DropMAE: Masked Autoencoders with Spatial-Attention Dropout for Tracking Tasks

Qiangqiang Wu[1] Tianyu Yang[2*] Ziquan Liu[1] Baoyuan Wu[4] Ying Shan[3] Antoni B. Chan[1]
[1]Department of Computer Science, City University of Hong Kong
[2]International Digital Economy Academy [3]Tencent AI Lab
[4]School of Data Science, The Chinese University of Hong Kong, Shenzhen
{qiangqwu2-c, ziquanliu2-c}@my.cityu.edu.hk, tianyu-yang@outlook.com
wubaoyuan@cuhk.edu.cn, yingsshan@tencent.com, abchan@cityu.edu.hk

Abstract

In this paper, we study masked autoencoder (MAE) pre-training on videos for matching-based downstream tasks, including visual object tracking (VOT) and video object segmentation (VOS). A simple extension of MAE is to randomly mask out frame patches in videos and reconstruct the frame pixels. However, we find that this simple baseline heavily relies on spatial cues while ignoring temporal relations for frame reconstruction, thus leading to sub-optimal temporal matching representations for VOT and VOS. To alleviate this problem, we propose DropMAE, which adaptively performs spatial-attention dropout in the frame reconstruction to facilitate temporal correspondence learning in videos. We show that our DropMAE is a strong and efficient temporal matching learner, which achieves better fine-tuning results on matching-based tasks than the ImageNet-based MAE with 2× faster pre-training speed. Moreover, we also find that motion diversity in pre-training videos is more important than scene diversity for improving the performance on VOT and VOS. Our pre-trained DropMAE model can be directly loaded in existing ViT-based trackers for fine-tuning without further modifications. Notably, DropMAE sets new state-of-the-art performance on 8 out of 9 highly competitive video tracking and segmentation datasets. Our code and pre-trained models are available at https://github.com/jimmy-dq/DropMAE.git.

1. Introduction

Recently, transformers have achieved enormous success in many research areas, such as natural language processing (NLP) [6, 22], computer vision [97] and audio generation [43, 73]. In NLP, masked autoencoding is commonly used to train large-scale generalizable NLP transformers containing billions of parameters. Inspired by the great success of self-supervised learning in NLP, recent advances [36, 89] in computer vision suggest that training large-scale vision transformers may undertake a similar trajectory with NLP. The seminal work MAE [36] reconstructs the input image from a small portion of patches. The learned representation in this masked autoencoder has been demonstrated to be effective in many computer vision tasks, such as image classification, object detection and semantic segmentation.

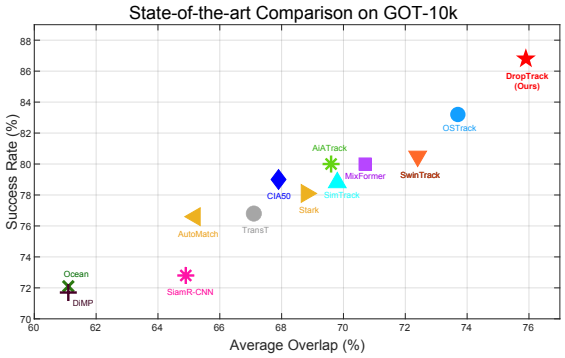

Figure 1. State-of-the-art comparison on GOT-10k [39] following the one-shot protocol. Our DropTrack with the proposed DropMAE pre-training achieves state-of-the-art performance without using complicated pipelines such as online updating.

In video object tracking (VOT), recently two works, SimTrack [10] and OSTrack [97], explore using an MAE pre-trained ViT model as the tracking backbone. Notably, these two trackers achieve state-of-the-art performance on existing tracking benchmarks without using complicated tracking pipelines. The key to their success is the robust pre-training weights learned by MAE on ImageNet [68]. In addition, [10, 97] also demonstrate that, for VOT, MAE *unsupervised* pre-training on ImageNet is more effective than *supervised* pre-training using class labels – this is mainly because MAE pre-training learns more fine-grained local

*Corresponding Author

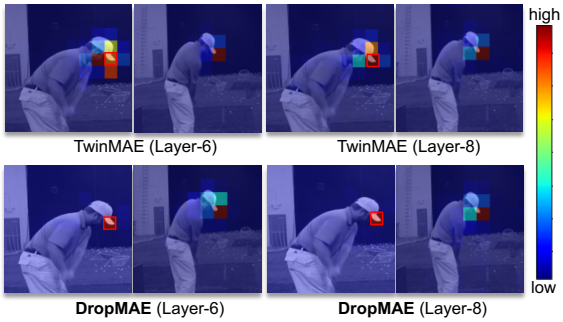

Figure 2. Visualization of the attention maps of the TwinMAE baseline and our DropMAE in the reconstruction of a random masked patch, which is denoted as a red bounding box in the left input frame. TwinMAE leverages the spatial cues (within the same frame) more than temporal cues (between frames) for reconstruction. Our proposed DropMAE improves the baseline by effectively alleviating co-adaptation between spatial cues in the reconstruction, focusing more on temporal cues, thus achieving better learning of temporal correspondences for VOT and VOS.

structures that are useful for accurate target localization required for VOT, whereas supervised training learns high-level class-related features that are invariant over appearance changes. Despite the promising performance achieved by [10,97], the MAE pre-training on ImageNet could still be sub-optimal for the tracking task due to the natural gap between images and videos, i.e., no prior temporal correspondence information can be learned in static images. However, previous tracking methods [3, 42, 83] have shown that temporal correspondence learning is the key in developing a robust and discriminative tracker. Thus there is an opportunity to further develop the MAE framework specifically for matching-base video tasks, such as VOT and VOS.

One simple way to extend MAE to videos is to randomly mask out frame patches in a video clip (i.e., video frame pairs) and then reconstruct the video clip. We denote this simple baseline as *twin MAE* (TwinMAE). Given a masked patch query, as illustrated in Figs. 2 & 4, we find that TwinMAE heavily relies on the spatially neighbouring patches *within the same frame* to reconstruct the masked patch, which implies a heavy co-adaptation of spatial cues (within-frame tokens) for reconstruction and may cause learning of sub-optimal temporal representations for matching-based downstream tasks like video object tracking and segmentation.

To address this issue with the TwinMAE baseline, we propose DropMAE specifically designed for pre-training a masked autoencoder for matching-based video downstream tasks (e.g., VOT and VOS). Our DropMAE adaptively performs spatial-attention dropout to break up co-adaptation between spatial cues (within-frame tokens) during the frame reconstruction, thus encouraging more temporal interactions and facilitating temporal correspondence learning in the pre-training stage. Interestingly, we obtain several important findings with DropMAE: 1) DropMAE is an effective and efficient temporal correspondence learner, which achieves better fine-tuning results on matching-based tasks than the ImageNet-based MAE with $2\times$ faster pre-training speed. 2) Motion diversity in pre-training videos is more important than scene diversity for improving the performance on VOT and VOS.

