

---

# SAMBA: SYNCHRONIZED SET-OF-SEQUENCES MODELING FOR MULTIPLE OBJECT TRACKING

**Anonymous authors**

Paper under double-blind review

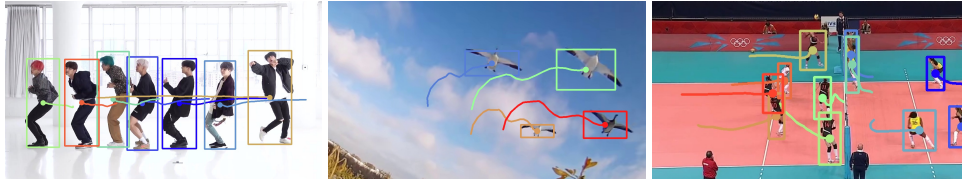
## ABSTRACT

Multiple object tracking in complex scenarios - such as coordinated dance performances, team sports, or dynamic animal groups - presents unique challenges. In these settings, objects frequently move in coordinated patterns, occlude each other, and exhibit long-term dependencies in their trajectories. However, it remains a key open research question on how to model long-range dependencies within tracklets, interdependencies among tracklets, and the associated temporal occlusions. To this end, we introduce Samba, a novel linear-time set-of-sequences model designed to jointly process multiple tracklets by synchronizing the multiple selective state-spaces used to model each tracklet. Samba autoregressively predicts the future track query for each sequence while maintaining synchronized long-term memory representations across tracklets. By integrating Samba into a tracking-by-propagation framework, we propose SambaMOTR, the first tracker effectively addressing the aforementioned issues, including long-range dependencies, tracklet interdependencies, and temporal occlusions. Additionally, we introduce an effective technique for dealing with uncertain observations (MaskObs) and an efficient training recipe to scale SambaMOTR to longer sequences. By modeling long-range dependencies and interactions among tracked objects, SambaMOTR implicitly learns to track objects accurately through occlusions without any hand-crafted heuristics. Our approach significantly surpasses prior state-of-the-art on the DanceTrack, BFT, and SportsMOT datasets. Our anonymous project page can be found at <https://anonymous-samba.github.io/>.

## 1 INTRODUCTION

Multiple object tracking (MOT) involves detecting multiple objects while keeping track of individual instances throughout a video stream. It is critical for multiple downstream tasks such as sports analysis, autonomous navigation, and media production (Luo et al., 2021). Traditionally, MOT methods are validated on relatively simple settings such as surveillance datasets (Milan et al., 2016), where pedestrians exhibit largely linear motion and diverse appearance, and rarely interact with each other in complex ways. However, in dynamic environments like team sports, dance performances, or animal groups, objects frequently move in coordinated patterns, occlude each other, and exhibit non-linear motion with long-term dependencies in their trajectories (Fig. 1). Modeling the long-term interdependencies between objects in these settings, where their movements are often synchronized or influenced by one another, remains an open problem that current methods fail to address.

Current *tracking-by-detection* methods (Bewley et al., 2016; Wojke et al., 2017; Zhang et al., 2022; Cao et al., 2023) often rely on heuristics-based models like the Kalman filter to independently model the trajectory of objects and predict their future location. However, these methods struggle with the non-linear nature of object dynamics such as motion, appearance, and pose changes. *Tracking-by-propagation* (Sun et al., 2020; Meinhardt et al., 2022; Zeng et al., 2022) offers an alternative by modeling tracking as an end-to-end autoregressive object detection problem, leveraging detection transformers (Carion et al., 2020; Zhu et al., 2020) to propagate track queries over time. Their flexible design fostered promising performance in settings with complex motion, pose, and appearance patterns, such as dance (Sun et al., 2022), sports (Cui et al., 2023), and bird (Zheng et al., 2024) tracking datasets. However, such methods only propagate the temporal information across adjacent frames, failing to account for long-range dependencies. MeMOTR (Gao & Wang, 2023) attempts a preliminary solution to this problem by storing temporal information through an external heuristics-



(a) DanceTrack (Sun et al., 2022) (b) BFT (Zheng et al., 2024) (c) SportsMOT (Cui et al., 2023)

Figure 1: **Tracking multiple objects in challenging scenarios** - such as coordinated dance performances (a), dynamic animal groups (b), and team sports (c) - requires handling complex interactions, occlusions, and fast movements. As shown in the tracklets above, objects may move in coordinated patterns and occlude each other. By leveraging the joint long-range dependencies in their trajectories, SambaMOTR accurately tracks objects through time and occlusions.

based memory. However, its use of an exponential moving average (EMA) to compress past history results in a suboptimal temporal memory representation, as it discards fine-grained long-range dependencies that are crucial for accurate tracking over time. Moreover, by processing each tracklet independently and overlooking tracklets interaction, current methods cannot accurately model objects' behavior through occlusions, resorting to naive heuristics to handle such cases: some (Zhang et al., 2023; Gao & Wang, 2023) freeze the track queries during occlusions and only rely on their last observed state during prolonged occlusions; others (Zeng et al., 2022) delegate occlusion management to the propagation module, which fails to estimate accurate track trajectories as it only propagates information across adjacent frames and does not account for historical information. We argue that effective long-term memory and interaction modeling allow for more accurate inference of occluded objects' behavior in complex environments, such as team sports or dance performances, by leveraging past information and understanding joint motion patterns.

To address these shortcomings, we propose Samba<sup>1</sup>, a novel linear-time set-of-sequences model that processes a set of sequences (*e.g.* multiple tracklets) simultaneously and compresses their histories into synchronized long-term memory representations, capturing interdependencies within the set. Samba adopts selective SSMs from Mamba (Gu & Dao, 2023) to independently model all tracklets, compressing their long-range histories into hidden states. We then propose to synchronize these memory representations across tracklets at each time step to account for interdependencies (*e.g.* interactions among tracklets). We implement synchronization via a self-attention mechanism (Vaswani et al., 2017) across the hidden states of all sequences, allowing tracklets to exchange information. This approach proves beneficial in datasets where objects move in coordinated patterns (Tabs. 1 to 3). The resulting set-of-sequences model, Samba, retains the linear-time complexity of SSMs while modeling the joint dynamics of the set of tracklets.

By integrating Samba into a tracking-by-propagation framework (Zeng et al., 2022), we present SambaMOTR, an end-to-end multiple object tracker that models long-range dependencies and interactions between tracklets to handle complex motion patterns and occlusions in a principled manner. SambaMOTR complements a transformer-based object detector with a novel set-of-queries propagation module based on Samba, which accounts for both individual tracklet histories and their interactions when autoregressively predicting the next track queries.

Additionally, some queries result in uncertain detections due to occlusions or challenging scenarios (see Fig. 2, Occlusion). To prevent these detections from compromising the memory representation and accumulating errors during query propagation with Samba, we propose MaskObs. MaskObs blocks unreliable observations from entering the set-of-queries propagation module while updating the corresponding hidden states and track queries using only the long-term memory of their tracklets and interactions with confidently tracked objects. Unlike previous methods that freeze track queries during occlusions, MaskObs leverages both temporal and spatial context - *i.e.* past behavior and interdependencies with other tracklets - to more accurately predict an object's future state. Consequently, SambaMOTR tracks objects through occlusions more effectively (Tab. 4, line d).

<sup>1</sup>Samba is named for its foundation on the Synchronization of Mamba's selective state-spaces (Gu & Dao, 2023). Its name reflects the coordinated motion of tracklets (Fig. 1), much like the synchronized movements in the samba dance. By synchronizing the hidden states across sequences, our approach is disentangled from Mamba's selective state-space models (SSMs) and can, in principle, be applied to any sequence model that includes an intermediate memory representation, such as other SSMs or recurrent neural networks.

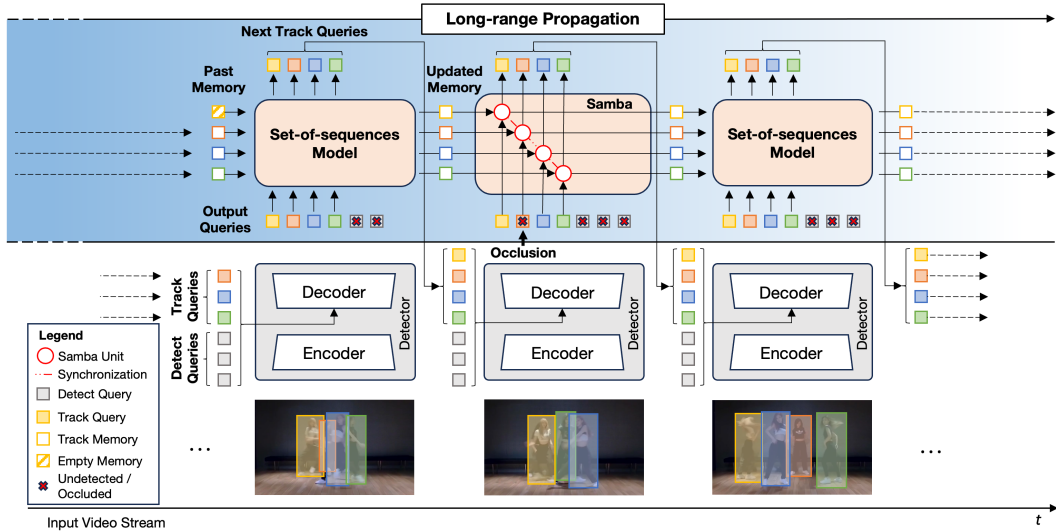


Figure 2: **Overview of SambaMOTR.** SambaMOTR combines a transformer-based object detector with a set-of-sequences Samba model. The object detector’s encoder extracts image features from each frame, which are fed into its decoder together with detect and track queries to detect newborn objects or re-detect tracked ones. The Samba set-of-sequences model is composed of multiple synchronized Samba units that simultaneously process the past memory and currently observed output queries for all tracklets to predict the next track queries and update the track memory. The hidden states of newborn objects are initialized from zero values (barred squares). In case of occlusions or uncertain detections, the corresponding query is masked (red cross) during the Samba update.

