results. Finally, Section 5 concludes the paper and outlines the directions for future research.

2. Related Work

2.1. Multiple Object Tracking

Tracking-by-Detection (TbD) remains the most prevalent and effective strategy for addressing the MOT problem, although there are alternative paradigms. For instance, end-to-end approaches, such as MOTR [36], perform the detection and association steps jointly. MOTR extends the transformer-based detector DETR [5] by incorporating an additional query mechanism for tracking. Its successor, MOTRv2 [38], improves performance by integrating a pretrained object detector to bootstrap the capabilities of MOTR. While these methods are competitive, they still lag behind the best-performing TbD-based approaches.

Recently, new transformer-based MOT paradigms that emphasize learned identity reasoning rather than heuristic association have emerged. In particular, SambaMOTR [28] introduces a synchronized set-of-sequences model capable of modeling long-range temporal dependencies and inter-tracklet interactions, demonstrating strong performance in scenarios with coordinated motion and heavy occlusions. Its capacity to propagate memory across tracklets aligns with the growing need for identity reasoning beyond geometric heuristics. On the other hand, MOTIP [14] proposes MOT as an in-context ID prediction problem, directly decoding identity labels for detections based on historical track information, eliminating handcrafted cost matrices and association rules.

Moving back to the TbD paradigm, one of the foundational approaches is SORT [3], which introduced a simple, online, and real-time tracker. SORT combines a basic object detector with an association mechanism that relies

solely on positional information to propagate object identities across frames. SORT is a reference point for numerous subsequent developments that incorporate new cues and mechanisms. DeepSORT [33] enhances SORT by including appearance information through deep learning, which improves the association process. StrongSORT [10] addresses limitations in both association and detection: it introduces an appearance-free link model to improve association robustness, and applies Gaussian-Smoothed Interpolation (GSI) to refine bounding box predictions in cases of missed detections.

OC-SORT [4], also built upon SORT, analyzes its limitations and proposes observation-centric improvements, including a re-update strategy and an additional momentum term in the association cost, to mitigate common tracking failures. HybridSORT [34] is a more recent extension of SORT that enriches the association process by incorporating additional cues. Beyond motion and appearance, it introduces three weak signals: object velocity, height state, and object confidence.

ByteTrack [37] is another relevant tracker within the TbD paradigm. It introduces a two-stage association strategy, dividing detections into high- and low-confidence groups. The first stage matches tracks with the group of high-confidence detections, whereas unmatched tracks are subsequently associated with the group of lower-confidence detections. New tracks are initialized only from unmatched detections in the first stage, improving robustness without relying on appearance features.

A complementary line of work modifies the evidence available to the tracker by introducing additional sensing modalities. MMOT [21] introduces the first benchmark for drone-based multispectral MOT, motivated by aerial

views in which small targets, occlusions and cluttered backgrounds degrade the cues that RGB trackers rely on. RGB-based trackers adapted for multi-spectral input improve over their RGB counterparts. The axis of that work is the sensing modality, while our approach keeps RGB input and instead changes the representation extracted from it.

Forecasting has also been explored to bridge occlusion gaps. Quo Vadis [8] works over a bird’s eye view, and generates trajectory predictions for tracks left without detection, improving long-term tracking as the search space for subsequent associations is restricted. JLA [18] trains tracking and forecasting together for predicting current and future locations in a single architecture, and uses the predictions to refine object locations to associate objects under occlusion. These methods predict where an object will be; our ORP module assesses whether it is still present.

Other works strengthen the association stage itself by learning richer motion or appearance models. UMA [35] combines object motion and object similarity in a single network trained with multi-task learning, joining a single-object tracking approach for motion estimation with similarity prediction to build a discriminative feature for association. DiffMOT [23] predicts non-linear motion using a diffusion-based model conditioned on each track’s historical motion information, improving the matching process. General-Track [24] introduces a relation framework that captures point-wise relations and translates them to the instance level. All of these methods focus on the matching among detections and tracks. Our work leaves the association module unchanged and focuses on recovering tracks that are lost after a detection failure, using segmentation masks to verify identity before a track is

recovered.

In line with the TbD paradigm, our work also builds upon this approach. To address the limitations caused by detector failures, we propose the integration of a dedicated module, ORP, which operates on unmatched tracks after the association step. Unlike the rest of the pipeline, which relies on bounding-box information, ORP uses segmentation masks, primarily for the richer object-level information they provide, in addition to their spatial precision. This enables a more effective characterization of the objects of interest, which is central to deciding whether a track should be recovered. This perspective positions our method within the recent shift toward models capable of performing explicit identity-consistency reasoning. Unlike end-to-end detectors or propagation-based transformers, our approach builds on the TbD framework, adding a learned verification step (MIR) for detection failure scenarios.

2.2. Segmentation

Segmentation methods aim to delineate objects within a frame by identifying the specific pixels they occupy. Early approaches to segmentation can be seen as extensions of object detection frameworks, such as Faster R-CNN [26]. Building upon this, Mask R-CNN [17] added a parallel branch dedicated to pixel-level instance segmentation, improving object localization beyond bounding boxes.

In recent years, the Segment Anything Model (SAM) [19] has emerged as one of the most relevant approaches in the field of segmentation. SAM is a class-agnostic model capable of segmenting arbitrary objects based on a variety of prompts, including bounding boxes, points, textual descriptions or

even another mask. Architecturally, SAM comprises an image encoder and a prompt encoder, which generate embeddings for the visual input and the user-provided prompt, respectively. These embeddings are then processed by a mask decoder to produce the final segmentation mask. To address potential ambiguities, SAM generates three candidate masks per prompt.

More recently, SAM 2 [25] has been introduced as an evolution of SAM, improving in both performance and versatility. It is the first model capable of handling segmentation tasks in both images and videos. While its architecture builds upon the original SAM design, it incorporates a streaming memory mechanism that enables temporal consistency across frames in video segmentation. Given a video and a prompt provided in a specific frame, SAM 2 can propagate the segmentation mask to subsequent frames.

Beyond their role in precise localization, segmentation masks also offer stable shape cues that remain informative under occlusions, complementing the mask-based identity reasoning introduced in our framework. Therefore, inspired by recent advances in segmentation models, our ORP module integrates SAM 2 to generate object masks across video frames, alongside a Vision Transformer-based network, the Mask Instance Resolver (MIR), enabling the recovery of relevant entities throughout the sequence. Section 3 details this component’s role in the overall architecture.

3. Description of the system

The architecture of SGAR-MOT is illustrated in Fig. 2. It builds upon a conventional Tracking-by-Detection (TbD) approach. Beyond that, SGAR-MOT is a hybrid system that couples box-level detection with mask-level

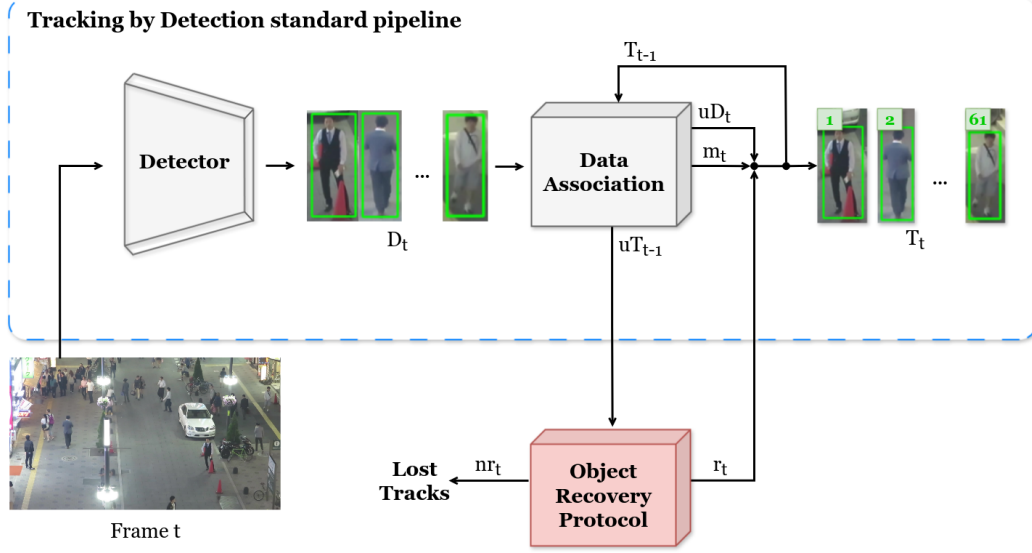


Figure 2: SGAR-MOT architecture. It includes the TbD pipeline (in blue), and the Object Recovery Protocol (ORP) stage (in red).

identity reasoning.