We conduct downstream task evaluation on 9 competitive VOT and VOS benchmarks, achieving state-of-the-art performance on these benchmarks. In particular, our trackers with DropMAE pre-training obtain 75.9% AO on GOT-10k, 52.7% AUC on LaSOT$_{ext}$, 56.9% AUC on TNL2K and 92.1%/83.0% $\mathcal{J}\&\mathcal{F}$ scores on DAVIS-16/17, w/o using complicated online updating or memory mechanisms.

In summary, the main contributions of our work are:

- To the best of our knowledge, we are the first to investigate masked autoencoder video pre-training for temporal matching-based downstream tasks. Specifically, we explore various video data sources for pre-training and build a TwinMAE baseline to study its effectiveness on temporal matching tasks. Since none exists, we further build a ViT-based VOS baseline for fine-tuning.
- We propose DropMAE, which adaptively performs spatial-attention dropout in the frame reconstruction to facilitate effective temporal correspondence learning in videos.
- Our trackers with DropMAE pre-training sets new state-of-the-art performance on 8 out of 9 highly competitive video tracking and segmentation benchmarks without complicated tracking pipelines.

2. Related Work

Visual object tracking and segmentation. Given an annotated bounding box in the first frame of a test video, visual object tracking (VOT) aims to accurately predict the target's bounding boxes in the following frames. Similarly, for visual object segmentation (VOS), given an annotated binary mask in the first frame, VOS aims to predict dense target masks in the remaining frames. In the early development of VOT, correlation filter-based approaches [19, 21, 31, 37, 46, 47, 51, 53, 82, 84] are dominant trackers due to their favorable ability in modeling target appearance variation. With the development of deep learning, deep Siamese networks [70, 78] are introduced to VOT. The representative work SiamFC [3] takes template and search images as input for target localization. Based on SiamFC, many improvements have been made, e.g., scale regression [41,42,94], online template updating [34,93,99], multi-level feature fusion [29], loss design [23] and backbone design [10,41,97,103]. For VOS, matching-based approaches, e.g., STM [60], AOT [96] and STCN [15], achieve promising results on existing VOS benchmarks. Recent improve-

ments [14, 50] on online memory design further improve the previous SOTA results in VOS. Recent studies including SimTrack [10] and OSTrack [97] show that the ViT backbone [24] with MAE pre-training on ImageNet is effective for object tracking.

Despite the great success of ViT with MAE pre-training on tracking, this static ImageNet-based pre-training still lacks temporal correspondence learning. Moreover, the developments of VOT and VOS show that learning strong temporal matching ability is essential for video tracking tasks. To the best of our best knowledge, we are the first to investigate masked autoencoder self-supervised video pre-training for tracking tasks. Our proposed DropMAE can learn reliable temporal correspondences, which we demonstrate are effective for downstream tracking tasks.

Self-supervised Learning. Self-supervised learning has received significant interest in the past few decades. There are many manually-designed pretext tasks for pre-training, such as image colorization [100], jigsaw puzzle solving [59], future frame prediction [69, 75] and rotation prediction [33]. Contrastive learning approaches [7, 11, 25, 35, 86, 88] are the mainstream self-supervised methods in recent years. However, these methods are sensitive to the type and strength of applied data augmentation, which makes them hard to train. Inspired by masked language modeling [6, 22], masked image modeling (MIM) approaches are proposed for learning unsupervised image [36, 89] or video representations [30, 71], which have been shown to be effective for many downstream tasks including image classification, object detection and video action recognition. However, there is no specifically designed pre-training approaches for temporal matching-based task, such as VOT and VOS. In this work, we propose DropMAE to explore this direction.

3. Method

We propose a self-supervised video pre-training method to learn robust representations for temporal matching-based downstream tasks, including VOT and VOS tasks. We firstly introduce a simple extension of MAE to temporal matching representation learning from video, (TwinMAE). We then illustrate the limitations of the TwinMAE baseline and propose a spatial-attention dropout strategy to facilitate temporal correspondence learning, denoted as DropMAE. The overall pipeline of both DropMAE and TwinMAE is shown in Fig. 3. Finally, we introduce the VOT and VOS methods used for fine-tuning the downstream tasks.

3.1. Temporal Masked Autoencoder (TwinMAE)

The masked autoencoder (MAE) model [36] consists of an encoder and a decoder. Its basic idea is to randomly mask out a large portion (e.g., 75%) of patches in an image and then reconstruct the image pixels. Specifically, the encoder only takes visible patches as input for feature learning, and then the decoder is input with both visible and masked patches to produce the image reconstruction. In order to adapt to downstream video matching-based tasks, one naive extension is to directly apply MAE on concatenated video frames, hoping to learn temporal matching representations from video frame pairs, which we denote as TwinMAE.

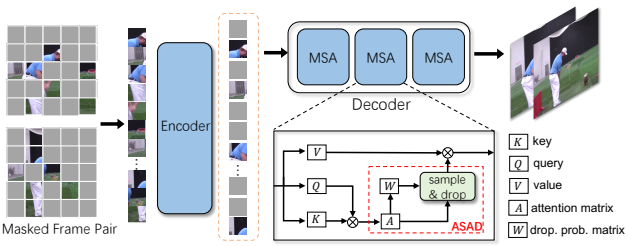

Figure 3. An illustration of our DropMAE. The proposed adaptive spatial-attention dropout (ASAD) facilitates temporal correspondence learning for temporal matching tasks. TwinMAE follows the same pipeline except that the ASAD module is not used.

It should be noted that existing works [30, 71] that extend MAE to video representation learning are mainly designed for the downstream task of video action recognition, where a long video clip (e.g., 16 frames) is used for reconstruction-based pre-training. To keep consistent with our downstream tasks of VOT/VOS, we follow the general training settings used in object tracking [3, 97], where two frames are sampled from one video as input to TwinMAE for pre-training. This adaptation significantly reduces the computational and memory cost compared to existing video pre-training approaches [30, 71], due to the quadratic complexity of ViTs.

Patch embedding. Firstly, we randomly sample 2 frames within a video with a predefined maximal frame gap. For each frame, we follow the vanilla ViT to divide it into non-overlapping patches. The patches extracted from the two frames are then concatenated together to form the overall patch sequence. We then randomly mask out patches in the patch sequence until a predefined mask ratio is reached. Note that we use the same mask ratio (i.e., 75%) with the original MAE, since the information redundancy of two frames should be similar to a single image. The visible patches are embedded by linear projection [24], and the masked patches are embedded using a shared learnable mask token. All the embedded patches are added with positional embeddings [24].

Frame identity embedding. To distinguish between the masked tokens in the same spatial location of the two frames, we use two learnable frame identity embeddings to indicate the two input frames. The corresponding frame identity embedding is added to each embedded patch.

