Scalable Video Object Segmentation with Simplified Framework

Qiangqiang Wu[1] Tianyu Yang[2*] Wei Wu[1] Antoni B. Chan[1]
[1]Department of Computer Science, City University of Hong Kong
[2]International Digital Economy Academy
{qiangqwu2-c, weiwu56-c-c}@my.cityu.edu.hk, tianyu-yang@outlook.com
abchan@cityu.edu.hk

Abstract

The current popular methods for video object segmentation (VOS) implement feature matching through several hand-crafted modules that separately perform feature extraction and matching. However, the above hand-crafted designs empirically cause insufficient target interaction, thus limiting the dynamic target-aware feature learning in VOS. To tackle these limitations, this paper presents a scalable Simplified VOS (SimVOS) framework to perform joint feature extraction and matching by leveraging a single transformer backbone. Specifically, SimVOS employs a scalable ViT backbone for simultaneous feature extraction and matching between query and reference features. This design enables SimVOS to learn better target-ware features for accurate mask prediction. More importantly, SimVOS could directly apply well-pretrained ViT backbones (e.g., MAE [21]) for VOS, which bridges the gap between VOS and large-scale self-supervised pre-training. To achieve a better performance-speed trade-off, we further explore within-frame attention and propose a new token refinement module to improve the running speed and save computational cost. Experimentally, our SimVOS achieves state-of-the-art results on popular video object segmentation benchmarks, i.e., DAVIS-2017 (88.0% $\mathcal{J}\&\mathcal{F}$), DAVIS-2016 (92.9% $\mathcal{J}\&\mathcal{F}$) and YouTube-VOS 2019 (84.2% $\mathcal{J}\&\mathcal{F}$), without applying any synthetic video or BL30K pre-training used in previous VOS approaches. Our code and models are available at `https://github.com/jimmy-dq/SimVOS.git`.

1. Introduction

Video Object Segmentation (VOS) is an essential and fundamental computer vision tasks in video analysis [30, 47, 48, 57, 49, 51] and scene understanding [28, 12, 46, 22, 15, 45]. In this paper, we focus on the semi-supervised VOS task, which aims to segment and track the objects of interest in each frame of a video, using only the mask annotation of the target in the first frame as given. The key challenges in VOS mainly lie in two aspects: 1) how to effectively distinguish the target from the background distractors; 2) how to accurately match the target across various frames in a video.

In the past few years, modern matching-based VOS approaches have gained much attention due to their promising performance on popular VOS benchmarks [54, 32, 33]. The typical method STM [30] and its following works [8, 57, 7] mainly use several customized modules to perform semi-supervised VOS, including feature extraction, target matching and mask prediction modules. The whole mask prediction process in these approaches can be divided into two sequential steps: 1) feature extraction on the previous frames (i.e., memory frames) and the new incoming frame (i.e., search frame); and 2) target matching in the search frame, which is commonly achieved by calculating per-pixel matching between the memory frames' embeddings and the search frame embedding.

Despite the favorable performance achieved by the above matching-based approaches, the separated feature extraction and matching modules used in these methods still have several limitations. Firstly, the separate schema is unable to extract dynamic target-aware features, since there is no interaction between the memory and search frame embeddings during the feature extraction. In this way, the feature extraction module is treated as the fixed feature extractor after offline training and thus cannot handle objects with large appearance variations in different frames of a video. Secondly, the matching module built upon the extracted features needs to be carefully designed to perform sufficient interaction between query and memory features. Recent works (e.g., FEELVOS [42] and CFBI [56]) explore to use local and global matching mechanisms. However, their performance is still degraded due to the limited expressive power of the extracted fixed features.

To solve the aforementioned problems, this paper presents a *Simplified VOS framework* (SimVOS) for joint

[*] Corresponding Author

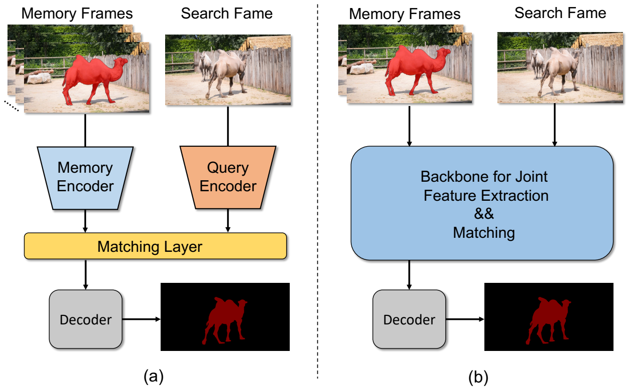

Figure 1: A comparison between the pipelines of (a) traditional VOS approaches [30, 8], and (b) our proposed SimVOS. The previous approaches predict segmentation masks by leveraging the customized separate feature extraction and matching modules. Our SimVOS removes hand-crafted designs and employs a unified transformer backbone for joint feature extraction and matching, which provides a simplified framework for accurate VOS.

feature extraction and matching. This basic idea allows SimVOS to learn dynamic target-aware features for more accurate VOS. Inspired by the recent successes on replacing hand-crafted designs [3, 11, 59] with general-purpose architectures in computer vision, we implement SimVOS with a ViT [11] backbone and a mask prediction head. As can be seen in Fig. 1, this new design removes the customized separate feature extraction and matching modules used in previous matching-based VOS approaches, thus facilitating the development of VOS in a more general and simpler system.

Besides providing a simple yet effective VOS baseline, the other goal of our work is to bridge the gap between the VOS and large-scale self-supervised pretraining communities. Recently, significant progress [21, 34, 20, 4] have been made in showing the superior performance of large-scale self-supervised models on some downstream tasks, including object classification [21], detection [23] and tracking [59, 3]. However, existing VOS approaches often rely on task-specific customized modules, and their architectures are specifically designed with VOS-specific prior knowledge, which makes it difficult for these approaches to utilize standard large-scale self-supervised models for VOS. As far as we know, leveraging a large-scale self-supervised model to develop a general-purpose architecture for VOS has not been explored. Our work moves towards a simple yet effective VOS framework that could naturally benefit from large-scale self-supervised pretraining tasks.

