Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052329

ARTICLE

Masked Autoencoders as Single Object Tracking Learners

Chunjuan Bo[1,*], Xin Chen[2] and Junxing Zhang[1]

[1]School of Information and Communication Engineering, Dalian Minzu University, Dalian, 116600, China

[2]School of Information and Communication Engineering, Dalian University of Technology, Dalian, 116024, China

*Corresponding Author: Chunjuan Bo. Email: bcj@dlnu.edu.cn

Received: 30 March 2024 Accepted: 30 May 2024 Published: 18 July 2024

ABSTRACT

Significant advancements have been witnessed in visual tracking applications leveraging ViT in recent years, mainly due to the formidable modeling capabilities of Vision Transformer (ViT). However, the strong performance of such trackers heavily relies on ViT models pretrained for long periods, limiting more flexible model designs for tracking tasks. To address this issue, we propose an efficient unsupervised ViT pretraining method for the tracking task based on masked autoencoders, called TrackMAE. During pretraining, we employ two shared-parameter ViTs, serving as the appearance encoder and motion encoder, respectively. The appearance encoder encodes randomly masked image data, while the motion encoder encodes randomly masked pairs of video frames. Subsequently, an appearance decoder and a motion decoder separately reconstruct the original image data and video frame data at the pixel level. In this way, ViT learns to understand both the appearance of images and the motion between video frames simultaneously. Experimental results demonstrate that ViT-Base and ViT-Large models, pretrained with TrackMAE and combined with a simple tracking head, achieve state-of-the-art (SOTA) performance without additional design. Moreover, compared to the currently popular MAE pretraining methods, TrackMAE consumes only 1/5 of the training time, which will facilitate the customization of diverse models for tracking. For instance, we additionally customize a lightweight ViT-XS, which achieves SOTA efficient tracking performance.

KEYWORDS

Visual object tracking; vision transformer; masked autoencoder; visual representation learning

1 Introduction

Single object tracking is a fundamental task within the field of computer vision, aiming to persistently track an arbitrary target object across a video sequence starting from its initial condition [1–3]. In particular, the user provides the bounding box of a target object in the initial frame of a video. Subsequently, the algorithm forecasts the bounding box of this target in the following frames. The past few years have seen remarkable progress in this domain, primarily fueled by the evolution of deep learning technologies [4–8]. Notably, the introduction and integration of the Vision Transformer (ViT) [9] have been pivotal in this wave of advancements, developing numerous tracking algorithms that leverage ViT's architecture to achieve superior performance.

ViT-based trackers [10–12] have shown superior performance, yet they harbor significant shortcomings. Primarily, these trackers depend on ViT models that have undergone extensive pretraining, making any subsequent modifications both costly and labor-intensive. Moreover, the orientation of ViT models' pre-training is generally towards tasks like image classification [13–16], focusing predominantly on static imagery. This approach does not necessarily cater to the nuanced requirements of tracking tasks, which demand a comprehensive grasp of both appearance and motion. Thus, the development of an efficient pretraining method, specifically for the tracking task, is imperative to enhance the tracker's effectiveness.

To address these challenges, this paper introduces a new unsupervised ViT pretraining method for the tracking task, namely TrackMAE. This method utilizes a masked ViT autoencoder to simultaneously consider appearance and motion information, thereby optimizing the ViT model for tracking applications. The rationale behind this approach is predicated on the indispensability of comprehending both appearance and motion for visual object tracking. To address the pretraining's previous effectiveness in motion information modeling, TrackMAE significantly enhances the pretraining process's efficiency and effectiveness.

To achieve the aforementioned objective, we develop a dual-branch pretraining framework comprising an appearance branch and a motion branch. The appearance branch is equipped with an appearance encoder and an appearance decoder, while the motion branch features a motion encoder and a motion decoder. The encoder employs the ViT model, and the decoder employs a smaller transformer model. In the appearance branch, the process is executed on images to facilitate the learning of appearance understanding. Specifically, an image undergoes random masking, with the unmasked tokens inputted into the appearance encoder. Subsequently, the appearance decoder endeavors to reconstruct the original image in the pixel domain. In parallel, the motion branch operates on pairs of video frames to glean insights into the motion relationship between frames. Here, two frames from the same video are randomly masked, and the unmasked tokens are concatenated and fed into the motion encoder. The motion decoder then aims to reconstruct the original frame pair in the pixel domain. By sharing model parameters between the appearance and motion branches, the framework engenders simultaneous learning of both appearance and motion understanding, thereby enhancing the model's proficiency in visual object tracking.

Our experiments demonstrate the efficiency and effectiveness of the proposed TrackMAE method, achieving new state-of-the-art results across multiple tracking benchmarks. In summary, this manuscript includes the following two main contributions.

First, we introduce a new unsupervised pretraining method for ViT models in the visual tracking task, predicated upon a masked autoencoder. This approach is designed to cultivate a nuanced comprehension of both appearance and motion dynamics, thereby facilitating an efficient and potent paradigm for training ViT models specifically tailored for tracking applications.

Second, we unveil a new suite of pretrained ViT models, alongside a corresponding new lineage of ViT-based tracking models, which strike a good trade-off between speed and tracking accuracy. Experiments verify the effectiveness of the proposed new models.

2 Related Work

2.1 Single Object Tracking

In the realm of single object tracking, Siamese-based methods [17–21] have garnered significant popularity. These methods typically utilize a dual-backbone architecture with shared parameters to

concurrently extract features from both the template and search region images. This is followed by the application of a correlation-based network for feature interaction, culminating in the use of head networks for the final prediction task. Some learning-based methods further enhance the Siamese-based trackers. For instance, Zhang et al. [22] proposed to learn better feature representation through a multi-task loss function, enhancing tracking accuracy and robustness. Zhang et al. [23] introduced channel and spatial attention-guided residual learning, which enhances feature representations in Siamese networks and addresses overfitting issues, resulting in improved tracking performance. Innovations such as Transformer Tracking (TransT) [24] and Transformer Meets Tracker (TMT) [25], along with the subsequent enhancements [26,27], have augmented tracking efficacy by integrating the transformer architecture [7–9] for enhanced feature interaction. More recently, the advent of a one-stream framework has set new state-of-the-art performance in tracking, exemplified by Sequence to sequence Tracking (SeqTrack) [10], One-Stream Tracking (OSTrack) [11], Simplified Tracking (SimTrack) [12], Mixed Transformer (MixFormer) [28], and Single Branch Transformer (SBT) [29]. This framework amalgamates feature extraction and fusion within a singular backbone network, a strategy that is both simplistic and effective, leveraging the pretrained capabilities of backbone networks initially optimized for image classification tasks. Pretrained ViT backbone models play a pivotal role in the superior performance of one-stream trackers. Thus, we develop a new specialized pretraining method tailored for the tracking task, further enhancing the tracking performance.