Given a frame at time step t , a detector identifies all objects present in the scene, producing a set of detections denoted as D_t . The detections are then matched with the tracks from the previous frame — T_{t-1} —, through an association step —Data association (Fig. 2). To perform this matching, similarity metrics are computed between the previous tracks and the current detections. These values populate a cost matrix, which is subsequently solved using the Hungarian Algorithm [20].

Tracks that are successfully **matched** with a detection — m_t — are updated to the current frame. Unmatched detections — uD_t — are used to initialize new tracks. Together, these two groups form the set of active tracks for the current frame — T_t . In the standard TbD pipeline (Fig. 2), **unmatched**

tracks resulting from the data association step $—uT_{t-1}—$ are immediately considered lost. In SGAR-MOT, this step is modified through the introduction of a dedicated module, the Object Recovery Protocol (ORP), which processes these unmatched tracks: a track is **lost** only if ORP fails to recover it. While the core pipeline relies on bounding boxes, ORP introduces segmentation masks as a complementary representation, enabling more precise and semantically rich object characterization. In particular, our framework introduces a learned verification step that evaluates identity consistency from segmentation evidence before recovering a track.

With the ORP module, some of these unmatched tracks are **recovered** $—r_t—$. In such cases, they are added to the set of tracks for the current frame $—T_t—$, alongside the matched tracks and the unmatched detections obtained from the initial data association step $—m_t$ and uD_t , respectively. On the other hand, tracks that cannot be recovered $—nr_t—$ are the so-called lost tracks. This design complements the base TbD pipeline with the capability of reasoning about object consistency in detector failure scenarios.

3.1. Object Recovery Protocol (ORP)

Fig. 3 illustrates the architecture of the ORP and its operation for a given unmatched track resulting from the initial association step. The module comprises a video segmentation method and a transformer-based architecture named Mask Instance Resolver (MIR). For video segmentation, we adopt SAM 2 [25] as the selected method.

To generate segmentation masks for the objects processed through ORP in the current frame, the method relies on the information from the previous frame. Specifically, SAM 2 requires the position of each object in the previous

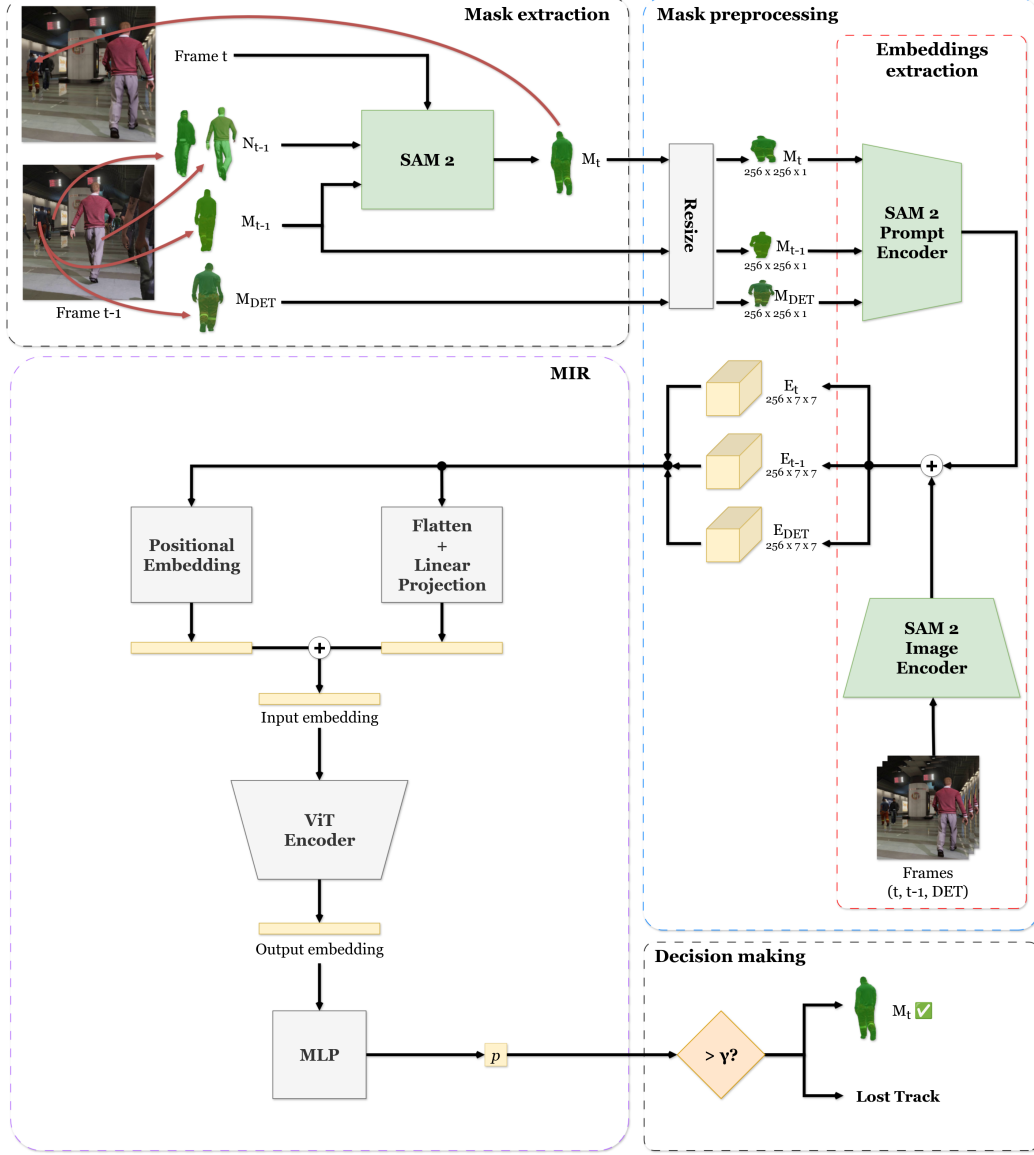


Figure 3: Object Recovery Protocol (ORP) architecture. It includes the video segmentation method (SAM 2 [25]) and the Mask Instance Resolver (MIR) transformer-based architecture with the preprocessing stage. For simplicity, the processing of only one single object is shown.

frame, along with the frame itself. The first time a track enters ORP, its position is represented by a bounding box, since up to that point the track was handled by the bounding-box-based pipeline and no mask had been generated for it. The box is taken from the previous frame, the last one in which the track was matched to a detection, and is used only to prompt SAM 2; the identity of the object is subsequently assessed by MIR from the masks. In later frames, this position is provided by the mask from the previous frame—denoted as M_{t-1} —, which SAM 2 propagates. Additionally, we provide the neighboring tracks present in the previous frame—denoted as N_{t-1} —, allowing the model to account for the spatial context surrounding each target track. The generated masks are constrained to avoid overlapping with one another, which improves spatial consistency across outputs.

Internally, SAM 2 processes this information through a sequence of components. Information from the previous frame—both from the target track and its neighboring tracks—is fed into the prompt encoder. In parallel, the image encoder receives the current frame and combines its output with the memory accumulated from previously processed frames. The resulting embeddings are then used by the mask decoder component of SAM 2 to generate segmentation masks for the target object. Notably, the output is encoded and stored in memory for future iterations—these steps are represented as SAM 2 in Fig. 3.

In ORP, SAM 2 functions as a semantic state estimator, providing temporally consistent, fine-grained evidence about the target object, beyond simple mask generation. By conditioning on the previous observation and the neighborhood context, SAM 2 supplies cues that can remain informative even when

detections fail.

Segmentation masks are generated at three distinct time points: the current frame — M_t —, the previous frame — M_{t-1} —, and the detection frame, which corresponds to a frame where the object was successfully detected and matched during the initial data association — M_{DET} ¹. Using three time-aligned masks provides complementary views of the same identity: the detection frame anchors the trajectory to a trusted past observation, the previous frame encodes the most recent state, and the current frame tests the continuity of the track. Together, these three masks provide stronger identity evidence than a single mask. The selection of the detection frame, along with other design choices related to the ORP module and SGAR-MOT, is discussed in the implementation details section (Section 4.1).