Autoencoder and Training. Following the autoencoding pipeline in the original MAE [36], the encoder only takes visible embedded patches as input, and the decoder is input with all the embedded patches for masked patch reconstruction. We use the same normalized pixel loss from MAE for

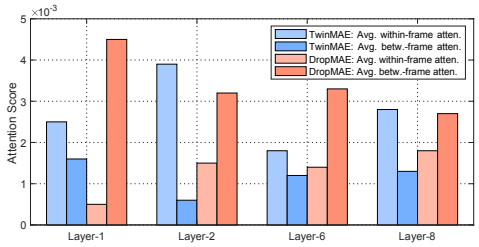

Figure 4. The average within-frame and between-frame attention scores obtained by TwinMAE and DropMAE in different decoder layers. The attention score is calculated on 20 randomly sampled K400 validation videos, and is averaged on all heads and locations.

training the whole network architecture.

3.2. Limitation of TwinMAE Baseline

The visualization of the reconstruction for our TwinMAE baseline is shown in Fig. 2. We also quantitatively compare the average within-frame and between-frame attentions during the reconstruction in Fig. 4. Interestingly, we find the TwinMAE reconstruction heavily relies on within-frame patches or spatial cues, which may lead to sub-optimal temporal representations for matching-based video tasks.

When only using within-frame spatial cues, the decoder will perform the reconstruction using only context information in the neighboring patches, and thus the learned encoder representations will embed context information. In contrast, when using between-frame cues, the decoder will learn to perform matching of patches *between* frames so as to recover the corresponding target patch in the other frame. Thus, decoding with between-frame cues will make the encoder learn representations that support temporal matching between frames. Previous works in object tracking [3, 83] also suggest that temporal correspondence learning plays a key role in developing a robust and discriminative tracker. Since TwinMAE relies more on context information, it is still suboptimal for downstream tracking tasks.

3.3. Adaptive Spatial-Attention Dropout

To address issue of TwinMAE discussed in Sec. 3.2, we propose a *Adaptive Spatial-Attention Dropout* (ASAD) to facilitate the temporal correspondence learning in the temporal MAE. Given a query token, our basic idea is to adaptively drop a portion of its within-frame cues in order to facilitate the model to learn more reliable temporal correspondence, i.e., between-frame cues. That is, we restrict the interactions between the query token and tokens in the same frame, and encourage more interactions with tokens in the other frame, through manipulation of the computed spatial-attention in the transformer. Therefore, to minimize the reconstruction loss, the model is facilitated to learn a better temporal matching ability, which is essential in matching-based video tasks.

Before introducing the proposed ASAD, we firstly revisit

(a) layer-4 (b) layer-6

Figure 5. Visualization of the temporal matching function f_{tem} on an example frame pair. A large value of $f_{tem}(i)$ indicates that the i-th pixel matches well to a pixel in the other frame.

the multi-head self-attention in ViT [24]. Let $\mathbf{z} \in \mathbb{R}^{N\times D}$ be the input sequence of the two concatenated input frames, N denotes the total patch number in the two frames and D is the feature dimension. The standard multi-head self-attention [24] can be formulated as:

$$[\mathbf{q},\mathbf{k},\mathbf{v}] = \mathbf{z}\mathbf{U}_{qkv}, \quad \text{SA}(\mathbf{z}) = \text{softmax}(\tfrac{1}{\sqrt{D_k}}\mathbf{q}\mathbf{k}^T), \quad (1)$$

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(\mathbf{z}); \text{SA}_2(\mathbf{z}); \cdots; \text{SA}_k(\mathbf{z})]\mathbf{U}_m, \quad (2)$$

where $\mathbf{U}_{qkv} \in \mathbb{R}^{D\times 3D_k}$ and $\mathbf{U}_m \in \mathbb{R}^{k\cdot D_k \times D}$. Let $A = \frac{\mathbf{q}\mathbf{k}^T}{\sqrt{D_k}} \in \mathbb{R}^{N\times N}$ denote the *attention matrix*. Our ASAD performs spatial-attention dropout on A so as to remove some within-frame interactions.

Temporal matching probability. We first need to consider the best tokens on which to apply ASAD. Intuitively, a query token that has a strong match in the other frame should be a good candidate, since, in the absence of within-frame cues, it can still be reconstructed well using the temporal cues in the other frame. Here, we define a temporal matching function $f_{tem}(\cdot)$ to measure the temporal matching probability of the i-th query token:

$$f_{tem}(i) = \max_{j\in\Omega_t(i)} (\hat{A}_{i,j}), \quad \hat{A} = \text{softmax}_{\text{row}}(A), \quad (3)$$

where the softmax function is applied on each row of A, $f_{tem}(i) \in [0,1]$, and $\Omega_t(i)$ denotes the *temporal* index set of the i-th query token, which contains all the token indices of the other frame. A larger value of $f_{tem}(i)$ indicates a larger probability that the i-th query token is well-matched in the other frame, and thus a good candidate for ASAD. A visualization of $f_{tem}(\cdot)$ is shown in Fig. 5.

Overall dropout probability measurement. The overall spatial-attention dropout probability at location (i,j) is measured by using both the temporal matching probability and the normalized spatial importance:

$$W_{i,j} = f_{tem}(i)\frac{\hat{A}_{i,j}}{\sum_{j\in\Omega_s(i)}\hat{A}_{i,j}}, \quad (4)$$

where $\Omega_s(i)$ is the *spatial* index set that contains all the other token indices (i.e., excluding the query index itself) in the same frame as the i-th query. When $W_{i,j}$ is large, the i-th query token has a good between-frame match, and meanwhile the j-th within-frame token is important for the i-th

query. In this case, dropping the attention element (i, j) in A facilitates the model to use between-frame (temporally-matched) tokens for token learning or reconstruction. Finally, we set the dropout probability for self-attention and temporal-self-attention to be 0, i.e., $W_{i,i} = W_{i,i+N/2} = 0$.

Note that there are $N(N/2-1)$ (i.e., excluding self-attention elements) spatial-attention elements in total, and only these spatial-attention elements are considered for dropout. With a pre-defined dropout ratio P, we globally drop a total of $N_d = PN(N/2-1)$ attention elements from A.

Sampling for Dropout. We draw N_d elements from a multinomial distribution based on the dropout probability matrix W. Then we drop the elements in A with the corresponding indices by setting their values to $-\infty$. After applying the softmax function in (1), the corresponding spatial-attention weights are removed. The other operations are the same with the original multi-head self attention mechanisms used in ViT. The PyTorch-like pseudocode is presented in the Supplemental.

Our ASAD method has negligible additional time cost compared with TwinMAE, due to the efficient matrix operation in GPUs. We apply ASAD to each layer in the decoder during the pre-training stage, so as to learn encoder representations that support temporal matching. In the next section, we introduce downstream task fine-tuning based on the well pre-trained ViT model.

