Supplementary Material
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

In this supplementary material, we provide the additional implementation details, pseudocode, overall pipeline, additional ablation study and qualitative visualization. Sec. A shows the PyTorch-like pseudocode of the proposed adaptive spatial-attention dropout (ASAD). Sec. B details the proposed VOS baseline with the ViT backbone. Sec. C illustrates the implementation details in both pre-training and downstream fine-tuning stages. More ablation studies are conducted in Sec. D. Sec. E shows the qualitative visualization of downstream VOT and VOS results, the video reconstruction results of our DropMAE and the additional quantitative VOS results. We finally discuss the limitation and future work of our DropMAE in Sec. F.

A. Algorithm

We show the pseudocode of our proposed adaptive spatial-attention dropout (ASAD) in Algorithm 1. As can be seen, the implementation is simple and neat, which could be flexibility incorporated into existing approaches like MAE [5]. The implementation mainly consists of three steps: 1) the calculation of temporal matching probability f_tem; 2) the calculation of spatially normalized within-frame attention A_spa; 3) sampling for dropout from a multinomial distribution. Our code and pre-trained models will be publicly available once the paper is published.

Algorithm 1: ASAD Pseudocode, PyTorch-like

```
# Input:  attention matrix A, sequence
 length N, drop number N_d
W = torch.zeros_like(A) # N-by-N
A = A.detach().softmax(dim=-1)) # N-by-N

# get temporal attentions in each row of A
A_tem = temporal_index(A) # N-by-N//2
f_tem = A_tem.max(dim=-1).values # N-by-1

# get spatial attentions in each row of A
A_spa = spatial_index(A) # N-by-N//2
# avoid self-attention dropout
A_spa[0:N//2, 0:N//2].fill_diagonal_(0)
A_spa[N//2:, 0:N//2].fill_diagonal_(0)
A_spa=A_spa/A_spa.sum(dim=-1, keepdim=True)

# calculate overall dropout probability
f_all = f_tem * A_spa # N-by-N//2

# put back to probability matrix W
W[0:N//2, 0:N//2] = f_all[0:N//2, 0:N//2]
W[N//2:, N//2:]  = f_all[N//2:, 0:N//2]
# sample N_d elements based on W

indices=torch.multinomial(W.view(1,-1),N_d)
return indices
```

B. VOS Baseline

Since there is no ViT-based VOS approach, we build a simple and effective ViT-based VOS baseline, namely DropSeg, in order to demonstrate the effectiveness of our DropMAE pre-training in VOS.

Architecture. The overall pipeline of our DropSeg is shown in Fig. 1, which mainly consists of the pre-trained ViT backbone and the mask prediction head. Note that the frame identity embeddings are two randomly initialized learnable vectors. We use the standard ViT-B/16 model [4] as the backbone and initialized it with our DropMAE pre-trained model, and the same decoder used in [3, 11] is employed. Since the decoder requires multi-resolution features for mask prediction, we follow [10] to upsample the updated search features to $2\times$ and $4\times$ sizes via two deconvolutional modules.

Fine-tuning. During the fine-tuning stage, given a video, we randomly sample a template frame with the mask anno-

*Corresponding Author

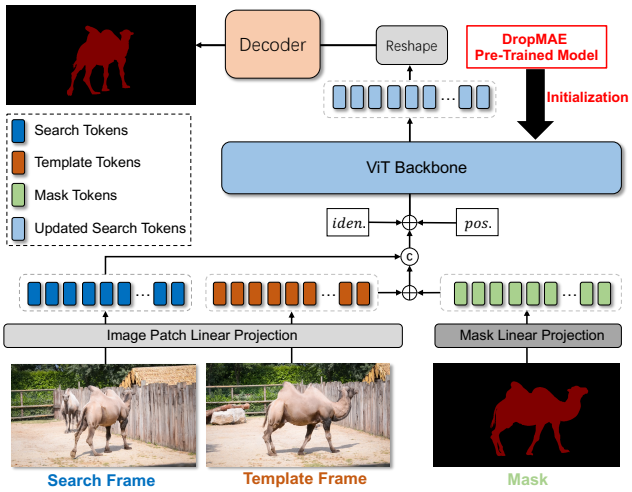

Figure 1. An overall pipeline of the proposed DropSeg for VOS. *iden* and *pos* indicate frame identity embeddings and positional embeddings, respectively. Our DropSeg with the DropMAE pre-trained model sets new state-of-the-art one-shot segmentation results on the DAVIS-16 [12] and DAVIS-17 [13] datasets.

