Self-supervised Video Object Segmentation with Distillation Learning of Deformable Attention

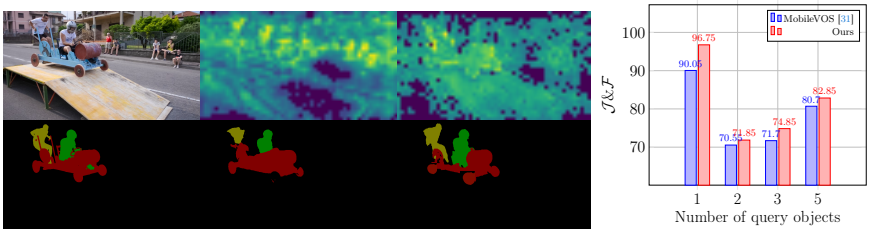
Quang-Trung Truong[1], Duc Thanh Nguyen[2], Binh-Son Hua[3], and Sai-Kit Yeung[1]

[1] Hong Kong University of Science and Technology
[2] Deakin University
[3] Trinity College Dublin
qttruong@connect.ust.hk

Abstract. Video object segmentation is a fundamental research problem in computer vision. Recent techniques have often applied attention mechanism to object representation learning from video sequences. However, due to temporal changes in the video data, attention maps may not well align with the objects of interest across video frames, causing accumulated errors in long-term video processing. In addition, existing techniques have utilised complex architectures, requiring highly computational complexity and hence limiting the ability to integrate video object segmentation into low-powered devices. To address these issues, we propose a new method for self-supervised video object segmentation based on distillation learning of deformable attention. Specifically, we devise a lightweight architecture for video object segmentation that is effectively adapted to temporal changes. This is enabled by deformable attention mechanism, where the keys and values capturing the memory of a video sequence in the attention module have flexible locations updated across frames. The learnt object representations are thus adaptive to both the spatial and temporal dimensions. We train the proposed architecture in a self-supervised fashion through a new knowledge distillation paradigm where deformable attention maps are integrated into the distillation loss. We qualitatively and quantitatively evaluate our method and compare it with existing methods on benchmark datasets including DAVIS 2016/2017 and YouTube-VOS 2018/2019. Experimental results verify the superiority of our method via its achieved state-of-the-art performance and optimal memory usage.

1 Introduction

Video object segmentation (VOS) is a fundamental task in computer vision, aiming to segregate object(s) of interest from a background across frames in a video sequence. The task has attracted considerable attention from the research community, resulting in various models developed in recent years [14]. In the perspective of deep learning, designing an architecture that can well learn features of an object of interest adaptively to temporal changes while maintaining optimal memory usage is still an open research problem. Literature has shown



(a) (b)

Fig. 1: (a) From left to right: input frame and ground-truth segmentation masks, distilled feature map and segmentation masks by the distillation strategy used in MobileVOS [31], distilled feature map and segmentation masks by our method. (b) Segmentation accuracy ($\mathcal{J}\&\mathcal{F}$) of MobileVOS [31] and our method with different numbers of query objects in DAVIS-17 val (see more details in Section 4).

a substantial body of work dedicated to developing deep learning models towards this goal [14]. Among these, the Vision Transformer (ViT) in [12] has been commonly adopted in recent VOS research, and made significant progress. Examples include the works in [13,16,49,51,54]. The reason for this success is the ability of the attention mechanism in the ViT in object representation learning. Specifically, unlike convolutional neural networks (CNNs) which obtain a global receptive field for an object by a pooling operator [39], the ViT captures global context via self-attention layers.

Despite such progress, there still exist issues in the current research. First, we found that attention layers are not well adapted to temporal changes, causing accumulated errors in processing of long-term video sequences. To mitigate this issue, several methods, e.g., [43, 52], have utilised optical flows in the attention module and achieved promising results. Additional motion information from optical flows is a useful guideline to define an object in the query of the attention module. In particular, optical flows within the same object should be smooth while flows across the object boundaries should be disruptive. Similarly, if an object moves differently from the background, the motion boundaries would be indicative of the object boundaries. Hence, optical flows would facilitate precise locating of object boundaries and vice versa. However, these methods require an accurate motion estimation model to be given in advance. Unfortunately, this requirement is not always fulfilled, especially when VOS is applied to challenging scenarios such as underwater applications.

Second, another major challenge in VOS is object forgetting in long-term video processing. The issue gets more critical in segmenting objects under severe occlusion. Several methods have been developed to tackle this challenge, e.g., [30, 34]. For instance, Park et al. [34] indicated that memory updates in short-term intervals with several frames, also known as clip-wise mask propagation, are more powerful than updates with a nearby frame. However, one has to deal with clip-level optimisation and parallel computation of multiple frames.

Third, existing methods are computationally expensive, limiting their applicability to low-powered devices. In particular, the computational complexity of ViT-based methods grows quadratically with their token length. The excessive

number of keys to attend per query patch yields high computational cost and slow convergence, increasing the risk of over-fitting. Recently, MobileVOS [31] applied knowledge distillation to create a lightweight VOS model. The core idea of their distillation strategy is to constrain the similarity between distilled feature maps in a teacher and a student network via the similarity between their correlation matrices. However, we found such an approach is still affected by fast motion (under sudden changes due to fast motion, correlation matrices in distilled layers may be significantly diverse). We illustrate this issue in Fig. 1.

In this paper, we propose a VOS method to address these aforementioned issues. Specifically, we make the following contributions in our work.

- We propose a deformable attention module for VOS to improve attention learning such that learnt attention maps are adaptive to both spatial and temporal changes. Our idea is motivated by Deformable Convolution Networks [10], that learn a deformable receptive field for each convolution filter. We found such a learning approach is applicable to learning of attention maps and can be effective for driving attention scores to more informative regions, considering both the spatial and temporal dimensions. Here, we devise such a deformable attention-like pattern for VOS, where the positions of the key and value in the attention layer are not fixed but can be optimised from data.
- We propose a lightweight architecture for VOS that can be trained using self-supervised learning. The learning process aims to transfer object representations learnt from a large model with full access to ground-truth labels to a smaller one with pseudo labels. We formulate this tranfer learning process as knowledge distillation (KD). However, unlike existing KD methods which constrain only the consistency of the logits produced by the teacher and student networks, we further constrain intermediate attention maps in both the networks.
- We prove the robustness of our method via extensive experiments on every aspect of its designs. In particular, we rigorously validate the core components including deformable attention, distillation learning of attention maps. We investigate various loss functions. We examine the distillation at different layers. We also compare our method with existing ones on bechmark datasets including YouTube-VOS 2019/2018 and DAVIS 2017/2016. Experimental results confirm the superiority of our method, showing its state-of-the-art performance and optimal memory usage over the baselines.