3.2. Mask Instance Resolver (MIR)

Once a segmentation mask has been generated for the target track in the current frame t , its correctness must be verified. To this end, we introduce a transformer-based architecture named Mask Instance Resolver (MIR) (Fig. 3). MIR addresses an identity verification task across multiple temporal views. A transformer encoder is well-suited to aggregate tokenized mask embeddings and model long-range dependencies and cross-view interactions, enabling the network to learn a notion of identity consistency rather than relying on fixed geometric thresholds. In this way, MIR receives as input three

¹The mask for the current frame is always generated, whereas the masks from the previous and detection frames are only required the first time the object enters ORP. In subsequent iterations, these masks are retrieved from memory to avoid redundant computation.

masks from three different frames and determines whether they correspond to the same object track. Fig. 4 illustrates an example of the input masks provided to MIR. By using the information encoded in these segmentation masks, MIR assesses their consistency and, if they represent the same object, enables the recovery of the corresponding track.

To perform this verification, MIR uses the mask from the previous frame — M_{t-1} —, which reflects the most recent prediction for the object, and the mask from the current frame — M_t —, which is relevant for maintaining the presence of the object in the scene. However, relying on the current and previous masks alone may not be sufficient to confirm the identity of the object. To enhance reliability, we incorporate information from an earlier frame —the detection frame, M_{DET} —. In this frame, the object was successfully tracked and associated with a detection during the standard tracking process. Specifically, we select the detection frame as the time step $t_{last} - 1$, where t_{last} denotes the last frame in which the object was associated in the standard tracking process.

The architecture of MIR builds upon the design of the Vision Transformer encoder [9]. Instead of feeding raw masks directly into the network, they are first preprocessed and embedded into a suitable feature representation. To accomplish this, we partially reuse components from SAM 2, specifically its encoders. The preprocessing pipeline and the main components of MIR are also illustrated in Fig. 3.

Given the three segmentation masks corresponding to the object, the preprocessing step (see Fig. 3) draws on two encoders from SAM 2: the prompt and the image encoder. While the prompt encoder can accept segmenta-



Figure 4: Three masks (in green) at different time instants for a person mostly occluded, which are the input to MIR. The real position of the person is represented by bounding boxes. The sequence belongs to the MOTSynth dataset.

tion masks as input, it requires a fixed input resolution of 256×256 pixels, rather than the original frame resolution. Therefore, an initial resizing step is applied to adapt the masks accordingly.

Once the three masks have been resized to the appropriate resolution, we proceed with the embedding extraction process (Fig. 3). Each mask is passed through the prompt encoder of SAM 2, and its output is combined with the corresponding frame representation obtained from the image encoder. This process mirrors the embedding generation strategy used in SAM 2 to feed the mask decoder. In our case, however, the resulting embeddings are used as input to MIR.

After extracting the embeddings for the detection mask — E_{DET} —, the previous frame mask — E_{t-1} — and the current frame mask — E_t —, they are flattened and projected to a suitable input size for the ViT encoder via a linear transformation —Flatten + Linear Projection (Fig. 3). Subsequently, a positional embedding is computed for each token —Positional Embedding

(Fig. 3)— and added to the projected representation. Following the approach proposed in [31], our design computes a spatiotemporal embedding $\{E^t, E^s, E^d\}$ that encodes temporal distance, object scale and spatial distance between each candidate α and a reference anchor κ . In our case, $\kappa = \{x_\kappa, y_\kappa, w_\kappa, h_\kappa, t_\kappa\}$ corresponds to the last observation of the object that was successfully matched to a detection, and includes the object coordinates in top-left-width-height format along with the time step. The embeddings for each candidate α are computed as follows:

$$E^t = \sigma^t(t_\alpha - t_\kappa) \quad (1)$$

$$E^s = \sigma^s \left(\log \left(\frac{w_\alpha}{w_\kappa} \right) + \log \left(\frac{h_\alpha}{h_\kappa} \right) \right) \quad (2)$$

$$E^d = \sigma^d \log \sqrt{\left(\frac{x_\alpha - x_\kappa}{w_\kappa} \right)^2 + \left(\frac{y_\alpha - y_\kappa}{h_\kappa} \right)^2} \quad (3)$$

where scaling factors $\sigma^t, \sigma^s, \sigma^d$ are applied to each embedding. These three values are used to compute the embeddings that form the final positional encoding added to each token [31]. This encoding scheme extends the standard spatial encoding used in ViT by incorporating object scale and temporal distance, which are particularly relevant for multi-frame object tracking, and replaces handcrafted association priors with a learnable bias: instead of hard-coded thresholds on motion or scale, MIR learns how temporal distance, relative displacement and size changes are useful for determining the identity of objects.

With the positional embedding added to the flattened embeddings — input embedding—, the transformer encoder —ViT encoder (see Fig. 3)— processes the sequence and produces an output of the same internal dimensionality —output embedding (in the same figure). This output is then

passed through a multi-layer perceptron (MLP), which maps the transformer output to a single scalar value p . This value represents the probability that all three masks correspond to the same object. Formally, MIR implements a learned decision function $f(E_{\text{DET}}, E_{t-1}, E_t) \rightarrow p \in [0, 1]$ that maps multi-time mask embeddings to an identity-consistency score. A track is recovered only if $p > \gamma$, a calibrated operating point balancing recovery gains—fewer false negatives—against the risk of identity switch. We consider that there is sufficient evidence to confirm the identity of the object across frames in that situation, recovering it as present in the current frame. Otherwise, the track is considered lost.

In practice, we observe that MIR remains reliable even when masks are imperfect, as it relies on shape evolution and context encoded in the embeddings rather than requiring high mask overlap in each frame. This robustness is essential under partial occlusions and crowded scenarios. In Section 4, visual examples of object recovery with segmentation masks are shown.

Finally, recovered tracks must be assigned a bounding box, as the main TbD tracker continues to operate using this representation. This ensures consistency with its standard association process. Although segmentation masks are used internally for object recovery, we perform tracking with bounding boxes, which remain as the final output. To address this, we estimate the bounding box by displacing the one from the previous frame according to the relative movement between the segmentation masks of the previous and current frames.

4. Experiments

4.1. Implementation details

SGAR-MOT consists of a Tracking-by-Detection standard pipeline followed by the ORP module. For the TbD pipeline, we rely on ByteTrack [37]: we adopt its association strategy and configuration parameters, as well as the detection model, YOLOX [16]. Building upon this baseline, we incorporate the ORP module to enable the recovery of object tracks that have been maintained for at least $F = 10$ consecutive frames. For video segmentation, we employ SAM 2 [25].

The MIR follows the structure of a Vision Transformer [9], specifically adopting the configuration of a ViT-Base model: 12 layers, 12 attention heads, a feed-forward output size of 3,072, and an internal embedding dimension of 768. The scaling factors of the positional embedding $\sigma^t, \sigma^s, \sigma^d$ were set to 1, 15, and 15, respectively. The transformer was trained for 600 epochs, using a batch size of 1,024, a binary cross-entropy loss function, and the SGD optimizer with a learning rate of 0.1.

The threshold γ for accepting a set of three masks as belonging to the same object was set to 0.5. All experiments were conducted on an NVIDIA A100 GPU.

4.1.1. Datasets

The MIR block was trained using samples extracted from MOTSynth [11], a synthetic dataset developed as part of the MOTChallenge. To generate training samples, we select situations in which the initial association performed by the TbD tracker fails. MOTSynth contains over 700 videos.

We generated training samples from all of them, except for a subset of 11 videos that were reserved for initial validation experiments. We selected the validation subset by choosing the videos with the poorest tracking metrics for the baseline tracker.

We tested SGAR-MOT with three datasets commonly used for MOT: MOT20 [7], which is particularly challenging due to its crowded scenes and severe occlusions, as people frequently cross paths; SportsMOT [6], which features scenes with individuals of similar appearances and dynamic movement patterns—in contrast to the more regular motion in MOT20—; and VisDrone [40], which includes smaller objects and a mix of vehicles and pedestrians.

4.2. Mask Instance Resolver analysis

The goal of this analysis is to isolate and evaluate the contribution of the transformer-based MIR module, independently from other elements of the tracking framework.

4.2.1. Precision analysis

We analyze the performance of the transformer-based MIR module across the selected datasets using the Average Precision (AP) score. We take the probabilities p provided by MIR for each sample indicating if the object should be recovered or not. Fig. 5 displays the Precision-Recall curves, as well as the AP results obtained for MOT20, VisDrone, MOTSynth, and SportsMOT. Validation subsets were used for all datasets except VisDrone, for which the test set was employed due to the availability of public annotations.