Taking both memory and query frames as input to the vanilla ViT for mask prediction may cause a large computational cost (quadratic complexity). To achieve a better performance-speed trade-off, we propose a token refinement module to reduce the computational cost and improve the running speed of SimVOS. This variant can run 2× faster than the SimVOS baseline, with a small reduction in VOS performance. We conduct experiments on various popular VOS benchmarks and show that our SimVOS achieves state-of-the-art VOS performance. In summary, this paper makes the following contributions:

- We propose a *Simplified VOS framework* (SimVOS), which removes the hand-crafted feature extraction and matching modules in previous approaches [30, 8], to perform joint feature extraction and interaction via a single scalable transformer backbone. We also demonstrate that large-scale self-supervised pre-trained models can provide significant benefits to the VOS task.
- We proposed a new token refinement module to achieve a better speed-accuracy trade-off for scalable video object segmentation.
- Our SimVOS achieves state-of-the-art performance on popular VOS benchmarks. Specifically, without applying any synthetic data pre-training, our variant SimVOS-B sets new state-of-the-art performance on DAVIS-2017 (88.0% $\mathcal{J}\&\mathcal{F}$), DAVIS-2016 (92.9% $\mathcal{J}\&\mathcal{F}$) and YouTube-VOS 2019 (84.2% $\mathcal{J}\&\mathcal{F}$).

2. Related Work

Video Object Segmentation. Traditional VOS methods follow the basic idea of online fine-tuning at test time to adapt to online tracked objects. Typical works include OSVOS [2], OnAVIS [43], MoNet [52], MaskTrack [31] and PReMVOS [26]. However, the time-consuming fine-tuning step limits their applicability to real-time applications, and meanwhile, the limited number of online training samples still degrades online fine-tuning. To improve the inference efficiency, OSMN [55] employs a meta neural network to guide mask prediction and uses a single forward pass to adapt the network to a specific test video. PML [5] formulates VOS as a pixel-wise retrieval task in the learned feature embedding space and uses a nearest-neighbor approach for real-time pixel-wise classification.

The typical matching-based approach STM [30] employs an offline-learned matching network, and treats past frame predictions as memory frames for the current frame matching. CFBI [56] and FEELVOS [42] further improve the matching mechanism by leveraging foreground-background integration and local-global matching. AOT [57] employs a long short-term transformer and an identification mechanism for the simultaneously multi-object association. STCN [8] uses the L2 similarity to replace the dot product used in STM and establishes correspondences only on images to further improve the inference speed. To efficiently encode spatiotemporal cues, SSTVOS [14] uses a sparse spatiotemporal transformer. XMEM [7] further proposes an Atkinson-Shiffrin memory model to enable STM-

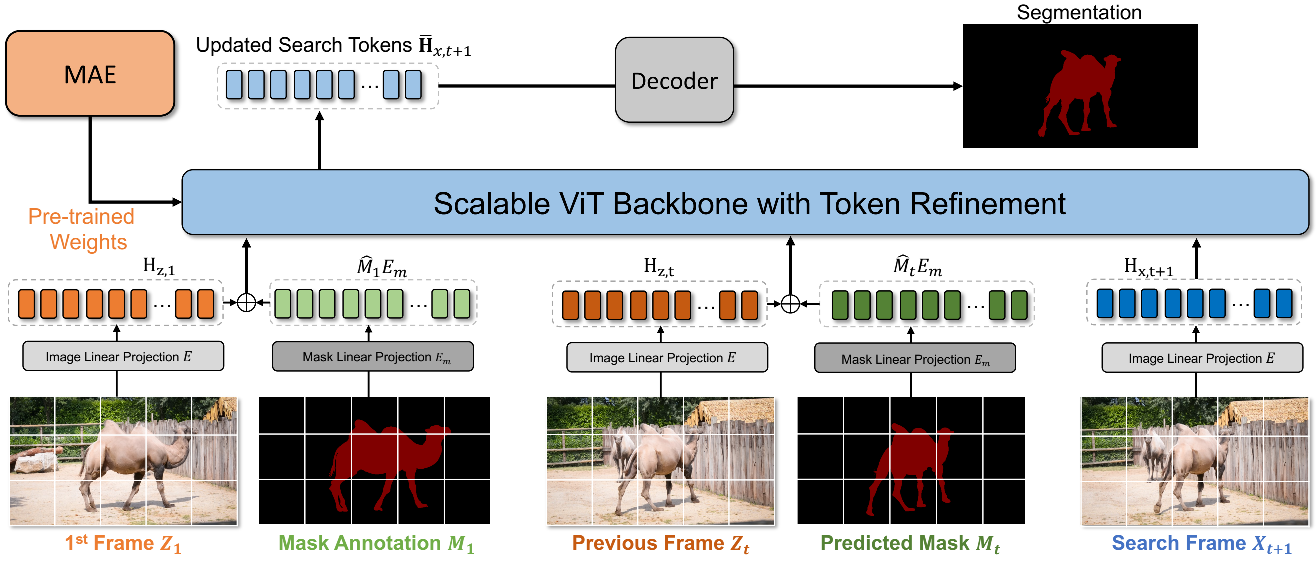
MAE
Pre-trained Weights
Updated Search Tokens $\bar{\mathbf{H}}_{x,t+1}$
Decoder
Segmentation
Scalable ViT Backbone with Token Refinement
$\mathrm{H}_{z,1}$
$\hat{M}_1 E_m$
$\mathrm{H}_{z,t}$
$\hat{M}_t E_m$
$\mathrm{H}_{x,t+1}$
Image Linear Projection E
Mask Linear Projection E_m
Image Linear Projection E
Mask Linear Projection E_m
Image Linear Projection E
1st Frame Z_1
Mask Annotation M_1
Previous Frame Z_t
Predicted Mask M_t
Search Frame X_{t+1}

Figure 2: The overall architecture of the proposed Simplified Video Object Segmentation (SimVOS) framework. Our SimVOS consists of a scalable ViT backbone with token refinement for template and search token interaction, a decoder for segmentation mask prediction, and image/mask projection modules.

like approaches to perform long-term VOS. Although successes have been achieved by these matching-based approaches, their performance is still limited due to the separate feature extraction and interaction designs. In this work, we show that the above hand-crafted designs can be effectively replaced with a general-purpose transformer architecture, i.e., a vanilla ViT [11] backbone with joint feature extraction and interaction, which can greatly benefit from existing large-scale pre-trained models (e.g., MAE [21]) and thus improve the state-of-the-art VOS performance [32, 33].

