GMOT-MAMBA: MAMBA-BASED MODEL PREDICTION FOR GENERIC MULTIPLE OBJECT TRACKING

Shashikant Verma[1], *Nicu Sebe*[2], *Shanmuganathan Raman*[1*]

[1]Indian Institute of Technology Gandhinagar, India
[2]University of Trento, Italy
shashikant.verma@iitgn.ac.in, niculae.sebe@unitn.it, shanmuga@iitgn.ac.in

ABSTRACT

We introduce GMOT-Mamba, a novel Mamba-based model prediction framework for Generic Multiple Object Tracking (GMOT) in video sequences. Our approach features a Weighted Feature Pooling (WFP) layer, which processes encoded target states, and an innovative encoder-decoder architecture that leverages Vision-Mamba (ViM) to predict filter weights. We train our model on combinations of large-scale datasets to capture strong priors and discriminative features necessary for generic object tracking. Through extensive experiments and ablation studies, we demonstrate the effectiveness of our approach, showcasing its competitive performance against state-of-the-art GMOT methods while outperforming SOT methods in both accuracy and inference speed. Our findings underscore the potential of Mamba for enhancing model prediction in visual tracking applications.

Index Terms— Generic Object Tracking, Vision Mamba, State Space Models, Multiple Object Tracking.

1. INTRODUCTION

Visual object tracking is one of the fundamental problems in computer vision. The task focuses on determining the state of a target object throughout each frame of a video sequence, starting from a given initial location. Traditionally, tracking tasks are approached by drawing on techniques from object detection [1], segmentation [2], and discriminative correlation filtering (DCF) [3, 4, 5]. Recent developments in the field are extending capabilities to track multiple generic objects simultaneously. Recent developments in tracking are expanding the capabilities to track multiple generic objects simultaneously. Although there is a rich history of research in this area, most Generic Object Tracking (GOT) methods and benchmarks have concentrated on tracking a single object in a video, leading to the introduction of the term Single Object Tracking (SOT). Nevertheless, GOT extends beyond single-object tracking; tracking multiple objects simultaneously can improve the robustness of the tracker through combined reasoning. Additionally, Generic Multiple Object Tracking (GMOT) approaches significantly reduce computational costs by utilizing shared components rather than employing separate Single Object Tracking (SOT) methods for each target.

Among the current approaches for generic object tracking, DCF-based methods have achieved considerable success. These methods learn a target model by optimizing an objective function that incorporates both foreground and background information from previous frames, enabling practical global reasoning during model learning. In parallel, Transformers have emerged as powerful tools for providing strong global reasoning across multiple frames, capturing prior information through self and cross-attention mechanisms. As a result, numerous studies have successfully utilized Vision Transformers for tracking tasks [6]. More recent research [5, 4] has explored the integration of learned priors into the DCF framework, moving beyond traditional methods that rely solely on minimizing an objective based on previous frames by leveraging the capabilities of Transformers.

In this work, we aim to leverage the powerful capabilities of Mamba [7], a time-variant state space model (SSM). Mamba employs a selective scanning mechanism similar to the attention mechanisms used in Transformers and has demonstrated improved inference efficiency and reduced computational complexity. Studies employing Mamba for vision tasks have demonstrated better capabilities to capture richer temporal and global context when compared to Transformers while achieving nearly linear complexity during inference. We propose a Mamba-based Model Prediction framework for Generic Multiple Object Tracking and demonstrate that our approach, incorporating Mamba, achieves improved performance over conventional vision backbones. Additionally, we provide a comprehensive comparison with

* This work is supported by Jibaben Patel Chair in AI.