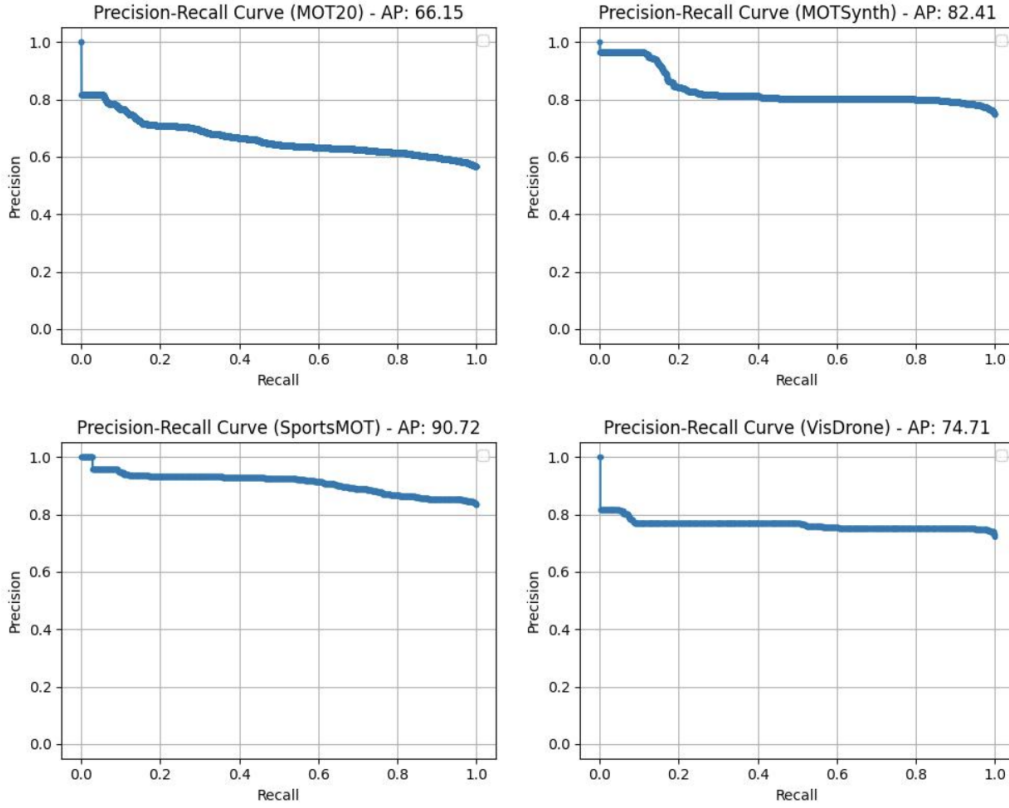


Figure 5: Precision-Recall curves of MIR in MOT20, MOTSynth and SportsMOT validation datasets, and in VisDrone test set.

MIR consistently achieves AP scores above 65% across all evaluated datasets. The lowest results are observed in the MOT20 dataset, which features highly crowded scenes and thus presents greater challenges for object recovery. However, despite its complexity, MIR still achieves a reasonably high precision. VisDrone ranks the second lowest, also characterized by dense and variable scenarios. In contrast, SportsMOT yields the highest AP score, exceeding 90%, followed by MOTSynth which scores above 80%. SportsMOT includes sports scenes with clearer object separation, while MOTSynth typi-

cally contains less crowded environments compared to the extreme conditions of MOT20. Overall, these results suggest that MIR is particularly effective in occlusion scenarios with moderate to low object density, and that it also operates under severe crowding. On average, the AP score across all datasets is approximately 78%, indicating consistent and reliable behavior of the module under diverse conditions.

To analyze the quality of the masks accepted by MIR, we used the MOT-Synth dataset. For that, we extracted the ground-truth masks that MIR should ideally accept, and compared them with the masks generated by SAM 2 and subsequently accepted by MIR. In particular, we computed the Intersection over Union (IoU) between each accepted mask and the ground-truth. As the analysis covers only masks accepted by MIR, we report the precision.

Fig. 6 shows the precision of MIR within each IoU interval. The lowest precision is obtained in the first interval, $[0, 0.01)$, which corresponds to masks with almost no overlap with the ground-truth; in those cases the precision falls to around 60%. These residual errors originate in the segmentation stage: when SAM 2 returns a mask disjoint from the target, MIR receives no valid evidence and the decision is less reliable. In the remaining intervals the precision is close to 100%, which indicates that MIR accepts a mask correctly once it covers any significant part of the target.

4.2.2. Analysis of inputs

We investigate the impact of various design choices on the performance of the MIR module regarding their inputs (Table 1). All experiments were conducted on the MOTSynth validation dataset.

One aspect we examined was the impact of varying the number and type

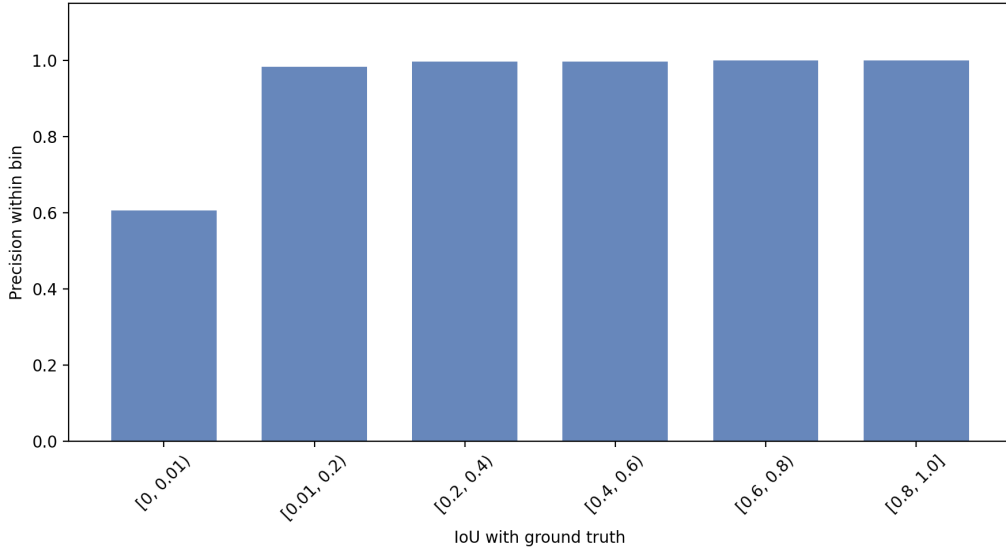


Figure 6: Precision of MIR across IoU intervals with respect to the ground-truth masks. Experiments conducted on MOTSynth validation subset.

of input masks provided to MIR (Table 1a). While the final configuration uses three masks —Det+Prev+Curr standing for detection, previous and current frame—, we also evaluated two alternative setups in which one mask was removed, resulting in only two inputs. In the first configuration —Det+Curr— the mask from the previous frame was excluded, and in the second —Prev+Curr— the detection frame mask was omitted. The results indicate that excluding the detection frame mask leads to a performance drop of 0.61 points in terms of AP, and 1.25 points in terms of $P_{R=0.5}$, the larger effect of the two. Additionally, the exclusion of the previous frame mask leads to a small decrease in performance in terms of AP —0.09 points— and of 0.61 in $P_{R=0.5}$, so the previous frame mask contributes modestly but consistently across both metrics. The three-mask configuration is best under

Transformer input	AP $\uparrow$	$P_{R=0.5}$ $\uparrow$
Det+Prev+Curr	75.83	79.59
Det+Curr	75.74	78.98
Prev+Curr	75.22	78.34

(a) Influence of different input combinations to MIR —being Det = Detection Frame, Prev = Previous Frame and Curr = Current Frame.

Det. frame	AP $\uparrow$	$P_{R=0.5}$ $\uparrow$
Previous to last det., $t_{det} = t_{last} - 1$	75.83	79.59
Last det., $t_{det} = t_{last}$	75.72	79.18
Initial det., $t_{det} = t_{init}$	74.13	79.32

(b) Influence of detection frame time (t_{det}) selection.

Table 1: Analysis of MIR with different input combinations (a) and with several detection frame times (b), on the MOTSynth validation subset. AP is the average precision; $P_{R=0.5}$ is the precision at a recall of 0.5. Higher is better for both.

both.