Finally, we introduce an efficient training recipe to scale SambaMOTR to longer sequences by sampling arbitrarily long sequences, computing tracking results, and applying gradients only on the last five frames. This simple strategy enables us to learn longer-range dependencies for query propagation, improving the tracking performance while maintaining the same GPU memory requirements as previous methods (Zeng et al., 2022; Gao & Wang, 2023).

We validate SambaMOTR on the challenging DanceTrack (Sun et al., 2022), SportsMOT (Cui et al., 2023), and BFT (Zheng et al., 2024) datasets. Owing to our contributions, we establish a new state of the art on all datasets. We summarize them as follows: **(a)** we introduce Samba, our novel linear-time set-of-sequences model based on synchronized SSMs; **(b)** we introduce SambaMOTR, the first tracking-by-propagation method that leverages past tracklet history in a principled manner to learn long-range dependencies, tracklets interaction, and occlusion handling; **(c)** we introduce MaskObs, a simple technique for dealing with uncertain observations in SSMs and an efficient training recipe that enables learning stronger sequence models with limited compute.

## 2 RELATED WORK

**Tracking-by-detection.** *Tracking-by-detection* is a popular paradigm in MOT, consisting of an object detection stage followed by data association to yield object trajectories throughout a video. Motion and appearance cues are typically utilized to match detections to tracklets through hand-crafted heuristics. The motion-based tracker SORT (Bewley et al., 2016) relies on Intersection over Union (IoU) to assign the tracklet locations predicted with a Kalman filter to object detections. ByteTrack (Zhang et al., 2022) introduces a two-stage matching scheme to associate low-confidence detections. OC-SORT (Cao et al., 2023) models non-linear motion by taking care of noise accumulation under occlusion. Alternatively, appearance descriptors can be used alone (Pang et al., 2021; Li et al., 2022; 2024) or in combination with motion (Wojke et al., 2017; Wang et al., 2020; Zhang et al., 2021; Aharon et al., 2022) to match detections to tracklets according to a similarity metric. Due to the disentangled nature of the two stages, tracking-by-detection methods historically leveraged state-of-the-art object detectors to top the MOT challenge (Dendorfer et al., 2021). However, by relying on hand-crafted heuristics, such methods struggle with non-linear motion and appearance patterns (Sun et al., 2022; Cui et al., 2023; Zheng et al., 2024; Li et al., 2023), and require domain-specific hyperparameters (Segu et al., 2023). Recent transformer-based methods have eased the bur-

den of heuristics. TransTrack (Sun et al., 2020) decodes track and detect queries with siamese transformer decoders and associates them with simple IoU matching. MeMOT (Cai et al., 2022) fuses a large memory bank into a tracklet descriptor with a transformer-based memory aggregator. However, it requires storing and processing with quadratic complexity the historical information from up to 27 past frames. GTR (Zhou et al., 2022) matches static trajectory queries to detections to generate tracklets, but fails to model object motion and tracklet interaction. In contrast, SambaMOTR implicitly learns motion, appearance, and tracklet interaction models by autoregressively predicting the future track queries with our set-of-sequences model Samba.

**Tracking-by-propagation.** Recent work (Meinhardt et al., 2022; Zeng et al., 2022) introduced a more flexible and end-to-end trainable *tracking-by-propagation* design that treats MOT as an autoregressive problem where object detection and query propagation are tightly coupled. Leveraging the transformer-based Deformable DETR (Zhu et al., 2020) object detector, TrackFormer (Meinhardt et al., 2022) and MOTR (Zeng et al., 2022) autoregressively propagate the detection queries through time to re-detect (track) the same object in subsequent frames. MOTRv2 (Zhang et al., 2023) leverages a pre-trained YOLOX object detector to provide anchors for Deformable DETR and boost its detection performance. However, these approaches only propagate queries across adjacent frames, failing to fully leverage the historical information. MeMOTR (Gao & Wang, 2023) first attempts to utilize the temporal information in tracking-by-propagation by aggregating long- (EMA of a tracklet’s queries through time) and short-term memory (fusion of the output detect queries across the last two observed frames) in a temporal interaction module. By collapsing the tracklet history with an EMA and by freezing the last observed state of track queries and memory under occlusions, MeMOTR cannot accurately estimate track query trajectories through occlusions. Finally, by modeling query propagation independently for each tracklet, it does not model tracklet interaction. In contrast, our proposed Samba set-of-sequences model relies on individual SSMs to independently model each tracklet as a sequence and it synchronizes the memory representations across the set of tracklets to enable tracklet interaction. Equipped with Samba, SambaMOTR autoregressively predicts future queries aware of long-range dynamics and of other tracklets’ motion and appearance.

### 3 PRELIMINARIES

Before introducing SambaMOTR (Sec. 4), we present the necessary background and notation on selective state-space models (Sec. 3.1) and tracking-by-propagation (Sec. 3.2).

#### 3.1 SELECTIVE STATE-SPACE MODELS

Inspired by classical state-space models (SSMs) (Kalman, 1960), structured SSMs (S4) (Gu et al., 2021) introduce a sequence model whose computational complexity scales linearly, rather than quadratically, with the sequence length. This makes S4 a principled and efficient alternative to transformers (Vaswani et al., 2017). By further introducing a selection mechanism - *i.e.* rendering the SSM parameters input-dependent - Mamba (Gu & Dao, 2023) can model time-variant systems, bridging the performance gap with transformers (Vaswani et al., 2017).

We here formally define selective SSMs (S6) (Gu & Dao, 2023). Let  $x(t)$  be the input signal at time  $t$ ,  $h(t)$  the hidden state, and  $y(t)$  the output signal. Given the system  $\mathbf{A}$ , control  $\mathbf{B}$ , and output  $\mathbf{C}$  matrices, we define the continuous linear time-variant SSM in Eq. (1). The discrete-time equivalent system (Eq. (2)) of the defined SSM is obtained through a discretization rule. The chosen discretization rule is typically the zero-order hold (ZOH) model:  $\bar{\mathbf{A}}(t) = \exp(\Delta(t)\mathbf{A}(t))$ ,  $\bar{\mathbf{B}}(t) = (\Delta(t)\mathbf{A}(t))^{-1}(\exp(\Delta(t)\mathbf{A}(t)) - \mathbf{I}) \cdot \Delta(t)\mathbf{B}(t)$ :

$$\begin{aligned} h'(t) &= \mathbf{A}(t)h(t) + \mathbf{B}(t)x(t) & (1) & & h_t &= \bar{\mathbf{A}}(t)h_{t-1} + \bar{\mathbf{B}}(t)x_t & (2) \\ y(t) &= \mathbf{C}(t)h(t) & & & y_t &= \mathbf{C}(t)h_t & \end{aligned}$$

$x_t$ ,  $h_t$ ,  $y_t$  are the observations sampled at time  $t$  of the input signal  $x(t)$ , hidden state  $h(t)$ , and output signal  $y(t)$ . While S4 learns a linear time-invariant (LTI) system with  $\Delta(t) = \Delta$ ,  $\mathbf{A}(t) = \mathbf{A}$ ,  $\mathbf{B}(t) = \mathbf{B}$  and  $\mathbf{C}(t) = \mathbf{C}$ , S6 introduces selectivity to learn a time-variant system by making  $\Delta(t)$ ,  $\mathbf{B}(t)$  and  $\mathbf{C}(t)$  dependent on the input  $x(t)$ , *i.e.*  $\Delta(t) = \tau_\Delta(\Delta + s_\Delta(x(t)))$ ,  $\mathbf{B}(t) = s_B(x(t))$ ,  $\mathbf{C}(t) = s_C(x(t))$ , where  $\tau_\Delta = \text{softplus}$ , and  $s_\Delta$ ,  $s_B$ ,  $s_C$  are learnable linear mappings.

In this paper, we propose to treat tracking-by-propagation as a sequence modeling problem. Given the discrete sequence of historical track queries for a certain tracklet, our query propagation mod-

216 ule Samba (Sec. 4.2) leverages SSMs to account for the historical tracklet information in a prin-  
217 ciple manner. By recursively compressing all tracklet history into a long-term memory, Samba’s  
218 complexity scales linearly with the number of frames, enabling efficient training on long sequences  
219 while processing indefinitely long tracklets at inference time.