Large-scale self-supervised pre-training. Large-scale self-supervised pre-training has achieved significant progress in recent years. Traditional approaches mainly focus on designing various pretext tasks for unsupervised representation learning, e.g., solving jigsaw puzzle [29], coloring images [60] and predicting future frame [39, 44] or rotation angle [17]. Recent advances [4, 20, 50, 40] show that more discriminative unsupervised feature representations can be learned in a contrastive paradigm. However, these approaches may ignore modeling of local image structures, thus being limited in some fine-grained vision tasks, e.g., segmentation. The generative MAE [21] approach further improves on the contrastive learning-based methods by learning more fine-grained local structures, which are beneficial for localization or segmentation-based downstream tasks. In this work, our proposed SimVOS can directly apply the pre-trained models learned by existing self-supervised methods, which effectively bridges the gap between the VOS and the self-supervised learning communities. We also show that MAE can serve as a strong pre-trained model for the VOS task.

3. Methodology

In this section, we present our proposed *Simplified VOS* (SimVOS) framework. An overview of SimVOS is shown in Fig. 2. We firstly introduce the basic SimVOS baseline with joint feature extraction and interaction for accurate video object segmentation in Sec. 3.1. Then, in order to reduce the computational cost and improve the inference efficiency, multiple speed-up strategies are explored in Sec. 3.2, including the usage of within-frame attention and a novel token refinement module for reducing the number of tokens used in the transformer backbone.

3.1. Simplified Framework

As shown in Fig. 2, our basic SimVOS mainly consists of a joint featrue extraction module and a mask prediction head. We use a vanilla ViT [11] as the backbone of SimVOS, which is mainly because: 1) ViT naturally performs the joint feature extraction and interaction, which perfectly meets our design; and 2) a large amount of pre-trained ViT models can be directly leveraged in the VOS task, without needing time-consuming model-specific synthetic video pre-training commonly used for previous VOS methods [8, 7, 30].

In our SimVOS, the memory frames for online matching consist of the initial template frame $Z_1 \in \mathbb{R}^{3\times H\times W}$ with ground truth mask $M_1 \in \mathbb{R}^{H\times W}$ and the previ-

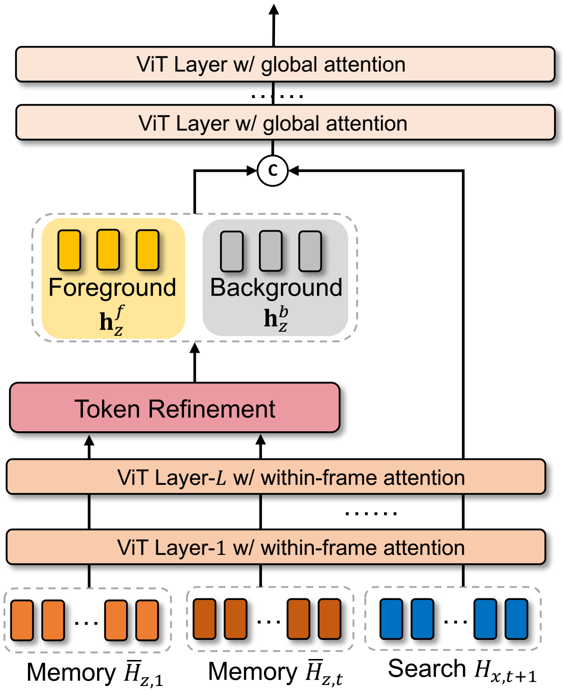

Figure 3: Overall pipeline of a scalable ViT backbone with within-frame attention applied to the first L layers, and our token refinement module applied after the L-th layer.

ous t-th frame $Z_t \in \mathbb{R}^{3\times H\times W}$ with the predicted mask $M_t \in \mathbb{R}^{H\times W}$. Given a current search frame X_{t+1}, the goal of SimVOS is to accurately predict the mask of X_{t+1} based on the memory frames. Instead of directly inputing the input frames to the ViT backbone for joint feature extraction and interaction, the input frames are first serialized into input sequences. Specifically, each input frame is reshaped into a sequence of flattened 2D patches with the size of $N \times 3P^2$ to obtain the reshaped memory sequences (i.e., $\hat{Z}_1 \in \mathbb{R}^{N\times 3P^2}$ and $\hat{Z}_t \in \mathbb{R}^{N\times 3P^2}$) and the search sequence $\hat{X}_{t+1} \in \mathbb{R}^{N\times 3P^2}$, where $P \times P$ is the patch resolution and $N = HW/P^2$ is the number of patches. After applying the linear projection $\mathbf{E} \in \mathbb{R}^{3P^2\times C}$ to convert the 2D patches to 1D tokens with C dimensions and adding the sinusoidal positional embedding [11], $\mathbf{P} \in \mathbb{R}^{N\times C}$, we get the memory embeddings, $\mathbf{H}_{z,1} \in \mathbb{R}^{N\times C}$ and $\mathbf{H}_{z,t} \in \mathbb{R}^{N\times C}$, and the search embeddings $\mathbf{H}_{x,t+1} \in \mathbb{R}^{N\times C}$.

Encoding mask annotation. To encode the mask annotation in the joint feature extraction and interaction, we use a linear projection $\mathbf{E}_m \in \mathbb{R}^{P^2\times C}$ to convert the 2D mask map to mask embeddings, which can be alternatively regarded as target-aware positional embeddings. Following the image sequence generation, the 2D mask maps M_1 and M_t are firstly flattened into sequences, i.e., $\hat{M}_1, \hat{M}_t \in \mathbb{R}^{N\times P^2}$. Next, the mask annotation are incorporated into the input memory embeddings,

$$\bar{\mathbf{H}}_{z,1} = \hat{M}_1\mathbf{E}_m + \mathbf{H}_{z,1}, \quad (1)$$

$$\bar{\mathbf{H}}_{z,t} = \hat{M}_t\mathbf{E}_m + \mathbf{H}_{z,t}. \quad (2)$$

The obtained memory embeddings and search embeddings are concatenated together to form the input embeddings $\mathbf{H}^0 = [\bar{\mathbf{H}}_{z,1}; \bar{\mathbf{H}}_{z,t}; \mathbf{H}_{x,t+1}]$ to the vanilla ViT for joint feature extraction and interaction.