Another aspect we explored was the effect of selecting different time steps as the detection frame input for MIR (Table 1b). Three configurations were tested: using the last frame in which the object was associated with a detection —denoted as t_{last} in Table 1—, the frame immediately preceding it — $t_{last} - 1$ in Table 1—, and the initial frame where the object was first matched to a detection — t_{init} in Table 1. The results show that using $t_{last} - 1$ yields the best performance, and this configuration was selected for MIR. Using t_{last} can lead to overlap between the detection and previous frame inputs

—particularly during the first time a track is processed through ORP, where t_{last} corresponds to $t - 1$. This results in a slight drop of 0.11 AP points. Selecting the initial detection frame performs the worst, yielding a drop in AP of more than 1 point. We computed the median temporal distance between t_{init} and the current frame t , which is approximately 62 frames—a considerable separation that negatively impacts the ability of MIR to correctly associate masks. In contrast, the median distance between $t_{last} - 1$ and t is around 3 frames, considerably lower. On $P_{R=0.5}$, the three configurations lie within 0.41 points; this is consistent with the low sensitivity to this choice, and the selected configuration remains best on both metrics.

4.2.3. Hyperparameter analysis

We also analyze the influence of other hyperparameters that impact the final performance of SGAR-MOT (Table 2). As in previous experiments, this analysis is conducted on the MOTSynth dataset. However, in this case, our objective is to evaluate the overall behavior of the tracker, taking as reference the MOTA —Multiple Object Tracking Accuracy [2]—, HOTA —Higher Order Tracking Accuracy [22]— and IDF1 —Identification F1 score [2]— scores, which are the standard metrics used to assess MOT systems.

We first vary the number of frames F that an object must be maintained before it becomes eligible for recovery through ORP (Table 2a). We evaluate several values, ranging from 5 to 25 in increments of 5, and observe that the best performance is obtained with $F = 10$ which is therefore selected as the default configuration. Lower or higher values lead to a slight degradation in performance.

The other experiment consists of varying the γ value (Table 2b), which

F value	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FN $\downarrow$	FP $\downarrow$
5	23.97	19.20	22.32	383,739	12,407
10	23.98	19.22	22.32	384,031	12,103
15	23.98	19.18	22.28	384,267	11,812
20	23.96	18.96	22.07	384,625	11,559
25	23.94	18.96	22.14	384,770	11,509

(a) Influence of different F values.

γ value	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FN $\downarrow$	FP $\downarrow$
0.3	23.97	19.28	22.40	383,827	12,329
0.4	23.99	19.29	22.33	383,799	12,275
0.5	23.98	19.22	22.32	384,031	12,103
0.6	23.96	19.29	22.32	384,326	11,896
0.7	23.91	19.27	22.32	384,853	11,661

(b) Influence of different γ thresholds for recovering objects on MIR output.

Table 2: SGAR-MOT hyperparameter analysis on MOTSynth validation subset. Standard tracking metrics are reported for all the experiments.

controls the acceptance threshold for recovering an object through the MIR module. The standard setting is $\gamma = 0.5$, as the module addresses a binary classification problem. However, we also evaluated values between 0.3 and 0.7 to assess the sensitivity of the system to this parameter. The results show no substantial differences across the tested range, with performance starting to decline at $\gamma = 0.7$. Although the configuration with $\gamma = 0.4$ yields slightly better results, we retain the standard value of 0.5 as our final choice.

4.3. Comparison with baseline

We evaluate the performance of SGAR-MOT with standard tracking metrics —MOTA [2], HOTA [22], and IDF1 [2]. We place particular emphasis on the MOTA, as it is the most widely used indicator for evaluating the overall performance of MOT systems. However, we also consider IDF1 and HOTA, which are more closely related to association accuracy. While MOTA primarily reflects detection quality, IDF1 and HOTA provide complementary insights into the system’s ability to maintain consistent object identities over time. Additionally, we analyze the number of false positives (FP), false negatives (FN), and identity switches (IDSW), as these indicators offer valuable insight into the behavior and evolution of the main metrics. In particular, we combine FP and FN to represent the total number of errors, which have a direct impact on the standard metrics. For this comparison, we include both our baseline (ByteTrack [37]) and SGAR-MOT. Table 3 presents the results on the test sets of three datasets: MOT20, SportsMOT, and VisDrone.

Overall, SGAR-MOT demonstrates improved performance compared to our baseline. In all three datasets, the number of false negatives is reduced, primarily due to the contribution of the ORP module. While this module also introduces some false positives —by incorrectly recovering certain objects—the reduction in false negatives outweighs the increase in false positives, resulting in a net decrease in total errors. Specifically, we reduce the number of errors by more than 2,000 in MOT20 and SportsMOT, and by over 500 in the VisDrone dataset —as highlighted in the *Errors* column in Table 3. This reduction in errors is reflected in a MOTA improvement of 0.2 in SportsMOT and VisDrone, and 0.4 in MOT20.

MOT20	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FP $\downarrow$	FN $\downarrow$	Errors $\downarrow$	IDSW $\downarrow$
ByteTrack	74.0	59.2	72.6	16,749	116,927	133,676	1,069
SGAR-MOT	74.4	59.0	72.1	23,085	108,483	131,568	1,090
Difference	+0.4	-0.2	-0.5	+6,336	-8,444	-2,108	+21

(a) Comparison on MOT20 dataset.

SportsMOT	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FP $\downarrow$	FN $\downarrow$	Errors $\downarrow$	IDSW $\downarrow$
ByteTrack	93.9	62.6	69.6	28,780	30,160	58,940	3,525
SGAR-MOT	94.1	63.4	70.7	30,885	26,016	56,901	3,265
Difference	+0.2	+0.8	+1.1	+2,105	-4,144	-2,039	-260

(b) Comparison on SportsMOT dataset.

VisDrone	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FP $\downarrow$	FN $\downarrow$	Errors $\downarrow$	IDSW $\downarrow$
ByteTrack	31.2	34.2	40.2	12,061	198,533	210,594	1,260
SGAR-MOT	31.4	34.4	40.5	12,613	197,458	210,071	1,265
Difference	+0.2	+0.2	+0.3	+552	-1,075	-523	+5

(c) Comparison on VisDrone dataset.

Table 3: Comparison of SGAR-MOT with the baseline (ByteTrack). Metrics include MOTA, HOTA, IDF1, False Positives (FP), False Negatives (FN), Errors (FP+FN), and Identity Switches (IDSW).

SGAR-MOT also achieves higher HOTA and IDF1 scores for SportsMOT and VisDrone datasets. On VisDrone, HOTA improves by 0.2 and IDF1 by 0.3. On SportsMOT, the improvement is larger —0.8 points in HOTA and 1.1 in IDF1—, which we attribute to the reduction in IDSW. Owing to the ORP module, tracks that would otherwise be considered lost —and later

re-initialized with a new identity— are recovered. This reduction in IDSW is reflected in the metrics that emphasize identity preservation. On MOT20, HOTA and IDF1 decrease by 0.2 and 0.5 points, and identity switches increase by 21. This trade-off follows from a huge number of objects per frame on MOT20 —123.2 on average [7], compared to 10.8 for SportsMOT [6]. In such scenarios, each track has more neighbors, so SAM 2 is more likely to propagate a mask onto an adjacent instance and MIR to recover a drifting track, which introduces identity switches reflected in HOTA and IDF1. Even under these more difficult conditions, the effect of ORP on MOT20 remains positive, since the total number of errors is still reduced.

In summary, SGAR-MOT, through the integration of the ORP module, enhances the tracking process compared to the baseline. It successfully recovers tracks that would otherwise be considered lost, reducing the overall number of errors. It further helps prevent identity switches in cases where a track is lost and later re-initialized with a different detection. These improvements lead to increases in the common MOT metrics —MOTA, HOTA and IDF1— and contribute to more stable and accurate tracking performance across datasets.

4.4. Ablation study

To quantify the contribution of MIR to the ORP module, we report an ablation study in Table 4. This study was performed on the MOTSynth validation subset and also in MOT20 test set.

We compare the full SGAR-MOT pipeline with a configuration in which MIR is removed, leaving the recovery decision to SAM 2: a track is recovered whenever a mask is generated for it. Without MIR, recovery is more frequent

Approach	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FP $\downarrow$	FN $\downarrow$	Errors $\downarrow$	IDSW $\downarrow$
ORP without MIR	21.73	18.67	21.53	31,544	376,202	407,746	2,921
SGAR-MOT	23.98	19.22	22.32	12,103	384,031	396,134	2,753
Difference	+2.25	+0.55	+0.79	-19,441	+7,829	-11,612	-168

(a) MOTSynth validation subset.