Config	Value
optimizer	AdamW [8]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$ [2]
batch size	4096
learning rate schedule	cosine decay
warmup epochs	40
augmentation	RandomResizedCrop
dropout ratio p	0.1
maximum sampling frame gap	50

Table 1. **The pre-training setting for DropMAE.**

tation, and a search frame of a training video within a predefined maximum frame gap (i.e., 25). Following the convention [11], DAVIS-17 [13] and YouTube-19 [16] datasets are used for training. The detailed training hyper-parameters are in Table 3. We use the same bootstrapped cross entropy loss in [3, 11] for supervision.

C. Implementation Details

In this section, we detail the implementation details of our DropMAE pre-training, and fine-tuning details of downstream VOT and VOS tasks.

C.1. DropMAE Pre-Training

We use the standard ViT-B/16 [4] as our backbone for pre-training. The detailed pre-training hyper-parameters are in Table 1, which mainly follows the training settings used in the original MAE [5]. For the dropout ratio P, we set

Config	Value
optimizer	AdamW [8]
learning rate in head	2.5e-4
learning rate in backbone	2.5e-5
weight decay	0.0001
droppath rate	0.1
batch size	128
epoch	300
learning rate decay epoch	240
learning rate decay factor	10
learning rate schedule	steplr
maximum sampling frame gap	200

Table 2. **The fine-tuning setting for DropTrack.**

Config	Value
optimizer	AdamW [8]
base learning rate	2e-5
weight decay	1e-7
droppath rate	0.1
batch size	32
Iterations	210,000
learning rate decay iteration	125,000
learning rate schedule	steplr
maximum sampling frame gap	25

Table 3. **The fine-tuning setting for DropSeg.**

$P = 0.1$ following the ablation study in Sec. 6 of the main paper. The pre-training is conducted on 64 NVIDIA V100 GPUs. As illustrated in Table 1 of the main paper, the 1600-epoch pre-training takes about 84 hours on the K400 dataset [7]. The whole training time can be further reduced to 58 hours by using 64 NVIDIA A100 GPUs.

C.2. Downstream VOT Fine-Tuning

Following the tracking baseline OSTrack [17], we use the template and search sizes of 192×192 and 384×384 pixels, respectively, and fine-tune our DropMAE model with the tracking specific data (see Sec. 4.1 of the main paper) on 4 NVIDIA A100 GPUs. The candidate elimination module proposed in OSTrack are also used for a fair comparison. For the full 300-epoch fine-tuning, the detailed hyper-parameters are in Table 2. For fine-tuning on GOT-10k [6], the total training epoch is reduced to 100 and the learning rate decays at 80 epoch. The inference speed of our DropTrack is the same as the baseline OSTrack [17], which is 58.1 FPS measured on a single GPU.

C.3. Downstream VOS Fine-Tuning

The detailed fine-tuning setting of DropSeg is shown in Table 3. We use 8 A100 GPUs for fine-tuning, and the whole training takes about 16 hours. The inference speed