3.4. Downstream Temporal Matching Tasks

After obtaining the pre-trained DropMAE model, we fine-tune the well-learned encoder (i.e., the ViT model) on downstream temporal matching tasks, i.e., VOT and VOS.

3.4.1 Video Object Tracking

Recently, the MAE ViT models pre-trained on ImageNet are applied to VOT and show impressive results. We use the state-of-the-art tracker OSTrack [97] as our baseline tracker for fine-tuning. In OSTrack, the cropped template and search images are firstly serialized into sequences and concatenated together. Then the overall sequence is added with the positional embeddings and input to the ViT backbone for joint feature extraction and interaction. Finally, the updated search features are input to a prediction head to predict the target bounding box.

During the fine-tuning stage, we use our pre-trained DropMAE encoder weights to initialize the ViT backbone used in OSTrack. Meanwhile, two frame identity embeddings are respectively added to template and search embeddings, in order to keep consistency with the pre-training stage. We use the same training losses of the original OSTrack. The detailed training parameters are shown in Supplementary for reference.

3.4.2 Video Object Segmentation

For VOS, there are currently no methods based on ViT. Thus, we build a simple VOS baseline with a ViT backbone.
Input serialization. Given a template frame with a binary mask, VOS aims to segment the object-of-interest in each frame of a video. Similar to the pre-training stage, the binary mask map, template and search frames are firstly converted to patch sequences, and then linearly projected and added with positional embeddings. Two frame identity embeddings are added to the template and search embeddings, and the mask embeddings are added to the template embeddings for mask encoding.
Joint feature extraction and interaction. The template and search embeddings are concatenated together and input to the ViT backbone for joint feature extraction and matching. We use the updated search features extracted from the last layer of ViT for mask prediction.
Mask prediction. The existing VOS approaches [14,15,60,96] employ multi-resolution features for mask prediction. However, the updated search features are single-resolution. We follow [45] to upsample the search features to $2\times$ and $4\times$ sizes via two deconvolutional modules. Finally, we use the same decoder used in [15,60] for mask prediction.
Training loss. We use the commonly-used cross entropy loss [15,60] to train the whole network architecture.
Online inference. During the online inference, we use the first frame with the mask annotation as the memory frame for online target matching in the search frame.
The pipeline figure and more implementation details can be found in Supplementary.

4. Experiments

4.1. Implementation Details

Datasets. In the pre-training stage, we explore various large-scale video data sources to pre-train our DropMAE model, including Kinetics-400 [40] (K400), Kinetics-600 [8] (K600), Kinetics-700 [9] (K700), Moments in Time [56] (MiT) and WebVid-2M [1]. The detailed performance comparison using different pre-training datasets is shown in the ablation study. For fine-tuning on VOT, we follow the training settings used in our baseline OSTrack [97]. We use the training splits of LaSOT [28], COCO [49], TrackingNet [57] and GOT-10k [39] for training. For the GOT-10k evaluation, we follow the one-shot evaluation and only fine-tune the model on the training split of GOT-10k. For VOS fine-tuning, we use Youtube-VOS [90] and Davis [64] datasets for fine-tuning following the standard convention [15,60].
Pre-training and fine-tuning. For DropMAE pre-training, the default input is two frames with a spatial size of 224×224 pixels. The frames are randomly sampled from each video within a maximum frame gap of 50. During the pre-training, one epoch is counted when all videos are sampled

Methods	Pre-training Data	Epochs	Pre-train. Time (h)	GOT-10k (VOT)			DAVIS-17 (VOS)		
				AO	$SR_{0.5}$	$SR_{0.75}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
No Pre-training	-	-	-	62.7	72.8	53.7	69.5	66.9	72.2
Supervised IN1k [72]	IN1K	300	-	69.7	79.0	65.6	78.0	74.8	81.1
Supervised IN21k [66]	IN21K	80	-	70.2	80.7	65.4	78.5	75.4	81.7
CLIP [65]	IN1K	32	-	67.4	76.8	60.0	73.6	70.5	76.7
MOCO-v3 [12]	IN1K	300	-	70.1	80.1	65.3	78.4	75.4	81.5
BeiT [2]	IN1K	800	103.1	67.4	76.8	60.0	76.1	72.7	79.4
MAE [36]	IN1K	1600	84	73.7	83.2	**70.8**	81.7	78.5	84.9
TwinMAE	K400	400	20.7	72.2	83.2	65.9	79.3	76.4	82.3
TwinMAE	K400	800	41.3	72.9	83.6	68.5	80.7	77.9	83.6
TwinMAE	K400	1600	82.7	74.2	84.9	69.4	81.2	78.1	84.2
DropMAE	K400	400	21.1	73.2	83.9	67.5	81.3	78.5	84.0
DropMAE	K400	800	42.2	74.8	85.4	70.5	82.7	**79.7**	85.6
DropMAE	K400	1600	84.4	**75.8**	**86.4**	**72.0**	**83.1**	**80.2**	**86.0**
DropMAE	K700	800	92.4	**75.9**	**86.8**	**72.0**	**83.0**	**80.2**	**85.7**

Table 1. Comparison of pre-training methods on downstream VOT and VOS tasks on GOT-10k [39] and DAVIS-17 [64]. All methods adopt the ViT-B/16 model [24] with 224×224 input images for pre-training. The pre-training time is measured on 64 NVIDIA V100 GPUs. The best two results are shown in **red** and **blue**.