2 Related work

Online-learning vs offline-learning. Existing VOS methods can be categorised into online-learning methods or offline-learning methods, depending on how they train their model to segment a target object. Online-learning methods [2,17,35] perform fine-tuning of a VOS model during the testing phase to incorporate specific information about the target object. Despite promising results, these methods are often experienced with over-fitting, i.e., they can learn the target

object very well from the first frame, but fail to segment it in following frames. In addition, these methods are not practical for real-time applications as re-training of a model for a new object is time consuming. On the other hand, offline-learning methods [8, 26] aim to train a network that can work on any video without the need of re-training to adapt the model with a new object during testing. Our method follows the offline-learning approach, where we formulate VOS as label propagation over time.

Self-supervised learning. This field receives the special attention of research community, recently. Self-supervised techniques utilise unlabeled videos with masks of the first frame given to identify query objects and then segment upcoming sequence frames in training. Generally, temporal correspondence learning is adopted to maintain temporal coherence in video segmentation methods such as Mining [19] and LIIR [24]. LIIR [24] belongs to temporal correspondence learning based on an additional video reconstruction. Inter-video and intra-video reconstruction scheme are used to formulate the contrast over inter-video and intra-video affinities to handle discriminating instances in the pixel-wise representation learning. Another family of temporal correspondence learning methods further execute both forward and backward tracking and penalize discrepancies between the initial and final positions of the considered pixels and regions, often known as cycle consistency based methods such as CorrFlow [21] and Self-cycle [41]. Self-cycle [41] addresses noisy labels caused by the path selection problem in constructing a graph from a video. It benefits from the cycle-based temporal correspondences and hard negative mining in multi-hop concurrent path consideration.

Another class of methods is mask correspondence learning. Recently, there are different mask-guided solutions aiming to generate a mask for self-supervised correspondence learning. Mask-VOS [23] creates pseudo-label data by adopting k-means clustering in order to enforce mask correspondence via a mask embedding scheme. Mask correspondence learning methods benefit from effective VOS methods belonging to semi-supervised learning that is trained under fully supervised fashion with real ground truth, e.g., DeAOTT [50]. MAST [22] shares the same with the matching-based method exploiting memory by calculating affinity matrix of query frames and reference frames. However, this framework is trained without manual annotation given. It is observed that the high-fidelity segmentation masks generated by semi-supervised VOS methods recently are effective to be exploited in self-supervised learning setting. The excellent matching-based architectures i.e., XMEM [7] or attention-based semi-VOS i.e., AOT [49], DeAOT [50] enable to produce discriminative attention maps shown the superiority in modelling space-time correspondences. However, there is the lack of self-supervised methods explicitly gaining the advantage of the excellent architectures to reduce performance discrepancies between supervised learning and self-supervised leaning VOS methods. To the best of our knowledge, we are the first, coming up with a solution to exploit explicitly available powerful semi-supervised models in self-supervised fashion. Specifically, mask correspondences are established between frames via attention paradigm DeAOT [50]. Only pseudo

labels are used in our proposed method with a simple but effective knowledge distillation framework to conduct attention transfer and logit transfer from a large teacher net to a student net for VOS.

Vision Transformers. Vision Transformer (ViT), a network architecture inspired by the Transformer in [44], has shown its ability in various computer vision tasks, e.g., image recognition [12], semantic segmentation [42], and object detection [3]. Such ability is enabled by attention mechanism [4], aiming to learn an attention map (score map) for every representation (e.g., a local image region) within a given context (e.g., an entire image). However, many areas in an attention map of a large-sized input may be dismissed during the training. To address this issue, several methods apply sliding window partition to the input data [1, 11, 29]. Dense attention in ViT is beneficial for learning of large receptive fields, but also incurs expensive memory usage and computational cost. To overcome this challenge, Xia et al. [46] proposed deformable attention, where the offsets of the keys and values in the self-attention module are not fixed in a regular grid, but determined from data. Similarly, Pan et al. [32] proposed slide-transformer, which allows location shifting of the key and value offsets. To shift the keys and values accordingly with depth-wise convolutions, the authors replaced the original column-based view to calculate the key and value matrices by a row-based view. There are methods combining both CNNs and ViTs. For instance, Xiao et al. [47] applied convolutions in early stages of a ViT to enhance the stability of the model training. CSwin Transformer proposed in [11] employed convolution-based positional encoding and demonstrated significant improvement. These convolution-based techniques have the potential to be applied in conjunction with deformable attention to further enhance performance. In this paper, we devise a deformable attention-like pattern for ViT-based VOS. In sharp contrast to our proposed architecture, [46] proposed a large deformable attention transformer architecture in which the number of parameters of its variants vary between 48M and 69M parameters. Specifically, the model [46] consists of four stages of attention for image segmentation and detection tasks. Multiple-head deformable attention block is placed in the two last stages and the first two stages contain local attention and shift-window attention. On the contrary, our single-head deformable attention is designed right after the encoder block. Our light-weight attention student is effective since attention distillation is leveraged to mimic teacher's discriminative attention maps and transfer to student net in data-driven manner.

Knowledge distillation. Knowledge distillation (KD) is a powerful machine learning technique that aims to transfer knowledge learnt from a large-sized model (teacher) to a smaller-sized one (student). KD has often been applied to self-supervised/weakly-supervised learning. For instance, Cheng et al. [9] adopted KD for instance segmentation where only box-level labels are available for training. Many recent KD frameworks [5, 18, 33, 55] have focused on design of loss functions or adapters. MobileVOS [31] proposed a distillation loss based on pixel-wise multiplication of correlation matrices of distilled feature maps. Unlike ex-

isting methods, in this paper, we propose a KD scheme for VOS training, where the knowledge transfer between the teacher and student architectures happens not only in logit layers but also in attention maps.

3 Proposed method

3.1 Overview

Our method aims at performing effective knowledge distillation (KD) for video object segmentation (VOS). We examine our method with DeAOT, the state-of-the-art VOS in [50]. Specifically, we opt DeAOTL as our teacher network as this is the largest model among all the variants of the DeAOT's family. In addition, we build our student network upon DeAOTT, the smallest model in this group. An overview of our knowledge distillation framework is shown in Fig. 2(a).

Both the teacher and student networks learn attention maps to share between two network branches: visual branch and ID branch. The visual branch aims to match objects by passing embeddings stored in memory across adjacent frames. The ID branch propagates object-specific knowledge learnt from past frames to the current frame to associate objects of the same ID across frames. In the teacher model, the shared attention maps are learnt by the Gated Propagation Module (GPM) [50]. We refer the readers to our supplementary material for the implementation details of the GPM. To improve the adaptivity of attention maps to temporal changes, we propose to replace the Gated Attention function, used in the GPM, by our Gated Deformable Attention function (see Fig. 2(b)) which is implemented via deformable attention (see Fig. 2(c)). We present the deformable attention in Section 3.2.