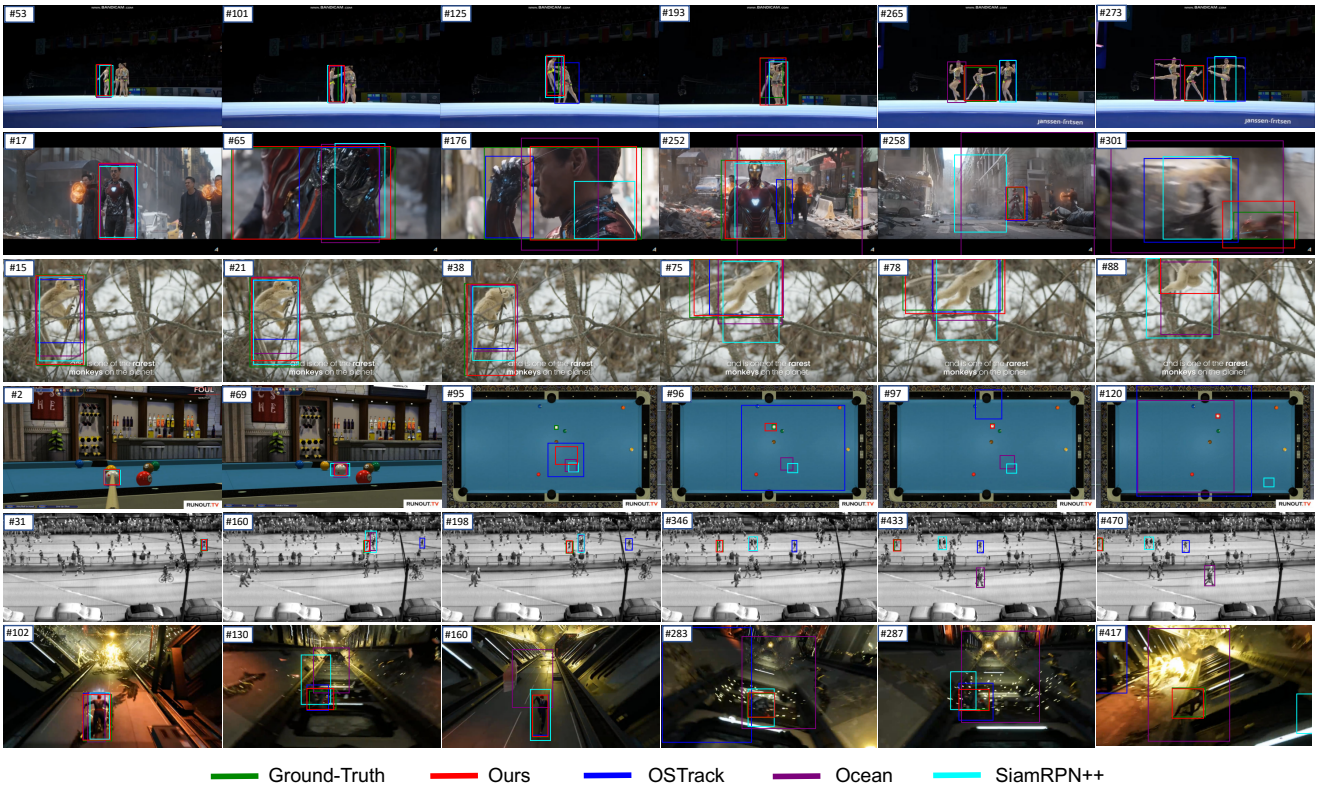

Figure 2. Qualitative tracking results of our DropTrack and state-of-the-art tracking methods, including OSTrack [17], Ocean [18] and SiamRPN++ [9]. The six video sequences are collected from TNL2K [15] (from top to down are *SportGirl_video_01_done*, *test_031_IronMan_transform_05_done*, *Monkey_BBC_video_01-Done*, *ZhuoQiu_video_02-Done*, *INF_crow1* and *Zhizhuxia_09-Done*, respectively). The frame number is shown in the *top-left* of each frame.

of DropSeg is about 10 FPS, which is measured on a single A100 GPU.

D. Additional Ablation Studies

In this section, we provide additional ablation studies on parameter selection and effectiveness analysis. We use the model pre-trained on K400 with 400 epochs for the ablation study.

Effect of maximum sampling frame gap. During the pre-training, we randomly sample two frames of a training video with a predefined maximal sampling frame gap g. Here, we study its effect on the downstream VOT task. As shown in Table. 4, the VOT task benefits more from the large sampling frame gap, i.e., 50. This is because the stronger temporal matching ability can be learned by using the relatively large sampling frame gap. Since the limited performance improvements from $g = 10$ to $g = 50$, we directly use $g = 50$ for all the pre-training experiments without further searching for the parameters.

Learning static frame representation from K400. To demonstrate the temporal correspondence learning in the pre-training is the key to the success of downstream tracking tasks, we treat K400 [7] as a static image dataset and perform the original MAE pre-training on it. We denote this baseline as MAE-K400-static. Specifically, in each training iteration, one frame image is randomly sampled of a video for masked autoencoding pre-training. To make a fair comparison with our DropMAE, we double the video number in this baseline such that the total sampled frame number in one epoch training is the same as DropMAE. The comparison between MAE-K400-static and DropMAE is shown in

Maximum Sampling Frame Gap	GOT-10k		
	AUC	$SR_{0.5}$	$SR_{0.75}$
1	72.2	82.7	65.7
10	72.8	83.4	67.2
50	**73.2**	**83.9**	**67.5**

Table 4. The effect of maximum sampling frame gap on the downstream tracking task.