Approach	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FP $\downarrow$	FN $\downarrow$	Errors $\downarrow$	IDSW $\downarrow$
ORP without MIR	72.06	57.36	69.97	42,477	100,877	143,324	1,220
SGAR-MOT	74.4	59.0	72.1	23,085	108,483	131,568	1,090
Difference	+2.34	+1.64	+2.13	-19,392	+7,606	-11,756	-130

(b) MOT20 test set.

Table 4: Contribution of MIR to the Object Recovery Protocol across two datasets. In ORP without MIR, a track is recovered whenever SAM 2 returns a mask.

and the number of false negatives is lower; however, false positives increase by 19,441 in MOTSynth and 19,392 in MOT20, and the total number of errors —as defined in Section 4.3— increases by more than 11,000 in both datasets with respect to SGAR-MOT. MOTA drops by 2.25 points, HOTA by 0.55 and IDF1 by 0.79 in MOTSynth. In MOT20, the drop is similar in MOTA —2.34 points— and larger in the other metrics —1.64 points in HOTA and 2.13 points in IDF1. Section 4.2.1 shows that MIR recovers tracks even when the quality of the mask provided by SAM 2 is poor —Fig. 6—, so its decisions do not reduce to a mask quality criterion.

4.5. Comparison with SoTA

Table 5 shows the comparison of SGAR-MOT with the state-of-the-art methods in MOT20, SportsMOT and VisDrone datasets. We chose meth-

ods that had been assessed in at least one of the selected datasets in their original publications, and we conducted additional experiments on the other datasets. Moreover, to show that ORP can be combined with other trackers, we also integrate it with SORT, using the same weights of the MIR module in SGAR-MOT, without retraining. These results are reported as SORT + ORP in Table 5. ORP’s design decisions and training were made with the ByteTrack data association procedure, so SORT + ORP does not exploit its full capabilities; even so, SORT + ORP improves MOTA and HOTA on all three datasets and reduces the overall number of errors.

For fairness, all experiments used the same detector and the default configuration and parameters of each method, except for the end-to-end trackers SambaMOTR [28] and MOTIP [14], which were trained on the corresponding datasets following the guidelines given by the authors.

Table 5a presents the comparison on the MOT20 test set. Our method ranks third in terms of MOTA, behind HybridSORT and QDTrack, while outperforming the remaining approaches. However, SGAR-MOT reduces identity switches by more than 300 relative to HybridSORT.

For SportsMOT (Table 5b), our method outperforms QDTrack, but still ranks third overall, behind HybridSORT and OC-SORT. Regarding the remaining metrics, SGAR-MOT exhibits a higher number of ID switches than HybridSORT. However, it achieves the lowest number of false negatives among all reported results, improving upon HybridSORT by more than 4,000 missed detections.

A final evaluation was conducted on the VisDrone test set for all trackers (Table 5c). In this scenario, ByteTrack and SGAR-MOT achieve the best

Method	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FP $\downarrow$	FN $\downarrow$	IDSW $\downarrow$
SORT [†] (ICIP 2016) [3]	73.2	61.2	75.4	13,423	124,227	1,236
SORT + ORP [†]	73.4	61.4	75.5	13,809	122,575	1,240
QDTrack (TPAMI 2023) [13]	74.7	60.0	73.8	-	-	-
OC-SORT (CVPR 2023) [4]	73.1	60.5	74.4	-	-	1,307
HybridSORT [†] (AAAI 2024) [34]	75.4	62.1	75.9	26,681	99,103	1,452
SambaMOTR [‡] (ICLR 2025) [28]	48.7	49.6	59.4	29,748	234,209	1,735
MOTIP ^{†‡} (CVPR 2025) [14]	10.5	14.2	10.2	7,207	62,405	734
ByteTrack [†] (ECCV 2022) [37]	74.0	59.2	72.6	16,749	116,927	1,069
SGAR-MOT [†] (Ours)	74.4	59.0	72.1	23,085	108,483	1,090

(a) MOT20

Method	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FP $\downarrow$	FN $\downarrow$	IDSW $\downarrow$
SORT [†] (ICIP 2016) [3]	93.0	62.2	64.9	20,879	46,019	4,642
SORT + ORP [†]	93.2	62.4	65.0	21,124	43,836	4,642
QDTrack (TPAMI 2023) [13]	90.1	60.4	62.3	-	-	-
OC-SORT (CVPR 2023) [4]	96.5	73.7	74.0	-	-	-
HybridSORT [†] (AAAI 2024) [34]	94.3	73.1	73.5	24,281	30,917	2,756
SambaMOTR [‡] (ICLR 2025) [28]	90.3	69.8	71.9	-	-	-
MOTIP [†] (CVPR 2025) [14]	92.4	72.6	77.1	-	-	-
ByteTrack [†] (ECCV 2022) [37]	93.9	62.6	69.6	28,780	30,160	3,525
SGAR-MOT [†] (Ours)	94.1	63.4	70.7	30,885	26,016	3,265

(b) SportsMOT

Method	MOTA $\uparrow$	HOTA $\uparrow$	IDF1 $\uparrow$	FP $\downarrow$	FN $\downarrow$	IDSW $\downarrow$
SORT [†] (ICIP 2016) [3]	-27.3	7.0	2.1	89,593	302,224	184
SORT + ORP [†]	-26.3	7.1	2.1	86,322	302,286	174
QDTrack [†] (TPAMI 2023) [13]	0.1	1.4	0.2	93	298,916	8,562
OC-SORT [†] (CVPR 2023) [4]	-31.8	9.0	3.1	105,900	299,368	263
HybridSORT [†] (AAAI 2024) [34]	-40.6	9.3	3.3	134,051	298,425	299
SambaMOTR ^{†‡} (ICLR 2025) [28]	9.1	27.1	26.6	33,416	245,617	835
MOTIP ^{†‡} (CVPR 2025) [14]	12.5	30.2	32.7	88,848	158,621	21,981
ByteTrack [†] (ECCV 2022) [37]	31.2	34.2	40.2	12,061	198,533	1,260
SGAR-MOT [†] (Ours)	31.4	34.4	40.5	12,613	197,458	1,265

(c) VisDrone

Table 5: Comparison of SGAR-MOT with state-of-the-art methods across three datasets. Metrics include MOTA, HOTA, IDF1, False Positives (FP), False Negatives (FN), and Identity Switches (IDSW). Best, second-best, and third-best MOTA scores are highlighted in black, blue, and red respectively. [†] indicates results produced by our own execution. All remaining results were available online. [‡] represents end-to-end methods.

performance when handling the detections produced by the YOLOX detector, whereas several other methods yielded negative MOTA scores. YOLOX generates a large number of detections with varying confidence values, including false positives. Most trackers fail to properly manage these detections, resulting in a substantial number of tracking errors. Such errors are reflected in the large count of false positives and false negatives, which in turn result in negative MOTA values. In addition, end-to-end methods exhibit intermediate performance, ranking between ByteTrack and the lowest-performing approaches. Under these conditions, SGAR-MOT ranks first, followed by ByteTrack and MOTIP, while HybridSORT is the lowest-performing tracker in this evaluation. Regarding other metrics, SGAR-MOT obtains the highest scores in both HOTA and IDF1, achieving one of the lowest counts of false negatives and the lowest number of total errors —combining false negatives and false positives.

Finally, to assess the overall performance of the selected trackers across the three datasets, we report the average ranking in Table 6, computed as an average of the individual rankings of each tracker in each dataset. Although SGAR-MOT ranks first only in VisDrone, it achieves the best global average. ByteTrack ranks second, while HybridSORT follows in third place. These results show that our method achieves the best average ranking, placing within the top three in each of the tested datasets.