We apply KD in training of our VOS framework. We opt to distill the attention map and logit layer in the 3rd GPM of the ID branch from the teacher network to the student one. We describe our KD scheme in Section 3.3. Unlike existing KD methods which transfer logit layers from the teacher to student model, here we also transfer attention maps during the training phase. In particular, the teacher model first transfers intermediate attention maps to the student model. These attention maps are calculated using our proposed deformable attention module. We constrain the attention map transfer via a Centered Kernel Alignment (CKA)-based loss [20]. Probability distributions of the logits (i.e., output of the softmax layer) are then transferred via intra-object and inter-object losses.

3.2 Deformable attention module

Vanilla self-attention Since we focus on image-like data formats, for the convenience in presentation, we describe the vanilla self-attention for tensor-based inputs, e.g., a high-dimensional feature map $X \in \mathbb{R}^{H \times W \times C}$ where $H \times W$ represent the spatial dimensions and C represents the number of channels. A more general definition can be found from [44].

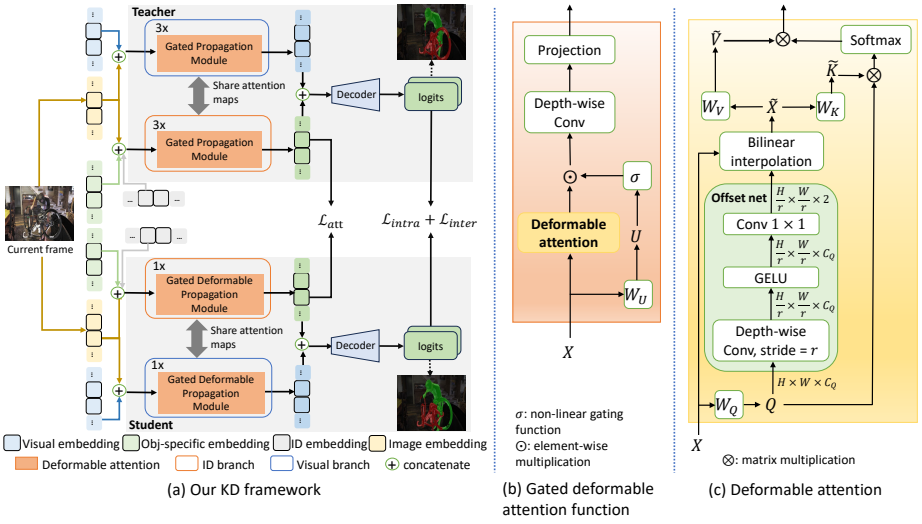


Fig. 2: Summary of our proposed VOS method. (a) Overview of our knowledge distillation method. The teacher model transfers intermediate attention maps to the student model. This transfer is enforced by a CKA-based loss $\mathcal{L}_{att}$. At the same time, probability distributions of logits are transferred using intra-object and inter-object losses $\mathcal{L}_{intra}$ and $\mathcal{L}_{inter}$. Both the teacher and student models make use of Gated Propagation Module (GPM) [50], aiming to propagate spatio-temporal information across frames via the attention mechanism. (b) Our proposed Gated Deformable Attention function, which is used to implement the GPM. (c) Deformable attention module, which is used to replace the vanilla attention in the Gated deformable attention function.

Given a feature map X, the query Q, key K, and value K in a single-head attention module are calculated as,

$$Q = W_q X, K = W_k X, V = W_v X \tag{1}$$

where W_q, W_k, and W_v contain learnable parameters.

The single-head self-attention transforms each query by calculating a weighted sum of values. The weights are computed by taking the dot product between the query and its corresponding keys, followed by a normalisation step as,

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V \tag{2}$$

where $\sqrt{d}$ is a scaling factor in the attention mechanism.

Deformable attention Inspired by Deformable Convolution Networks [10], deformable attention [46] allows the offsets of the keys and values in the self-attention mechanism flexible yet learnable from data. Particularly, deformable

attention calculates an attention map for an input feature map in 3 steps: 1) initialise reference points (locations for the keys and values), 2) generate offsets, 3) re-sample the features regarding to new reference points shifted the generated offsets.

Initialise reference points. Deformable attention uses a set of irregular points, called reference points, to locate the query and key in the given feature map X. Those reference points are initialised from a uniform grid $R = \{(0,0),..,(H_g-1,W_g-1)\}$ where $H_g = H/g$ and $W_g = W/g$, g is a grid-size factor. The reference points are then normalised into $[-1,1]$.

Generate offsets. The query Q is calculated as in Eq. (2) and then partitioned evenly along its feature channels. In particular, let $Q \in \mathbb{R}^{H\times W\times C_Q}$, Q is partitioned into S sub-feature maps $\{Q_i \in \mathbb{R}^{H\times W\times C_{Q_i}}\}_{i=1}^{S}$, where $\sum_i^S C_{Q_i} = C_Q$ and $C_{Q_i} = C_{Q_j}$, $\forall i,j \in \{1,...,S\}$.

Each sub-feature map Q_i is passed to an offset network $M(Q_i)$ to generate a set of offsets $\Delta_i = \{\delta_{i,r} \in \mathbb{R}^2 | \forall r \in R\} \in \mathbb{R}^{2\times H_g\times W_g}$. The offset network is a convolutional neural network consisting of 2 convolutional layers with GELU activation functions in between (see our supplementary material for the details). Finally, given Q, we can calculate a set of offsets $\Delta = \{\Delta_i\}_{i=1}^{S} \in \mathbb{R}^{2\times S\times H_g\times W_g}$.

Re-sample the features. We re-sample the features in X at new locations made by shifting the reference points with their offsets in Δ. In particular, let $\tilde{X}_i$ be the re-sampled feature map corresponding to the offsets Δ_i. Let $\mathbf{p}_r \in \mathbb{R}^2$ be a location relative to a reference point $r \in R$. We calculate $\tilde{X}_i(\mathbf{p}_r)$ using bilinear interpolation as follows,

$$\tilde{X}_i(\mathbf{p}_r) = \sum_{\mathbf{q}\in R} I(\mathbf{p}_r + \delta_{i,r}, \mathbf{q}) X(\mathbf{q}) \tag{3}$$

where I is defined as,

$$\begin{aligned} I(\mathbf{p},\mathbf{q}) = &\max(0, 1-|\mathbf{p}_x - \mathbf{q}_x|)\times \\ &\max(0, 1-|\mathbf{p}_y - \mathbf{q}_y|) \end{aligned} \tag{4}$$

where $\mathbf{p} = (\mathbf{p}_x, \mathbf{p}_y) \in \mathbb{R}^2$ and $\mathbf{q} = (\mathbf{q}_x, \mathbf{q}_y) \in \mathbb{R}^2$.