Joint feature extraction and matching. Previous VOS approaches extract the features of memory and search frames firstly, and then employ a manually-designed matching layer to attend the memory features to the search features for the final mask prediction. In SimVOS, this feature extraction and matching step can be simultaneously implemented in a more elegant and general way via the multi-head self-attention used in the vanilla ViT.

Mask Prediction. The updated search embedding $\bar{\mathbf{H}}_{x,t+1}$ output from the last layer of ViT is further reshaped to a 2D feature map. Following the previous approach STM [30], we use the same decoder that consists of several convolutional and deconvolutional layers for the final mask prediction. Since the decoder requires multi-resolution inputs, $\bar{\mathbf{H}}_{x,t+1}$ is firstly upsampled to 2× and 4× sizes via the deconvolution-based upsampling modules used in [30].

3.2. Speed-up Strategies

Despite the favorable segmentation results achieved by the proposed basic SimVOS, the computational and memory complexity for multi-head attention on the long sequence input $\mathbf{H}^0 \in \mathbb{R}^{3N\times C}$ can be very high when the frame resolution is large. To reduce the computational cost, we explore multiple speed-up strategies including within-frame attention and token refinement for foreground and background prototype generation. Fig. 3 illustrates a scalable ViT backbone with our speed-up strategies.

Within-frame attention. In the vanilla ViT, each query token globally attends to all other tokens, thus leading to quadratic complexity. However, it may be less necessary to perform the global token interaction in the early layers since the shallow features mainly focus more on the local structures instead of catching the long-range dependency. Therefore, for each query token, we restrict token attendance to only those within the same frame, and refer to this as *within-frame attention*. By applying the within-frame attention, the complexity of the computation in a specific layer reduces from $\mathcal{O}(9N^2)$ to $\mathcal{O}(N^2)$. In practice, within-frame attention is used in the first L layers of ViT.

Token refinement module. The within-frame attention reduces the overall complexity in the first L layers of ViT. However, the global self-attention used in the rest of the layers in ViT still causes the large quadratic complexity. Continuing to perform within-frame attention for deep layers

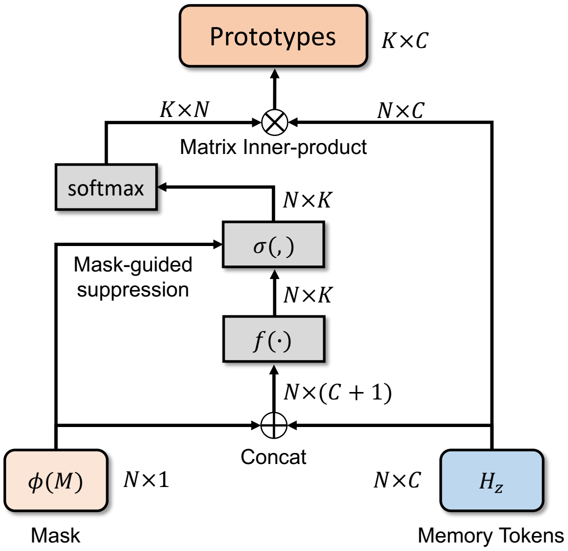

Figure 4: Detailed pipeline of the token refinement module, which effectively generates a small number of foreground and background prototypes guided by mask distribution.

alleviates this issue but also causes significant performance degradation, which is illustrated in Table 2. To address the aforementioned issues, we propose a token refinement module to further reduce the number of tokens in the memory embeddings, thus leading to a significant reduction of computational cost in the global self-attention layers.

Recent advances [18, 36, 13] in token reduction for efficient image classification show that the tokens can be effectively clustered by using a learnable convolutional module. In our work, given a memory embedding $\mathbf{H}_z \in \mathbb{R}^{N\times C}$ extracted from the first L layers of ViT using within-frame attention. our goal is to cluster or segment $\mathbf{H}_z$ into several foreground and background prototypes, in order to reduce the overall complexity in the following global self-attention layers. This can be achieved by proposing a learnable token refinement module for prototype generation guided by the segmentation mask M. Specifically, the generation of foreground prototypes can be formulated as:

$$W = f([\mathbf{H}_z, \phi(M)]) \in \mathbb{R}^{N\times K}, \tag{3}$$

$$\hat{W} = \text{softmax}(\sigma(W, \phi(M))) \in \mathbb{R}^{N\times K}, \tag{4}$$

where K is the number of generated prototypes. $[,]$ is the concatenation operation at the channel dimension, the softmax function is applied over each column of the input 2D matrix $\in \mathbb{R}^{N\times K}$, and $\phi()$ denotes a downsampling operation to reshape the mask M in order to meet the same spatial size with $\mathbf{H}_z$, i.e., $\phi(M) \in \mathbb{R}^{N\times 1}$. $f(\cdot)$ is a clustering function, which is implemented as a neural network module that consists of several convolutional layers and a fully-connected layer, (see supplementary for details). Inspired by [18, 36], the convolutional layers are firstly employed to map the high-dimensional input features to lower feature dimension, and the fully-connected layer predicts a prototype assignment matrix $\hat{W}$ in order to map the original features to K latent prototypes. We also use a post-processing function $\sigma(\cdot)$ to suppress weights at non-target locations in W, which is achieved by setting the corresponding rows in W to negative infinity, such that these elements can be suppressed after applying the softmax function. Finally, the generated prototypes are generated as:

$$\mathbf{h}_z = \hat{W}^T \mathbf{H}_z \in \mathbb{R}^{K\times C}. \tag{5}$$

For the background prototype generation, we simply replace the mask M used in (3) and (4) with the reverse mask i.e., $1 - M$. For clarification, we denote the foreground and background prototypes as $\mathbf{h}_z^f \in \mathbb{R}^{K_f\times C}$ and $\mathbf{h}_z^b \in \mathbb{R}^{K_b\times C}$ respectively, where K_f and K_b indicate their corresponding number of generated prototypes. As shown in Fig. 3, we then feed the concatenation of $\mathbf{h}_z^f$, $\mathbf{h}_z^b$ and the search tokens $\mathbf{H}_{x,t+1}$ to the remaining layers of ViT for global self-attention calculation.