### 221 3.2 TRACKING-BY-PROPAGATION

222 Tracking-by-propagation methods alternate between a detection stage and a propagation stage, rely-  
223 ing on a DETR-like (Carion et al., 2020) transformer object detector and a query propagation mod-  
224 ule. At a time step  $t$ , the backbone and transformer encoder extract image features for a frame  $I_t$ .  
225 The *detection stage* involves feeding both a fixed-length set of learnable detect queries  $Q_t^{det}$  to the  
226 transformer decoder to detect newborn objects and a variable-length set of propagated track queries  
227  $Q_t^{tck}$  to re-detect tracked ones. At time  $t = 0$ , the set of track queries is empty, *i.e.*  $Q_0^{tck} = E_0^{tck} = \emptyset$ .  
228 Detect and track queries  $[Q_t^{det}, Q_t^{tck}]$  interact in the decoder with image features to generate the cor-  
229 responding output embeddings  $[E_t^{det}, E_t^{tck}]$  and bounding box predictions  $[D_t^{det}, D_t^{tck}]$ . We denote  
230 the set of embeddings corresponding to newborn objects as  $\hat{E}_t^{det}$ , and  $\hat{E}_t^{tck} = [\hat{E}_t^{det}, E_t^{tck}]$  as the  
231 set of embeddings corresponding to the tracklets  $\mathcal{S}_t$  active at time  $t$ . During the *propagation stage*, a  
232 query propagation module  $\Theta(\cdot)$  typically takes as input the set of embeddings  $\hat{E}_t^{tck}$  and outputs re-  
233 fined tracked queries  $Q_{t+1}^{tck} = \Theta(\hat{E}_t^{tck})$  to re-detect the corresponding objects in the next frame.

234 Although prior work failed to properly model long-range history and tracklet interactions (Zeng  
235 et al., 2022; Gao & Wang, 2023; Meinhardt et al., 2022), and given that multiple objects often move  
236 synchronously (Fig. 1), we argue that the future state of objects in a scene can be better predicted by  
237 (i) considering both their historical positions and appearances, and (ii) estimating their interactions.  
238 In this work, we cast query propagation as a set-of-sequences modeling problem. Given a set of  
239 multiple tracklets, we encode the history of each tracklet in a memory representation using a state-  
240 space model and propose memory synchronization to account for their joint dynamics.

## 242 4 METHOD

243 In this section, we introduce SambaMOTR, an end-to-end multiple object tracker that combines  
244 transformer-based object detection with our set-of-sequences model Samba to jointly model the  
245 long-range history of each tracklet and the interaction across tracklets to propagate queries. First, in  
246 Sec. 3.2 we provide background on the tracking-by-propagation framework and motivate the need  
247 for better modeling of both temporal information and tracklets interaction. Then, we describe the  
248 SambaMOTR architecture (Sec. 4.1) and introduce Samba (Sec. 4.2), our novel set-of-sequences  
249 model based on synchronized state spaces that jointly models the temporal dynamics of a set of se-  
250 quences and their interdependencies. Finally, in Sec. 4.3 we describe SambaMOTR’s query propa-  
251 gation strategy based on Samba, our effective technique MaskObs to deal with occlusions in SSMs, a  
252 recipe to learn long-range sequence models with limited compute, and our simple inference pipeline.

### 254 4.1 ARCHITECTURE

255 Similar to other tracking-by-propagation methods (Meinhardt et al., 2022; Zeng et al., 2022; Gao &  
256 Wang, 2023), the proposed SambaMOTR architecture (Fig. 2) is composed of a DETR-like (Carion  
257 et al., 2020) object detector and a query propagation module. As object detector, we use Deformable-  
258 DETR (Zhu et al., 2020) with a ResNet-50 (He et al., 2016) backbone followed by a transformer  
259 encoder to extract image features and a transformer decoder to detect bounding boxes from a set of  
260 detect and track queries. As query propagation module, we use our set-of-sequences model Samba.  
261 Each sequence is processed by a Samba unit synchronized with all others. A Samba unit consists of  
262 two Samba blocks (Sec. 4.2) interleaved with LayerNorm (Ba et al., 2016) and a residual connection.

### 263 4.2 SAMBA: SYNCHRONIZED STATE-SPACE MODELS FOR SET-OF-SEQUENCES MODELING

264 Set-of-sequences modeling involves simultaneously modeling a set of temporal sequences and the  
265 interdependencies among them. In MOT, set-of-sequences models can capture long-range temporal  
266 relationships within each tracklet as well as complex interactions across tracklets. To this end, we  
267 introduce Samba, a linear-time set-of-sequences model based on the synchronization of multiple  
268 state-space models. In this paper, we leverage Samba as a set-of-queries propagation network to  
269 jointly model multiple tracklets and their interactions in a tracking-by-propagation framework.

270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323

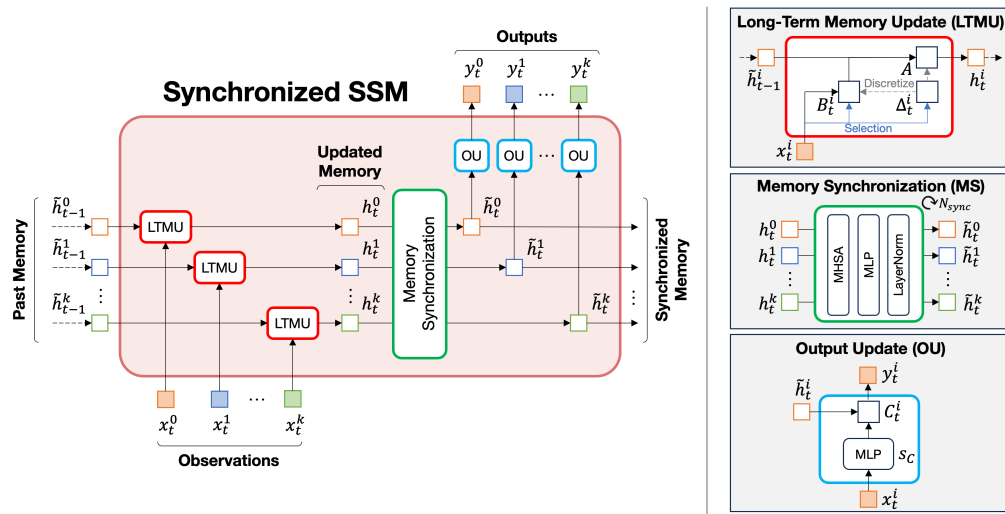


Figure 3: **Synchronized State-Space Models.** We illustrate a set of  $k$  synchronized SSMs. A Long-Term Memory Update block updates each hidden state  $\tilde{h}_{t-1}^i$  based on the current observation  $x_t^i$ , resulting in the updated memory  $h_t^i$ . The Memory Synchronization block then derives the synchronized hidden state  $\tilde{h}_t^i$ , which is fed into the Output Update module to predict the output  $y_t^i$ .

**Synchronized Selective State-Space Models.** Let  $x_t^i$  be the discrete observation at time  $t$  of the  $i$ -th input sequence from a set of sequences  $\mathcal{S}$ . We choose selective SSMs (Gu & Dao, 2023) to model each sequence through a hidden state  $h_t^i$  (Eq. (3a)) encoding long-term memory, but our approach applies to any other SSM. Given the memory  $\tilde{h}_{t-1}^i$ , we define a long-term memory update (LTMU) function that updates  $\tilde{h}_{t-1}^i$  based on the current observation  $x_t^i$ , resulting in the updated memory  $h_t^i$ . We propose a memory synchronization (MS) function  $\Gamma_{i \in \mathcal{S}}(\cdot)$  that produces a set of synchronized hidden states  $\tilde{h}_t^i \forall i \in \mathcal{S}$ , modeling interactions across the set of sequences  $\mathcal{S}$  (Eq. (3b)). Finally, we derive the output  $y_t^i$  for each sequence through the output update (OU) function (Eq. (3c)).

$$\text{LTMU: } h_t^i = \bar{\mathbf{A}}^i(t)h_{t-1}^i + \bar{\mathbf{B}}^i(t)x_t^i \quad (3a)$$

$$\text{MS: } \tilde{h}_t^i = \Gamma_{i \in \mathcal{S}}(h_t^i) \quad (3b)$$

$$\text{OU: } y_t^i = \mathbf{C}(t)\tilde{h}_t^i \quad (3c)$$

An ideal memory synchronization function should be flexible regarding the number of its inputs (hidden states) and equivariant to their order. Thus, we propose to define the memory synchronization function  $\Gamma(\cdot) = [FFN(MHSA(\cdot))]_{\times N_{sync}}$  as a set of  $N_{sync}$  stacked blocks with multi-head self-attention (MHSA) (Vaswani et al., 2017) followed by a feed-forward network (FFN). A schematic illustration of the proposed synchronized state-space model layer is in Fig. 3.

**Set-of-sequences Model.** Refer to App. B.1 for a detailed description of how our synchronized SSM is used in the Samba units that model each sequence in the set-of-sequences Samba model.