Methods	GOT-10k		
	AUC	$SR_{0.5}$	$SR_{0.75}$
DropMAE	**73.2**	**83.9**	**67.5**
MAE-K400-static	70.4	80.7	65.6

Table 5. The comparison between DropMAE and MAE-K400-static on GOT-10k [6].

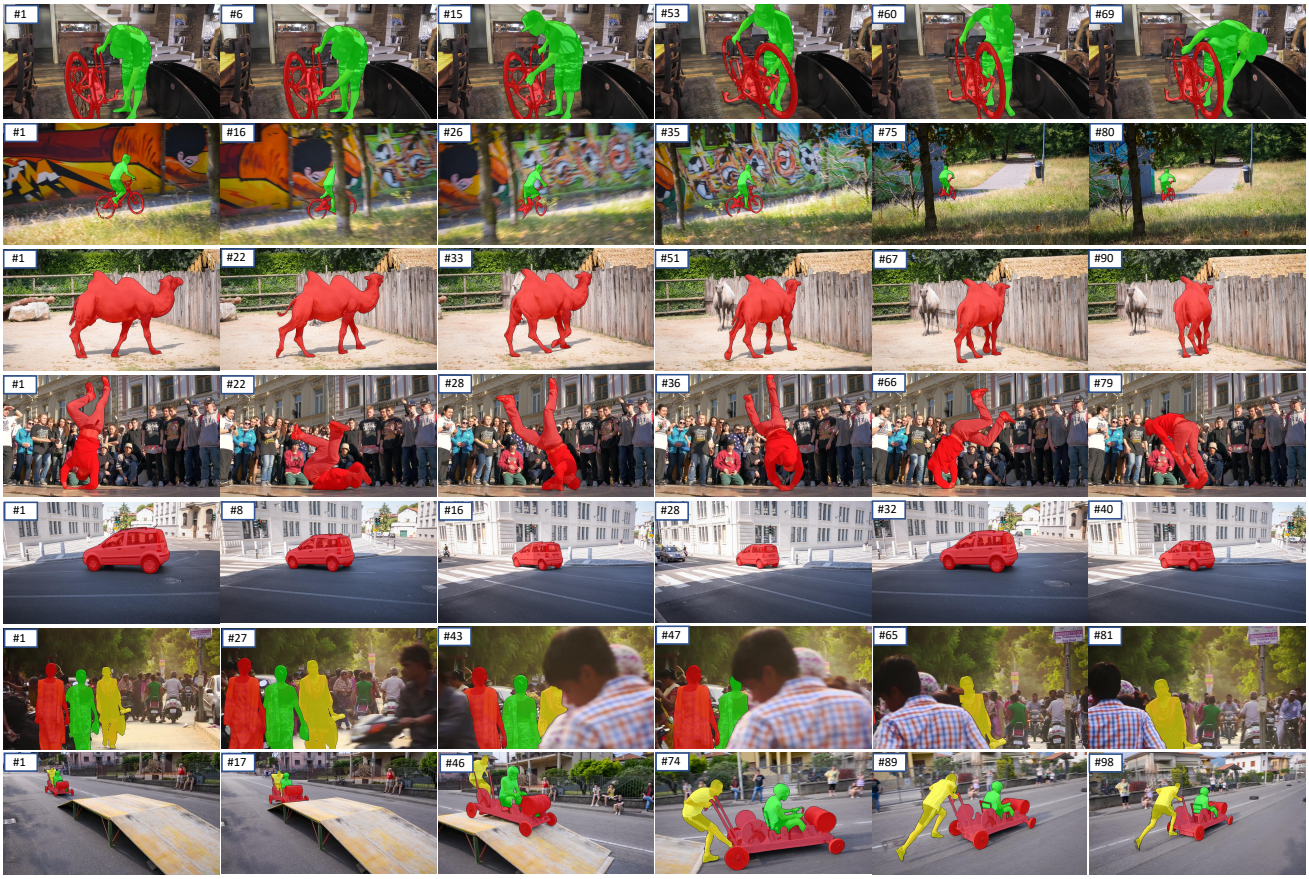

Figure 3. Qualitative results of our one-shot approach DropSeg on seven sequences in DAVIS-17 [13], which are respectively *bike-packing*, *bmx-trees*, *camel*, *breakdance*, *car-shadow*, *india* and *soapbox*. The frame number is shown in the *top-left* of each frame, and the ground-truth mask annotation is given in the first frame. Our DropSeg shows favorable segmentation results without using complicated designs such as online fine-tuning and memory mechanisms.