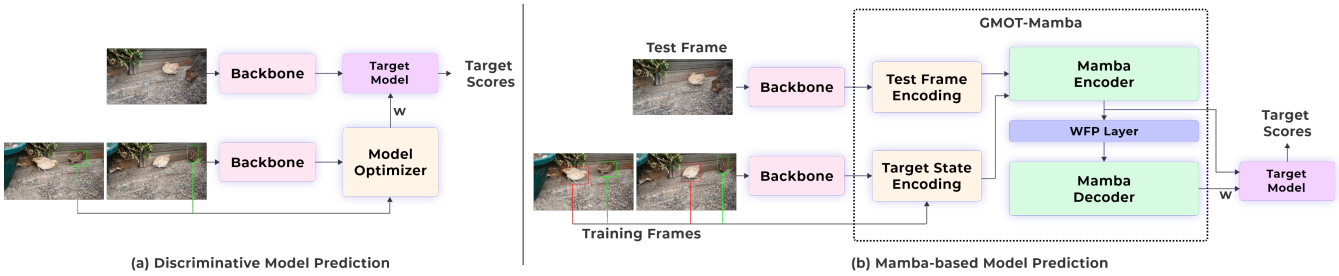

Fig. 1. The existing generalized model prediction framework used by discriminative correlation filtering-based trackers is illustrated in (a), compared to the proposed GMOT-Mamba in (b), where we utilize Vision-Mamba as a foundational component to learning target model weights. Additionally, we extend single object tracking (SOT) to multiple object tracking (MOT) by predicting individual target model weights, each specific to the object to be tracked simultaneously.

Transformer-based methods to evaluate the impact and advantages of integrating Mamba in this context.

Contributions. In summary, our contributions are:

- We propose a Mamba-based Model Prediction framework for Generic Multiple Object Tracking and demonstrate its competitive performance in target model prediction.
- We introduce a Weighted Feature Pooling (WFP) Layer for processing encoded target states of multiple objects, along with an encoder-decoder architecture that utilizes Vision-Mamba to predict target model weights.
- We perform extensive ablation studies and a thorough comparison with Transformer-based methods to assess the impact and benefits of integrating Mamba in the context of model prediction-based tracking.

2. RELATED WORK

Visual Object Tracking. Object tracking is a fundamental and extensively studied task in computer vision. Siamese tracking methods [8] have gained popularity for their simplicity and speed. More recently, DCF-based approaches [3, 5, 4] have emerged and become widely adopted. These trackers work by solving an optimization problem to estimate filter weights that effectively differentiate the target from other objects and the background. Over the years, research efforts have primarily focused on two distinct task definitions: Generic Object Tracking (GOT) [9, 10] and Multiple Object Tracking (MOT) [11]. Most MOT trackers utilize semantic class information and employ a tracking-by-detection approach, while most GOT-based methods are mainly explored in single object tracking (SOT) settings. In this work, we take a step towards tracking multiple generic objects simultaneously by expanding on a DCF-based tracking framework.

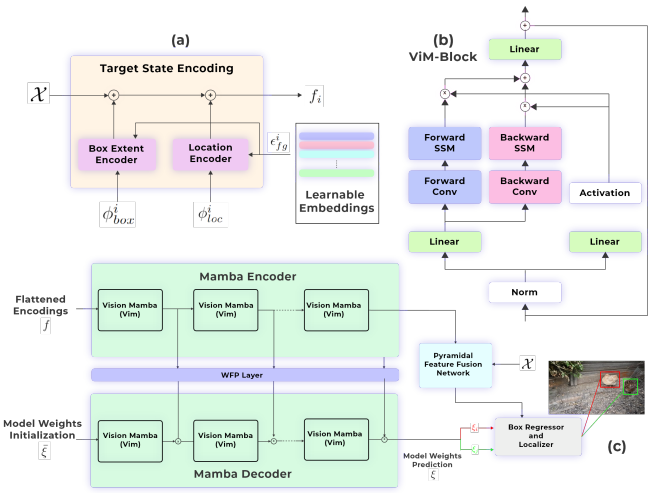

Fig. 2. Main building blocks of the proposed GMOT-Mamba architecture. In (a), we demonstrate the calculation of Target State Encodings, initialized using a bounding box in LTRB representation, ϕ_{loc}, along with a Gaussian map centered on the bounding box's center, ϕ_{loc}. (b) ViM Block. (c) presents a high-level overview of the Mamba-based model prediction network, which leverages Vision-Mamba (ViM). Specifically, we implement the ViM block following [12] and adopt the efficient selective scanning approach from [7].