#### 4.3 SAMBAMOTR: END-TO-END TRACKING-BY-PROPAGATION WITH SAMBA

**Query Propagation with Samba.** As described in Sec. 3.2, the query propagation module  $\Theta(\cdot)$  takes as input the decoder output embeddings  $\hat{E}_t^{tck}$  and outputs refined track queries  $Q_{t+1}^{tck} = \Theta(\hat{E}_t^{tck})$ . SambaMOTR extends this paradigm by accounting for the temporal information and tracklets interaction. In particular, we use our Samba module  $\Phi(\cdot)$  to compress the history of each tracklet into a hidden state  $h_t^i$  and synchronize it across tracklets to derive the synchronized memory  $\tilde{h}_t^i$ . Notice that  $\tilde{h}_t^i = \mathbf{0}$  for a newborn object  $i$ . At time  $t$ , we first enrich the detector output embeddings  $\hat{E}_t^{tck}$  with position information by summing to them sine-cosine positional encodings  $PE(\cdot)$  of the corresponding bounding boxes coordinates  $\hat{D}_t^{tck}$  to implicitly model object motion and appearance, obtaining the set of input observations  $X_t^{tck} = \hat{E}_t^{tck} + PE(\hat{D}_t^{tck})$ . Given the set of input observations  $X_t^{tck}$  and past synchronized hidden states  $\tilde{H}_{t-1}$  for all tracklets in the set

$S_t$  of tracklets at time  $t$ , we feed them into Samba  $(Y_t, \tilde{H}_t) = \Phi(X_t^{tck}, \tilde{H}_{t-1})$  to obtain the output embeddings  $Y_t$  and updated synchronized hidden states  $\tilde{H}_t$ . Finally, we use the output embeddings  $Y_t$  and a learnable mapping  $s_y$  to predict a residual  $\Delta Q_t^{tck} = s_y(Y_t)$  to the past track queries  $Q_t^{tck}$  and generate the new ones, *i.e.*  $Q_{t+1}^{tck} = Q_t^{tck} + \Delta Q_t^{tck}$ . By recursively unfolding this process over time, SambaMOTR can track multiple objects while compressing indefinitely long tracklet histories into their long-term memory representations, effectively modeling object motion and appearance changes, and tracklets interactions.

**MaskObs: Dealing with Uncertain Observations.** Tracking-by-propagation may occasionally deal with occluded objects or uncertain detections. Given a function  $conf(\cdot)$  to estimate the predictive confidence of an input observation  $x_t^i$ , we propose MaskObs, a strategy to handle uncertain observations. MaskObs masks uncertain observations from the state update (Eq. (4)), thus defining the system dynamics solely based on its history and the interdependencies with other sequences:

$$h_t^i = \bar{\mathbf{A}}^i(t)h_{t-1}^i + \bar{\mathbf{B}}^i(t)x_t^i \cdot \mathbb{1}[conf(x_t^i) > \tau_{mask}] \quad (4)$$

$\mathbb{1}[\cdot]$  is the indicator function, and  $\tau_{mask}$  is the confidence threshold, *e.g.*  $\tau_{mask} = 0.5$ . We implement  $conf(x_t^i)$  as the predictive confidence  $conf(d_t^i)$  of the corresponding bounding box  $d_t^i$ . In our work, this design choice allows us to better model query propagation through occlusions (Tab. 4, line b).

**Efficiently Learning Long-range Sequence models.** Previous MOTR-like approaches are trained end-to-end on a sequence of 5 consecutive frames sampled at random intervals. While SambaMOTR’s set-of-sequences model Samba already shows impressive generalization performance to long sequences at inference time (Tab. 4, line c), we propose to train on longer sequences (*i.e.* 10 frames) and only apply gradients to the last 5 frames (Tab. 4, line d). We hypothesize that this strategy allows us to learn better history compression for late observations in a sequence, resulting in even better tracking performance while being trained with similar GPU memory requirements. A schematic illustration of our training scheme proposal is in Fig. B.

**Inference Pipeline.** At a given time step  $t$ , we jointly input the learnable detect queries  $Q_{det}$  and track queries  $Q_t^{tck}$  ( $Q_0^{tck} = \emptyset$ ) into the transformer decoder to produce detection embeddings  $E_t^{det}$  and tracking embeddings  $E_t^{tck}$  and the corresponding bounding boxes. Each detection bounding box with a confidence score higher than a threshold  $\tau_{det}$  will initialize a newborn track  $\hat{E}_t^{det}$ . We then propagate the embeddings of newborn  $\hat{E}_t^{det}$  and tracked  $E_t^{tck}$  objects together with the track memory  $\tilde{H}_{t-1}$  to generate the updated track queries  $Q_{track}^{t+1}$  and synchronized memory  $\tilde{H}_{t+1}$ . To deal with occlusions and lost objects, we consider an individual track query  $q_t^{i,track}$  inactive if its corresponding bounding box confidence  $conf(d_t^i)$  at time  $t$  is lower than  $\tau_{track}$ . If a track query is inactive for more than  $N_{miss}$  frames, it is deemed lost and dropped.

Unlike MeMOTR (Gao & Wang, 2023), which does not update the track embedding and long-term memory for an object with low detection confidence at a time step  $t$ , our approach employs a principled query propagation scheme that can hallucinate likely track query trajectories under occlusions by relying on its past history or attending to other trajectories. Thus, we always update the memory and track query for any tracklet - even when occluded - as long as it is not deemed lost.

## 5 EXPERIMENTS

In this section, we present experimental results to validate SambaMOTR. We describe our evaluation protocol (Sec. 5.1) and report implementation details (Sec. 5.2). We then compare SambaMOTR to the previous state-of-the-art methods (Sec. 5.3) and conduct an ablation study (Sec. 5.4) on the method components. We provide more ablations in the appendix. Qualitative results can be found in Fig. 1 and at the anonymous project page <https://anonymous-samba.github.io/>.

### 5.1 EVALUATION PROTOCOL

**Datasets.** To evaluate SambaMOTR, we select a variety of challenging datasets exhibiting highly non-linear motion in crowded scenarios, with frequent occlusions and uniform appearances. All datasets present scenes with objects moving synchronously. Thus, they represent a suitable benchmark for assessing the importance of modeling tracklet interaction. DanceTrack (Sun et al., 2022) is a multi-human tracking dataset composed of 100 group dancing videos. The Bird Flock Tracking (BFT) (Zheng et al., 2024) dataset includes 106 clips from the BBC documentary series Earthflight (Downer & Tennant, 2011). SportsMOT (Cui et al., 2023) consists of 240 video sequences

378 from basketball, volleyball, and soccer scenes. Due to the highly linear motion in MOT17 (Milan  
379 et al., 2016), its small size (only 7 videos), and the subsequent need for training on additional de-  
380 tection datasets, end-to-end tracking methods do not provide additional advantages over more naive  
381 Kalman-filter-based methods. We report its results in the Appendix.

382 **Metrics.** Following prior work, we measure the overall tracking performance with the  
383 HOTA (Luiten et al., 2021) metric and disentangle detection accuracy (DetA) and association accu-  
384 racy (AssA). We report the MOTA (Bernardin & Stiefelhagen, 2008) and IDF1 (Ristani et al., 2016)  
385 metrics for completeness. Since our objective is improving association performance and the overall  
386 tracking quality, HOTA and AssA are the most representative metrics.

## 388 5.2 IMPLEMENTATION DETAILS

389  
390 Following prior works (Gao & Wang, 2023; Zhang et al., 2023), we apply random resize, random  
391 crop, and photometric augmentations as data augmentation. The shorter side of the input image is  
392 resized to 800 preserving the aspect ratio, and the maximum size is restricted to 1536. For a fair  
393 comparison with prior work (Sun et al., 2020; Zeng et al., 2022; Gao & Wang, 2023), we use the  
394 Deformable-DETR (Zhu et al., 2020) object detector with ResNet-50 (He et al., 2016) and initialize  
395 it from COCO (Lin et al., 2014) pre-trained weights. Similar to MeMOTR (Gao & Wang, 2023),  
396 we inject track queries after one decoder layer. We run our experiments on 8 NVIDIA RTX 4090  
397 GPUs, with batch size 1 per GPU. Each batch element contains a video clip with 10 frames, and we  
398 compute and backpropagate the gradients only over the last 5. We sample uniformly spaced frames  
399 at random intervals from 1 to 10 within each clip. We utilize the AdamW optimizer (Loshchilov &  
400 Hutter, 2017) with initial learning rate of  $2.0 \times 10^{-4}$ . For simplicity,  $\tau_{det} = \tau_{track} = \tau_{mask} = 0.5$ .  
401  $N_{miss}$  is 35, 20, and 50 on DanceTrack, BFT, and SportsMOT, respectively, due to different dataset  
402 dynamics. On DanceTrack (Sun et al., 2022), we train SambaMOTR for 15 epochs on the training  
403 set and drop the learning rate by a factor of 10 at the 10<sup>th</sup> epoch. On BFT (Sun et al., 2022), we  
404 train for 20 epochs and drop the learning rate after 10 epochs. On SportsMOT (Cui et al., 2023), we  
405 train for 18 epochs and drop the learning rate after 8 and 12 epochs. SambaMOTR’s inference runs  
406 at 16 FPS on a single NVIDIA RTX 4090 GPUs.

## 407 5.3 COMPARISON WITH THE STATE OF THE ART

408 We compare SambaMOTR with multiple tracking-by-detection and tracking-by-propagation ap-  
409 proaches on the DanceTrack (Tab. 1), BFT (Tab. 2) and SportsMOT (Tab. 3) datasets. All meth-  
410 ods are trained without using additional datasets. Since trackers use various object detectors with  
411 different baseline performance, we report the detector used for each method. For fair comparison,  
412 we report the performance of tracking-by-propagation methods with Deformable DETR (Zhu et al.,  
413 2020), marking the best in **bold**. We underle the overall best result. Tracking-by-detection methods  
414 often use the stronger YOLOX-X (Ge et al., 2021), but tracking-by-propagation consistently outper-  
415 forms them, with SambaMOTR achieving the highest HOTA and AssA across all datasets.