Intuitively, Eq. (3) re-samples $\tilde{X}_i(\mathbf{p}_r)$ based on the four discrete locations closest to $\mathbf{p}_r + \delta_{i,r}$. Next, we calculate deformable key $\tilde{K}$ and deformable value $\tilde{V}$ as,

$$\tilde{K} = W_k\tilde{X}, \tilde{V} = W_v\tilde{X} \tag{5}$$

Finally, a deformable attention map is achieved as,

$$\text{DefAtt}(Q,\tilde{K},\tilde{V}) = \text{softmax}\left(\frac{Q\tilde{K}^\top}{\sqrt{d}}\right)\tilde{V} \tag{6}$$

To prioritise important tokens in a video sequence, the deformable attention map is element-wise multiplied with a gating embedding as,

$$\text{GatedDefAtt}(Q,\tilde{K},\tilde{V},U) = \text{DefAtt}(Q,\tilde{K},\tilde{V}) \odot \sigma(U) \tag{7}$$

where $U = W_u X \in \mathbb{R}^{W \times H \times C_u}$ with W_u as a learnable parameter matrix, $\sigma(\cdot)$ is a non-linear gating function (e.g., SiLU/Swish [38,46]), and $\odot$ is an element-wise product.

3.3 Knowledge distillation

Although we expect the student model to be smaller in size and faster at inference, we also aim to make it strong in terms of performance. To achieve this, we propose a new knowledge distillation scheme that guides the student network to learn not only logits from the teacher network but also intermediate attention maps. In addition, to further strengthen the distillation process, we enforce relational matching between the predictions of the student and the teacher network. In particular, we apply the intra-object and inter-object relations in [18] in our distillation loss. These object-based relations fit well our purpose (object segmentation) and are proven to strengthen the prediction ability of our method against fast motion and deformable shapes.

Let F_T^k and F_S^l be attention maps in the teacher and student networks at encoder block k and l respectively. In our implementation, we distill the last attention map in the teacher network. We enforce the consistency between F_T^k and F_S^l during the distillation process. Feature distillation has often been performed via projectors [5, 28], where KL-divergence is utilised to reduce the discrepancy between corresponding layers in both the teacher and student models. However, we found that the KL-divergence loss is usually anisotropic. To effectively transfer attention maps between the teacher and the student networks, we define our loss based on the Centered Kernel Alignment (CKA) [20], which is proven isotropic with respect to all dimensions regardless of scales, and reliant only on the feature distributions. Here we summarise the main steps of CKA calculation and refer the readers to the work by [20] for more details. First, we apply a linear kernel to both F_T^k and F_S^l to obtain Gram matrices G_T^k and G_S^l. Let $H(G_T^k, G_S^l)$ be the Hilbert-Schmidt Independence Criterion-based metric [15] between the Gram matrices G_T^k and G_S^l. The CKA score between F_T^k and F_S^l is defined as,

$$\mathrm{CKA}(F_T^k, F_S^l) = \frac{H(G_T^k, G_S^l)}{\sqrt{H(G_T^k, G_S^l) \cdot H(G_T^k, G_S^l)}} \tag{8}$$

The loss for attention distillation is finally defined as,

$$\mathcal{L}_{att} = \sum_k \sum_l a_{k,l}(1 - \mathrm{CKA}(F_T^k, F_S^l)) \tag{9}$$

where $a_{k,l} = 1$ if F_T^k is transferred to F_S^l, and $a_{k,l} = 0$, otherwise.

Next, we present how to distill logit values. Let $\mathbf{Z}_S \in \mathbb{R}^{B \times N}$ and $\mathbf{Z}_T \in \mathbb{R}^{B \times N}$ be the logit values from the student and teacher model on a batch of B samples and N objects. Let $\mathbf{Y}_S \in \mathbb{R}^{B \times N}$ and $\mathbf{Y}_T \in \mathbb{R}^{B \times N}$ be the probability distributions over the N objects achieved from $\mathbf{Z}_S$ and $\mathbf{Z}_T$ using the softmax

operator, i.e., $\mathbf{Y}_{S/T}[i,:] = \text{softmax}(\mathbf{Y}_{S/T}[i,:])$, $i = 1, ..., B$. The distillation loss for the inter-object and intra-object relations are defined as,

$$\mathcal{L}_{inter} = \frac{1}{B}\sum_{i=1}^{B} d_p(\mathbf{Y}_S[i,:], \mathbf{Y}_T[i,:]) \tag{10}$$

$$\mathcal{L}_{intra} = \frac{1}{N}\sum_{j=1}^{N} d_p(\mathbf{Y}_S[:,j], \mathbf{Y}_T[:,j]) \tag{11}$$

where d_p is the Pearson's distance measuring the discorrelation between two probability distributions.

The final loss of our distillation method is calculated as,

$$\mathcal{L} = \mathcal{L}_{inter} + \mathcal{L}_{intra} + \lambda\mathcal{L}_{att} \tag{12}$$

where λ is a balancing factor.

4 Experiments

4.1 Datasets

We conducted our experiments on VOS benchmark datasets including DAVIS 2016 [36], DAVIS 2017 [37], YouTube-VOS 2018 and 2019 [48]. The DAVIS 2016 consists of video sequences with single objects of interest. This dataset has 30 videos for training and 20 videos for validation with high-quality ground truth segmentation for salient objects. The DAVIS 2017 is an improved version of the DAVIS 2016 with 60 videos for training and 30 videos for validation.

The YouTube-VOS [48] is a large-scale dataset for segmenting multiple objects. It has 3,471 videos for training, 474 and 507 videos for validation in the 2018 and 2019 version, respectively. The training set has 65 categories, and the validation set further includes 26 unseen categories.

4.2 Experimental setup

Implementation details We adopted the pre-trained DeAOTL model from [50] with ResNet50 backbone as the teacher network. We chose the DeAOTT also from [50] with MobileNet-V2 backbone as the student network. We set $\lambda = 1.5$ in Eq. (12).

We performed the distillation in a self-supervised fashion, i.e., no access to the ground-truth labels during the training of the student model. Specifically, we applied the teacher model to generate pseudo labels that are used to train the student model. Following the setting in [50], we first performed the distillation on synthetic video sequences generated from 28,732 images from BIG [6], DUTS [45], ECSSD [40], FSS-1000 [25], HRSOD [53] datasets. We then trained the student model on the VOS datasets (DAVIS [36, 37], YouTube-VOS [48]). Data augmentation was applied to both the training steps. We conducted all

Table 1: Comparison of the vanilla attention and deformable attention in attention learning in VOS. Best performances are highlighted.

Attention mechanism	DAVIS-16 Val			DAVIS-17 Val			YT-VOS18	YT-VOS19
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$
Vanilla attention	83.35	83.30	83.40	69.10	66.60	71.60	68.61	69.26
Deformable attention	**85.75**	**84.90**	**86.60**	**72.75**	**69.90**	**75.60**	**73.18**	**73.95**

Table 2: Comparison of state-of-the-art KD methods with self-supervised setting (i.e., without access to ground-truth labels). Best performances are highlighted. For [5], we re-executed the supplied code from the original work, but were not able to produce reasonable results on the Davis-16 and YT-VOS18/19 datasets. We, therefore, fill the results of [5] on those datasets with "-".