Baseline Architectures for Tracking. Traditional Siamese architectures, while effective at capturing semantic context, struggle to generalize to out-of-distribution objects typical in Generic Object Tracking. With the rise of Transformers and their strong ability to model global context and capture robust priors, Vision Transformers have been directly applied to tracking tasks [6]. These have been further enhanced by integrating them with DCF-based methods [5, 3, 4] and meta-learning techniques [2]. Recently, with the advances in Struc-

tured State Space (S4) models and their effectiveness in sequential modeling, these models have been integrated into various vision tasks [13, 12]. However, limited focus has been on applying these models specifically to tracking. This work uses Mamba as the baseline architecture to develop a weight prediction model and thoroughly evaluate state space models' effectiveness in tracking tasks.

3. BACKGROUND

We begin with an overview of existing model predictors and the key principles of Mamba, followed by a detailed explanation of our proposed approach.

Mamba. The structured state space models (S4) [14] and Mamba, a selective state space model [7], have recently emerged as a promising class of architectures for sequence modeling. SSMs transform an input sequence $x(t) \in \mathbb{R}$ into an output sequence $y(t) \in \mathbb{R}$ through a hidden latent state $h(t) \in \mathbb{R}^N$. The discretized **S4** models can be represented as recurrent formulation: $y_t = Ch_t$, where $h_t = \bar{A}h_{t-1} + \bar{B}x_t$ or through global convolution: $y = x * \bar{K}$, where $\bar{K} = (C\bar{B}, C\overline{AB}, \ldots, C\overline{A^k B}, \ldots)$.

$$\bar{A} = exp(\Delta A) \qquad \bar{B} = (\Delta A)^{-1}(exp(\Delta A) - I) \cdot \Delta B \quad (1)$$

Mamba incorporates a selection mechanism into **S4** models by making the parameters (Δ, B, C) input-dependent through a series of linear layers [7]. This input dependence transforms the traditionally linear-time invariant **S4** models into time-variant models, which means that state space models can no longer be parallelized using the global convolution approach discussed earlier. We leverage the selection mechanism introduced by [7] to overcome the limitations of LTI models and utilize their hardware-aware state expansion implementation of selective scan to mitigate the computational bottlenecks arising from the time-variant nature of Mamba.

Discriminative Model Prediction. These approaches, illustrated in Figure 1(a), learn a target model from train-frames $\mathcal{S}_{train}$ to localize object in the test-frame $\mathcal{S}_{test}$. A widely adopted formulation by these approaches is to solve an optimization problem such that the target model produces the desired target state $y_i \in \mathcal{Y}$ given training-frames $\mathcal{S}_{train} \in \{(x_i, y_i)\}_{i=1}^{k}$. Here, $x_i \in \mathcal{X}$ are deep feature maps of the i-th input frame, and k is a total number of training frames. The optimization problem is defined in Equation 2.

$$w = \arg\min_{w} \sum_{(x,y)\in\mathcal{S}_{train}} f(\tau(\bar{w}; x), y) + \lambda r(\bar{w}) \quad (2)$$

Here, we seek to optimize for w by minimizing error measured by function f between the target model's output $\tau(\bar{w}; x)$ and ground-truth labels y. The term $r(\bar{w})$ represents the regularization weighted by parameter λ.