416 **DanceTrack.** The combination of highly irregular motion and crowded scenes with frequent occlu-  
417 sions and uniform appearance historically made DanceTrack challenging for tracking-by-detection  
418 methods. Despite their higher DetA when using the strong object detector YOLOX-X (Ge et al.,  
419 2021), tracking-by-propagation significantly outperforms them (see MeMOTR (Gao & Wang, 2023)  
420 and SambaMOTR). SambaMOTR sets a new state of the art, with +3.8 HOTA and +5.2 AssA  
421 on the strongest competitor MeMOTR. Our method owes this performance improvement to its bet-  
422 ter modeling of the historical information, our effective strategy to learn accurate sequence models  
423 through occlusions, and our modeling of tracklets interaction (group dancers move synchronously).

424 **BFT.** Bird flocks present similar appearance and non-linear motion. For this reason, OC-SORT  
425 works best among tracking-by-detection methods. Nevertheless, bird flocks move synchronously,  
426 and interaction among tracklets is an essential cue for modeling joint object motion. Thanks to our  
427 proposed sequence models synchronization, SambaMOTR achieves +2.8 HOTA and +4.9 AssA  
428 over the best competitor overall (OC-SORT), and an impressive +6.3 HOTA and +12.5 improve-  
429 ment over the previous best tracking-by-propagation method TrackFormer (Meinhardt et al., 2022).

430 **SportsMOT.** Sports scenes typically present non-linear motion patterns that the Kalman filter strug-  
431 gles to model, hence the underwhelming performance of ByteTrack (Zhang et al., 2022). For this  
reason, trackers that model non-linear motion either explicitly (OC-SORT (Cao et al., 2023)) or im-



Table 1: **State-of-the-art comparison on DanceTrack** (Sun et al., 2022) without additional training data. Best tracking-by-propagation method in **bold**; best overall underlined.

Methods	Detector	HOTA	AssA	DetA	IDF1	MOTA
<i>Tracking-by-detection:</i>						
FairMOT (Zhang et al., 2021)	CenterNet (Duan et al., 2019)	39.7	23.8	66.7	40.8	82.2
CenterTrack (Zhou et al., 2020)		41.8	22.6	78.1	35.7	86.8
TraDeS (Wu et al., 2021)		43.3	25.4	74.5	41.2	86.2
TransTrack (Sun et al., 2020)	Deformable DETR (Zhu et al., 2020)	45.5	27.5	75.9	45.2	88.4
GTR (Zhou et al., 2022)	CenterNet2 (Zhou et al., 2021)	48.0	31.9	72.5	50.3	84.7
ByteTrack (Zhang et al., 2022)	YOLOX-X (Ge et al., 2021)	47.7	32.1	71.0	53.9	89.6
QDTrack (Pang et al., 2021)		54.2	36.8	80.1	50.4	87.7
OC-SORT (Cao et al., 2023)		55.1	38.3	80.3	54.6	<u>92.0</u>
C-BIoU (Yang et al., 2023)		60.6	45.4	<u>81.3</u>	61.6	91.6
<i>Tracking-by-propagation:</i>						
MOTR (Zeng et al., 2022)	Deformable DETR (Zhu et al., 2020)	54.2	40.2	73.5	51.5	79.7
MeMOTR (Gao & Wang, 2023)		63.4	52.3	77.0	65.5	85.4
SambaMOTR (ours)		<b>67.2</b>	<b>57.5</b>	<b>78.8</b>	<b>70.5</b>	<b>88.1</b>

Table 2: **State-of-the-art comparison on BFT** (Zheng et al., 2024) without additional training data. Best tracking-by-propagation method in **bold**; best overall underlined.

Method	Detector	HOTA	AssA	DetA	IDF1	MOTA
<i>Tracking-by-detection:</i>						
FairMOT (Zhang et al., 2021)	CenterNet (Duan et al., 2019)	40.2	28.2	53.3	41.8	56.0
CenterTrack (Zhou et al., 2020)		65.0	54.0	58.5	61.0	60.2
SORT (Wojke et al., 2017)	YOLOX-X (Ge et al., 2021)	61.2	62.3	60.6	77.2	75.5
ByteTrack (Zhang et al., 2022)		62.5	64.1	61.2	<u>82.3</u>	<u>77.2</u>
OC-SORT (Cao et al., 2023)		66.8	68.7	65.4	79.3	77.1
TransCenter (Xu et al., 2022)	Deformable DETR (Zhu et al., 2020)	60.0	61.1	<u>66.0</u>	72.4	74.1
TransTrack (Sun et al., 2020)		62.1	60.3	64.2	71.4	71.4
<i>Tracking-by-propagation:</i>						
TrackFormer (Meinhardt et al., 2022)	Deformable DETR (Zhu et al., 2020)	63.3	61.1	<u>66.0</u>	72.4	<b>74.1</b>
SambaMOTR (ours)		<b>69.6</b>	<b>73.6</b>	<b>66.0</b>	<b>81.9</b>	72.0

licity (TransTrack (Sun et al., 2020)) perform well. Notably, our tracking-by-propagation SambaMOTR enables implicit joint modeling of motion, appearance, and tracklet interaction, obtaining the best HOTA overall (69.8) despite the lower DetA of our Deformable-DETR detector compared to OC-SORT’s YOLOX-X. Moreover, SambaMOTR exhibits a significant +1.6 AssA over the best tracking-by-propagation method and an impressive +4.6 AssA over OC-SORT.

#### 5.4 ABLATION STUDIES

We ablate the effect of each component of our method in Tab. 4, as detailed in Sec. 4 and illustrated in Fig. B. Additional ablation studies are presented in App. C.2.

**SSM.** Line (a) shows the benefits of a sequential representation for tracking. We use a vanilla sequence model, such as Mamba, as the baseline for query propagation, establishing a robust foundation that outperforms MeMOTR’s EMA-based history and temporal attention module.

**MaskObs.** Handling track queries during occlusions (line b) with MaskObs - which masks uncertain observations from the state update and relies on long-term memory and interactions with visible tracklets - leads to significant overall improvements (+1.3 HOTA), highlighting the effectiveness of managing occluded objects.

**Sync.** Making tracklets aware of each other through our synchronization mechanism (line c) results in over 1% improvement across all metrics, demonstrating how modeling interactions between tracklets enhances tracking accuracy by capturing joint dynamics and coordinated movements.

Table 3: **State-of-the-art comparison on SportsMOT** (Cui et al., 2023) without additional training data. Best tracking-by-propagation method in **bold**; best overall underlined.

Methods	Detector	HOTA	AssA	DetA	IDF1	MOTA
<i>Tracking-by-detection:</i>						
FairMOT (Zhang et al., 2021)	CenterNet (Duan et al., 2019)	49.3	34.7	70.2	53.5	86.4
QDTrack (Pang et al., 2021)		60.4	47.2	77.5	62.3	90.1
ByteTrack (Zhang et al., 2022)	YOLOX-X (Ge et al., 2021)	62.1	50.5	76.5	69.1	93.4
OC-SORT (Cao et al., 2023)		68.1	54.8	<u>84.8</u>	68.0	<u>93.4</u>
TransTrack (Sun et al., 2020)	Deformable DETR (Zhu et al., 2020)	68.9	57.5	82.7	71.5	92.6
<i>Tracking-by-propagation:</i>						
MeMOTR (Gao & Wang, 2023)	Deformable DETR (Zhu et al., 2020)	68.8	57.8	82.0	69.9	90.2
SambaMOTR (ours)		<b>69.8</b>	<b>59.4</b>	<b>82.2</b>	<u>71.9</u>	<b>90.3</b>

Table 4: **Ablation on method components** on the DanceTrack test set. Compared to prior work (in gray), we introduce a long-range query propagation module based on state-space models (SSM), we mask uncertain queries during the state update (MaskObs), we synchronize memory representations across tracklets (Sync), and we learn from longer sequences (Longer).

Method		SSM	MaskObs	Sync	Longer	HOTA	AssA	DetA	IDF1	MOTA
SambaMOTR (Ours)	(a)	✓	-	-	-	63.5	53.8	75.1	67.0	81.7
	(b)	✓	✓	-	-	64.8	54.3	77.7	68.1	85.7
	(c)	✓	✓	✓	-	65.9	55.6	78.4	68.7	87.4
	(d)	✓	✓	✓	✓	<b>67.2</b>	<b>57.5</b>	<b>78.8</b>	<b>70.5</b>	<b>88.1</b>
MOTR (Zeng et al., 2022)	(e)	-	-	-	-	54.2	40.2	73.5	51.5	79.7
MeMOTR (Gao & Wang, 2023)	(f)	-	-	-	-	63.4	52.3	77.0	65.5	85.4

**Long-sequence training.** Efficiently incorporating longer sequences during training (line d) helps the model to properly utilize its long-term memory, enabling generalization to indefinitely long sequences and leading to a notable +1.9 improvement in AssA.

Our final query propagation method (line d) improves MeMOTR’s association accuracy by +5.2 (line f), and MOTR’s by an impressive +17.3 (line e).