2.2 Self-Supervised Learning

In recent decades, self-supervised learning has emerged as a domain of considerable interest, with a myriad of manually crafted agent tasks devised for pretraining, including but not limited to image colorization [30], future frame anticipation [31], and rotation estimation [32]. Amidst this backdrop, contrastive learning methods [16,33–35] have ascended to prominence as the leading self-supervised strategies, albeit their effectiveness often relies on large pretraining data. On the other hand, drawing inspiration from the natural language processing domain, specifically masked language models [36], the concept of Masked Image Modeling (MIM) has been introduced, aiming at the autonomous learning of image representation. MAE [13] stands as a quintessential exemplar of MIM methodologies. This strategy has demonstrated its efficacy across an extensive range of downstream applications, encompassing image classification, object detection, and video comprehension.

Notably, ViT models pretrained via the MAE method have showcased exemplary efficacy in the realm of visual object tracking. Nonetheless, this pretraining approach is not inherently tailored for tracking purposes, failing to capitalize on tracking-specific data and to fully harness the potential of ViT for tracking tasks. Furthermore, the substantial training costs associated with this method render model modifications prohibitively expensive. In light of these considerations, our work introduces a new efficient pretraining method for tracking. This method uniquely integrates appearance and motion information, thereby optimizing both pretraining efficiency and tracking performance.

3 Proposed Method

The overall framework of the proposed method is illustrated in Fig. 1, which comprises three main stages: pretraining, fine-tuning, and tracking.

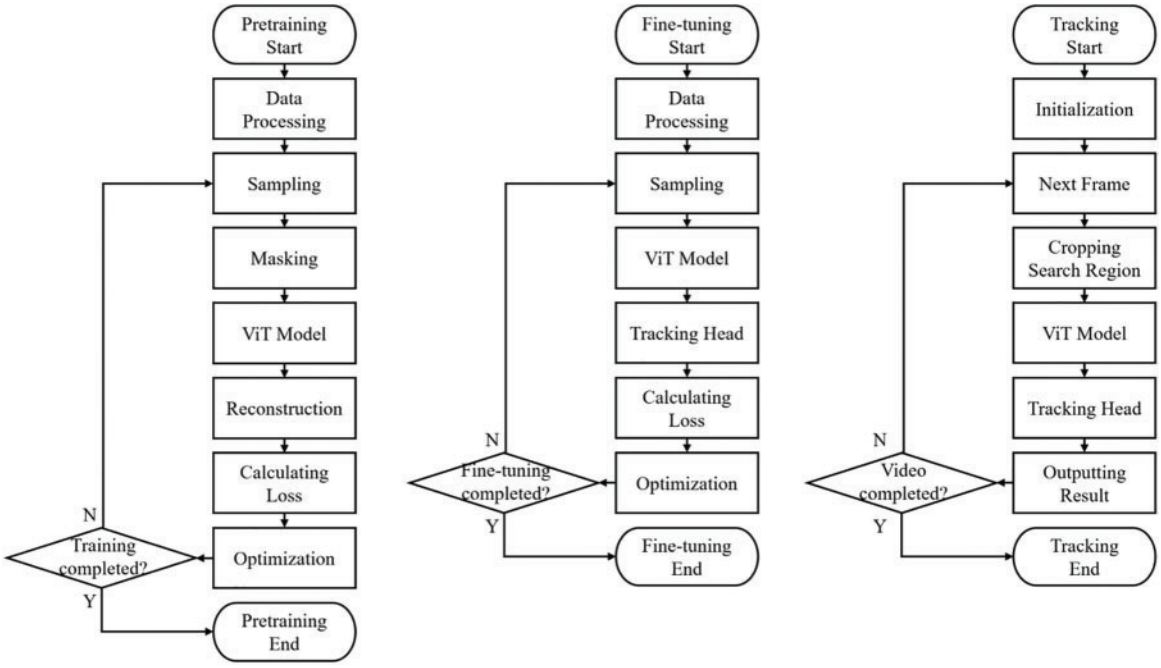

Figure 1: Flowchart of model pretraining, fine-tuning, and tracking

For pretraining, the aim is to enhance the model's feature representation capability. To achieve this, we propose the TrackMAE method, which is the focus of this work. Firstly, we preprocess the collected training data to obtain a collection of input sample units for the model. Then, we sample the required number of samples for one iteration from this collection. Subsequently, random masking is applied, followed by feeding the masked data into the ViT model to obtain latent features. Next, we reconstruct the original image from the latent features and calculate the loss compared to the original image, optimizing the ViT model through gradient descent. This process iterates until the specified number of iterations for pretraining is reached. More details will be discussed in subsequent sections.

The goal of fine-tuning is to teach the model to execute the tracking task. For fine-tuning, commonly used methods in the tracking field are adopted. Initially, we preprocess the collected training data to obtain a collection of sample pairs (consisting of one template image and one search region image). Then, we sample from this collection and input it into the ViT Model. The features outputted by the ViT Model are then input into the tracking head to predict the bounding box on the search region. Subsequently, the loss is computed between the predicted bounding box and the ground truth bounding box, optimizing the ViT Model as well as the tracking head. Similar to pretraining, this process iterates until the specified number of iterations for fine-tuning is reached. The ViT model for fine-tuning initially loads the optimized weights from the pretraining stage.

After pretraining and fine-tuning, the obtained model is capable of performing the tracking task. Specifically, for a video sequence, in the first frame, the user specifies the bounding box of the target, initializing the algorithm to crop the target template from the frame. In the subsequent frames, the search region image is obtained by expanding the bounding box from the previous frame. The search region image and the template image are then input into the ViT model, followed by the Tracking Head, which outputs the tracking result for the current frame. This process repeats for each frame until the end of the video sequence.

In the aforementioned steps, the main focus of our work lies in improving the pretraining phase to obtain better initial weights for the ViT model. The fine-tuning and tracking phases follow popular methods in tracking. Therefore, the subsequent sections will focus on introducing our TrackMAE pretraining method.

3.1 Overview of TrackMAE Pretraining

As illustrated in Fig. 2, Our TrackMAE pretraining method is an autoencoding paradigm that reconstructs the original signal from its partial observation. This methodology bifurcates into two distinct branches: one dedicated to appearance and the other to motion. Within the motion branch, a pair of video frames constitute the original signal, whereas the appearance branch utilizes a single image for this purpose. Each branch is equipped with an encoder that translates the observed signal into a latent representation, coupled with a decoder responsible for the regeneration of the original signal from this latent space. By employing shared parameters across the two encoders, our model concurrently learns to discern the dynamic interplay between two video frames and to capture the static appearance characteristics inherent in a single image.