In Table 3, we explore multiple settings of these two hyper-parameters K_f and K_b, which can be set to relatively small numbers without degrading performance much. Since $K_f + K_b \ll N$, the overall complexity is further reduced in the global self-attention layers based on the proposed token refinement module. For example, given a memory frame with the size of 480×960, there are 1800 tokens in total when $P = 16$. Based on our method, only 512 prototypes are generated ($K_f = K_b = 256$), which is about 3.5 times less than the original variant and achieves a better speed-accuracy trade-off.

We show the overall pipeline of our token refinement module in Fig. 4 and further visualize the assignment probability matrix $\hat{W}$ for both foreground and background prototype generation in Fig. 6. Interestingly, we find that the token refinement (TR) module aims to aggregate boundary features for prototype generation. This observation provides a potential explanation on which kinds of spatial features are more beneficial for online matching in video object segmentation. It shows that the boundary structures provide more useful cues for accurate online segmentation.

3.3. Training and Inference

Training on video datasets. Previous approaches [30, 8, 7, 57] commonly adopt two-stage or three-stage training including synthetic data pre-training using static image datasets [25, 10, 16, 19, 38], BL30K [6] pre-training and main training on video datasets. In this work, we observe that the proposed SimVOS can be well learned by only using the single stage of main training on video datasets (e.g., DAVIS2017 [33] and YouTube-VOS 2019 [54]), which fur-

Pre-trained Method	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
Random	68.8	66.3	71.3
ImageNet1K [41]	81.3	78.8	83.8
ImageNet22K [35]	83.5	80.5	86.6
MoCo-V3 [20]	81.5	79.0	83.9
MAE [21]	**88.0**	**85.0**	**91.0**

Table 1: The ablation study on the DAVIS-2017 val set using various pre-trained models for SimVOS with ViT-Base backbone.

Backbone	L	TR	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	FPS
ViT-Base	0		88.0%	85.0%	91.0%	4.9
ViT-Large	0		88.5%	85.4%	91.5%	2.0
ViT-Base	2		87.4%	84.4%	90.4%	5.7
ViT-Base	2	✓	86.0%	83.2%	88.9%	9.4
ViT-Base	4		86.9%	84.0%	89.8%	6.5
ViT-Base	4	✓	87.1%	84.1%	90.1%	9.9
ViT-Base	6		86.8%	83.7%	89.9%	7.6
ViT-Base	6	✓	86.5%	83.4%	89.5%	10.7
ViT-Base	8		86.5%	83.6%	89.4%	8.8
ViT-Base	8	✓	86.0%	83.0%	89.1%	11.4

Table 2: The performance of SimVOS variants on the DAVIS-2017 validation set. L denotes the number of layers using within-frame attention. TR indicates the usage of the token refinement module, where the default numbers of generated foreground/background prototypes are 384/384.

ther simplifies the training process. Specifically, we randomly sample two frames in a video clip with a predefined maximum frame gap (i.e., 10), in order to construct the template and search frames during the training. Following the convention [30, 8], the same bootstrapped cross entropy loss is used for supervision.

Online inference. During the online inference, we use the first frame and the predicted previous frame as the memory frames. The overall pipeline is shown in Fig. 2. There are no additional online adaptation or fine-tuning steps used.

4. Implementation Details

Evaluation metric. We use the official evaluation metrics, $\mathcal{J}$ and $\mathcal{F}$ scores, to evaluate our method. Note $\mathcal{J}$ is calculated as the average IoU between the prediction and ground-truth masks. $\mathcal{F}$ measures the boundary similarity measure between the prediction and ground-truth. The $\mathcal{J}\&\mathcal{F}$ score is the average of the above two metrics.

Training and evaluation. The proposed SimVOS is evaluated on multiple VOS benchmarks: DAVIS-2016 [32], DAVIS-2017 [33] and YouTube-VOS 2019 [54]. For a fair comparison with previous works [30, 57], we train our method on the training set of YouTube-VOS 2019 for YouTube-VOS evaluation. For DAVIS evaluation, we train SimVOS on both DAVIS-2017 and YouTube-VOS 2019. In the evaluation stage, we use the default 480P 24 FPS videos for DAVIS and 6 FPS videos for YouTube-VOS 2019 on an NVIDIA A100 GPU.

K_f	K_b	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	FPS
384	**384**	87.1%	84.1%	90.1%	9.9
256	256	86.7%	83.6%	89.8%	11.4
128	128	85.3%	82.1%	88.4%	13.5
512	256	86.0%	83.0%	89.1%	9.9
256	512	86.6%	83.7%	89.5%	9.9

Table 3: The ablation study on the number of generated foreground (K_f) and background (K_b) prototypes used in the TR module of our SimVOS, which employs the ViT-Base backbone and uses the first L=4 layers for within-frame attention. The default prototype number used in SimVOS is shown in bold.

Variant	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
w/o $\sigma(,)$	84.9	81.9	88.0
w/ $\sigma(,)$	87.1	84.1	90.1

Table 4: The ablation study on the DAVIS-2017 val set w/ and w/o using the post-processing function $\sigma(,)$ in Eq. (4).

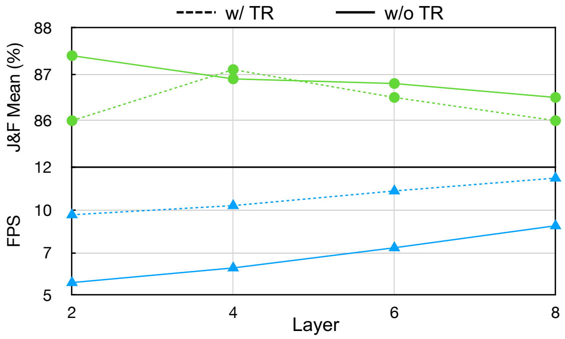

Figure 5: The plot of speed and $\mathcal{J}\&\mathcal{F}$ score versus layer index L for within-frame atttention. The dashed and solid lines represent the variant w/ and w/o applying the TR module, respectively.

We use randomly cropped 384×384 patches from video frames for training. We use the mini-batch size of 32 with a learning rate of $2\text{e}{-5}$. The number of training iterations is set to $210,000$ and the learning rate is decayed to $2\text{e}{-4}$ at half the iterations. The predefined maximum frame gap is set to a fixed value, i.e., 10, without using the curriculum learning schedule as described in [30, 8, 57] for simplicity. More training details can be found in the supplementary.