## 6 LIMITATIONS

Following the tracking-by-propagation paradigm, our model drops tracklets that are inactive for more than  $N_{miss}$  frames to decrease the risk of ID switches. However, in some datasets like SportsMOT (Cui et al., 2023) football players may disappear from the camera view for multiple seconds, outliving the  $N_{miss}$  threshold. We argue that future work should complement tracking-by-propagation with long-term re-identification to tackle this issue. Furthermore, in this paper, we introduced Samba, a set-of-sequences model. Our ablation study (Tab. 4) shows that Samba significantly outperforms the already strong SSM baseline. However, this comes with the trade-off of increased computational complexity. In particular, SSMs have linear complexity in time and linear complexity in the number of sequences (tracklets) independently modeled. Samba retains linear-time complexity, which enables it to track for indefinitely long-time horizons, but quadratic complexity in the number of sequences due to the use of self-attention in memory synchronization. Our ablations show that this trade-off is worth the performance improvement.

## 7 CONCLUSION

The proposed SambaMOTR fully leverages the sequential nature of the tracking task by using our set-of-sequences model, Samba, as a query propagation module to jointly model the temporal history of each tracklet and their interactions. The resulting tracker runs with linear-time complexity and can track objects across indefinitely long sequences. SambaMOTR surpasses the state-of-the-art on all benchmarks, reporting significant improvements in association accuracy compared to prior work.

540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593

---

## REFERENCES

- Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.
- Magda Amiridi, Gregory Darnell, and Sean Jewell. Latent temporal flows for multivariate analysis of wearables data. *arXiv preprint arXiv:2210.07475*, 2022.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- 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 Uprocft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468. IEEE, 2016.
- 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 (CVPR)*, pp. 8090–8100, 2022.
- Jinkun Cao, Jiangmiao Pang, Xinchuo 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 (CVPR)*, 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213–229. Springer, 2020.
- Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. Sportsmot: A large multi-object tracking dataset in multiple sports scenes. *arXiv preprint arXiv:2304.05170*, 2023.
- Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision (IJCV)*, 129:845–881, 2021.
- John Downer and David Tennant. Earthflight, 2011. URL <https://www.bbc.co.uk/programmes/b018xscl>. BBC.
- Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6569–6578, 2019.
- 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 (ICCV)*, pp. 9901–9910, 2023.
- Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.
- Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

- 
- 594 Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):  
595 1735–1780, 1997.
- 596
- 597 Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic*  
598 *Engineering*, 1960.
- 599
- 600 Siyuan Li, Martin Danelljan, Henghui Ding, Thomas E Huang, and Fisher Yu. Tracking every thing  
601 in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 498–  
602 515. Springer, 2022.
- 603
- 604 Siyuan Li, Tobias Fischer, Lei Ke, Henghui Ding, Martin Danelljan, and Fisher Yu. Ovtrack: Open-  
605 vocabulary multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer*  
606 *Vision and Pattern Recognition (CVPR)*, pp. 5567–5577, 2023.
- 607
- 608 Siyuan Li, Lei Ke, Martin Danelljan, Luigi Piccinelli, Mattia Segu, Luc Van Gool, and Fisher Yu.  
609 Matching anything by segmenting anything. In *Proceedings of the IEEE/CVF Conference on*  
610 *Computer Vision and Pattern Recognition (CVPR)*, 2024.
- 611
- 612 Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr  
613 Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer*  
614 *Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014,*  
615 *Proceedings, Part V 13*, pp. 740–755. Springer, 2014.
- 616
- 617 Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-  
618 detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022.
- 619
- 620 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*  
621 *arXiv:1711.05101*, 2017.
- 622
- 623 Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé,  
624 and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International*  
625 *Journal of Computer Vision (IJCV)*, 129:548–578, 2021.
- 626
- 627 Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple  
628 object tracking: A literature review. *Artificial intelligence*, 293:103448, 2021.
- 629
- 630 Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer:  
631 Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF Conference on Com-*  
632 *puter Vision and Pattern Recognition (CVPR)*, pp. 8844–8854, 2022.
- 633
- 634 Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark  
635 for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.
- 636
- 637 Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense  
638 similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on*  
639 *Computer Vision and Pattern Recognition (CVPR)*, pp. 164–173, 2021.
- 640
- 641 Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural  
642 networks. In *International conference on machine learning*, pp. 1310–1318. Pmlr, 2013.
- 643
- 644 Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. Motiontrack: Learn-  
645 ing robust short-term and long-term motions for multi-object tracking. In *Proceedings of the*  
646 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17939–17948,  
647 2023.
- 648
- 649 Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance mea-  
650 sures and a data set for multi-target, multi-camera tracking. In *Proceedings of the European Con-*  
651 *ference on Computer Vision (ECCV)*, pp. 17–35. Springer, 2016.
- 652
- 653 Mattia Segu, Bernt Schiele, and Fisher Yu. Darth: Holistic test-time adaptation for multiple object  
654 tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*,  
655 pp. 9717–9727, 2023.

- 
- 648 Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping  
649 Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*,  
650 2020.
- 651 Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack:  
652 Multi-object tracking in uniform appearance and diverse motion. In *Proceedings of the IEEE/CVF*  
653 *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20993–21002, 2022.
- 654 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,  
655 Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information*  
656 *Processing Systems (NeurIPS)*, 30, 2017.
- 657 Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell,  
658 and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE/CVF Interna-*  
659 *tional Conference on Computer Vision (ICCV)*, pp. 4534–4542, 2015.
- 660 Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-  
661 object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp.  
662 107–122. Springer, 2020.
- 663 Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun.  
664 Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.
- 665 Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep  
666 association metric. In *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–  
667 3649. IEEE, 2017.
- 668 Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect  
669 and segment: An online multi-object tracker. In *Proceedings of the IEEE/CVF Conference on*  
670 *Computer Vision and Pattern Recognition (CVPR)*, pp. 12352–12361, 2021.
- 671 Yuxia Wu, Yuan Fang, and Lizi Liao. On the feasibility of simple transformer for dynamic graph  
672 modeling. *arXiv preprint arXiv:2401.14009*, 2024.
- 673 Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-  
674 Pineda. Transcenter: Transformers with dense representations for multiple-object tracking. *IEEE*  
675 *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 45(6):7820–7835, 2022.
- 676 Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular  
677 motions and similar appearances? make it easier by buffering the matching space. In *Proceedings*  
678 *of the IEEE/CVF Winter conference on Applications of Computer Vision (WACV)*, pp. 4799–4808,  
679 2023.
- 680 Xitong Yang, Palghat Ramesh, Radha Chitta, Sriganesh Madhvanath, Edgar A Bernal, and Jiebo  
681 Luo. Deep multimodal representation learning from temporal data. In *Proceedings of the*  
682 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5447–5455,  
683 2017.
- 684 Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-  
685 to-end multiple-object tracking with transformer. In *Proceedings of the European Conference on*  
686 *Computer Vision (ECCV)*, pp. 659–675. Springer, 2022.
- 687 Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Infor-*  
688 *mation Processing Systems*, 32, 2019.
- 689 Yifu Zhang, Chunyu Wang, Xinggong Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the  
690 fairness of detection and re-identification in multiple object tracking. *International Journal of*  
691 *Computer Vision (IJCV)*, 129:3069–3087, 2021.
- 692 Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu  
693 Liu, and Xinggong Wang. Bytetrack: Multi-object tracking by associating every detection box. In  
694 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1–21. Springer, 2022.

---

702 Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object  
703 tracking by pretrained object detectors. In *Proceedings of the IEEE/CVF Conference on Computer  
704 Vision and Pattern Recognition (CVPR)*, pp. 22056–22065, 2023.

705  
706 Guangze Zheng, Shijie Lin, Haobo Zuo, Changhong Fu, and Jia Pan. Nettrack: Tracking highly  
707 dynamic objects with a net. *arXiv preprint arXiv:2403.11186*, 2024.

708  
709 Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *Proceedings  
710 of the European Conference on Computer Vision (ECCV)*, pp. 474–490. Springer, 2020.

711  
712 Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv  
713 preprint arXiv:2103.07461*, 2021.

714  
715 Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transform-  
716 ers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition  
717 (CVPR)*, pp. 8771–8780, 2022.

718  
719 Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr:  
720 Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755

---

## APPENDIX

In this appendix, we report additional discussions and experiments. First, we provide background on sequence models in App. A. Then, we report additional implementation details for SambaMOTR in App. B. We show a schematic illustration of the Samba block in Fig. A and our method components in Fig. B. Finally, we provide additional results App. C, conducting several ablation studies on specific design choices that contributed to SambaMOTR’s performance.

### A BACKGROUND ON SEQUENCE MODELS.

**Sequence Models.** Sequence models are a class of machine learning models dealing with sequential data, *i.e.* where the order of elements is important. Applications of sequence models are widespread across different fields, such as natural language processing (Vaswani et al., 2017; Gu & Dao, 2023), time series forecasting (Wen et al., 2022) and video analysis (Venugopalan et al., 2015). Several architectures have been proposed to process sequences, each with its own strengths and limitations. *Recurrent neural networks (RNNs)* handles sequential data by maintaining a hidden state that updates as the network processes each element in a sequence. However, RNNs often struggle with long sequences due to issues like vanishing or exploding gradients (Pascanu et al., 2013). *Long short-term memory (LSTM)* networks (Hochreiter & Schmidhuber, 1997) introduce gating units to mitigate RNN’s vanishing gradient problem. *Transformers* (Vaswani et al., 2017) rely on self-attention mechanisms to weigh the importance of different parts of the input data. Unlike RNNs and LSTMs, transformers process entire sequences simultaneously, making them efficient at modeling long-range dependencies at the cost of quadratic computational complexity wrt. sequence length. Building on the idea of modeling temporal dynamics like RNNs and LSTMs, structured state-space models (Gu et al., 2021) introduce a principled approach to state management inspired by classical SSMs (Kalman, 1960). Despite excelling at modeling long-range dependencies in continuous signals, structured SSMs lag behind transformers on discrete modalities such as text. Recently, selective state-space models (Mamba) (Gu & Dao, 2023) improved over prior work by making the SSM parameters input-dependent, achieving the modeling power of Transformers while scaling linearly with sequence length.