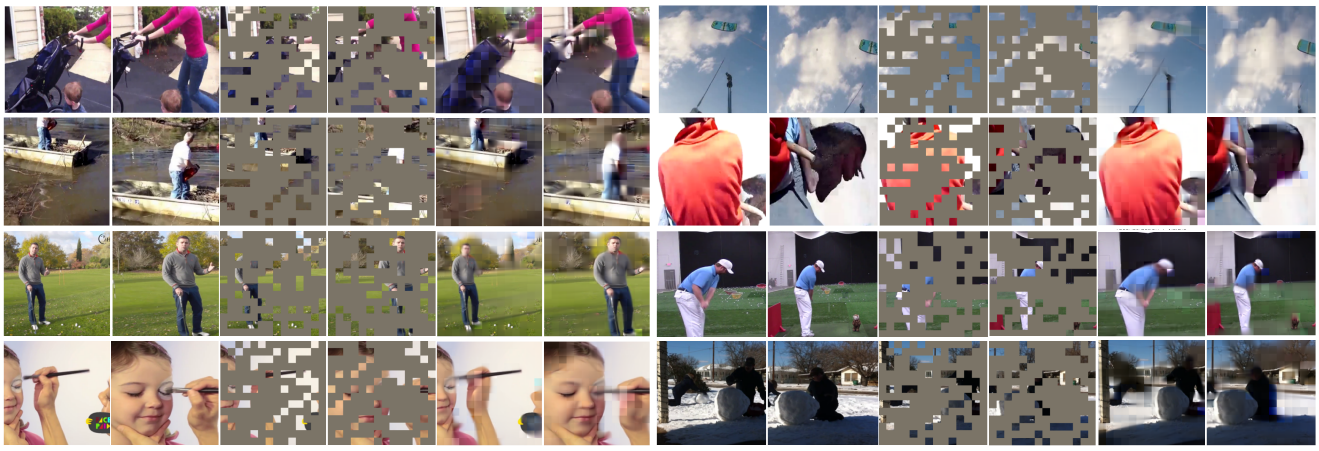

Figure 4. Video frame reconstruction results of DropMAE on K400 validation set. We show the original input frame pairs, masked frame pairs (i.e., with 75% mask ratio) and reconstruction results, sequentially.

Method	#Mem	Static	BL30K	DAVIS-17			YouTubeVOS-19 val				
				$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$
XMEM	∼27	✓	✓	87.7	84.0	91.4	85.8	84.8	89.2	80.3	88.8
STCN	27	✓	✓	85.3	82.0	88.6	84.2	82.6	87.0	79.4	87.7
XMEM	∼27	✓		86.2	82.9	89.5	85.5	84.3	88.6	80.3	88.6
STCN	27	✓		85.4	82.2	88.6	82.7	81.1	85.4	78.2	85.9
SWEM	27	✓		84.3	81.2	87.4	82.6	82.0	86.1	77.2	85.2
DropSeg^{+}	2	✓		86.5	83.5	89.5	83.4	82.9	87.3	77.7	85.6
STCN^{-}	27			82.5	79.3	85.7	-	-	-	-	-
DropSeg	1			83.0	80.2	85.7	-	-	-	-	-

Table 6. 'Static' and 'BL30K' indicate pre-training with static images and large-scale video BL30K. #Mem is the estimated number of memory frames, which is measured w/ the avg. video length in YouTubeVOS-19.

Methods	GOT-10k		
	AUC	$SR_{0.5}$	$SR_{0.75}$
DropMAE	**73.2**	**83.9**	**67.5**
RandDrop-MAE	71.7	82.4	66.2

Table 7. The comparison between DropMAE and RandDrop-MAE on GOT-10k [6].

Pre-trained MAE	GOT-10k			Davis-17
	AUC	$SR_{0.5}$	$SR_{0.75}$	$\mathcal{J}\&\mathcal{F}$
w/o	73.2	83.9	67.5	81.3
w/	**75.2**	**85.4**	**71.5**	**82.6**

Table 8. The ablation study on the usage of the pre-trained MAE model for DropMAE pre-training.

Table 5. As can be seen, without temporal correspondence learning, MAE-K400-static is significantly worse than our DropMAE, which further demonstrates the effectiveness of the temporal correspondence learning in the DropMAE pre-training.