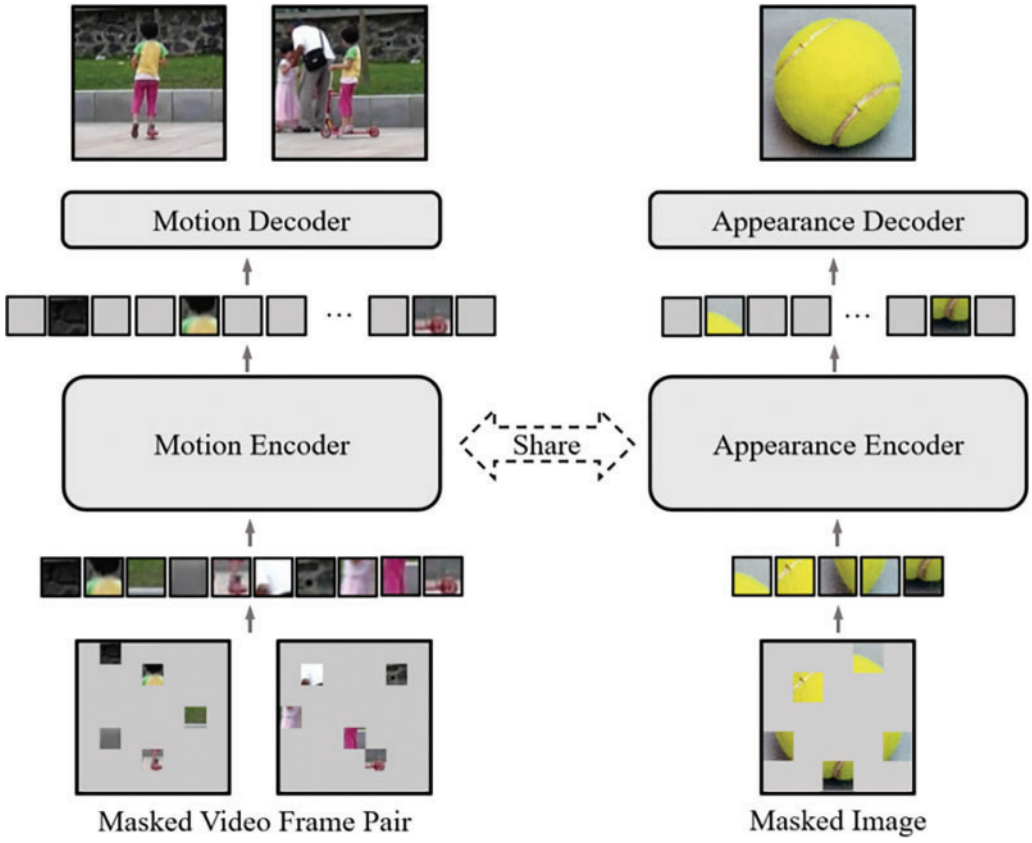

Figure 2: Overall paradigm of the proposed TrackMAE pretraining

3.2 Inputting and Masking

We partition the input comprising both the video frame pair $f_1, f_2 \in \mathbb{R}^{3\times H\times W}$ and the image $x \in \mathbb{R}^{3\times H\times W}$ into uniform, non-overlapping patches with size $3 \times P \times P$. Subsequently, a subset

of these patches is selected, while the remainder are obscured through masking. Following MAE [13], we employ the random sampling strategy, i.e., we sample random patches following a uniform distribution.

Within the appearance branch, the random sampling procedure is implemented with a substantial masking ratio (75% in our experiments). This deliberate reduction in available information cultivates a challenging reconstructive task, thereby enhancing the model's capacity to interpret and encode the nuanced facets of image appearance. Conversely, in the motion branch, we apply a moderate masking ratio of 60%. Given the inherent resemblance between the two video frames under consideration, the preserved information is comparatively abundant. Consequently, the model can organize the retained patches to reconstruct the original video frames. This strategic choice facilitates the model's ability to grasp the spatial relationships of the unmasked patches.

3.3 TrackMAE Encoder

Both the motion and appearance encoders employ a ViT model, leveraging shared weights. Frames and image patches undergo conversion into tokens via linear projection, subsequently augmented with positional embeddings. Then, the video frames' unmasked tokens are concatenated and processed with the motion encoder's Transformer blocks. The image's unmasked tokens are processed with the appearance encoder's Transformer blocks. Since the encoder only processes the unmasked token embeddings, the compute and memory are significantly reduced, allowing us to scale up the encoders.

3.4 TrackMAE Decoder

Both the motion and appearance decoders utilize the Transformer architecture. As shown in Fig. 2, the TrackMAE decoders are fed with the full set of frames and image tokens, which includes both the encoded unmasked patches and the masked tokens. Each masked token is a shared, trainable embedding, denoting the absent patch. Positional embeddings are additionally integrated into this set of tokens. The motion and appearance decoders, respectively processing the frames token set and the image token set, are tasked with predicting the original normalized pixels for each token. Considering the decoders' engagement with the full token set, a lightweight Transformer configuration is adopted to ensure efficiency. The lightweight Transformer consists of 4 blocks with a dimension of 512.

3.5 Reconstruct Target

Our TrackMAE is trained to reconstruct the normalized pixels corresponding to the original frames and image. The terminal layer of the TrackMAE decoder embodies a linear projection, which endeavors to forecast the normalized pixel vector for each patch, denoting the patch index. The output is subsequently reshaped into the form of the reconstructed frames and image. The adopted loss function is the Mean Squared Error (MSE) loss, which quantifies the discrepancy between the reconstructed and original signals within the normalized pixel domain. The loss is confined to the masked patches, summarized as follows.

$$\mathcal{L} = \sum_{j \in \boldsymbol{f1,f2}} \left[\lambda_m \mathcal{L}_{MSE}\left(p_j, \widehat{p_j}\right)\right] + \sum_{j \in \boldsymbol{x}} \left[\lambda_a \mathcal{L}_{MSE}\left(p_j, \widehat{p_j}\right)\right] \tag{1}$$

Here, j stands for the index of unmasked pixels within frames f_1, f_2, or image x. p_j represents the $j-th$ predicted pixel, and $\widehat{p_j}$ is the $j-th$ normalized original pixel. $\lambda_m = 1$ and $\lambda_a = 1$ are the regularization parameters in our experiments.

3.6 Evaluation Tracker

To assess performance on the downstream tracking task, we develop a straightforward tracking framework utilizing our pretrained ViT models. As depicted in Fig. 3, the template and search region images are partitioned into patches. Subsequently, these patches are transformed into tokens via a linear projection layer. The pretrained ViT blocks then proceed to extract feature representations from these tokens. Ultimately, we employ a rudimentary corner-based tracking head [26] to predict the bounding box of the target object. The loss function adheres to a conventional and simplistic paradigm [26] commonly adopted in tracking.

$$\mathcal{L} = \lambda_G \mathcal{L}_{\text{GIoU}}\left(b, \hat{b}\right) + \lambda_1 \mathcal{L}_1\left(b, \hat{b}\right) \tag{2}$$

Herein, $\mathcal{L}_{GIoU}$ signifies the generalized IoU loss, and $\mathcal{L}_1$ represents the $\mathcal{L}_1$-norm loss. b and $\hat{b}$ denote the predicted and ground-truth bounding boxes, respectively. $\lambda_G = 1$ and $\lambda_1 = 1$ are the regularization parameters. For the inference, we also follow the common method [26].