Method	Source	GOT-10k [39]			TNL2K [79]		$LaSOT_{ext}$ [27]			LaSOT [28]		
		AO	$SR_{0.5}$	$SR_{0.75}$	AUC	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P
SiamFC [3]	ECCVW16	34.8	35.3	9.8	29.5	28.6	23.0	31.1	26.9	33.6	42.0	33.9
MDNet [58]	CVPR16	29.9	30.3	9.9	-	-	27.9	34.9	31.8	39.7	46.0	37.3
ECO [19]	ICCV17	31.6	30.9	11.1	32.6	31.7	22.0	25.2	24.0	32.4	33.8	30.1
SiamPRN++ [41]	CVPR19	51.7	61.6	32.5	41.3	41.2	34.0	41.6	39.6	49.6	56.9	49.1
DiMP [4]	ICCV19	61.1	71.7	49.2	44.7	43.4	39.2	47.6	45.1	56.9	65.0	56.7
SiamR-CNN [74]	CVPR20	64.9	72.8	59.7	52.3	52.8	-	-	-	64.8	72.2	-
LTMU [18]	CVPR20	-	-	-	48.5	47.3	41.4	49.9	47.3	57.2	-	57.2
Ocean [104]	ECCV20	61.1	72.1	47.3	38.4	37.7	-	-	-	56.0	65.1	56.6
TrDiMP [76]	CVPR21	67.1	77.7	58.3	-	-	-	-	-	63.9	-	61.4
TransT [13]	CVPR21	67.1	76.8	60.9	50.7	51.7	-	-	-	64.9	73.8	69.0
AutoMatch [102]	ICCV21	65.2	76.6	54.3	47.2	43.5	37.6	-	43.0	58.3	-	59.9
STARK [92]	ICCV21	68.8	78.1	64.1	-	-	-	-	-	67.1	77.0	-
KeepTrack [55]	ICCV21	-	-	-	-	-	48.2	-	-	67.1	77.2	70.2
MixFormer-L [17]	CVPR22	70.7	80.0	67.8	-	-	-	-	-	70.1	79.9	76.3
SBT [87]	CVPR22	70.4	80.8	64.7	-	-	-	-	-	66.7	-	71.1
UAST [98]	ICML22	63.5	74.1	51.4	-	-	-	-	-	57.1	-	58.7
SwinTrack-384 [48]	NeurIPS22	72.4	80.5	67.8	**55.9**	**57.1**	49.1	-	55.6	**71.3**	-	76.5
AiATrack [32]	ECCV22	69.6	80.0	63.2	-	-	46.8	54.4	54.2	49.6	56.9	49.1
CIA50 [63]	ECCV22	67.9	79.0	60.3	50.9	57.6	-	-	-	66.2	-	69.6
SimTrack-L [10]	ECCV22	69.8	78.8	66.0	55.6	55.7	-	-	-	70.5	79.7	-
OSTrack-384 [97]	ECCV22	**73.7**	**83.2**	**70.8**	**55.9**	56.7	**50.5**	**61.3**	**57.6**	71.1	**81.1**	**77.6**
DropTrack	**Ours**	**75.9**	**86.8**	**72.0**	**56.9**	**57.9**	**52.7**	**63.9**	**60.2**	**71.8**	**81.8**	**78.1**

Table 2. Comparison with state-of-the-art VOT approaches on four large-scale challenging datasets. The best two results are shown in **red** and **blue**. For GOT-10k evaluation, all the methods follow the one-shot protocol, training only on the training set in GOT-10k.

once. For fair comparison, we use the same mask ratio (i.e., 75%) and training hyper-parameters of MAE [36] to pre-train the TwinMAE and DropMAE models. Following [14, 15], we use a bootstrapped cross entropy loss for training. The detailed pre-training and fine-tuning hyper-parameters are in the Supplementary.

4.2. Comparison with Pre-Training Methods

In Table 1, we compare our DropMAE with existing pre-training methods on the downstream tasks of VOT and VOS. DropMAE and TwinMAE are pre-trained using videos (K400, K700), while MAE and other methods are pre-trained on ImageNet 1k or 21k (IN1K, IN21K). The VOT and VOS baselines illustrated in Sec. 3.4 use the official pre-trained VIT-B/16 models provided by existing pre-training approaches (see Table 1) for fine-tuning. TwinMAE with 800-epoch training performs favorably against MAE on VOT, but achieves inferior results on VOS. There are two main reasons: 1) TwinMAE is not effective enough at learning temporal matches; 2) The number of object classes in K400 is limited, and meanwhile the object classes in DAVIS-17 are included in ImageNet. Thus MAE generalizes well to VOS. Our DropMAE, which is a stronger temporal matching learner, outperforms MAE on both the VOT and VOS tasks with 800-epoch training (i.e., 42.2 hours[1])

[1]In our implementation, we perform video decoding and extract the video frames on the fly in each training iteration, which could be further sped up using some tricks like repeated sampling [30].

Method	OTB100 AUC	ITB AUC	TrackingNet AUC	TrackingNet P_{Norm}
SiamFC [3]	58.3	44.1	57.1	66.3
Ocean [104]	68.4	47.7	-	-
ATOM [20]	68.3	47.2	70.3	77.1
DiMP [4]	53.7	339	74.0	80.1
TransT [13]	**69.5**	54.7	81.4	86.7
STARK [92]	68.1	57.6	82.0	86.9
OSTrack [97]	-	**64.8**	**83.9**	**88.5**
DropTrack	**69.6**	**65.0**	**84.1**	**88.9**

Table 3. Comparison with state-of-the-art VOT approaches on OTB100 [85], ITB [44] and TrackingNet [57]. The best two results are shown in **red** and **blue**.

by using the K400 dataset. This indicates that our DropMAE is 2× faster than MAE.

5. State-of-the-art Comparison

In this section, we compare our fine-tuned VOT and VOS models, denoted as DropTrack and DropSeg, with state-of-the-art approaches on VOT and VOS benchmarks. We use DropMAE trained on K700 with 800 epochs as the pre-training model for both VOT and VOS fine-tuning.

5.1. Video Object Tracking

To demonstrate the effectiveness of the proposed DropMAE for VOT, we compare our DropTrack with state-of-the-art trackers on 7 challenging tracking benchmarks.

GOT-10k. GOT-10k [39] is a challenging dataset that follows the one-shot evaluation protocol, where the trackers are required to be trained on its training split, and the test object classes have no overlap with the objects in the training split. As shown in Table 2, our DropTrack achieves state-of-the-art results on this dataset, outperforming OSTrack by 2.2% and 3.6% in terms of AO and $SR_{0.5}$. This implies that the temporal correspondence learning in the pre-training is beneficial for the downstream tracking task. Although there exists a domain gap between the pre-training data and the test data (i.e., a large portion of test objects in GOT-10k are animals, vehicles and object parts, whereas K700 only consists of human-centric action videos), the temporal matching ability learned by DropMAE can still be transferred to the downstream tracking task, improving the tracking performance.

LaSOT. LaSOT consists of 280 long test sequences, and our results are presented in Table 2. Our DropTrack sets a new record on this dataset with 71.8% AUC, 81.8% P_{Norm} and 78.1% P, which shows the great potential of our DropTrack in robust long-term visual tracking.

LaSOT$_{ext}$. LaSOT$_{ext}$ is an extension of LaSOT with more challenging video sequences for testing – similar to GOT-10k, the test split has a large gap with the training split, and sequences with novel object classes (i.e., not appeared in ImageNet) are used for evaluation. Our DropMAE outperforms the other trackers by large margins. This demonstrates that a tracker with DropMAE pre-training generalizes well to unseen objects in generic visual object tracking.