**Set-of-sequences Models.** Only few approaches (Yang et al., 2017; Amiridi et al., 2022; Wu et al., 2024) explore the task of set-of-sequences modeling, which we define as the task of simultaneously modeling multiple temporal sequences and their interdependencies to capture complex relationships and interactions across different data streams. Set-of-sequences modeling has applications in multivariate time series analysis (Amiridi et al., 2022), dynamic graph modeling (Wu et al., 2024), and sensor data fusion (Yang et al., 2017). However, existing techniques involve complex and expensive designs. We here introduce Samba, a linear-time set-of-sequences model based on the synchronization of multiple selective state-space models to account for the interaction across sequences.

### B SAMBAMOTR - ADDITIONAL DETAILS

SambaMOTR builds on Samba to introduce linear-time sequence modeling in tracking-by-propagation, treating each tracklet as a sequence of queries and autoregressively predicting the future track query. By inducing synchronization on the SSMs’ memories across an arbitrary number of sequences, Samba elegantly models tracklet interaction and query propagation under occlusions.

#### B.1 SAMBA

We illustrate a Samba set-of-sequences model in Fig. A. A Samba model (Fig. A) is composed of a set of siamese Samba units (one for each sequence being modeled) with shared weights. Each Samba unit is synchronized with others through our synchronized SSM layer. In particular, a Samba unit is composed of  $N$  non-linear Samba blocks. To obtain a non-linear Samba unitblock that can be embedded into a neural network, we wrap the synchronized SSM layer following the Mamba (Gu & Dao, 2023) architecture. A linear projection expands the input dimension  $D$  by an expansion factor  $E$ , followed by a causal convolution and a SiLU (Hendrycks & Gimpel, 2016) activation before being fed to the sync SSM layer. The output of a residual connection is passed to a SiLU before being multiplied by the output of the synchronized SSM and passed to an output linear projection. More-

810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863

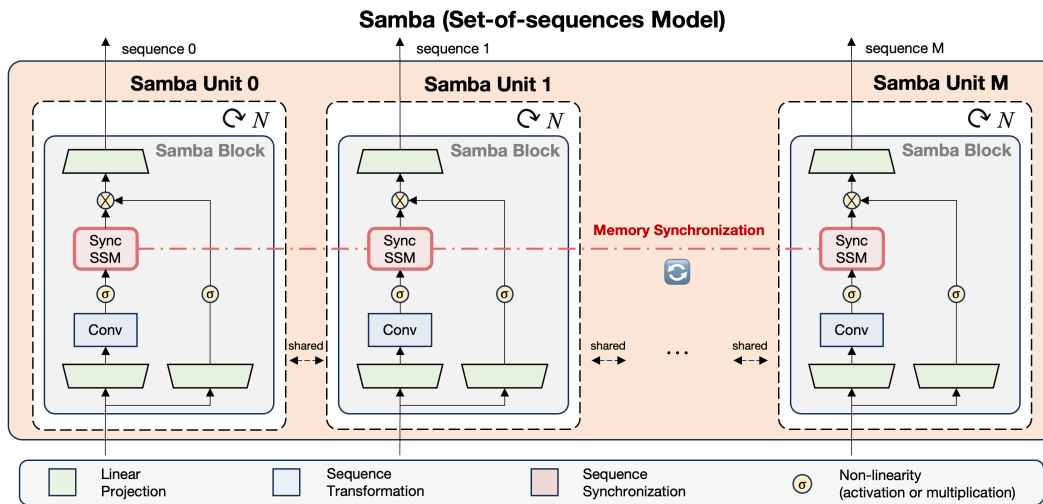


Figure A: **Illustration of our Set-of-sequences Model block.** Our set-of-sequences model Samba simultaneously processes an arbitrary number  $M$  of input sequences. Each sequence is processed by a Samba unit, synchronized with the others thanks to our synchronized state-space model. All Samba units share weights and are composed of a stack of  $N$  Samba blocks. A Samba block has the same architecture as a Mamba block, but it adopts our synchronized SSM to synchronize long-term memory representations across the individual state-space models.

over, we replace Mamba’s RMSNorm (Zhang & Sennrich, 2019) with LayerNorm (Ba et al., 2016) for consistency with the detector. Finally, we repeat  $N$  such Samba blocks, interleaved with standard normalization and residual connections, to form a Samba unit. The resulting set-of-sequences model is linear-time, supports a variable number of sequences that start and end at different time steps, and models long-range relationships and interdependencies across multiple sequences.

## B.2 SCHEMATIC ILLUSTRATION OF OUR CONTRIBUTIONS

We provide a schematic illustration of our contributions towards building Samba in Fig. B, disentangling them from one another to make the functioning of each component clear.

Mamba is the underlying sequence model, shown in the first row (Mamba). The second row depicts our strategy to deal with uncertain observations by ignoring them in the state update (Occlusion Masking). Synchronization across multiple sequence models using our synchronization module to let their hidden states communicate to model sequence interaction is shown in the third row (Sync). The last row illustrates our efficient training strategy to learn long-range dynamics from longer sequences at a comparable computational expense for backpropagation (Longer).

Each of these components is ablated in Tab. 4 by incrementally adding them within our framework, showing the effectiveness of each towards the impressive final performance of SambaMOTR.

## C ADDITIONAL RESULTS

We report additional results on the popular MOT17 pedestrian tracking benchmark in App. C.1. We extend our ablation study in App. C.2, investigating the effectiveness of synchronization, the use of positional embeddings and the effectiveness of residual prediction.

### C.1 MOT17

While MOT17 served as a benchmark of paramount importance to advance the state of current multiple object tracking algorithms, its very small size is reducing its significance as a training dataset. Since MOT17 only counts 7 training videos, modern tracking solutions complement its training with additional detection datasets and increasingly stronger detectors to improve the overall



864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917

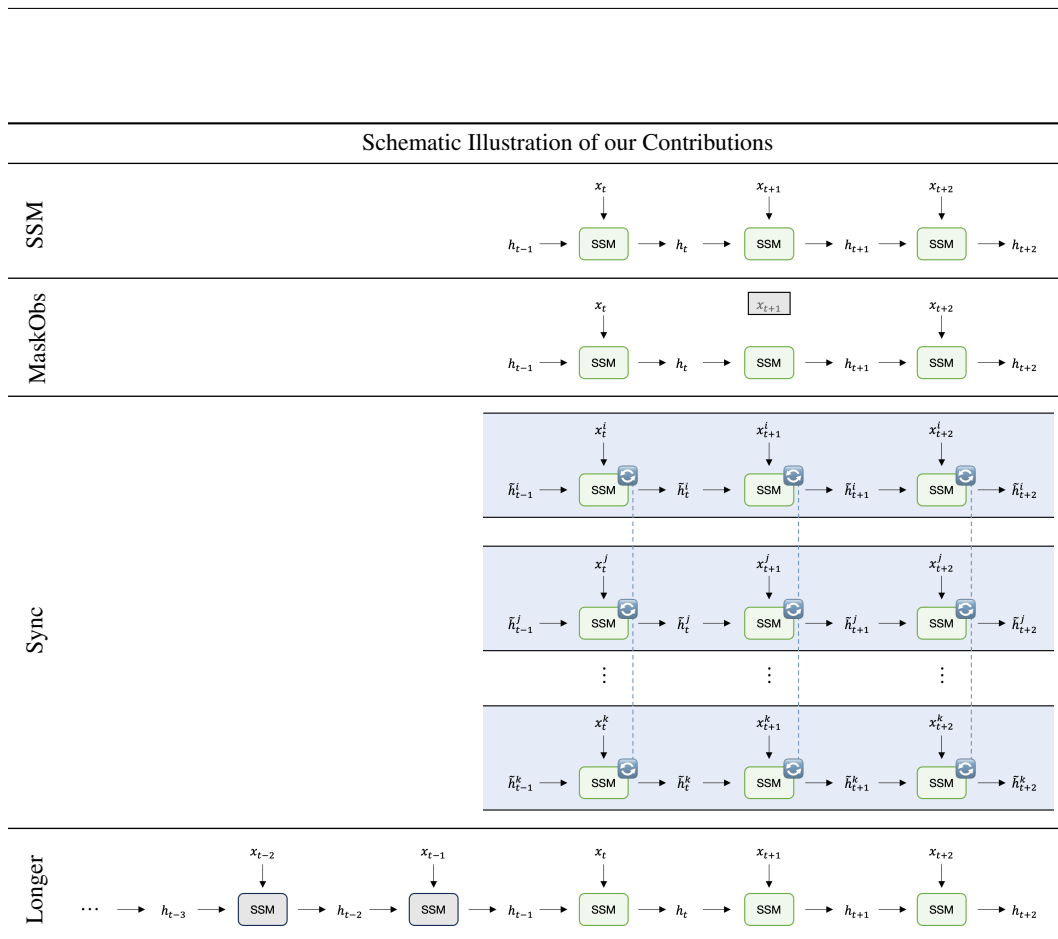


Figure B: Schematic illustration of our contributions (as ablated in Tab. 4). State-space model (SSM) blocks at timesteps with gradient applied are in green, and blocks without gradient are in grey.