We evaluated the runtime performance of SGAR-MOT on the SportsMOT dataset, which is the one with the largest number of videos and frames in the test set. On SportsMOT, the baseline runs at 15 FPS, while SGAR-MOT is still able to run at 7 FPS. Regarding the number of learned parameters,

Dataset	Ranking			
	MOT20	SportsMOT	VisDrone	Average
SORT (ICIP 2016) [3]	6	6	7	6.33
SORT + ORP	5	5	6	5.33
QDTrack (TPAMI 2023) [13]	2	9	5	5.33
OC-SORT (CVPR 2023) [4]	7	1	8	5.33
HybridSORT (AAAI 2024) [34]	1	2	9	4.00
SambaMOTR (ICLR 2025) [28]	8	8	4	6.67
MOTIP (CVPR 2025) [14]	9	7	3	6.33
ByteTrack (ECCV 2022) [37]	4	4	2	3.33
SGAR-MOT (Ours)	3	3	1	2.33

Table 6: Comparative ranking of SGAR-MOT with the state-of-the-art taking into account the results in the different test sets evaluated. We show the ranking in each dataset and the average (the lower the value, the better the position in the ranking). Best, second-best, and third-best average rankings are highlighted in black, blue, and red respectively.

SGAR-MOT adds 94.69 M parameters —MIR module— with respect to the baseline —SAM 2 is frozen. Measured end to end on SportsMOT on an NVIDIA A100 GPU in our evaluation setup, the peak allocated memory of ByteTrack was 54.27 GB, while in SGAR-MOT it rises to 57.28 GB.

To test whether the differences in tracking performance are statistically significant, we compare our method, SGAR-MOT, against the two other top-ranked trackers in Table 6, ByteTrack and HybridSORT, in a pairwise manner. Each comparison uses the per-sequence MOTA over the three datasets:

MOT20, SportsMOT and VisDrone².

We applied the Wilcoxon signed-rank test in both cases using the STAC platform [27], under the null hypothesis that the median of the per-sequence MOTA differences is zero. The null hypothesis is rejected in both comparisons with $p < 0.05$, indicating that the performance differences are statistically significant, and that SGAR-MOT is, overall, the best algorithm.

4.6. Visualizations on SGAR-MOT performance

We provide qualitative examples in Fig. 7 illustrating how SGAR-MOT utilizes segmentation masks to recover tracks that would otherwise be lost due to detector failures. These visualizations highlight the ability of the system to maintain trajectory continuity even in challenging detection scenarios.

Three frames are shown for each example, ordered from left to right, from older to more recent. The first frame —Previous situation (Fig. 7)— represents a moment in which the object was being tracked through the standard TbD pipeline, with the detector operating correctly. The second frame —Last detection (Fig. 7)— corresponds to the detection frame t_{det} , representing the object moments before the detector failure. Finally, the third frame —Recovery (Fig. 7)— illustrates the current situation, where the object has been recovered through ORP using the segmentation mask also shown in the frame. The first three examples depict successful recoveries in scenarios involving different types of occlusions. The last row illustrates

²We utilize the results per-sequence instead of per-dataset —MOT20, SportsMOT and VisDrone— in order to have enough data for the statistical test. We use the test sets of MOT20 and VisDrone, and the validation set of SportsMOT, for which per-sequence test results are not publicly available.

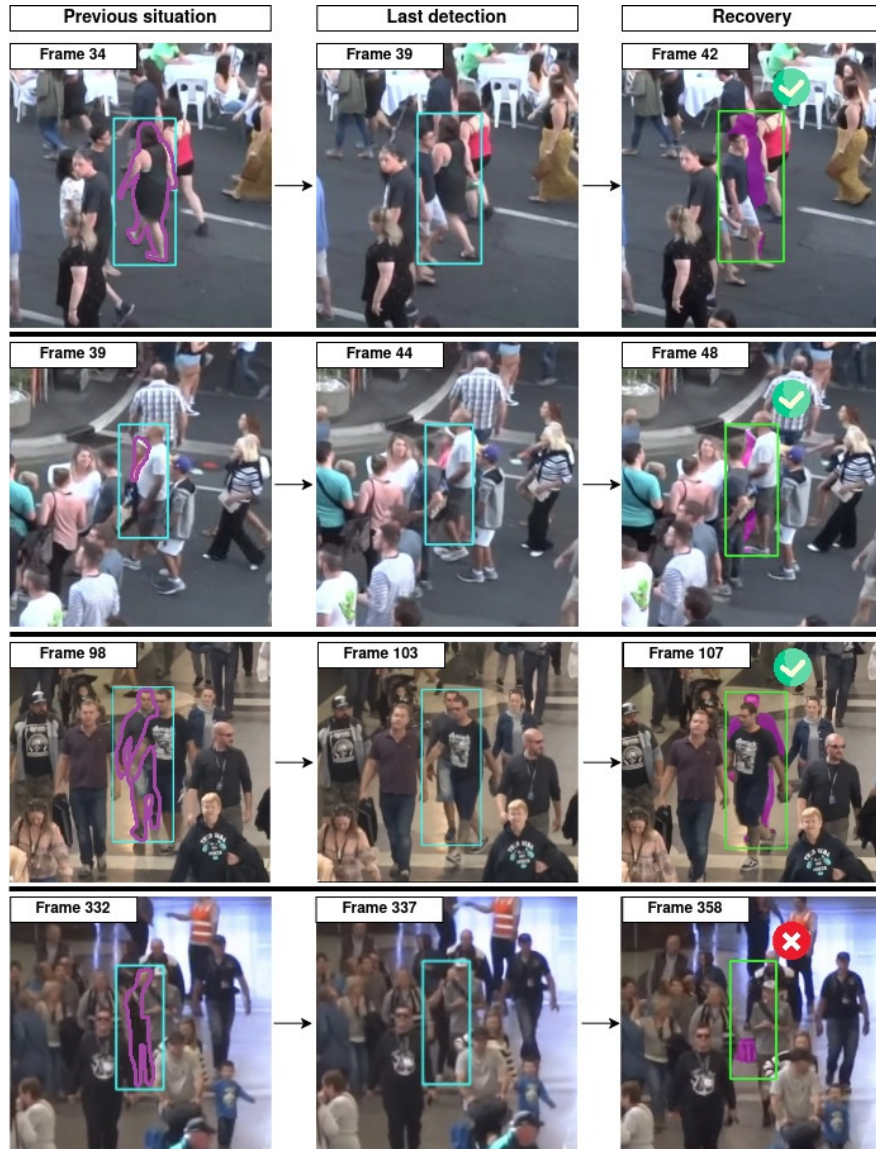


Figure 7: Examples on the use of SGAR-MOT for recovering tracks that would otherwise be lost after detector failures. Videos from MOT20 dataset.

an incorrect recovery, where the tracker maintains an object using a mask mistakenly extracted from a different instance in the scene.

5. Conclusions and future work

In this work, we introduced SGAR-MOT, a novel MOT system that builds upon a two-stage TbD framework. Our approach incorporates a new module, ORP, specifically designed to enhance tracking performance by recovering tracks that would otherwise be lost due to detector failures. The ORP module consists of two main components: a video segmentation network that generates masks for each unmatched track processed by ORP, and a transformer-based architecture named MIR that takes the segmentation masks at different time instants and decides whether they belong to the same object. Beginning from a standard TbD pipeline, SGAR-MOT integrates mask-level evidence to support decisions traditionally governed by heuristic persistence rules.

ORP provides a mechanism for verifying object identities under occlusions. By combining temporally consistent masks produced by SAM 2 with the multi-frame reasoning capabilities of MIR, the system learns to assess object permanence through latent spatiotemporal cues rather than relying solely on bounding-box geometry. In this way, SGAR-MOT couples detection, segmentation and learned identity verification within a single pipeline.

The results in terms of AP indicate that MIR successfully accepts the majority of valid recovery cases. Also, the comparison with the SoTA showed that SGAR-MOT reduced the number of tracking errors, achieving the best average ranking among the evaluated approaches in terms of MOTA. Our results confirm that incorporating segmentation masks helps to recover tracks that would otherwise be lost in a standard TbD pipeline.

These findings suggest that segmentation-guided identity reasoning ef-

fectively complements detection-based tracking, especially in settings with heavy occlusion or dense interactions. Moreover, SGAR-MOT contributes to the broader shift toward systems that integrate multiple forms of visual evidence and employ learned components to support high-level decisions typically handled with heuristics.

These observations suggest several directions for future work. For instance, the ORP module could be replaced with a fully mask-based association stage, moving towards a Tracking-by-Segmentation (TbS) paradigm. Our proposal may prove useful for generalizing MOT to open-vocabulary scenarios, which are already being successfully explored in object detection [15]. Another potential avenue could focus on correcting other types of detector failures. For example, we have observed cases in different datasets where objects are either not detected or assigned low confidence scores that lead to their direct rejection. Future refinements could aim to recover such objects and further improve tracking robustness.