5. Experiments

In this section, we conduct ablation studies, state-of-the-art comparison and qualitative visualization to demonstrate the effectiveness of our proposed SimVOS.

5.1. Ablation Study

Large-scale pre-trained models. We test several popular large-scale pre-trained models for initializing the ViT-Base backbone used in our SimVOS. As can be seen in Table

Method	OL	S	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
OSVOS-S [55]	✓	✓	68.0	64.7	71.3
OSVOS [2]	✓		60.3	56.6	63.9
OnAVOS [43]	✓		65.4	61.6	69.1
CINM [1]	✓		70.6	67.2	74.0
AFB-URR [24]		✓	76.9	74.4	79.3
STM [30]		✓	81.8	79.2	84.3
RMNet [53]		✓	83.5	81.0	86.0
HMMN [37]		✓	84.7	81.9	87.5
MiVOS* [6]		✓	84.5	81.7	87.4
STCN [8]		✓	85.4	82.2	88.6
STCN* [8]		✓	85.3	82.0	88.6
AOT [57]		✓	84.9	82.3	87.5
XMEM [7]		✓	86.2	82.9	89.5
XMEM* [7]		✓	87.7	84.0	$\underline{91.4}$
CFBI [56]			81.9	79.3	84.5
JOINT [27]			83.5	80.8	86.2
SSTVOS [14]			82.5	79.9	85.1
FAVOS [9]			58.2	54.6	61.8
STCN$^-$ [8]			82.5	79.3	85.7
XMEM$^-$ [7]			84.5	81.4	87.6
SimVOS-BS			87.1	84.1	90.1
SimVOS-B			$\underline{88.0}$	$\underline{85.0}$	91.0
SimVOS-L			**88.5**	**85.4**	**91.5**

Table 5: Comparisons with previous approaches on the DAVIS-2017 validation set. OL and S represent the online learning and synthetic data pre-training. $*$ denotes the BL30K [6] pre-training. $-$ means to remove synthetic data pre-training.

Method	OL	S	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
OSVOS [2]	✓		80.2	79.8	80.6
OnAVOS [43]	✓		85.7	-	-
CINM [1]	✓		84.2	-	-
STM [30]		✓	89.3	88.7	89.9
RMNet [53]		✓	88.8	88.9	88.7
HMMN [37]		✓	90.8	89.6	92.0
MiVOS* [6]		✓	91.0	89.6	92.4
STCN [8]		✓	91.6	90.8	92.5
STCN* [8]		✓	91.7	90.4	93.0
AOT [57]		✓	91.1	90.1	92.1
XMEM [7]		✓	91.5	90.4	92.7
XMEM* [7]		✓	92.0	90.7	93.2
FAVOS [9]			-	82.4	79.5
CFBI [56]			89.4	88.3	90.5
XMEM$^-$ [7]			90.8	89.6	91.9
SimVOS-BS			91.5	89.9	93.1
SimVOS-B			$\underline{92.9}$	$\underline{91.3}$	$\underline{94.4}$
SimVOS-L			**93.6**	**92.0**	**95.3**

Table 6: Comparisons with previous approaches on the DAVIS-16 validation set. OL and S represent the online learning and synthetic data pre-training. $*$ denotes the BL30K [6] pretraining. $-$ means without applying synthetic data pre-training.

1, the generative MAE [21] model is the optimal choice compared with the other pre-trained models, including the supervised ImageNet1K [41], ImageNet22K [35] and the contrastive learning-based approach MoCoV3 [20]. This is mainly because fine-grained local structures are learned in MAE, which is more beneficial for the pixel-wise VOS task. Training with random initialization severely degrades the performance, which indicates that the number of training videos in DAVIS and YouTube-VOS is not sufficient enough for learning a robust VOS model. Based on these observations, we use the MAE pre-trained model as the default initialization for our SimVOS.

Within-frame attention. The within-frame attention (i.e., $L = [2, 4, 6, 8]$) can reduce the overall computational cost and improve the inference speed. As shown in Table 2 and Fig. 5, when the number of within-frame attention layers L is increased, the variants are more efficient but also suffer from some performance degradation, due to the insufficient interaction between the memory and search tokens. Considering the performance-speed trade-off, we use $L = 4$ as our default setting for further token refinement.

Token refinement. The token refinement (TR) module can further reduce the overall complexity in the global multi-head self-attention layers of SimVOS. There are several observations in Table 2 and Fig. 5: 1) the TR module applied in the early layer (e.g., $L = 2$) may cause insufficient memory token encoding, thus leading to large performance drop; 2) When $L \geq 4$, the TR module improves the inference speed, and achieves comparable results with the baseline.

Number of prototypes. The number of generated foreground or background prototypes may affect the overall performance of SimVOS. We conduct this ablation study in Table 3. As can be seen, severely decreasing the prototype number from 384 to 128 causes a relatively large drop of 1.8% $\mathcal{J}\&\mathcal{F}$. This shows that the TR module needs enough prototypes (e.g., 384) to represent large foreground or background regions in a video frame.

Ratio of foreground and background prototypes. We fix the total number (i.e., 768) of foreground and background prototypes, and test different ratios (i.e., 1:1, 2:1, and 1:2) of these two types of prototypes in Table 3. We find that the variant with balanced foreground and background prototypes achieves the best performance, which is because the foreground and background prototypes are all essential for accurate VOS in future frames of a test video.

Impact of $\sigma(,)$. In Table 4, we study the impact of with (w/) or without (w/o) the usage of mask map for foreground or background prototype generation. For the variant w/o using the mask map, we remove the post-processing function $\sigma(,)$ in Eq. 4 and directly use W for prototype generation.

Method	S	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
RMNet [53]	✓	75.0	71.9	78.1
STCN [8]	✓	76.1	73.1	80.0
STCN* [8]	✓	79.9	76.3	83.5
AOT [57]	✓	79.6	75.9	83.3
MiVOS* [6]	✓	78.6	74.9	82.2
XMEM [7]	✓	81.0	77.4	84.5
XMEM* [7]	✓	81.2	77.6	84.7
CFBI [56]		75.0	71.4	78.7
CFBI+ [58]		78.0	74.4	81.6
XMEM$^-$ [7]		79.8	76.3	83.4
SimVOS-BS		79.3	75.1	83.6
SimVOS-B		80.4	76.1	84.6
SimVOS-L		**82.3**	**78.7**	**85.8**

Table 7: Comparisons with previous approaches on the DAVIS-2017 test-dev. S indicates the usage of synthetic data pre-training. $*$ denotes the BL30K [6] pre-training. $-$ means without applying synthetic data pre-training. We use 480p videos for evaluation.