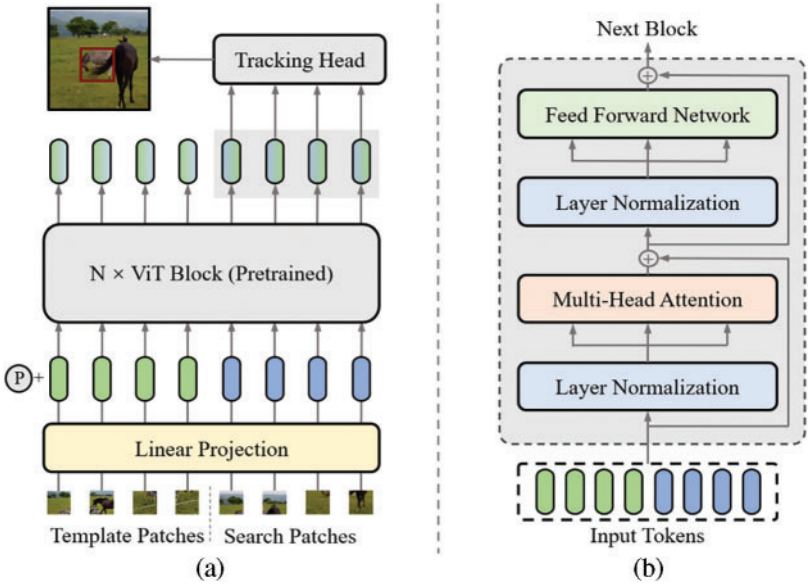

Figure 3: Framework of the evaluation tracker

4 Experiments

4.1 Implementation Details

Models. We employ our Track-MAE method to pretrain three variants of ViT models: ViT-L, ViT-B, and ViT-XS (L: Large, B: Base; XS: eXtra Small). The comprehensive configurations of these models are detailed in Table 1. For all models, the decoder's settings remain consistent: with 8 layers, a hidden size of 512, 16 attention heads, and an Multilayer Perceptron (MLP) size of 2048. Our implementation is carried out utilizing Python 3.8 and PyTorch 1.11.0.

Table 1: Details of TrackMAE pretrained models

Encoder	Layers	Hidden size	MLP size	Heads	Params (M)
ViT-L	24	1024	4096	16	307
ViT-B	12	768	3072	12	86
ViT-XS	4	768	3072	12	29

Pre-training. The datasets utilized for pretraining encompass the train-splits of ImageNet1k [37], COCO (Common Objects in Context) 2017 [38], TrackingNet [39], GOT-10k (Generic Object Tracking) [40], and LaSOT (Large-scale Single Object Tracking) [41]. Regarding the appearance branch, image sampling is conducted from ImageNet1k and COCO2017, with ImageNet1k being sampled four times more frequently than COCO2017. For the motion branch, frame pairs are sampled uniformly from TrackingNet, LaSOT, and GOT-10k. Common data augmentations, including scaling, translation, cropping, and jittering, are employed to enrich the dataset. All input images and frames undergo resizing to 256 × 256 resolution. The patch size is set to 16 × 16. The masked ratios for the appearance and motion encoders are set at 75% and 60%, respectively. The optimizer utilized is Adaptive Moment with Weight decay (AdamW) [42], incorporating a weight decay of 5e-2. The base learning rate is established at 7.5e-5, with a cosine decay schedule employed for learning rate decay. Training is executed on Intel Xeon CPU E5-2690 v4 @ 2.60 GHz with 512 GB RAM and 8 NVidia RTX 3090 GPUs, spanning 500 epochs with a batch size of 128, and each epoch comprises 200,000 samples. The initial 25 epochs integrate a warmup strategy [43].

Tracking Models. Based on our TrackMAE pretrained ViT models, we develop five variants of TrackMAE trackers with diverse encoders and input resolutions, as outlined in Table 2. We adopt ViT-L as our backbone models for TrackMAE-L384 and L256, ViT-B for TrarckMAE-B384 and B256, and ViT-XS for TrackMAE-XS. For the tracking head, we utilize the corner-based head implementation from Spatio-Temporal trAnsfoRmer Tracking (STARK) [26]. Furthermore, we provide details on model parameters, Floating Point Operations Per Second (FLOPs), inference speed, and latency in Table 2. The fine-tuning and testing strategy remains consistent with that of previous popular trackers [10,24,26]. The speed and latency are measured on an Intel Core i9-9900K CPU @ 3.60 GHz with 64 GB RAM and a single NVidia TITAN RTX GPU.

Table 2: Details of TrackMAE tracking models

Model	Encoder	Input resolution	Params (M)	FLOPs (G)	Speed (*fps*)	Latency (*ms*)
TrackMAE-L384	ViT-L	384 × 384	305	351	10	100
TrackMAE-L256	ViT-L	256 × 256	305	156	23	43
TrackMAE-B384	ViT-B	384 × 384	88	100	26	38
TrackMAE-B256	ViT-B	256 × 256	88	45	63	16
TrackMAE-XS	ViT-XS	256 × 256	31	15	202	5

Compared Methods. In our experiments, many popular tracking method are chosen for comparison, including Sequence to sequence Tracking (SeqTrack) [10], One-Stream Tracking (OSTrack) [11], Siamese Re-detection Convolutional Neural Network (SiamR-CNN) [18], Siamese Region Proposal Network Plus Plus (SiamRPN++) [19], Siamese Box Adaptive Network (SiamBAN) [21],

Transformer Tracking (TransT) [24], Transformer Meets Tracker (TMT) [25], STARK [26], Shifted Window Tracking (SwinTrack) [27], Mixed Transformer (Mixformer) [28], Attention in Attention Tracking (AiATrack) [44], Transforming Model Prediction (TMP) [45], Cyclic Shifting Window Transformer Tracking (CSWinTT) [46], Keep Tracking (KeepTrack) [47], Siamese Attention (SiamAttn) [48], Object-aware anchor-free networks (Ocean) [49], Discriminative Model Prediction (DiMP) [50], Multi-Domain Networks (MDNet) [51], Hierarchical Tracking (HiT) [52], Fast Efficient Accurate Robust tracking (FEAR) [53], Hierarchical Cross-Attention Tracking (HCAT) [54], Exemplar Transformers Tracking (E.T.Track) [55], Lightweight Tracking (LightTrack) [56].

4.2 State-of-the-Art Comparison

LaSOT. The state-of-the-art (SOTA) comparison results on LaSOT [41] are presented in Table 3. With the same encoder and input resolution, our TrackMAE-B256 surpasses previous SeqTrack-B256 and OSTrack-256 by 1.1 and 1.9 in Area Under Curve (AUC) score, respectively. It is noteworthy that our TrackMAE pretraining method significantly reduces training time, saving up to 5 times the training duration compared to their MAE pretraining method. Upon scaling up the model, TrackMAE-L384 achieves a new SOTA performance with a 72.5 AUC score. Fig. 4 presents the results of attribute-based evaluations on LaSOT, indicating the performance of the tracker under different challenging scenarios. The results demonstrate that our approach outperforms previous methods across various challenges, showcasing the comprehensive improvement brought by the pretraining of the base model. Particularly, our approach exhibits the most significant performance advantage under the challenges of fast motion and background clutter, highlighting its strong tracking stability and discriminability. Furthermore, as demonstrated in Table 4, TrackMAE-XS establishes a SOTA performance compared to previous efficient trackers, achieving a 66.0 AUC score and running 202 frames per second (*fps*).