Method	Source	OL	M	S	DAVIS-2016 [62] $\mathcal{J}\&\mathcal{F}$	DAVIS-2016 $\mathcal{J}$	DAVIS-2016 $\mathcal{F}$	DAVIS-2017 [64] $\mathcal{J}\&\mathcal{F}$	DAVIS-2017 $\mathcal{J}$	DAVIS-2017 $\mathcal{F}$
RANet [80]	ICCV19			✓	85.5	85.5	85.4	65.7	63.2	68.2
STM [60]	ICCV19		✓	✓	89.3	**88.7**	89.9	81.8	79.2	84.3
FRTM [67]	CVPR20	✓	✓		83.5	83.6	83.4	76.7	73.9	79.6
TVOS [101]	CVPR20		✓		-	-	-	72.3	69.9	74.7
LWL [5]	ECCV20	✓	✓		-	-	-	81.6	79.1	84.1
CFBI [95]	ECCV20		✓		89.4	88.3	**90.5**	81.9	79.1	84.6
UniTrack [81]	NeurIPS21		✓		-	-	-	-	58.4	-
STCN$^-$ [15]	NeurIPS21		✓		-	-	-	**82.5**	79.3	**85.7**
SSTVOS [26]	CVPR21		✓		-	-	-	**82.5**	**79.9**	85.1
SWEM$^-$ [50]	CVPR22		✓		**89.5**	-	-	81.9	-	-
RTS [61]	ECCV22	✓	✓		-	-	-	80.2	77.9	82.6
OSMN [54]	TPAMI18				73.5	74.0	72.9	54.8	52.5	57.1
FAVOS [16]	CVPR18				81.0	82.4	79.5	58.2	54.6	61.8
VideoMatch [38]	ECCV18				-	81.0	-	56.5	-	-
SiamMask [77]	CVPR19				69.8	71.7	67.8	56.4	54.3	58.5
D3S [52]	CVPR20				74.0	75.4	72.6	60.8	57.8	63.8
Siam R-CNN [52]	CVPR20				-	-	-	70.6	66.1	75.0
Unicorn [91]	ECCV22				87.4	86.5	88.2	69.2	65.2	73.2
DropSeg	**Ours**				**92.1**	**90.9**	**93.3**	**83.0**	**80.2**	**85.7**

Table 4. Comparison with state-of-the-art VOS approaches on the validation sets of DAVIS-2016 [62] and DAVIS-2017 [64]. OL, M and S indicate **O**nline **L**earning, using **M**emory mechanism, and using **S**ynthetic videos for pre-training.

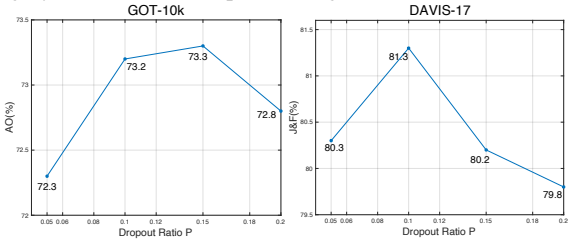

Figure 6. Ablation study of the dropout ratio P on the GOT-10k (VOT) and DAVIS-17 (VOS) datasets.

TNL2K. TNL2K is a large-scale evaluation dataset that consists of 700 test videos with various challenges, such as significant appearance variation and manually added adversarial samples. As illustrated in Table 2, our DropMAE significantly outperforms the other trackers on this dataset.

ITB, TrackingNet and OTB100. In Table 3, we evaluate our DropTrack on ITB [44], OTB100 [85] and TrackingNet [57], achieving state-of-the-art performance on each one. DropTrack is slightly better than OSTrack on ITB and TrackingNet. We believe the main reason is the fully overlapped training and test object classes in these two datasets, which reduces the effect of pre-training. A competitive tracker on these two datasets can be learned even using supervised ImageNet weights, which has been shown in [97].

On all 7 VOT datasets, our DropTrack outperforms the baseline OSTrack, which demonstrates that our DropMAE pre-training on videos learns better temporal-matching representations than the MAE model trained on ImageNet, resulting in more a robust tracker that generalizes well to both unseen and seen objects.

5.2. Video Object Segmentation

In Table 4, we compare our DropSeg with existing VOS approaches on the DAVIS-16/17 [62, 64].

DAVIS-16. DAVIS-16 is composed of 20 manually anno-

Datasets	No. Videos	No. Actions	VOT AO	VOT $SR_{0.5}$	VOT $SR_{0.75}$	VOS $\mathcal{J}\&\mathcal{F}$
K400 [40]	240,000	400	73.2	83.9	67.5	82.7
K600 [8]	390,000	600	74.5	85.5	69.5	82.8
K700 [9]	526,768	700	**75.6**	**86.2**	**71.4**	**83.0**
MiT [56]	802,244	339	75.1	85.5	70.6	82.8
WebVid [1]	240,000	-	72.8	83.4	67.3	81.5
WebVid [1]	960,000	-	73.4	85.0	69.5	82.9

Table 5. The downstream VOT and VOS performance on GOT-10k and DAVIS-17 obtained by using our DropMAE pre-trained on various video datasets. VOS uses 800 epochs pre-training.

tated test sequences. As shown in Table 4, our one-shot DropSeg approach, without using any online learning and complicated memory mechanisms, achieves the best $\mathcal{J}\&\mathcal{F}$ score of 92.1%, which significantly outperforms the other compared one-shot approaches and is even better than the approaches with complicated pipelines (i.e., OL, M and S). This implies that the pixel-wise correspondence learned during the pre-training is effective for capturing long-range dependencies between various frames in VOS.

DAVIS-17. DAVIS-17 is an extension of DAVIS-16, comprising more challenging videos and supports multi-object segmentation. In Table 4, our DropSeg achieves competitive results of 83.0% $\mathcal{J}\&\mathcal{F}$, 80.2% $\mathcal{J}$ and 85.7% $\mathcal{F}$, which shows its superiority in handling more challenging videos. More VOS results and comparisons are included in the **supplementary**.

6. Ablation Studies

In this section, we conduct ablation studies to provide more detailed analysis of our method. We use DropMAE with 400-epoch pre-training for the ablation study.

The effect of dropout ratio P. We study the effect of dropout rate P in Fig. 6. A relatively small dropout ratio of $P = 0.1$ works well on both VOT and VOS tasks. Meanwhile, dropping too many spatial cues (e.g., P=0.2) degrades the downstream tasks, which is mainly because the spatial cues are also useful for accurate localization and segmentation. $P = 0.1$ is the optimal setting, and thus we adopt it in the following experiments.

Pre-training video sources. Since we are the first to explore masked autoencoder pre-training for temporal matching tasks, it is not clear which video dataset is the optimal choice for pre-training. Here, we use five popular video datasets for pre-training, including K400 [40], K600 [8], K700 [9], MiT [56] and WebVid-2M [1]. For WebVid-2M, we randomly sample 240k and 960k videos for fair comparison and faster validation. The downstream tracking results are reported in Table 5. Performance is not favorable even using 960,000 videos in WebVid for pre-training. This indicates that WebViD is not a good choice for tracking pre-training, which is mainly because it is a video caption dataset that focuses on scene diversity and lacks rich object motion. From the experiments on the K400/600/700 and MiT, tracking benefits from pre-training with from rich action classes (i.e., 700 action classes of K700), from which the model can learn stronger temporal matching ability.