4. METHODOLOGY

4.1. GMOT-Mamba: Mamba-Based Model Prediction Network

Target state encodings. Given the training frames $\mathcal{S}_{train}$, we encode the bounding boxes of all m targets in $xywh$ format using the LTRB representation, where each $\phi^i_{box} \in \mathbb{R}^{H\times W\times 4}$. These encodings are subsequently passed through a series of MLPs, projecting them into a higher-dimensional space, denoted as k. Additionally, for each target object, we represent its location with a Gaussian map, $\phi^i_{loc} \in \mathbb{R}^{H\times W}$, centered on the bounding box's center. We adopt a similar encoding formulation as presented in [5, 4] to enable simultaneous tracking of multiple objects. To develop a tracker capable of tracking m objects, we construct a pool of learnable embeddings, $\epsilon_{fg} \in \mathbb{R}^{m\times k}$. The final encoding for the i-th object in $\mathcal{S}_{train}$ is then defined in Equation 3.

$$f = \mathcal{X} \ + \sum_{i=0}^{m} \epsilon^i_{fg} \cdot \phi^i_{loc} \ + \sum_{i=0}^{m} \epsilon^i_{fg} \cdot \mathrm{MLP}(\phi^i_{box}) \quad (3)$$

Here, $\mathcal{X} \in \mathbb{R}^{H_f\times W_f\times k}$ is a high dimensional feature map of the training frame extracted from the backbone network ResNet-50. Additionally, we incorporate test-frame features into the encoding as $f_{test} = x_{test} + \epsilon_{test}$, where ϵ_{test} is a learnable token, similar to ϵ_{fg}. A visual representation of target state encoding estimation is shown in Figure 2(a).

Vision Mamba (ViM). We expand the main components of GMOT-Mamba, illustrated in Figure 1 in a generalized fashion with a more detailed depiction provided in Figure 2. The original Mamba block proposed by [7] is designed for 1-D sequences and can not be employed for vision-related tasks directly. However, similar to Vision Transformers (ViT) [15], the Mamba block can be adapted for images with modifications [12], as shown in Figure 2(b). This involves transforming the image space into flattened image patches. For additional details on the ViM architecture, please refer to [12]. In our approach, the target state encoding in Equation 3 represents a flattened pixel feature representation, where $f \in \mathbb{R}^{H_f\cdot W_f\times k}$, and serves as input to the initial ViM block of the encoder branch, as shown in Figure 2(c). Simultaneously, the decoder branch receives a zero-initialized weight matrix $\bar{\xi}$, which is processed by another set of ViM blocks and fused with the encoder branch features using the WFP Layer. Starting with $\bar{\xi} = \mathbf{0}_{m\times k}$, our model prediction network learns the model weights ξ from encodings, incorporating data-driven learned priors that traditional DCF trackers lack.

Weighted Feature Pooling Layer. We propose the Weighted Feature Pooling (WFP) Layer, which consists of a single Xavier-initialized learnable transformation