Random dropout. The vanilla ViT [4] implements dropout [14] in each multi-head self-attention layer. To see whether this random dropout works in our masked autoencoding pre-training setting, we build a baseline called RandDrop-MAE, which adopts the random dropout in each self-attention layer of the decoder during the pre-training. Different from our adaptive dropout strategy (i.e., ASAD), RandDrop-MAE randomly drops between-frame or within-frame attentions. For a fair comparison, we use the same dropout ratio (i.e., 0.1) for RandDrop-MAE. As shown in Table 7, RandDrop-MAE degrades the performance compared with our DropMAE. We believe this is because the random dropout may excessively drop some attention elements that are essential for reconstruction and thus degrade the learning.

Pre-trained MAE. The downstream VOT and VOS tasks consist of large amounts of objects with diverse classes for evaluation. Considering that K400 is composed of human-action videos, there still exists domain gap between the pre-training and fine-tuning stages. In order to alleviate this gap, we use the original MAE trained on ImageNet as the pre-training weights of our DropMAE, and then we further pre-train our DropMAE on K400 for temporal correspondence learning. From Table 8, we can find that our DropMAE benefits from the pre-trained MAE on both VOT and VOS tasks, which is mainly because the diverse object classes learned in MAE are beneficial for generic object tracking and segmentation. This also shows the potential that the better downstream performance can be achieved by using the pre-trained MAE and larger video data sources (e.g., K700 [1]).

E. Qualitative Visualization

In this section, we show the qualitative results of our DropTrack and DropSeg on the VOT and VOS tasks, respectively.

E.1. Video Object Tracking

In Fig. 2, we show the qualitative tracking results obtained by our DropTrack and the other 3 compared trackers. The selected sequences contain various challenges including significant appearance variation, background cluster, illumination variation and similar objects. Our DropTrack handles these challenges well due to the robust DropMAE pre-trained model.

E.2. Video Object Segmentation

The qualitative visualization of our DropSeg is shown in Fig. 3. Even without using online fine-tuning or complicated memory mechanisms, our DropSeg can still provide accurate segmentation results in the following frames by only using the mask annotation in the first frame, which is mainly due to the favorable temporal matching ability learned in the DropMAE pre-training.

E.3. Frame Reconstruction

We show the video frame reconstruction results obtained by our DropMAE in Fig. 4. As can be seen, although less

spatial cues are leveraged in the reconstruction, our DropMAE still achieves favorable reconstruction results by exploring temporal cues or between-frame patches.

E.4. Additional Comparison on VOS

In this subsection, we provide the additional comparisons on DAVIS-17 and YouTube-VOS 19 datasets, which are illustrated in Table 6. DropSeg does not use online memory mechanisms, so we have used short videos (e.g., DAVIS-16/17) to focus evaluation on the learned representations. To more fairly compare on the longer YouTube-VOS videos, we employ an improved variant DropSeg$^+$, which 1) uses static images for video pre-training (as in [3, 11]); 2) uses the first and previously predicted frames as memory. In Tab. 6, although our DropSeg$^+$ only uses 2 memory frames, it is better than STCN w/ static image pre-training on DAVIS-17 & YouTube-19. DropSeg$^+$ is comparable to XMEM on DAVIS-17, but worse on YouTube-19, which is mainly due to the lack of complex online memory and BL30K pre-training.

F. Limitation and Future Work

Due to the limited object classes in video datasets (e.g., K400 [7] and K700 [1]), the pre-training video sources still have large domain gap with the downstream VOT and VOS tasks, which results in a sub-optimal pre-trained model. In Table 8, we find that this gap can be alleviated by using the pre-trained MAE model during the DropMAE pre-training. In the future work, we will perform more large-scale DropMAE pre-training (i.e., w/ the MAE pre-trained model) on larger video sources in order to provide more robust pre-trained models for VOT and VOS communities.

References

[1] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics700 human action dataset. In *arXiv:1907.06987*, 2019. 5, 6

[2] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 2

[3] 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. 1, 2, 6

[4] A. Dosovitskiy, L. Beyer, and A. Kolesnikov. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 5

[5] K. He, X. Chen, and S. Xie. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1, 2

[6] 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. 2, 3, 5

[7] 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. 2, 3, 6

[8] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *arXiv:1412.6980*, 2014. 2

[9] 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. 3

[10] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022. 1

[11] 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. 1, 2, 6

[12] 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. 2

[13] 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. 2, 4

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *The journal of machine learning research*, 2014. 5

[15] 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. 3

[16] 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. 2

[17] 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. 2, 3

[18] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 3