Settings	GOT-10k
DropTrack-K400-400E	73.2/67.5
w/ ASAD in Encoder	73.1/68.1
w/ domain specific data	73.4/68.8
w/o frame identity embed	72.9/67.4

Table 6. The tracking performance of AO/$SR_{0.75}$ on GOT10-k reported by variants with different settings.

Applying ASAD to the encoder. Here we test applying ASAD to all layers including both encoder and decoder of masked autoencoder. As shown in Table 6, this variant gains improvements (0.6% in $SR_{0.75}$) over the original baseline. Considering its additional cost and limited performance improvement, we only apply ASAD to the decoder.

Domain specific data. We also add the tracking training data (without using box annotations), including TrackingNet, LaSOT, and GOT-10k, into K400 for pre-training. The downstream tracking performance by using the larger pre-training set is 73.4/68.8, which is better than the baseline. It shows that the domain-specific data is helpful to bridge the domain gap, which can be considered as future work to extend Kinetic datasets with more tracking videos.

Frame identity embedding. During pre-training, the frame identity embedding is used to identify masked patches in the same 2D location of the two frames. From Table 6, we can find that downstream fine-tuning without the frame identity embedding performs worse than with it, since not using it is inconsistent with the pre-training stage.

7. Conclusion

This paper investigated masked autoencoding pre-training for temporal matching-based downstream tasks. Specifically, an adaptive spatial-attention dropout method is proposed to facilitate temporal correspondence learning in self-supervised pre-training on videos. We show that our proposed pre-training method DropMAE can achieve better downstream performance on VOT and VOS than the image-based MAE, while using 50% less pre-training time. In addition, as the first work investigating this problem, we show the guidelines of selecting video sources for pre-training, i.e., selecting videos with rich motion information is more beneficial for temporal matching-based downstream tasks. The experimental results on VOT and VOS show that our DropMAE sets new state-of-the-art results on 8 out of 9 tracking benchmarks. We expect our DropMAE to serve as a strong pre-trained backbone in future work.

8. Acknowledgment

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11215820).

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 5, 8

[2] H. Bao, L. Dong, and F. Wei. Beit: Bert pre-training of image transformers. In *arXiv:2106.08254*, 2021. 6

[3] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H.S. Vedaldi. Fully-convolutional siamese networks for object tracking. In *ECCVW*, pages 850–865, 2016. 2, 3, 4, 6, 7

[4] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pages 6182–6191, 2019. 6, 7

[5] G. Bhat, F.J. Lawin, M. Danelljan, A. Robinson, M. Felsberg, L.V. Gool, and R. Timofte. Learning what to learn for video object segmentation. In *ECCV*, 2020. 7

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, and Arvind Neelakantan. Language models are few-shot learners. In *NeurIPS*, 2020. 1, 3

[7] Yue Cao, Zhenda Xie, Bin Liu, Youtong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. In *NeurIPS*, 2020. 3

[8] João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. In *arXiv:1808.01340*, 2018. 5, 8

[9] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics700 human action dataset. In *arXiv:1907.06987*, 2019. 5, 8

[10] B. Chen, P. Li, L. Bai, L. Qiao, Q. Shen, and B. Li. Backbone is all your need: A simplified architecture for visual object tracking. In *ECCV*, 2022. 1, 2, 3, 6

[11] T. Chen, S. Kornblith, and M. Norouzi. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 3

[12] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 6

[13] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu. Transformer tracking. In *CVPR*, pages 8126–8135, 2021. 6, 7

[14] H. K. Cheng and A G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 3, 5, 6

[15] H. K. Cheng, Y. W. Tai, and C. K. Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, pages 11781–11794, 2021. 2, 5, 6, 7

[16] J. Cheng, Y.H. Tsai, W.C. Hung, S. Wang, and M.H. Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 7

[17] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*, 2022. 6

[18] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang. High-performance long-term tracking with meta-updater. In *CVPR*, 2020. 6

[19] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, pages 21–26, 2017. 2, 6

[20] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, pages 4660–4669, 2019. 7

[21] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: learning continuous convolution operators for visual tracking. In *ECCV*, pages 472–488, 2016. 2

[22] Acob Devlin, Ming-Wei Chang, Kenton Lee, , and Kristina Toutanova. Language models are few-shot learners. In *NAACL*, 2019. 1, 3

[23] X. Dong and J. Shen. Triplet loss in siamese network for object tracking. In *ECCV*, 2018. 2

[24] A. Dosovitskiy, L. Beyer, and A. Kolesnikov. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3, 4, 6

[25] Alexey Dosovitskiy, Jost Tobias, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, pages 766–774, 2014. 3

[26] Brendan Duk, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W. Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *CVPR*, 2021. 7

[27] H. Fan, H. Bai, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, M. Huang, J. Liu, and Y. Xu. Lasot: A high-quality large-scale single object tracking benchmark. In *IJCV*, 2021. 6

[28] H. Fan, L. Lin, and F. Yang. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, pages 5374–5383, 2019. 5, 6

[29] H. Fan and H. Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *ICCV*, 2019. 2

[30] C. Feichtenhofer, H. Fan, Y. Li, and K. He. Masked autoencoders as spatiotemporal learners. In *arXiv:2205.09113*, 2022. 3, 6

[31] H. Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017. 2

[32] Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Attention in attention for transformer visual tracking. In *ECCV*, 2022. 6

[33] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *arXiv:1803.07728*, 2018. 3

[34] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, pages 1763–1771, 2017. 2

[35] K. He and H. Fan adn Y. Wu. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 3

[36] K. He, X. Chen, and S. Xie. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1, 3, 6

[37] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 37(3):583–596, 2015. 2

[38] Y.T. Hu, J.B. Huang, and A.G. Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 7

[39] L. Huang, X. Zhao, and K. Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1, 5, 6, 7

[40] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. In *arXiv:1705.06950*, 2017. 5, 8

[41] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019. 2, 6

[42] B. Li, W. Wu, Z. Zhu, and J. Yan. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018. 2

[43] N. Li, S. Liu, and Y. Liu. Neural speech synthesis with transformer network. In *AAAI*, pages 6706–6713, 2019. 1

[44] X. Li, Q. Liu, W. Pei, Q. Shen, Y. Wang, H. Lu, and M.H. Yang. An informative tracking benchmark. In *arXiv:2112.06467*, 2021. 7

[45] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022. 5

[46] Y. Liang, Q. Wu, Y. Liu, and H. Wang. Robust correlation filter tracking with shepherded instance-aware proposals. In *ACM MM*, 2018. 2

[47] Y. Liang, Q. Wu, Y. Liu, Y. Yan, and H. Wang. Deep correlation filter tracking with shepherded instance-aware proposals. In *IEEE Transactions on Intelligent Transportation Systems*, 2021. 2