Table 3: Comparison with state-of-the-art trackers on the LaSOT benchmark. The best results are shown in underline font. P: precision, norm: normalization

	AUC	P_{norm}	P
TrackMAE-L384	72.5	81.3	78.6
TrackMAE-L256	72.4	81.6	78.5
TrackMAE-B384	70.5	79.2	75.8
TrackMAE-B256	71.0	80.5	77.1
SeqTrack-B256 [10]	69.9	79.7	76.3
OSTrack-256 [11]	69.1	78.7	75.2
Mixformer-L [28]	70.1	79.9	76.3
Mixformer-22k [28]	69.2	78.7	74.7
SwinTrack [27]	71.3	–	76.5
AiATrack [44]	69.0	79.4	73.8
TMP [45]	68.5	79.2	73.5
CSWinTT [46]	66.2	75.2	70.9
KeepTrack [47]	67.1	77.2	70.2
STARK [26]	67.1	77.0	–
TransT [24]	64.9	73.8	69.0
TMT [25]	63.9	–	61.4
SiamAttn [48]	56.0	64.8	–

(Continued)

Table 3 (continued)

	AUC	P_{norm}	P
SiamBAN [21]	51.4	59.8	–
Ocean [49]	56.0	65.1	56.6
SiamR-CNN [18]	64.8	72.2	–
DiMP [50]	56.9	65.0	56.7
SiamRPN++ [19]	49.6	56.9	49.1
MDNet [51]	39.7	46.0	37.3

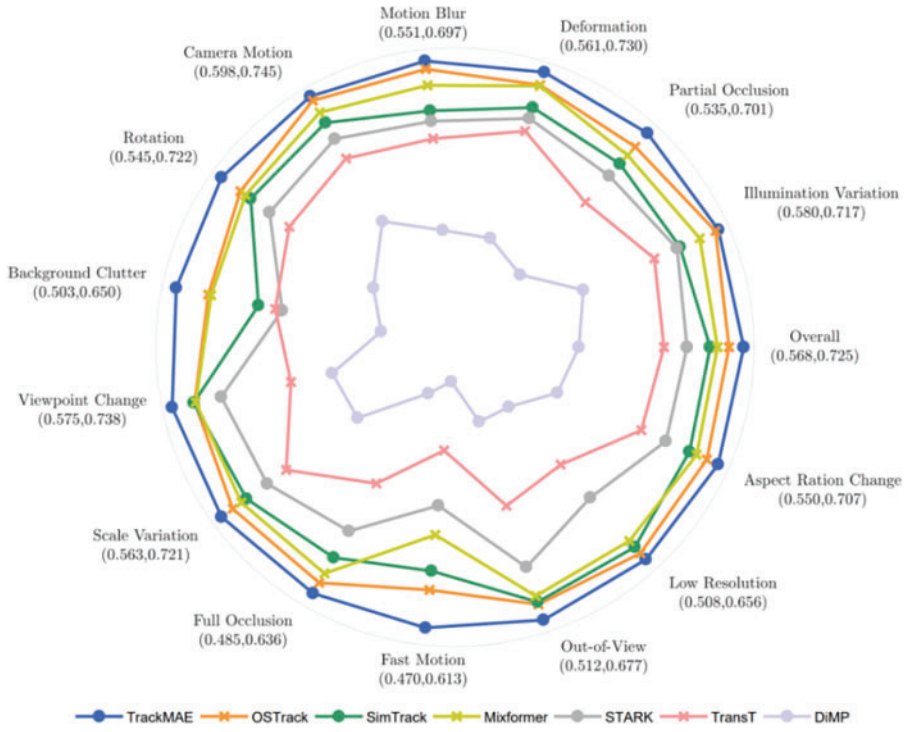

Figure 4: AUC scores of different attributes on the LaSOT benchmark

Table 4: Comparison with state-of-the-art efficient trackers on the LaSOT benchmark. The best results are shown in underline font. P: precision, norm: normalization

	AUC	P_{norm}	P	Speed (*fps*)
TrackMAE-XS	66.0	74.1	69.1	202
HiT-Small [52]	60.5	68.3	61.5	192
FEAR [53]	53.5	–	54.5	105
HCAT [54]	59.3	68.7	61.0	195

(Continued)

Table 4 (continued)

	AUC	P_{norm}	P	Speed (*fps*)
E.T.Track [55]	59.1	–	–	40
LightTrack [56]	53.8	–	53.7	128

TrackingNet. The SOTA comparison results on TrackingNet [39] are detailed in Table 5. Our TrackMAE-L384 achieves the highest performance with an AUC score of 85.4. Furthermore, TrackMAE-B256 exhibits superior performance compared to the aligned OSTrack-256 and SeqTrack-B256. As outlined in Table 6, TrackMAE-XS surpasses previous efficient trackers.

Table 5: Comparison with state-of-the-art trackers on the TrackingNet benchmark. The best results are shown in underline font. P: precision, norm: normalization

	AUC	P_{norm}	P
TrackMAE-L384	85.4	89.3	85.5
TrackMAE-L256	84.8	88.8	84.2
TrackMAE-B384	84.9	89.0	84.4
TrackMAE-B256	83.8	88.2	82.6
SeqTrack-B256 [10]	83.3	88.3	82.2
OSTrack-256 [11]	83.1	87.8	82.0
Mixformer-L [28]	83.9	88.9	83.1
Mixformer-22k [28]	83.1	88.1	81.6
SwinTrack [27]	84.0	–	82.8
AiATrack [44]	82.7	87.8	80.4
TMP [45]	81.5	86.4	78.9
CSWinTT [46]	81.9	86.7	79.5
STARK [26]	82.0	86.9	–
TransT [24]	81.4	86.7	80.3
TMT [25]	78.4	83.3	73.1
SiamAttn [48]	75.2	81.7	–
SiamR-CNN [18]	81.2	85.4	80.0
DiMP [50]	74.0	80.1	68.7
SiamRPN++ [19]	73.3	80.0	69.4
MDNet [51]	60.6	70.5	56.5

GOT-10k. The SOTA comparison results on GOT-10k [40] are elaborated in Tables 7 and 8. Our TrackMAE model family exhibits competitive performance compared to previous trackers. Specifically, TrackMAE-L384 attains the top result with a score of 77.0 in average overlap (AO) score, while TrackMAE-XS achieves the top efficient performance with a score of 66.9 in AO score.

Table 6: Comparison with state-of-the-art efficient trackers on the TrackingNet benchmark. The best results are shown in underline font. P: precision, norm: normalization

	AUC	P_{norm}	P	Speed (*fps*)
TrackMAE-XS	79.8	84.5	76.5	202
HiT-Small [52]	77.7	81.9	73.1	192
HCAT [54]	76.6	82.6	72.9	195
E.T.Track [55]	75.0	80.3	70.6	40
LightTrack [56]	72.5	77.8	69.5	128

Table 7: Comparison with state-of-the-art trackers on the GOT-10k benchmark. The best results are shown in underline font.