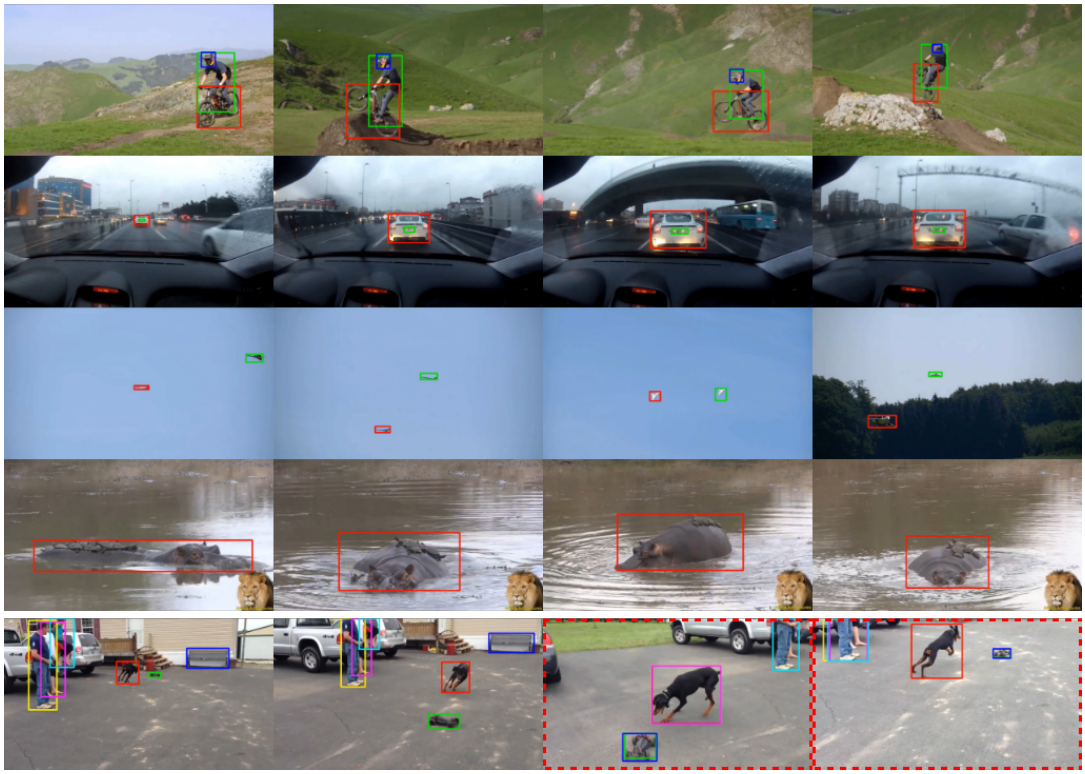

Fig. 3. Tracking results using our proposed approach on validation split sequences of LaGOT. Each frame shown is at least 200 frames apart from the previous one. The last row displays a failure case, and sub-optimal tracking is obtained where the target object either exits the frame or becomes occluded. For better clarity, zoom-in is recommended. Tracked instances are colored with the same color boxes across frames.

matrix, $\mathcal{W} \in \mathbb{R}^{H_f \cdot W_f \times m}$. The WFP layer maps the output from the ViM blocks of the encoder module to the space of ξ, i.e., it transforms from $\mathbb{R}^{H_f \cdot W_f \times d}$ to $\mathbb{R}^{m \times d}$. Here, d represents a high-dimensional space at a certain network depth. The transformed *input-aware* and *target-state aware* features are then combined to decoder ViM outputs through an addition operation, as shown in Figure 2(c).

Box Regression and Target Localization. After estimating the target model weights ξ from the decoder module for filtering, we adopt neural architecture for the Box-Regressor and localizer modules and approach as presented in [4, 5] for target localization and bounding box regression. The key difference is that rather than applying filtering on the low-resolution output from the Mamba-encoder block, we use a Pyramidal Feature Fusion Network [16] first to obtain high-resolution features $\mathcal{X}_H$. We use the same ξ as weights for both localization filtering and box-regression filtering, i.e., $\xi = \xi_{loc} = \xi_{box}$. The filtering operation for localizer defined as $\phi^i_{loc} = \xi^i_{loc} * \mathcal{X}_\mathcal{H}$ and for box regression as $\phi^i_{box} = \xi^i_{box} * \mathcal{X}_\mathcal{H}$.

Loss Functions To supervise our network, we employ two losses, specifically for target localization and bounding box regression. For the ground truth bounding box annotation of object i in a frame, we first generate ϕ^i_{loc} as a Gaussian map centered on the bounding box. A localization loss [3, 4] is applied between ϕ^i_{loc} and the map predicted by the localizer, $\hat{\phi}^i_{loc}$, for the i-th object. If only n instances are tracked out of m possible tracklets in a given training iteration, the target scores for the remaining $m - n$ embeddings should produce low values. Based on this, we define the localization loss in Equation 4. For the bounding box, we utilize IoU Loss [17] for predictions corresponding to n targets and ignore others. In our implementation, we use a pool of $m = 10$ embeddings to supervise our network in tracking a maximum of ten objects simultaneously.

$$L_{loc} = \Sigma_{i=0}^{n} L_{focal}(\hat{\phi}^i_{loc}, \phi^i_{loc}) + \Sigma_{i=n}^{m} L_{focal}(\hat{\phi}^i_{loc}, \mathbf{0}) \quad (4)$$