tracking performance and top the leaderboard. However, such expedients are deviating from the study of fundamental tracking solutions and focusing more on engineering tricks. Moreover, due to its highly linear motion, its small size (only 7 videos), and the subsequent need for training on additional detection datasets, end-to-end tracking methods do not provide additional advantages over more naive Kalman-filter-based methods. For this reason, we preferred to it other more modern and meaningful datasets in the main paper, *i.e.* DanceTrack (Sun et al., 2022), SportsMOT (Cui et al., 2023) and BFT (Zheng et al., 2024), which allows us to study the importance of modeling tracklets interaction and of implicitly learning motion and appearance models to cope with the underlying non-linear motion, appearance and pose changes of the objects. Nevertheless, we here compare with the state-of-the-art for completeness and show comparable performance to previous tracking-by-propagation methods.

## C.2 ABLATION STUDIES

We here complement the ablation study in Sec. 5.4 with additional experiments on specific SambaMOTR’s design choices. All ablations are based on the final version of our method, including all contributions as in Tab. 4 line d.

**Ablation on different formulations of synchronization.** In Tab. C, we ablate on different formulations of state synchronization and report the corresponding state update equation for each option. In particular, the first row (Sync: -) does not apply state synchronization and is equivalent to using Mamba as a query propagation module together with our occlusion masking and efficient longer training strategy as explained in Sec. 4.3. Since this option does not model tracklet interaction, it reports the lowest performance. We then compare synchronizing the hidden state before (prior) or after (posterior) and find that synchronizing the posterior is more effective. We attribute this to the

Table A: State-of-the-art comparison on MOT17 (Milan et al., 2016). Best tracking-by-propagation method in **bold**; best overall underlined. For a fair comparison with the only MeMOTR’s published result, we also adopt DAB-Deformable-DETR.

Methods	Detector	HOTA	AssA	DetA	IDF1	MOTA
<i>Tracking-by-detection:</i>						
CenterTrack (Zhou et al., 2020)	CenterNet (Duan et al., 2019)	52.2	51.0	53.8	64.7	67.8
FairMOT (Zhang et al., 2021)		59.3	58.0	60.9	72.3	73.7
TransTrack (Sun et al., 2020)	Deformable DETR (Zhu et al., 2020)	54.1	47.9	61.6	63.9	74.5
TransCenter (Xu et al., 2022)		54.5	49.7	60.1	62.2	73.2
MeMOT (Cai et al., 2022)		56.9	55.2	-	69.0	72.5
GTR (Zhou et al., 2022)	CenterNet2 (Zhou et al., 2021)	59.1	57.0	61.6	71.5	75.3
DeepSORT (Wojke et al., 2017)	YOLOX-X (Ge et al., 2021)	61.2	59.7	63.1	74.5	78.0
SORT (Bewley et al., 2016)		63.0	62.2	64.2	78.2	80.1
ByteTrack (Zhang et al., 2022)		63.1	62.0	64.5	77.3	80.3
OC-SORT (Cao et al., 2023)		63.2	63.4	63.2	77.5	78.0
QDTrack (Pang et al., 2021)		63.5	62.6	64.5	77.5	78.7
C-BIoU (Yang et al., 2023)		64.1	63.7	64.8	79.7	81.1
MotionTrack (Qin et al., 2023)		65.1	65.1	65.4	80.1	81.1
<i>Tracking-by-propagation:</i>						
TrackFormer (Meinhardt et al., 2022)	Deformable DETR (Zhu et al., 2020)	-	-	-	68.0	74.1
MOTR (Zeng et al., 2022)		57.2	55.8	58.9	68.4	71.9
MeMOTR (Gao & Wang, 2023)	DAB-Deformable DETR (Liu et al., 2022)	58.8	58.4	59.6	71.5	72.8
SambaMOTR (ours)		58.8	58.2	59.7	71.0	72.9

opportunity to compensate for occluded observations in the current frame with the dynamics from other unoccluded tracklets to better model track query propagation through occlusions.

**Ablation on the effect of synchronization on hard DanceTrack sequences.** In Tab. B, we report the performance on hard sequences of the DanceTrack test set for two SambaMOTR with and without synchronization. We select the top-6 hardest sequences for the version without synchronization and show that utilizing synchronization greatly improves the overall metrics.

**Ablation on the use of query positional embeddings in Samba.** In Tab. D, we ablate on the addition of positional embeddings to the track embeddings before feeding them to Samba. We find that positional embeddings are very beneficial to Samba, arguably because they enable to implicitly learn non-linear motion models.

**Ablation on the prediction of residual vs. full queries with Samba.** In Tab. E, we ablate on the output format of our Samba-based query propagation module. We compare two versions: one that directly outputs the final track queries with Samba, and one that predicts a residual over the track queries used to detect in the current frame. We find that learning a residual is significantly more effective than directly predicting the final track query.

**Ablation on the query propagation strategy through occlusions.** We compare two query propagation strategies for occluded track queries in Tab. F. First, we evaluate our model with MeMOTR’s (Gao & Wang, 2023) query propagation strategy (Freeze), which freezes the last observed state - *i.e.* the last track query that generated a confident detection - and memory until the tracklet is detected again in a new frame. Next, we compare this with actively propagating occluded track queries and their memory through occlusions using our MaskObs strategy (Sec. 4.3). We find that MaskObs outperforms Freeze: by inferring a tracklet’s future state during occlusions using only its past memory and interactions with other observed objects, it keeps tracklets alive longer.

972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025

Table B: Ablation on the effect of synchronization on difficult sequences on DanceTrack test.

Sequence	Synchronization					
	✗			✓		
	HOTA	DetA	AssA	HOTA	DetA	AssA
dancetrack0046	34.5	54.0	22.1	39.8	60.7	26.2
dancetrack0085	39.6	60.9	25.9	40.3	63.9	25.5
dancetrack0050	41.9	66.4	26.5	42.4	69.8	25.8
dancetrack0036	42.9	74.7	24.7	48.4	77.5	30.3
dancetrack0028	43.1	71.9	25.8	47.8	73.6	31.1
dancetrack0009	43.4	73.6	25.7	48.0	75.2	30.7
<i>average</i>	40.9	66.9	25.1	<b>44.5</b>	<b>70.1</b>	<b>28.3</b>

Table C: Ablation on memory synchronization positioning. We report the state equation and performance on the DanceTrack test set for: (i) the baseline without synchronization (-); (ii) synchronization on the updated state prior to input contribution (Prior); (iii) synchronization on the fully-updated state (Posterior).

Sync	State Equation	HOTA	AssA	DetA	IDF1	MOTA
-	$h_t^i = \bar{\mathbf{A}}^i(t)h_{t-1}^i + \bar{\mathbf{B}}^i(t)x_t^i$	66.0	56.4	77.5	69.5	86.7
Prior	$h_t^i = \Gamma_{i \in \mathcal{T}}(\bar{\mathbf{A}}^i(t)h_{t-1}^i) + \bar{\mathbf{B}}^i(t)x_t^i$	66.1	56.7	77.3	70.0	86.4
Posterior	$h_t^i = \Gamma_{i \in \mathcal{T}}(\bar{\mathbf{A}}^i(t)h_{t-1}^i + \bar{\mathbf{B}}^i(t)x_t^i)$	<b>67.2</b>	<b>57.5</b>	<b>78.8</b>	<b>70.5</b>	<b>88.1</b>

Table D: Ablation on the use of positional embeddings. We ablate on the addition of positional embeddings to the observed queries fed as input to the Samba module.

Query Position	HOTA	AssA	DetA	IDF1	MOTA
-	65.6	56.2	76.7	69.3	85.4
✓	<b>67.2</b>	<b>57.5</b>	<b>78.8</b>	<b>70.5</b>	<b>88.1</b>

Table E: Ablation on the use of residual prediction. We evaluate two formats for the output of SambaMOTR’s Samba module, *i.e.* direct query prediction (-) and prediction of a residual wrt. the track query from the previous frame (✓).

Residual	HOTA	AssA	DetA	IDF1	MOTA
-	64.2	54.0	76.7	67.0	84.5
✓	<b>67.2</b>	<b>57.5</b>	<b>78.8</b>	<b>70.5</b>	<b>88.1</b>

Table F: Ablation on strategies for tracking through occlusions. We evaluate two strategies for tracking objects through occlusions: freezing the last observed track state (Freeze) as in MeMOTR, and propagating queries through occlusions using our MaskObs strategy.

Strategy	HOTA	AssA	DetA	IDF1	MOTA
Freeze	65.9	56.6	76.8	69.7	86.3
MaskObs	<b>67.2</b>	<b>57.5</b>	<b>78.8</b>	<b>70.5</b>	<b>88.1</b>