	AO	$SR_{0.5}$	$SR_{0.75}$
TrackMAE-L384	77.0	85.4	76.5
TrackMAE-L256	76.4	85.2	74.2
TrackMAE-B384	74.9	83.8	72.2
TrackMAE-B256	74.0	83.5	71.9
SeqTrack-B256 [10]	73.7	–	–
OSTrack-256 [11]	73.6	82.8	71.4
SimTrack [12]	69.8	78.8	66.0
Mixformer-22k [28]	70.7	80.0	67.8
SwinTrack [27]	72.4	–	67.8
AiATrack [44]	69.6	80.0	63.2
SBT [29]	70.4	80.8	64.7
CSWinTT [46]	69.4	78.9	65.4
STARK [26]	68.8	78.1	64.1
TransT [24]	72.3	82.4	68.2
TMT [25]	68.8	80.5	59.7
Ocean [49]	61.1	72.1	47.3
SiamR-CNN [18]	64.9	72.8	59.7
DiMP [50]	61.1	71.7	49.2
SiamRPN++ [19]	51.7	61.6	32.5
MDNet [51]	29.9	30.3	9.9

4.3 Ablation Study

We use TrackMAE-B256 as the baseline model in our ablation study. The result of the baseline model is reported in Table 9 (#1).

Table 8: Comparison with state-of-the-art efficient trackers on the GOT-10k benchmark. The best results are shown in underline font.

	AO	$SR_{0.5}$	$SR_{0.75}$	Speed (*fps*)
TrackMAE-XS	66.9	76.2	60.0	202
HiT-Small [52]	62.6	71.2	54.4	192
FEAR [53]	61.9	72.2	–	105
HCAT [54]	65.1	76.5	56.7	195
LightTrack [56]	61.1	71.0	–	128

Table 9: Ablation study on LaSOT and GOT-10k. Δ denotes the performance change (averaged over benchmarks) compared with the baseline

#	Method	LaSOT	GOT-10k	Δ
1	**Baseline**	71.0	74.0	–
2	w/o motion branch	67.5	70.0	**−3.8**
3	w/o appearance branch	69.6	73.2	**−1.1**
4	TrackMAE → MAE	69.1	72.4	**−1.8**
5	TrackMAE → Scratch	62.3	61.2	**−10.8**
6	Decoder width: 512 → 256	69.5	72.4	**−1.6**
7	Decoder width: 512 → 768	70.4	73.5	**−0.6**
8	Decoder depth: 8 → 4	70.4	74.0	**−0.3**
9	Decoder depth: 8 → 12	71.3	74.4	**+0.4**
10	Appearance mask ratio: 75% → 65%	70.8	73.6	**−0.3**
11	Appearance mask ratio: 75% → 85%	70.1	73.3	**−0.8**
12	Motion mask ratio: 60% → 50%	70.9	74.0	**−0.1**
13	Motion mask ratio: 60% → 70%	70.3	73.7	**−0.5**

Component Analysis. We separately remove the motion branch and the appearance branch, then train the remaining branch using the same datasets. The results in Table 9 (#2 and #3) demonstrate a decrease in performance for both cases, indicating that learning to understand image appearance and inter-frame motion correlation is indispensable. Furthermore, it highlights that the impact of inter-frame motion correlations is more significant.

Pre-training Method. We conduct a comparative analysis between our TrackMAE pretraining method, the MAE pretraining method, and training from scratch. For the MAE pretraining, we utilize aligned datasets with those used in TrackMAE, following the original training schedule length as specified in its original paper [10], which incurs a training time five times longer than ours. As depicted in Table 9 (#4), although the MAE method requires a longer training duration, it yields inferior performance compared to our approach. In contrast, for the training from scratch method, we train the tracking model on tracking datasets with the same training time as TrackMAE. As illustrated in Table 9 (#5), this approach obtains significantly lower performance compared to our method, underscoring the efficacy of our approach.

Decoder Architecture. We report the influence of the decoder settings (depth and width) in Table 9 (#6–#9). After a thorough evaluation, considering factors such as performance and computational efficiency, we determined the optimal configuration with a decoder depth of 8 and width of 512.

Masking Ratio. Table 9 (#10–13) delineates the impact of the masking ratio. Notably, the optimal masking ratio for the appearance branch is relatively high (75%), while for the motion branch, it is modest (60%). This discrepancy can be attributed to the inherent reconstruction processes for images and frames. A higher masking ratio necessitates the model to develop a deeper understanding of appearance for image reconstruction. Conversely, a modest masking ratio prompts the model to organize the remaining patches to reconstruct frames, facilitating the learning of motion relationships.

4.4 Visualization Results

In this section, to enhance our understanding of the model's capabilities and limitations, we visualize several representative cases, encompassing both successful and unsuccessful instances.

Successful Cases. Fig. 5 illustrates representative successful cases. Our model demonstrates good tracking accuracy and robustness under various conditions such as changes in target appearance and viewpoint (#1), motion blur (#2), severe occlusion (#3), low illumination (#4), and adverse weather conditions (#5).

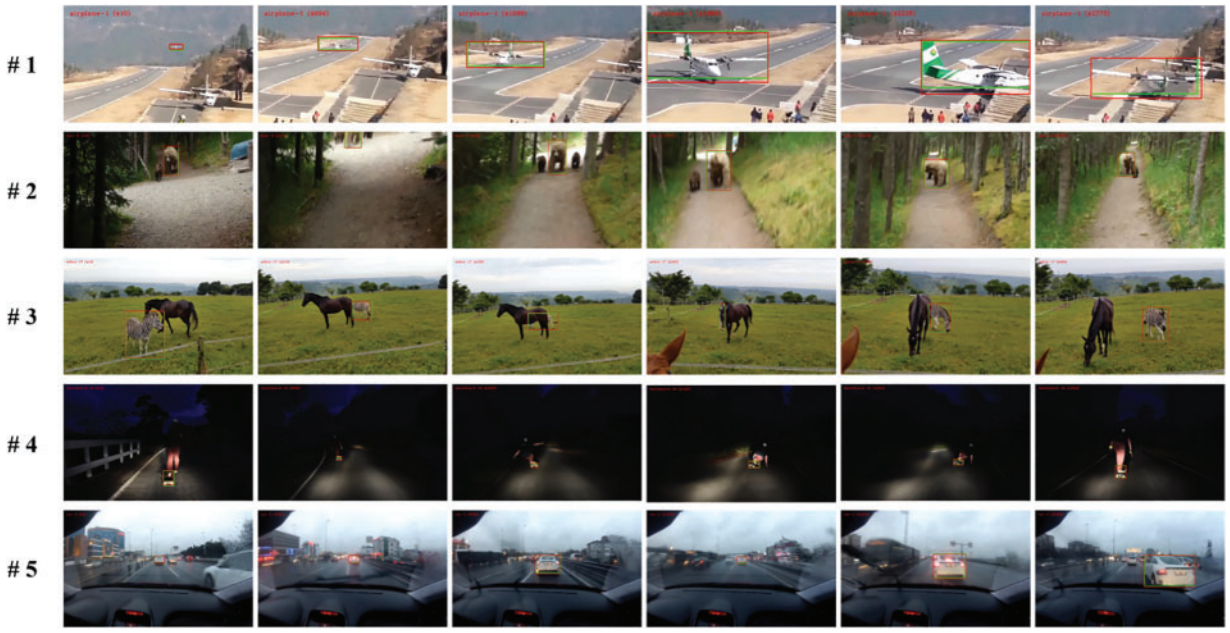

Figure 5: Visualization of representative successful cases

Failure Cases. Fig. 6 displays representative failure cases. The model is prone to tracking drift in the presence of interfering similar objects (#1), particularly when the correct target is occluded (#2). Additionally, when the background is cluttered (#3), there is a decrease in the precision of bounding box predictions.