Methods	DAVIS-16 Val			DAVIS-17 Val			YT-VOS18	YT-VOS19
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$
DIST [18]	84.50	84.00	85.00	71.05	68.50	73.60	71.40	72.90
PEFD [5]	-	-	-	57.20	54.80	59.60	-	-
MobileVOS [31]	80.1	79.5	80.7	70.3	67.9	72.7	72.68	72.96
Ours	**85.75**	**84.90**	**86.60**	**72.75**	**69.90**	**75.60**	**73.20**	**74.00**

experiments on two NVIDIA RTX-3090 GPUs. The training process on the synthetic and VOS datasets took 24 hours with 100K iterations and 18 hours with 130K iterations, respectively. We set the training batch size to 16 in the whole training process.

Evaluation metrics. We evaluated the performance of our method and other baselines using the standard VOS metrics including the region similarity score $\mathcal{J}$, the boundary accuracy $\mathcal{F}$, and their mean ($\mathcal{J}\&\mathcal{F}$). The $\mathcal{J}$ score is the average intersection-over-union ratio between predicted and the ground-truth masks. The $\mathcal{F}$ score is the average similarity between the boundary of predicted and the ground-truth masks). We also followed the evaluation protocol by [36] to calculate these metrics.

4.3 Evaluation and results

We first evaluate our proposed deformable attention in learning of attention maps in VOS. To do this, we experiment our framework (in Fig. 2(a)) with the vanilla attention and deformable attention. We report the results of this experiment in Table 1. As shown in the results, the deformable attention outperforms the vanilla one on all the evaluation metrics and across all the datasets.

Next, we evaluate the effectiveness of our proposed KD method. Recall that our method also distills the attention maps, in addition to the logits during the distillation process. Therefore, to validate this idea, we compare our KD strategy with the one in [18], which transfers only the logits from the teacher to

Fig. 3: Qualitative results of our method and a baseline (a DeAOTT model with the vanilla attention trained using the standard KD, i.e., only logit layers are transferred). As shown, compared with the baseline, our method can maintain the association of the objects and their IDs across frames (see frame 36 in the 1st and 2nd row). Our method also tends to be aware of parts of the same object (see frame 50 in the 3rd and 4th row). We hypothesize this success is due to the cross-frame adaptivity of deformable attention over its counterpart. More results are provided in our supplementary material.

the student model. This strategy is equivalent to setting λ (in Eq. (12)) to 0. In addition, we experiment with the state-of-the-art KD algorithm in [5]. For a fair comparison, we replicate the methods in [5, 18, 31] for the VOS scenario using the same experimental setup presented in Section 4.2. We summarise the results of this comparison in Table 2. The experimental results confirm the benefit of distillation of intermediate attention maps during the distillation process. The results also show the superiority of our proposed KD method over the state-of-the-art KD.

We compare our method with existing self-supervised VOS methods on common datasets and present the comparison results in Table 3. In addition to evaluating the methods using segmentation accuracy metrics, we also compare their computational speed using frame-per-second (FPS) metric. As shown in Table 3, our method ranks first on the YouTube-VOS 2018 dataset and second on the DAVIS 2017 in terms of the segmentation accuracy (evident by the $\mathcal{J}\&\mathcal{F}$ scores). However, compared with the first-ranked method, i.e., Mask-VOS [23], although our method incurs a lower accuracy ($<$ 2.5% of the $\mathcal{J}\&\mathcal{F}$ score), it takes much less memory due to the lightweight architecture yet offers a much faster inference speed (about 50 $\times$ of the FPS), making the VOS real-time and feasible to low-powered devices. We summarise the segmentation accuracy vs memory footprint of all the methods in Fig. 4. We visualise several qualitative results of our method in Fig. 3.

4.4 Ablation study

We conducted ablative experiments to explain the rationale behind the design and settings of our method.

Table 3: Comparison of self-supervised VOS methods in terms of segmentation accuracy ($\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, $\mathcal{F}$) and inference speed (frame-per-second - FPS) on DAVIS-17 Val dataset. Best and second-best performances are highlighted with bold and underlines, respectively. Works in [19, 41] do not provide executable code for re-production. Their inference speed, thus, is not reported. Methods in [21, 41], on the other hand, are not evaluated on the YouTube-VOS18 dataset.

Methods	DAVIS-17 Val				YouTube-VOS18
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	FPS $\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$
CorrFlow [21]	50.30	48.40	52.20	2.00	-
MAST [22]	65.50	63.30	67.60	2.06	64.20
Self-cycle [41]	70.50	67.40	73.60	-	-
Mining [19]	70.30	67.90	72.60	-	67.30
LIIR [24]	72.10	69.70	74.50	1.87	69.30
Mask-VOS [23]	**74.50**	**71.60**	**77.40**	1.77	<u>71.60</u>
DeAOTT [50]	69.10	66.60	71.60	**64.26**	68.61
Ours	<u>72.75</u>	<u>69.90</u>	<u>75.60</u>	<u>52.36</u>	**73.18**

Table 4: Comparison of KL-divergence and CKA. Best performances are highlighted.

Variants	DAVIS-17 Val		
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
KL-divergence	54.65	52.10	57.20
CKA	**72.75**	**69.90**	**75.60**

Table 5: Comparison of different combinations of attention maps ($\mathcal{L}_{att}$) and logit layers ($\mathcal{L}_{inter}+\mathcal{L}_{intra}$) used in the distillation process.

Variants	DAVIS-17 Val		
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
Att. map 1, logit	69.65	66.90	72.40
Att. map 2, logit	66.40	63.80	69.00
Att. map 3, logit	**72.75**	**69.90**	**75.60**

Loss functions Recall that we propose to use the CKA score [20] to define the loss for attention distillation in Eq. (9). We prove that such a selection is effective. Specifically, we compare the CKA with the commonly used KL divergence in implementing the attention distillation loss. We present this comparison in Fig. 5. We observe that the CKA loss is well below the KL loss and clearly shows its convergence. This result also illustrates the anisotropic property of the KL loss in KD. We also investigate the logit distillation loss ($\mathcal{L}_{inter}+\mathcal{L}_{inter}$) in Eq. (11) and the entire loss ($\mathcal{L}$) in Eq. (12) in Fig. 5. As shown, our loss functions converge during the KD process. We quantitatively evaluate these distance metrics in terms of the performance of VOS. The results of this study is reported in Table 4, which clearly confirms our choice (i.e., CKA) for the attention loss.

Distillation information An important question to KD applications is what information should be distilled. In this ablation study, we experiment with different combinations of attention and logit layers used in the distillation process. Specifically, we choose attention maps generated from the GPM of the ID branch from the teacher model (see Fig. 2(a)). Recall that our student network is trained by

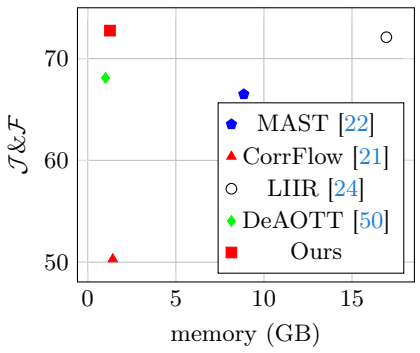


Fig. 4: Comparison of self-supervised VOS methods in terms of segmentation accuracy ($\mathcal{J}\&\mathcal{F}$) and memory footprint on DAVIS-17 Val dataset.