5. RESULTS AND DISCUSSION

Datasets. To train GMOT-Mamba for generic multiple object tracking, we use a combination of large-scale SOT and MOT datasets, including LaSOT [18], GOT-10k [19], TAO [20], and COCO-context [21]. Among these, TAO and COCO-context provide annotations for

Table 1. Comparison with state-of-the-art methods

Tracker	Model	LaGOT			LaSOT	
		F1	Suc.	HOTA	Precision	Suc.
GMOT	**TaMOS** [4]	0.634	63.8	62.2	74.8	76.9
GSOT	**ToMP-50** [5]	0.612	60.2	59.4	73.1	75.5
GSOT	**DiMP** [3]	0.578	57.4	56.6	63.9	65.1
GSOT	**KeepTrack** [22]	0.614	60.8	59.3	73.2	75.8
GSOT	**MixFormer** [23]	0.621	62.3	61.2	73.7	76.3
MOT	**QD-Track** [24]	0.187	20.7	22.8	29.1	30.7
MOT	**OV-Track** [25]	0.143	13.7	20.8	25.3	26.4
GMOT	**Ours**	0.618	61.7	59.8	73.2	76.2

Table 2. Inference (FPS) comparison with Model Prediction Methods

Method	Obj. (n) = 1	Obj. (n) = 3	Obj. (n) = 5	Obj. (n) = 10
DiMP[3]	37.2	18.6	7.44	3.72
ToMP[5]	39.6	19.8	7.92	3.96
TaMOS[4]	21.7	19.7	18.4	14.3
Ours	23.4	21.6	19.9	15.4

multiple instances. Since COCO-context is an image-based dataset, we apply random translations and rotations to the frames, simulating synthetic video for training. LaSOT and GOT-10k offer annotations for single object tracking. Since supervision needs to be robust for k objects where $1 \leq k \leq m$, training on a mixture of datasets is crucial. The LaGOT validation benchmark, introduced by [4], extends LaSOT by providing annotations for multiple instances within the videos. For GMOT validation, we utilize 100 sequences from LaSOT, averaging 2500 frames per video, with corresponding annotations provided by LaGOT. Additionally, we use same 100 sequences with original annotations of LaSOT for evaluating SOT metrics.

Training Details. We sample three frames from the video sequence sampled uniformly across all datasets above. The first two frames are designated training frames to generate target state encodings, while predictions are made on the remaining frame, which serves as the test frame. In each epoch, 50K videos are sampled uniformly from all datasets, and the network is trained on NVIDIA GeForce RTX 4090 GPUs for 80 epochs.

SOTA Comparisons We evaluate our approach against several existing state-of-the-art methods, including TaMOS [4], ToMP [5], DiMP [3], KeepTrack [22], MixFormer [23], QD-Track [24], and OV-Track [25]. Among these, TaMOS is a Generic-MOT tracker, while QD-Track and OV-Track are MOT trackers that are not well-suited for tracking unknown generic objects. We include them in our evaluation to highlight the contrast in performance for generic object tracking. In Table 1, we present a comparison of GMOT-Mamba (Ours) on the LaGOT and LaSOT datasets, which are widely adopted for benchmarking MOT and SOT performance, respectively. The comparison includes metrics such as F1 score, precision, Success, and the HOTA metric [26]. Compared to SOT-based methods, it falls behind only MixFormer [23]. However, it is important to note that SOT methods cannot track multiple objects simultaneously, unlike our approach. We report metrics on LaGOT by running SOT trackers m times, where m corresponds to the number of track IDs available for each sequence. Our method performs competitively with TaMOS [4] in tracking multiple objects simultaneously. In Table 2, we show inference speed obtained by different model prediction-based approaches. Our method achieves notable FPS gains when tracking multiple objects (n out of m) compared to SOT-based methods and demonstrates improved speed over TaMOS. In Figure 3, we present qualitative results obtained by GMOT-Mamba.