[48] L. Lin, H. Fan, Y. Xu, and H. Ling. Swintrack: A simple and strong baseline for transformer tracking. In *arXiv:2112.00995*, 2021. 6

[49] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[50] Z. Lin, T. Yang, M. Li, Z. Wang, C. Yuan, W. Jiang, and W. Liu. Swem: Towards real-time video object segmentation with sequential weighted expectation-maximization. In *CVPR*, pages 1362–1372, 2022. 3, 7

[51] Y. Liu, Y. Liang, Q. Wu, L. Zhang, and H. Wang. A new framework for multiple deep correlation filters based object tracking. In *ICASSP*, 2022. 2

[52] A. Lukezic, J. Matas, and M. Kristan. D3s: a discriminative single shot segmentation tracker. In *CVPR*, 2020. 7

[53] A. Lukezic and T. Vojir. Discriminative correlation filter woth channel and spatial reliability. In *CVPR*, 2017. 2

[54] K.K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taix´e, D. Cremers, and L. Van Gool. ideo object segmentation without temporal information. In *TPAMI*, 2018. 7

[55] C. Mayer, M. Danelljan, D.P. Paudel, and L. Van Gool. Learning target candidate association to keep track of what not to track. In *ICCV*, 2021. 6

[56] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, and A. Oliva. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):502–508, 2019. 5, 8

[57] M. Muller, A. Bibi, and Giancola S. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, pages 300–317, 2018. 5, 7

[58] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016. 6

[59] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 3

[60] S. W. Oh, J. Y. Lee, N. Xu, and S. J. Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9226–9235, 2019. 2, 5, 7

[61] Matthieu Paul, Martin Danelljan, Christoph Mayera, , and Luc Van Gool. Robust visual tracking by segmentation. In *ECCV*, 2022. 7

[62] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 7

[63] Zhixiong Pi, Weitao Wan, Chong Sun, Changxin Gao, Nong Sang, and Chen Li. Hierarchical feature embedding for visual tracking. In *ECCV*, 2022. 6

[64] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. In *arXiv:1704.00675*, 2017. 5, 6, 7

[65] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 6

[66] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor. Imagenet-21k pretraining for the masses. In *arXiv:2104.10972*, 2021. 6

[67] A. Robinson, F.J. Lawin, M. Danelljan, F.S. Khan, and M. Felsberg. Learning fast and robust target models for video object segmentation. In *CVPR*, 2020. 7

[68] O. Russakovsky, J. Deng, and et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[69] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 3

[70] R. Tao, E. Gavves, and A. W.M. Smeulders. Siamese instance search for tracking. In *CVPR*, pages 1420–1429, 2016. 2

[71] Z. Tong, Y. Song, J. Wang, and L. Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *arXiv:2203.12602*, 2022. 3

[72] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers distillation through attention. In *ICML*, 2021. 6

[73] T.D. Truong, C.N. Duong, and A.H. Pham. The right to talk: An audio-visual transformer approach. In *CVPR*, pages 1105–1114, 2021. 1

[74] P. Voigtlaender, J. Luiten, P.H.S. Torr, and B. Leibe. Siam r-cnn: Visual tracking by re-detection. In *CVPR*, pages 6578–6588, 2020. 6

[75] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 3

[76] N. Wang, W. Zhou, J. Wang, and H. Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *ICCV*, 2021. 6

[77] Q. Wang, L. Zhang, and L. Bertinetto. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 7

[78] X. Wang, C. Li, B. Luo, and J. Tang. Sint++: Robust visual tracking via adversarial positive instance generation. In *CVPR*, pages 4864–4873, 2018. 2

[79] X. Wang, X. Shu, Z. Zhang, B. Jiang, Y. Wang, Y. Tian, and F. Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *CVPR*, 2021. 6

[80] Z. Wang, J. Xu, L. Liu, F. Zhu, and L. Shao. Ranet: Ranking attention network for fast video object segmentation. In *ICCV*, 2019. 7

[81] Z. Wang, H. Zhao, Y.L. Li, S. Wang, P. Torr, and L. Bertinetto. Do different tracking tasks require different appearance models? In *NeurIPS*, 2021. 7

[82] Q. Wu and AB. Chan. Meta-graph adaptation for visual object tracking. In *ICME*, 2021. 2

[83] Q. Wu, J. Wan, and A. B. Chan. Progressive unsupervised learning for visual object tracking. In *CVPR*, pages 2993–3002, 2021. 2, 4

[84] Q. Wu, Y. Yan, Y. Liang, Y. Liu, and H. Wang. Dsnet: Deep and shallow feature learning for efficient visual tracking. In *ACCV*, pages 119–134, 2018. 2

[85] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015. 7

[86] Z. Wu, Y. Xiong, and S. Yu. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 3

[87] F. Xie, C. Wang, G. Wang, Y. Cao, W. Yang, and W. Zeng. Correlation-aware deep tracking. In *CVPR*, 2022. 6

[88] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, pages 16684–16693, 2021. 3

[89] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, and Z. Yao. Simmim: A simple framework for masked image modeling. In *CVPR*, pages 9653–9663, 2022. 1, 3

[90] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *arXiv:1809.03327*, 2018. 5

[91] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, 2022. 7

[92] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, pages 10448–10457, 2021. 6, 7

[93] T. Yang and A. B. Chan. Learning dynamic memory networks for object tracking. In *ECCV*, pages 152–167, 2018. 2

[94] T. Yang, P. Xu, and R. Hu. Roam: Recurrently optimizing tracking model. In *CVPR*, pages 6718–6727, 2020. 2

[95] Z. Yang, Y. Wei, and Y. Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020. 7

[96] Z. Yang, Y. Wei, and Y. Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, pages 2491–2502, 2021. 2, 5

[97] B. Ye, H. Chang, B. Ma, and S. Shan. Joint feature learning and relation modeling for tracking: A one-stream framework. In *ECCV*, pages 341–357, 2022. 1, 2, 3, 5, 6, 7

[98] Dawei Zhang, Yanwei Fu, and Zhonglong Zheng. Uast: Uncertainty-aware siamese tracking. In *ICML*, 2022. 6

[99] Lichao Zhang, Abel G.-Garcia, Joost van de Weijer, Martin Danelljan, Fahad Shahbaz, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *arXiv:1908.00855*, 2019. 2

[100] Richard Zhang, Phillip Isola, and Alexei A Efros. Image colorization. In *ECCV*, 2016. 3

[101] Y. Zhang, Z. Wu, H. Peng, and S. Lin. A transductive approach for video object segmentation. In *CVPR*, 2020. 7

[102] Z. Zhang, Y. Liu, X. Wang, B. Li, and W. Hu. Learn to match: Automatic matching network design for visual tracking. In *ICCV*, pages 19339–13348, 2021. 6

[103] Z. Zhang and H. Peng. Deeper and wider siamese networks for raal-time visual tracking. In *CVPR*, 2017. 2

[104] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 6, 7