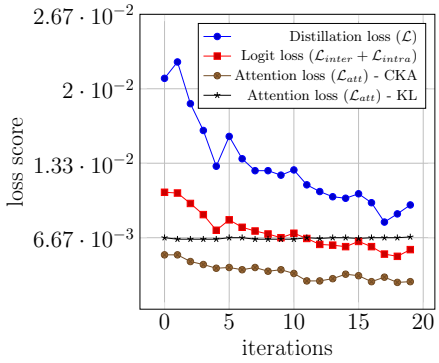


Fig. 5: Convergence analysis of the loss functions.

distillation of the attention map and the logit layer in the 3rd GPM from the teacher network. We report the performance of these combinations in Table 5, which clearly confirms the best performance of our student network.

5 Conclusion

We propose a novel method for video object segmentation (VOS) with self-supervised learning. The novelty of our work lies in improving attention learning to adapt with temporal changes in VOS via deformable attention that allows flexible feature locating, and a new knowledge distillation framework that enhances the distillation process via attention transfer.

We apply these technical innovations to create and train a lightweight VOS network in a self-supervised fashion. The network is shown to be adapted to both the spatial and temporal dimensions. We evaluate our method through extensive experiments on several benchmark datasets. Experimental results verify the robustness and efficiency of our method, and show that our method achieves state-of-the-art performance and optimal memory usage.

Appendices

–Supplementary Material–

In this supplementary material, we provide detailed descriptions of network architectures used in our work in Appendix A. We present the pseudo-code for our proposed knowledge distillation for VOS in Appendix B. Visualisations of the vanilla attention and our proposed deformable attention are illustrated in Appendix C. We present more quantitative analyses on various aspects of our

work in Appendix D, and qualitative comparisons of our method with existing video object segmentation methods in Appendix E. Limitations are discussed in Appendix F. Videos are also supplied alongside with this document. We will publish our code and pre-trained models.

A Network architectures

We refer the reader to Fig. 2 in the main paper where we depict the full pipeline of our method. In the following sub-sections, we provide technical details for main components in the pipeline.

A.1 Encoder

We utilised ResNet-50 and MobileNet-V2 as respective teacher's and student's backbones. To make the networks comparable with DeAOT [50], we modified the MobileNet-V2's encoder to include a dilation in the last stage and removed the stride from the first convolution in this last stage. We also removed the last stage in the ResNet-50 backbone.

A.2 Gated propagation module (GPM) and gated deformable propagation module (GDPM)

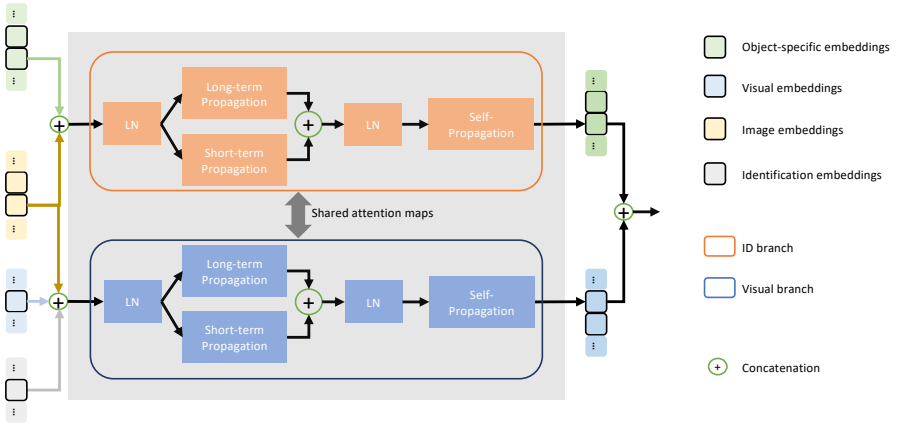


Fig. 6: Gated propagation module.

We present the architecture of the gated propagation module (GPM) in Fig. 6 in this supplementary material. Inputs for the GPM include visual embeddings, object-specific embeddings, identification embeddings and image embeddings. Those embedings are concatenated to be passed through a visual branch and an ID branch, each of which includes a series of long short-term transformers.

Specifically, image embeddings are produced by the image encoder, which we present in Section A.1. Visual embeddings and object-specific embeddings are the results of the visual and ID branch, respectively. Identification embeddings are generated from a decoder with respect to the previous frame. We describe the decoder in Section A.4 and Fig. 7. Note that, the visual branch and ID branch share the same attention maps.

The concatenated embedding inputs are fed into a Layer Normalization (LN). Long-term propagation, short-term propagation, and self-propagation modules in Fig. 6 are implemented using vanilla attention.

The gated deformable propagation module (GDPM) shares a similar architecture with the GPM. However, we replace the vanilla attention in the long-term propagation and self-propagation in both ID branch and visual branch by our deformable attention, which is described in the following sub-section.

A.3 Deformable attention

To lighten the network architecture of the student model, we implemented the deformable attention module in our GDPM using a single-head architecture. Our deformable attention offers flexible locations for the key and value via an offset network. The offset network consists of 2 convolutional layers with GELU activation functions in between (please refer to Figure 2(c) in the main paper). We set the resolution of the patch tokens of 16×16. The query $Q \in \mathbb{R}^{H \times W \times C_Q}$ fed into the offset network is calculated by a linear transformation as $Q = W_q X$ and partitioned uniformly along its feature channels into S groups, resulting sub-feature maps $\left\{Q_i \in \mathbb{R}^{H \times W \times C_{Q_i}}\right\}_{i=1}^{S}$. We set $C_Q = 512$ and $S = 4$. The partition of Q into smaller groups in the first convolution aims at controlling the connections between input feature channels and output channels. This convolution has kernel size of 5, stride of 1, padding of 2. The dimensions for the input and output channels are set to $\frac{C_{Q_i}}{S}$.

For the second convolutional layer, 1×1 convolution is applied. The resulting offsets $\Delta \in \mathbb{R}^{2 \times S \times H_g \times W_g}$ contain the offset values for both vertical and horizontal directions with respect to reference points. These offsets are then used to shift the reference points for locating the deformable key $\tilde{K}$ and value $\tilde{V}$, which are finally used to calculate a deformable attention map.

A.4 Decoder

Fig. 7 provides an overview of the decoder used in our paper. Inputs for the decoder are visual and object-specific embeddings, which are reshaped into a 2D form before being passed into the decoder. The decoder includes a series of convolutional layers with skip connections and bilinear upsampling layers. The final convolutional layer returns logits, which are then converted into a segmentation mask.