Method	S	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}_{seen}\uparrow$	$\mathcal{J}_{unseen}\uparrow$
MiVOS* [6]	✓	82.4	80.6	78.1
HMMN [37]	✓	82.5	81.7	77.3
STCN [8]	✓	82.7	81.1	78.2
STCN* [8]	✓	84.2	82.6	79.4
SwinB-AOT [57]	✓	84.5	84.0	78.4
XMEM [7]	✓	85.5	84.3	80.3
XMEM* [7]	✓	85.8	84.8	80.3
CFBI [56]		81.0	80.6	75.2
CFBI+ [58]		82.6	81.7	77.1
JOINT [27]		82.8	80.8	79.0
SSTVOS [14]		81.8	80.9	76.7
XMEM$^-$ [7]		**84.2**	**83.8**	78.1
SimVOS-B		**84.2**	83.1	**79.1**

Table 8: Comparisons with previous approaches on the YouTube-VOS 2019 validation set. S indicates the usage of synthetic data pre-training. $*$ denotes the BL30K [6] pretraining. $-$ means without applying synthetic data pre-training.

As we can see, w/o the usage of mask map, the generated prototypes are low quality, which degrades the performance with a margin of xx in terms of the $\mathcal{J}\&\mathcal{F}$ metric. This is mainly because this arbitrary generation may fix both foreground and background regions, thus generating ambiguous prototypes and causing more tracking failures.

5.2. State-of-the-art Comparison

In this section, we compare multiple variants of our SimVOS with state-of-the-art VOS approaches. Specifically, the variants employ the VIT-Base or ViT-Large as the backbone, and do not use within-frame attention or the TR module, which are respectively denoted as **SimVOS-B** and **SimVOS-L**. We also include a full-version variant that employs the ViT-Base backbone and all the speed-up strategies including the within-frame attention and the TR module, denoted as **SimVOS-BS**.

DAVIS-2017. DAVIS-2017 [33] is a typical VOS benchmark which has been widely for state-of-the-art comparisons in the VOS community. This dataset consists of 150 sequences and 376 annotated objects in total. The validation set of DAVIS-2017 contains 30 challenging videos and uses a multi-object setting, where multiple annotated objects in the initial video frame are required to track and segment in the following frames. The test set contains more challenging 30 videos for evaluation.

The comparison between our SimVOS and previous approaches on the DAVIS-2017 validation set is shown in Table 5. Without applying online learning and synthetic data pre-training, our SimVOS-B and SimVOS-L set new state-of-the-art $\mathcal{J}\&\mathcal{F}$ scores, i.e., 88.0% and 88.5%, which are even better than XMEM* that applies both BL30K and synthetic data pre-training. The results on DAVIS-2017 test-dev is shown in Table 7, our SimVOS-BS achieves comparable results to XMEM$^-$ that employs more memory frames. The SimVOS-L variant achieves 82.3% $\mathcal{J}\&\mathcal{F}$ score, which is the leading performance on this dataset by using 480P videos for evaluation. We believe the strong performance of SimVOS can be attributed to two main aspects: 1) the generative MAE initialization, and 2) the ViT backbone is suitable for memory and search interaction.

DAVIS-2016. DAVIS-2016 is a subset of DAVIS-2017 and it follows a single-object setting. For completeness, we also compare SimVOS with state-of-the-art approaches on the DAVIS-2016 validation set, which is illustrated in Table 6. The $\mathcal{F}$ scores achieved by our SimVOS-F, SimVOS-B and SimVOS-L are 93.1%, 94.4% and 95.3%, respectively. These results are significantly better than the previous SOTA approaches under the same training settings.

YouTube-VOS 2019. YouTube-VOS 2019 [54] is a large-scale VOS benchmark that consists of 507 validation videos for evaluation. In Table 8, we show that our SimVOS performs favorably against state-of-the-art approaches on the YouTube-VOS 2019 validation set. Note that our SimVOS-B is evaluated at the default 6 FPS videos, but still achieves comparable performance with XMEM$^-$ that uses all the frames for evaluation. Moreover, the $\mathcal{J}_{unseen}$ score of SimVOS-B is 79.1%, outperforming the others under the same setting (i.e., w/o synthetic data pre-training), which shows our method can generalize well to unseen objects during the testing. More results on YouTube-VOS 2019 and qualitative visualization are included in the supplementary.

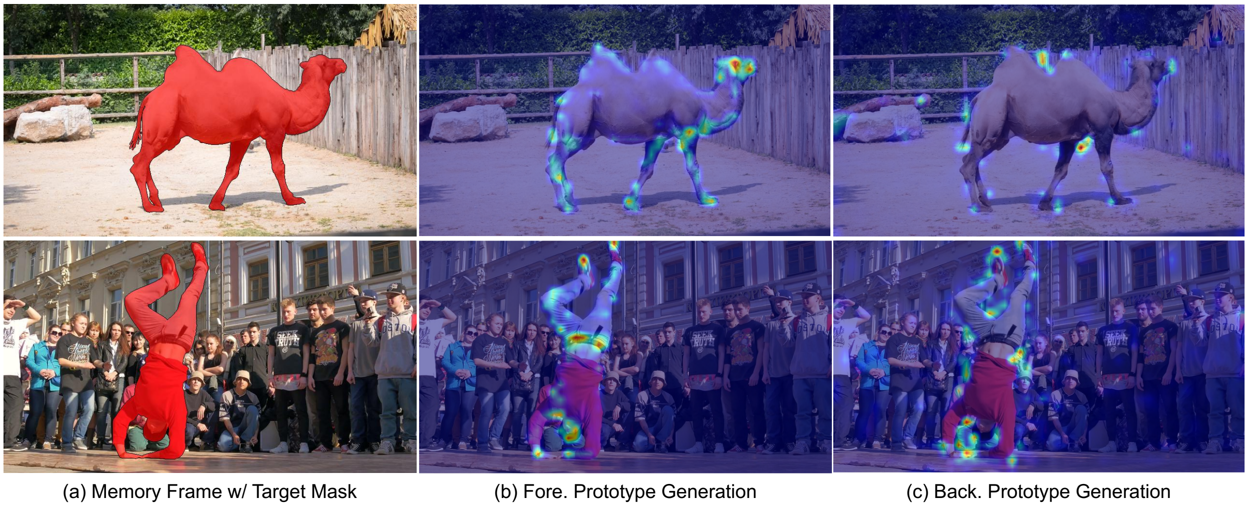
(a) Memory Frame w/ Target Mask
(b) Fore. Prototype Generation
(c) Back. Prototype Generation