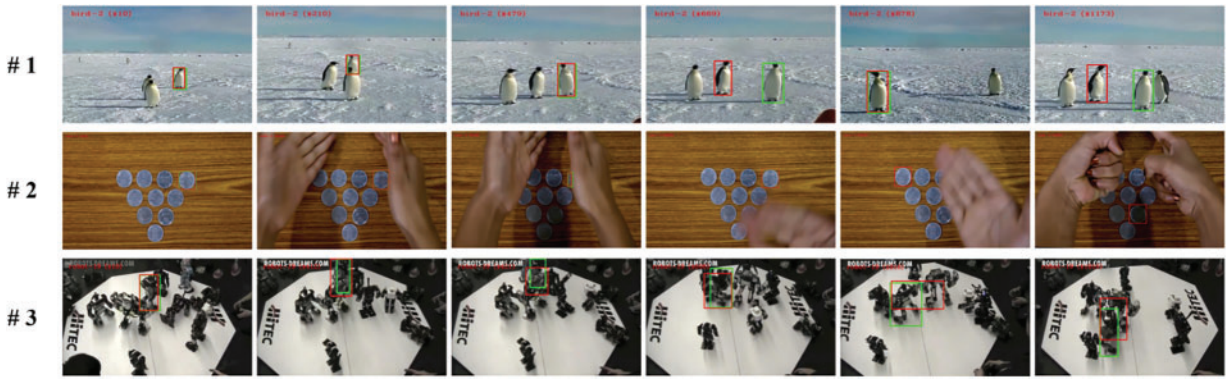

Figure 6: Visualization of representative failure cases

5 Conclusion

This work proposes a new ViT pretraining method for visual object tracking, i.e., TrackMAE. It employs a two-branch masked autoencoder method to train the ViT to understand both the appearance and the motion information, resulting in less training cost and higher tracking performance. Extensive experiments demonstrate that TrackMAE is effective, achieving competitive performance in the tracking task. We hope this work could provide a flexible way to train a tracking model, facilitating the model design and training of tracking.

Acknowledgement: The author is very grateful to everyone who contributed to providing relevant information for this article.

Funding Statement: The work is supported in part by National Natural Science Foundation of China (No. 62176041), and in part by Excellent Science and Technique Talent Foundation of Dalian (No. 2022RY21).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Chunjuan Bo, Xin Chen, Junxing Zhang; data collection: Chunjuan Bo; analysis and interpretation of results: Chunjuan Bo, Xin Chen; draft manuscript preparation: Chunjuan Bo, Xin Chen. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors. The data that support the findings of this study are available from the corresponding author, Chunjuan Bo, upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 76, no. 1–3, pp. 323–338, Apr. 2018. doi: 10.1016/j.patcog.2017.11.007.

[2] S. Liu, D. Liu, G. Srivastava, D. Połap, and M. Woźniak, "Overview of correlation filter based algorithms in object tracking," *Complex Intell. Syst.*, vol. 7, no. 4, pp. 1895–1917, Apr. 2021. doi: 10.1007/s40747-020-00161-4.

[3] Y. Liang, Y. Han, and L. Li, "Survey on deep-learning-based long-term object tracking algorithms," *Comput. Eng. Appl.*, vol. 59, no. 4, pp. 1–17, Feb. 2024. doi: 10.3778/j.issn.1002-8331.2206-0507.
[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," presented at the 2016 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 770–778.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," presented at the 2012 Adv. Neural Inf. Proces. Syst., Lake Tahoe, NV, USA, Dec. 3–6, 2012.
[6] C. Szegedy *et al.*, "Going deeper with convolutions," presented at the 2015 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Boston, MA, USA, Jun. 7–12, 2015, pp. 1–9.
[7] A. Vaswani *et al.*, "Attention is all you need," presented at the 2017 Adv. Neural Inf. Proces. Syst., Long Beach, CA, USA, Dec. 4–9, 2017, pp. 5998–6008.
[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," presented at the 2020 Eur. Conf. Comput. Vis., Glasgow, UK, Aug. 23–28, 2020, pp. 213–229.
[9] A. Dosovitskiy *et al.*, "An image is worth 16 × 16 words: Transformers for image recognition at scale," presented at the 2021 Int. Conf. Learn. Represent., May 3–7, 2021.
[10] X. Chen, H. Peng, D. Wang, H. Lu, and H. Hu, "Seqtrack: Sequence to sequence learning for visual object tracking," presented at the 2023 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Vancouver, BC, Canada, Jun. 17–24, 2023, pp. 14572–14581.
[11] B. Ye, H. Chang, B. Ma, S. Shan, and X. Chen, "Joint feature learning and relation modeling for tracking: A one-stream framework," presented at the 2022 Eur. Conf. Comput. Vis., Tel Aviv, Israel, Oct. 23–28, 2022, pp. 341–357.
[12] B. Chen *et al.*, "Backbone is all your need: A simplified architecture for visual object tracking," presented at the 2022 Eur. Conf. Comput. Vis., Tel Aviv, Israel, Oct. 23–28, 2022, pp. 375–392.
[13] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," presented at the 2022 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., New Orleans, LA, USA, Jun. 19–24, 2022, pp. 16000–16009.
[14] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," presented at the 2021 Inter. Conf. Mach. Learn., Jul. 18–24, 2021, pp. 10347–10357.
[15] H. Touvron, M. Cord, and H. Jégou, "DeiT III: Revenge of the ViT," presented at the 2022 Eur. Conf. Comput. Vis., Tel Aviv, Israel, Oct. 23–28, 2022, pp. 516–533.
[16] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," presented at the 2021 Inter. Conf. Mach. Learn., Jul. 18–24, 2021, pp. 8748–8763.
[17] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," presented at the 2019 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Long Beach, CA, USA, Jun. 16–20, 2019, pp. 4586–4595.
[18] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," presented at the 2020 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., USA, Jun. 14–19, 2020, pp. 6577–6587.
[19] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," presented at the 2019 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Long Beach, CA, USA, Jun. 16–20, 2019, pp. 4277–4286.
[20] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," presented at the 2020 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Jun. 14–19, 2020, pp. 6268–6276.
[21] Z. Chen *et al.*, "SiamBAN: Target-aware tracking with Siamese box adaptive network," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5158–5173, Apr. 2023. doi: 10.1109/TPAMI.2022.3195759.