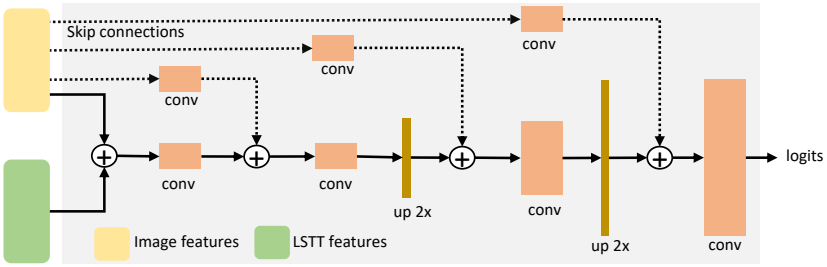


Fig. 7: Decoder architecture. We utilised the feature pyramid network in [27] to make our decoder.

B Knowledge distillation pseudo-code

We illustrate a PyTorch-style implementation of our distillation method in Algorithm 1. Note that our knowledge distillation method does not require ground-truth labels for training of the student network, but uses pseudo labels generated by the teacher model.

C Visualisation

To illustrate the adaptability of deformable attention under temporal changes in VOS, we visualise the learnt attention maps of the distilled attention layer of the student network in Fig. 8. Recall that, in our implementation, we distill the last attention layer in the GPM in the teacher network to the GDPM in the student network. As shown in Fig. 8, attention maps achieved by deformable attention focus on the foreground objects.

We further showcase keypoints obtained from attention maps in Fig. 9. Keypoints are image locations whose corresponding attention map values are greater than 0.85 of the maximum attention values. It can be seen from Fig. 9 that most important keypoints are located within the boundaries of query objects. This means that deformable attention is aware of object boundaries under variations overtime. This ability is critical for segmenting objects in cluttered backgrounds.

We visualise deformable attention map values for given specific query tokens in Fig. 10. In this experiment, each query token (represented by a star) is chosen from important keypoints and corresponding key tokens (top 200-similar key tokens) are shown. It is observed that high-similarity keys of the queries from the bus and camel often located in foreground objects. In contrast, keys of the queries from the swan or humans are scattered in both the foreground and background, showing the challenge in segmenting these objects.

Algorithm 1: PyTorch-style pseudocode for our proposed knowledge distillation framework for VOS

```
# f_s, f_t: student and pre-trained teacher networks
# y_s, y_t: soft labels of student and teacher networks
# a_s, a_t: attention maps of student and teacher networks
f_t.eval()
f_s.train()
for X,_ in dataloader:
    # Feed forward
    a_t, y_t = f_t(X)
    a_s, y_s = f_s(X)

    # inter-object and intra-object losses
    num_obj = y_s.shape[1]
    y_s = y_s.transpose(1, -1).reshape(-1, num_obj)
    y_t = y_t.transpose(1, -1).reshape(-1, num_obj)
    y_s = softmax(y_s, dim=1)
    y_t = softmax(y_t, dim=1)
    def inter_obj_loss(y_s, y_t):
        1 - pearson_corr(y_s, y_t).mean()
    inter_loss = inter_obj_loss(y_s, y_t)
    intra_loss = inter_obj_loss(y_s.transpose(0, 1),
     y_t.transpose(0, 1))

    # Attention loss
    att_loss = cka_score(a_s, a_t)
    loss = inter_loss + intra_loss + λ*att_loss

    # Optimisation step
    loss.backward()
    optimizer.step()
```

D Quantitative analysis

D.1 Loss functions

In our ablation study (section 4.4) in the main paper, we studied the impact of the loss functions by investigating their convergence during training (please see the learning curves in Fig. 5 in the main paper). Here, we verify their role in testing. In particular, we measure the $\mathcal{J}$, $\mathcal{F}$, and $\mathcal{J}\&\mathcal{F}$ scores of different combinations of the loss functions on the test sets. We report the results of this experiment in Table 6. As shown in the results, our defined loss, combining all $\mathcal{L}_{inter}$, $\mathcal{L}_{intra}$, and $\mathcal{L}_{att}$ achieves the best performance on all evaluation metrics across all datasets.

Fig. 8: Visualisation of the deformable attention maps of the distilled attention layer in the student network on DAVIS2017 Val dataset. For each test case, we present a query frame (frame ID is shown in the left-corner) and its corresponding ground-truth segmentation masks (top-right corner). For the ease in presentation, we only display pixels whose attention map value is greater than 0.5. As shown, pixels on the target objects are highlighted (i.e., they are often associated with high attention map values) on the deformable attention maps. We found that those pixels also correspond to huge temporal changes, showcasing the adaptability of the deformable attention mechanism to both the spatial and temporal dimensions.

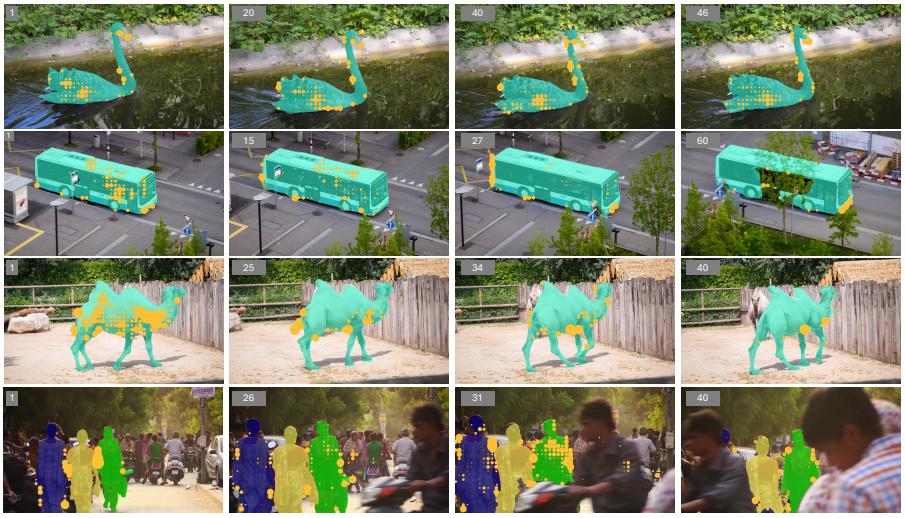


Fig. 9: Keypoints "●" from deformable attention maps on DAVIS2017 Val dataset. The radius of each keypoint "●●" is proportional to attention map values of the distilled attention layer in the student network at that keypoint.