Table 3. Ablation Study

Model Prediction	Bi-directional	Time-Invariant	SSM	LaGOT	
				F1-Score	Success
Mamba-ED	✓	✗	Selective	0.58	57.64
Mamba-ED	✗	✗	Selective	0.54	57.28
S4-ED	✓	✓	Structured	0.51	55.02
S4-ED	✗	✓	Structured	0.48	54.32

Ablation Study In Table 3, we present an exhaustive study of incorporating various types of SSMs in the model prediction approach, discussed in Section 4.1. Initially, we modify the proposed Encoder-Decoder (ED) modules by incorporating bi-directionality into the ViM Block [12]. Additionally, we eliminate the dependence on input for parameters (Δ, B, C), making the SSM time-invariant, following the structure of S4 Models (Structured SSM). The effects of these changes on ViM blocks, as illustrated in Figure 2(c) is detailed in the Table 3. All configurations presented in this table are evaluated after 40 training epochs for fair assessment. We find that SSMs perform best in bi-directional and selective state space settings, which we utilize to construct the ViM blocks of our proposed network, GMOT-Mamba.

6. CONCLUSION

This work introduces GMOT-Mamba, a model-based prediction approach leveraging Mamba for generic multiple-object tracking. We propose a novel encoder-decoder architecture featuring a standard ViM-block with a Weighted-Feature Pooling layer, which transforms and fuses input features into a learnable matrix to predict filter weights. Our experiments demonstrate that state space models, like Transformers, effectively learn discriminative features. GMOT-Mamba achieves competitive performance compared to state-of-the-art GMOT methods and surpasses SOT methods in both metrics and inference speed. Our ablation studies reveal that selective SSM (Mamba) outperforms traditional S4 Models, showing great promise for vision-related tasks, including tracking.

7. REFERENCES

[1] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei, "Motr: End-to-end multiple-object tracking with transformer," in *European Conference on Computer Vision*, 2022.

[2] Matthieu Paul, Martin Danelljan, Christoph Mayer, and Luc Van Gool, "Robust visual tracking by segmentation," in *European Conference on Computer Vision*. Springer, 2022, pp. 571–588.

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte, "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

[4] Christoph Mayer, Martin Danelljan, Ming-Hsuan Yang, Vittorio Ferrari, Luc Van Gool, and Alina Kuznetsova, "Beyond sot: Tracking multiple generic objects at once," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 6826–6836.

[5] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool, "Transforming model prediction for tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8731–8740.

[6] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1571–1580.

[7] Albert Gu and Tri Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[8] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[9] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, 2014.

[10] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu, "Towards grand unification of object tracking," in *European Conference on Computer Vision*. Springer, 2022, pp. 733–751.

[11] Mingzhen Huang, Xiaoxing Li, Jun Hu, Honghong Peng, and Siwei Lyu, "Tracking multiple deformable objects in egocentric videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1461–1471.

[12] Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang, "Video mamba suite: State space model as a versatile alternative for video understanding," *arXiv preprint arXiv:2403.09626*, 2024.

[13] Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré, "S4nd: Modeling images and videos as multidimensional signals with state spaces," *Advances in neural information processing systems*, pp. 2846–2861, 2022.

[14] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré, "Combining recurrent, convolutional, and continuous-time models with linear state space layers," *Advances in neural information processing systems*, vol. 34, pp. 572–585, 2021.

[15] Alexey Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[17] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[18] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[19] Lianghua Huang, Xin Zhao, and Kaiqi Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[20] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan, "Tao: A large-scale benchmark for tracking any object," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[22] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool, "Learning target candidate association to keep track of what not to track," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 13444–13454.

[23] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu, "Mixformer: End-to-end tracking with iterative mixed attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13608–13618.

[24] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu, "Quasi-dense similarity

learning for multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 164–173.

[25] Siyuan Li, Tobias Fischer, Lei Ke, Henghui Ding, Martin Danelljan, and Fisher Yu, "Ovtrack: Open-vocabulary multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 5567–5577.

[26] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, pp. 548–578, 2021.