[22] D. Zhang and Z. Zheng, "Joint representation learning with deep quadruplet network for real-time visual tracking," presented at the 2020 Int. Jt. Conf. Neural Netw., Virtual, Glasgow, UK, Jul. 19–24, 2020, pp. 1–8.
[23] D. Zhang, Z. Zheng, M. Li, and R. Liu, "CSART: Channel and spatial attention-guided residual learning for real-time object tracking," *Neurocomputing*, vol. 436, no. 4, pp. 260–272, May 2021. doi: 10.1016/j.neucom.2020.11.046.
[24] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang and H. Lu, "Transformer tracking," presented at the 2021 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., USA, Jun. 19–25, 2021, pp. 8126–8135.
[25] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," presented at the 2021 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., USA, Jun. 19–25, 2021, pp. 1571–1580.
[26] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," presented at the 2021 IEEE Int. Conf. Comput. Vis., Montreal, QC, Canada, Qct. 11–17, 2021, pp. 10428–10437.
[27] L. Lin, H. Fan, Z. Zhang, Y. Xu, and H. Ling, "Swintrack: A simple and strong baseline for transformer tracking," presented at the 2022 Adv. Neural Inf. Proces. Syst., New Orleans, LA, USA, Nov. 28–Dec. 9, 2022, pp. 16743–16754.
[28] Y. Cui, C. Jiang, L. Wang, and G. Wu, "Mixformer: End-to-end tracking with iterative mixed attention," presented at the 2022 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., New Orleans, LA, USA, Jun. 19–24, 2022, pp. 13598–13608.
[29] F. Xie, C. Wang, G. Wang, Y. Cao, W. Yang and W. Zeng, "Correlation-aware deep tracking," presented at the 2022 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., New Orleans, LA, USA, Jun. 19–24, 2022, pp. 8741–8750.
[30] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," presented at the 2016 Eur. Conf. Comput. Vis., Amsterdam, Netherlands, Oct. 8–16, 2016, pp. 649–666.
[31] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMS," presented at the 2015 Inter. Conf. Mach. Learn., Lile, France, Jul. 6–11, 2015, pp. 843–852.
[32] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," presented at the 2018 Int. Conf. Learn. Represent., Vancouver, BC, Canada, Apr. 30–May 3, 2018.
[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," presented at the 2020 Inter. Conf. Mach. Learn., Jul. 13–18, 2020, pp. 1575–1585.
[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," presented at the 2020 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., USA, Jun. 14–19, 2020, pp. 9726–9735.
[35] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," presented at the 2018 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Salt Lake City, UT, USA, Jun. 18–22, 2018, pp. 3733–3742.
[36] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," presented at the 2019 Conf. N. Am. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol., Minneapolis, MN, USA, Jun. 2–7, 2019, pp. 4171–4186.
[37] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," presented at the 2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Miami, FL, USA, Jun. 18–22, 2009, pp. 248–255.
[38] T. -Y. Lin *et al.*, "Microsoft COCO: Common objects in context," presented at the 2014 Eur. Conf. Comput. Vis., Zurich, Switzerland, Sep. 6–12, 2014, pp. 740–755.
[39] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild," presented at the 2018 Eur. Conf. Comput. Vis., Munich, Germany, Sep. 6–12, 2018, pp. 310–327.

[40] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021. doi: 10.1109/TPAMI.2019.2957464.

[41] H. Fan *et al.*, "Lasot: A high-quality benchmark for large-scale single object tracking," presented at the 2019 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Long Beach, CA, USA, Jun. 16–20, 2019, pp. 5374–5383.

[42] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," presented at the 2019 Int. Conf. Learn. Represent., New Orleans, LA, USA, May 6–9, 2019.

[43] P. Goyal *et al.*, "Accurate, large minibatch sgd: Training imagenet in 1 hour," arXiv preprint arXiv:1706.02677, 2017.

[44] S. Gao, C. Zhou, C. Ma, X. Wang, and J. Yuan, "AiATrack: Attention in attention for transformer visual tracking," in S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), *Lecture Notes in Computer Science*, Tel Aviv, Israel, Oct. 23–27, 2022, vol. 13682, pp. 146–164.

[45] C. Mayer *et al.*, "Transforming model prediction for tracking," presented at the 2022 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., New Orleans, LA, USA, Jun. 19–24, 2022, pp. 8721–8730.

[46] Z. Song, J. Yu, Y. P. P. Chen, and W. Yang, "Transformer tracking with cyclic shifting window attention," presented at the 2022 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., New Orleans, LA, USA, Jun. 19–24, 2022, pp. 8791–8800.

[47] C. Mayer, M. Danelljan, D. P. Paudel, and L. Van Gool, "Learning target candidate association to keep track of what not to track," presented at the 2021 IEEE Int. Conf. Comput. Vis., Montreal, QC, Canada, Jun. 19–24, 2021, pp. 13444–13454.

[48] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable siamese attention networks for visual object tracking," presented at the 2020 IEEE Int. Conf. Comput. Vis., USA, Jun. 19–24, 2020, pp. 6728–6737.

[49] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," presented at the 2020 Eur. Conf. Comput. Vis., Glasgow, UK, Jun. 14–19, 2020, pp. 6728–6737.

[50] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," presented at the 2020 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., USA, Jun. 14–19, 2020, pp. 7181–7190.

[51] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," presented at the 2016 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., Las Vegas, NV, USA, Jun. 26–Jul. 1, 2016, pp. 4293–4302.

[52] B. Kang, X. Chen, D. Wang, H. Peng, and H. Lu, "Exploring lightweight hierarchical vision transformers for efficient visual tracking," presented at the 2023 IEEE Int Conf. Comput. Vis., Paris, France, Oct. 2–6, 2023, pp. 9578–9587.

[53] V. Borsuk, R. Vei, O. Kupyn, T. Martyniuk, I. Krashenyi, and J. Matas, "FEAR: Fast, efficient, accurate and robust visual tracker," presented at the 2022 Eur. Conf. Comput. Vis., Tel Aviv, Israel, Oct. 23–27, 2022, pp. 644–663.

[54] X. Chen, B. Kang, D. Wang, D. Li, and H. Lu, "Efficient visual tracking via hierarchical cross-attention transformer," presented at the 2022 Eur. Conf. Comput. Vis., WSP, Tel Aviv, Israel, Oct. 23–27, 2022, pp. 461–477.

[55] P. Blatter, M. Kanakis, M. Danelljan, and L. Van Gool, "Efficient visual tracking with exemplar transformers," presented at the 2023 IEEE Winter Conf. Appl. Comput. Vis., Waikoloa, HI, USA, Oct. 23–27, 2023, pp. 1571–1581.

[56] B. Yan, H. Peng, K. Wu, D. Wang, J. Fu and H. Lu, "Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search," presented at the 2021 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., USA, Jun. 19–25, 2021, pp. 15175–15184.