Figure 6: Given the (a) memory frame w/ target mask, the visualization of assignment matrix $\hat{W} \in \mathbb{R}^{N \times K}$ (Eq. 4) for both (b) foreground and (c) background prototype generation. $\hat{W} \in$ is averaged over each row and then upsampled to the image size for visualization. The TR module tends to aggregate boundary features to generate prototypes for accurate online VOS. More visualization is shown in supplementary.

6. Conclusion

In this work, we present a scalable video object segmentation approach with simplified frameworks, called SimVOS. Our SimVOS removes hand-crafted designs (e.g., the matching layer) used in previous approaches and employs a single transformer backbone for joint feature extraction and matching. We show that SimVOS can greatly benefit from existing large-scale self-supervised pre-trained models (e.g., MAE) and can be served as a simple yet effective baseline for developing self-supervised pre-training tasks in VOS. Moreover, a new token refinement module is proposed to further reduce the computational cost and increase the inference speed of SimVOS. The proposed SimVOS achieves state-of-the-art performance on existing popular VOS benchmarks, and the simple design can inspire and serve as a baseline for future ViT-based VOS.

7. 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), and a Strategic Research Grant from City University of Hong Kong (Project No. 7005840).

References

[1] L. Bao, B. Wu, and W. Liu. Cnn in mrf: video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, 2018.

[2] S. Caelles, K.K. Maninis, J. Pont-Tuset, L. Leal-Taixé, and L. Van Gool. One-shot video object segmentation. In *CVPR*, pages 221–230, 2017.

[3] 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.

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

[5] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, pages 1189–1198, 2018.

[6] H.K. Cheng, Y.W. Tau, and C.K. Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *CVPR*, 2021.

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

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

[9] 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.

[10] M.M. Cheng, N.J. Mitra, X. Huang, P.H. Torr, and S.M. Hu. Global contrast based salient region detection. In *TPAMI*, 2014.

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

[12] Z. Dou, C. Lin, R. Xu, L. Yang, S. Xin, T. Komura, and W. Wang. Coverage axis: Inner point selection for 3d shape skeletonization. In *Computer Graphics Forum*, 2022.

[13] Z. Dou, Q. Wu, C. Lin, Z. Cao, Q. Wu, W. Wan, T. Komura, and W. Wang. Tore: Token reduction for efficient human mesh recovery with transformer. In *ICCV*, 2023.

[14] B. Duke, A. Ahmed, C. Wolf, and G. W. Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *CVPR*, pages 5912–5921, 2021.

[15] N. Dvornik, K. Shmelkov, and J. Mairal. Blitznet: A real-time deep network for scene understanding. In *ICCV*, 2017.

[16] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. In *IJCV*, 2010.

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

[18] R. Grainger, T. Paniagua, and X. Song. Learning patch-to-cluster attention in vision transformer. In *arXiv:2203.11987*, 2022.

[19] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Training data-efficient image transformers distillation through attention. In *ICCV*, 2011.

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

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

[22] Li-Jia Li, Richard Socher, and Fei-Fei Li. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.

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

[24] Y. Liang, X. Li, N. Jafari, and J. Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In *NeurIPS*, 2020.

[25] 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.

[26] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *CVPR*, pages 565–580, 2018.

[27] Y. Mao, N. Wang, W. Zhao, and H. Li. Joint inductive and transductive learning for video object segmentation. In *ICCV*, 2021.

[28] Y. Xia nad Y. Xu, S. Li, R. Wang, J. Du, D. Cremers, and U. Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *CVPR*, 2021.

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

[30] S.W. Oh, J.Y. Lee, N. Xu, and S.J. Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[31] F. Perazzi, A. Khoreva, R. Benenson, and B. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, pages 2663–2672, 2017.

[32] 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.

[33] 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.

[34] 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.

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

[36] M. Ryoo, A. J. Piergiovanni, and A. Arnab. Tokenlearner: Adaptive space-time tokenization for videos. In *NIPS*, 2021.

[37] H. Seeing, S.W. Oh, J.Y. Lee, S. Lee, S. Lee, and E. Kim. Hierarchical memory matching network for video object segmentation. In *ICCV*, 2021.

[38] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical image saliency detection on extended cssd. In *TPAMI*, 2015.

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

[40] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *arXiv:1906.05849*, 2019.

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

[42] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, and L.C. Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, pages 9481–9490, 2019.

[43] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017.

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

[45] J. Wan, Q. Wu, and AB Chan. Modeling noisy annotations for point-wise supervision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[46] Q. WU, J. Wan, and AB Chan. Dynamic momentum adaptation for zero-shot cross-domain crowd counting. In *The 29th ACM international conference on Multimedia*, 2021.

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

[48] 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.

[49] Q. Wu, T. Yang, Z. Liu, B. Wu, Y. Shan, and Antoni B. Chan. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In *CVPR*, 2023.

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

[51] Y. Xia, Q. Wu, W. Li, AB Chan, and U. Stilla. A lightweight and detector-free 3d single object tracker on point clouds. In *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[52] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, pages 1140–1148, 2018.

[53] H. Xie, H. Yao, S. Zhou, S. Zhang, and W. Sun. Efficient regional memory network for video object segmentation. In *CVPR*, 2021.

[54] 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.

[55] L. Yang, Y. Wang, and X. Xiong. Efficient video object segmentation via network modulation. In *CVPR*, pages 6499–6507, 2018.

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

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

[58] Z. Yang, Y. Wei, and Y. Yang. Collaborative video object segmentation by multi-scale foreground-background integration. In *TPAMI*, 2021.

[59] 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.

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