Acknowledgements

This work has received financial support from the Agencia Estatal de Investigación (Spain) (grant numbers PID2023-149549NB-I00 and PID2024-155219OB-C32), the “Cátedra Televés en Diseño Microelectrónico” by the PERTE Chip (grant number TSI-069100-2023-0010), the Galician Ministry for Education, Universities and Professional Training and the “ERDF A way of making Europe” through grants “Galician Research Centre Accreditation 2024-2027 ED431G-2023/04” and “Reference Competitive Group Accreditation 2026-2029 ED431C 2026/52”. Manuel Bendaña is supported by the

Spanish Ministerio de Universidades under the FPU national plan (grant number FPU22/01828).

References

- [1] Amosa, T.I., Sebastian, P., Izhar, L.I., Ibrahim, O., Ayinla, L.S., Bashaashwan, A.A., Bala, A., Samaila, Y.A., 2023. Multi-camera multi-object tracking: A review of current trends and future advances. *Neurocomputing* 552, 126558.
- [2] Bernardin, K., Stiefelhagen, R., 2008. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP J. Image Video Process.* 2008.
- [3] Bewley, A., Ge, Z., Ott, L., Ramos, F.T., Upcroft, B., 2016. Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016, IEEE. pp. 3464–3468.
- [4] Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K., 2023. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, IEEE. pp. 9686–9696.
- [5] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-End Object Detection with Transformers, in: Computer Vision - ECCV 2020 - 16th European Conference,

- Glasgow, UK, August 23-28, 2020, Proceedings, Part I, Springer. pp. 213–229.
- [6] Cui, Y., Zeng, C., Zhao, X., Yang, Y., Wu, G., Wang, L., 2023. SportsMOT: A Large Multi-Object Tracking Dataset in Multiple Sports Scenes, in: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023, IEEE. pp. 9887–9897.
 - [7] Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I.D., Roth, S., Schindler, K., Leal-Taixé, L., 2020. MOT20: A benchmark for multi object tracking in crowded scenes. CoRR abs/2003.09003. [arXiv:2003.09003](#).
 - [8] Dendorfer, P., Yugay, V., Osep, A., Leal-Taixé, L., 2022. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking?, in: Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022.
 - [9] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.
 - [10] Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H., 2023.

StrongSORT: Make DeepSORT Great Again. *IEEE Trans. Multim.* 25, 8725–8737.

- [11] Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Osep, A., Calderara, S., Leal-Taixé, L., Cucchiara, R., 2021. MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking?, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, IEEE. pp. 10829–10839.
- [12] Fernández-Sanjurjo, M., Bosquet, B., Mucientes, M., Brea, V.M., 2019. Real-time visual detection and tracking system for traffic monitoring. *Engineering Applications of Artificial Intelligence* 85, 410–420.
- [13] Fischer, T., Huang, T.E., Pang, J., Qiu, L., Chen, H., Darrell, T., Yu, F., 2023. QDTrack: Quasi-Dense Similarity Learning for Appearance-Only Multiple Object Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 15380–15393.
- [14] Gao, R., Qi, J., Wang, L., 2025. Multiple Object Tracking as ID Prediction, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025, Computer Vision Foundation / IEEE. pp. 27883–27893.
- [15] Garcia-Fernandez, P., Vaquero, L., Liu, M., Xue, F., Cores, D., Sebe, N., Mucientes, M., Ricci, E., 2025. Superpowering Open-Vocabulary Object Detectors for X-ray Vision, in: IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, Hawaii, October 19-23.

- [16] Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021. YOLOX: exceeding YOLO series in 2021. CoRR abs/2107.08430. [arXiv:2107.08430](#).
- [17] He, K., Gkioxari, G., Dollár, P., Girshick, R.B., 2017. Mask R-CNN, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, IEEE Computer Society. pp. 2980–2988.
- [18] Kesa, O., Styles, O., Sanchez, V., 2022. Multiple object tracking and forecasting: Jointly predicting current and future object locations, in: IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACV - Workshops, Waikoloa, HI, USA, January 4-8, 2022, IEEE. pp. 560–569.
- [19] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W., Dollár, P., Girshick, R.B., 2023. Segment Anything, in: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023, IEEE. pp. 3992–4003.
- [20] Kuhn, H.W., 1955. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly* 2, 83–97.
- [21] Li, T., Xu, T., Wang, Y., Qin, H., Lin, X., Li, J., 2025. MMOT: the first challenging benchmark for drone-based multispectral multi-object tracking, in: Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS

2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025.

- [22] Luiten, J., Osep, A., Dendorfer, P., Torr, P.H.S., Geiger, A., Leal-Taixé, L., Leibe, B., 2021. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.* 129, 548–578.
- [23] Lv, W., Huang, Y., Zhang, N., Lin, R., Han, M., Zeng, D., 2024. Diffmot: A real-time diffusion-based multiple object tracker with non-linear prediction, in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024, IEEE*. pp. 19321–19330.
- [24] Qin, Z., Wang, L., Zhou, S., Fu, P., Hua, G., Tang, W., 2024. Towards generalizable multi-object tracking, in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024, IEEE*. pp. 18995–19004. URL: <https://doi.org/10.1109/CVPR52733.2024.01797>, doi:10.1109/CVPR52733.2024.01797.
- [25] Ravi, N., Gabeur, V., Hu, Y., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C., Girshick, R.B., Dollár, P., Feichtenhofer, C., 2024. SAM 2: Segment Anything in Images and Videos. *CoRR* abs/2408.00714. [arXiv:2408.00714](https://arxiv.org/abs/2408.00714).
- [26] Ren, S., He, K., Girshick, R.B., Sun, J., 2017. Faster R-CNN: Towards

- Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149.
- [27] Rodríguez-Fdez, I., Canosa, A., Mucientes, M., Bugarín, A., 2015. STAC: A web platform for the comparison of algorithms using statistical tests, in: 2015 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2015, Istanbul, Turkey, August 2-5, 2015, IEEE. pp. 1–8.
- [28] Segù, M., Piccinelli, L., Li, S., Yang, Y., Gool, L.V., Schiele, B., 2025. Samba: Synchronized Set-of-Sequences Modeling for Multiple Object Tracking, in: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025, OpenReview.net.
- [29] Tsai, C.Y., Shen, G.Y., Nisar, H., 2023. Swin-JDE: Joint Detection and Embedding Multi-Object Tracking in Crowded Scenes Based on Swin-Transformer. *Engineering Applications of Artificial Intelligence* 119, 105770.
- [30] Vaquero, L., Mucientes, M., Brea, V.M., 2020. SiamMT: Real-Time Arbitrary Multi-Object Tracking, in: 25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021, IEEE. pp. 707–714.
- [31] Vaquero, L., Xu, Y., Alameda-Pineda, X., Brea, V.M., Mucientes, M., 2024. Lost and Found: Overcoming Detector Failures in Online Multi-object Tracking, in: Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXIII, Springer. pp. 448–466.

- [32] Wang, F., Yan, H., Zhang, L., Gao, K., 2023. CSCMOT: Multi-object tracking based on channel spatial cooperative attention mechanism. *Engineering Applications of Artificial Intelligence* 126, 107084.
- [33] Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017, IEEE. pp. 3645–3649.
- [34] Yang, M., Han, G., Yan, B., Zhang, W., Qi, J., Lu, H., Wang, D., 2024. Hybrid-SORT: Weak Cues Matter for Online Multi-Object Tracking, in: Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada, AAAI Press. pp. 6504–6512.
- [35] Yin, J., Wang, W., Meng, Q., Yang, R., Shen, J., 2020. A unified object motion and affinity model for online multi-object tracking, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, Computer Vision Foundation / IEEE. pp. 6767–6776.
- [36] Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y., 2022. MOTR: End-to-End Multiple-Object Tracking with Transformer, in: Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVII, Springer. pp. 659–675.

- [37] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X., 2022. ByteTrack: Multi-object Tracking by Associating Every Detection Box, in: Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXII, pp. 1–21.
- [38] Zhang, Y., Wang, T., Zhang, X., 2023. MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, IEEE. pp. 22056–22065.
- [39] Zhou, X., Wang, C., Wang, X., Liu, W., 2024. Video text tracking with transformer-based local search. *Neurocomputing* 609, 128420.
- [40] Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q., 2018. Vision Meets Drones: A Challenge. CoRR abs/1804.07437. [arXiv:1804.07437](https://arxiv.org/abs/1804.07437).