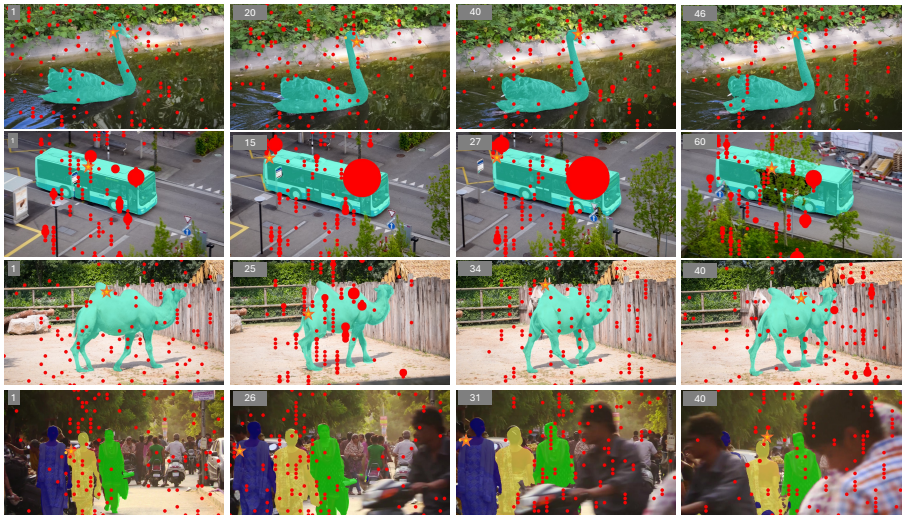


Fig. 10: Query tokens "⋆" and corresponding key tokens "●" identified from deformable attention maps on DAVIS2017 Val dataset. The radius of each key token "●●" is proportional to the similarity between that key token and its input query token "⋆".

D.2 Balance factor λ

Recall that our distillation loss is defined as,

$$\mathcal{L} = \mathcal{L}_{inter} + \mathcal{L}_{intra} + \lambda \mathcal{L}_{att}$$

where λ is a user-defined paramter to balance the logit and attention losses.

Table 6: Comparison of loss functions. Best performances are highlighted.

Loss function	DAVIS-16 Val			DAVIS-17 Val			YT-VOS18	YT-VOS19
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$
$\mathcal{L}_{inter}$	83.50	82.70	84.30	68.75	66.20	71.30	69.83	71.05
$\mathcal{L}_{inter} + \mathcal{L}_{intra}$	82.50	82.10	83.50	69.50	66.90	72.10	69.67	70.94
$\mathcal{L}_{inter} + \mathcal{L}_{intra} + \lambda\mathcal{L}_{att}$	**85.75**	**84.90**	**86.60**	**72.75**	**69.90**	**75.60**	**73.18**	**73.95**

Table 7: Parameter setting for λ. Best performances are highlighted.

Value for λ	DAVIS-16 Val			DAVIS-17 Val			YT-VOS18	YT-VOS19
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$
$\lambda = 0.5$	83.75	83.2	84.3	68.25	65.6	70.9	69.35	70.93
$\lambda = 1.0$	83.55	83	84.1	69.1	66.7	71.5	68.23	69.47
$\lambda = 1.5$	**85.75**	**84.90**	**86.60**	**72.75**	**69.90**	**75.60**	**73.18**	**73.95**
$\lambda = 2.0$	81.60	81.20	82.00	64.75	63.10	66.40	69.90	67.17

Table 8: Comparison of conventional KD and our proposed KD, applied to the model in [49]. Best performances are highlighted.

Methods	DAVIS-16 Val			DAVIS-17 Val			YT-VOS18	YT-VOS19
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$
Conventional KD	78.90	80.80	77.00	56.30	55.70	56.90	60.29	60.52
Ours	**81.50**	**83.00**	**80.00**	**67.25**	**66.70**	**67.80**	**67.30**	**67.60**

We experimented our method with various values for λ and present results in Table 7. Experimental results show that our current setting ($\lambda = 1.5$) achieves the best performance.

D.3 Generality

Our proposed knowledge distillation (KD) method can be applied to other attention-based VOS models. To prove this, we apply our KD strategy to the model in [49]. Specifically, we employ the model in [49] as the teacher model and distill the attention map in its last layer to the student model. We compare our KD strategy (i.e., distillation of both the logits and attention layers) and the conventional approach (i.e., distillation of the logits only) in Table 8. As shown in the results, our proposed KD outperforms the conventional one on all the test sets. Note that we train the student network using the self-supervised fashion (i.e., labels are generated from the teacher model).

D.4 A toy example on DAVIS-17 Val dataset.

To gain insights into knowledge distillation for VOS, we conducted an experiment using a toy example on DAVIS2017 Val dataset (see Fig. 1 in the main paper and Fig. 11 in this supplementary material). In this experiment, we randomly selected

4 videos from DAVIS2017 Val dataset, where each video contain a different number of query objects. Overall, our method outperforms existing knowledge distillation frameworks in most cases (for different numbers of query objects), except for a case where the test video has two query objects.

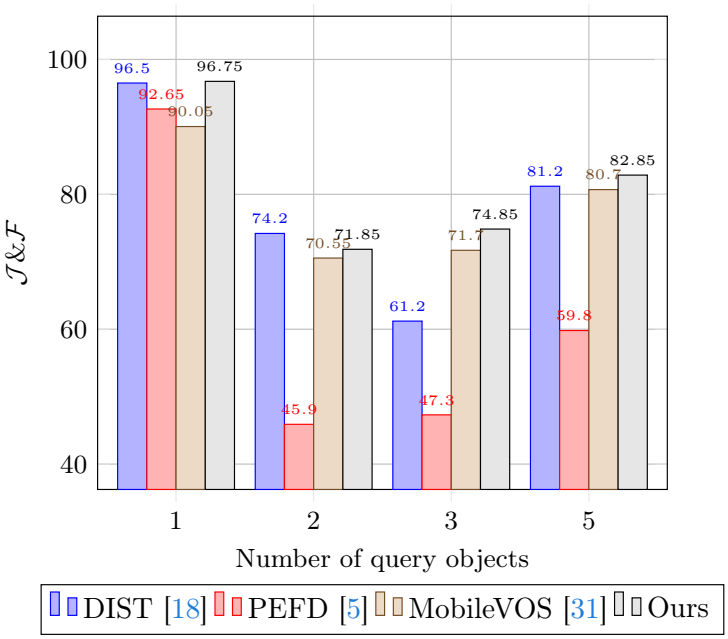


Fig. 11: Comparison of state-of-the-art knowledge distillation methods with self-supervised setting on our toy example.

E Qualitative analysis

We provide qualitative comparisons of our method with existing ones including CorrFlow [21], MAST [22], LIIR [24] in Fig. 12 and Fig. 13. Fig. 12 shows common cases while Fig. 13 present challenging cases such as fast motion, cluttered background, and high deformation.

F Limitations

Our method struggles with videos containing out-of-plane rotations and intra-target scale variations (see Fig. 14). Intra-target scale variations refer to the scenarios where a target object changes its scale significantly by suddenly moving closer/further from the camera. It is observed that, intra-target scale variations are also challenging for the teacher model (downsampling/upsamplinglayers in the encoder/decoder cannot tolerate sudden and huge changes of object appearance). As a result, attention and logit transfers from the teacher model to the student model could be easily deteriorated to learn discriminitive object representations for VOS.

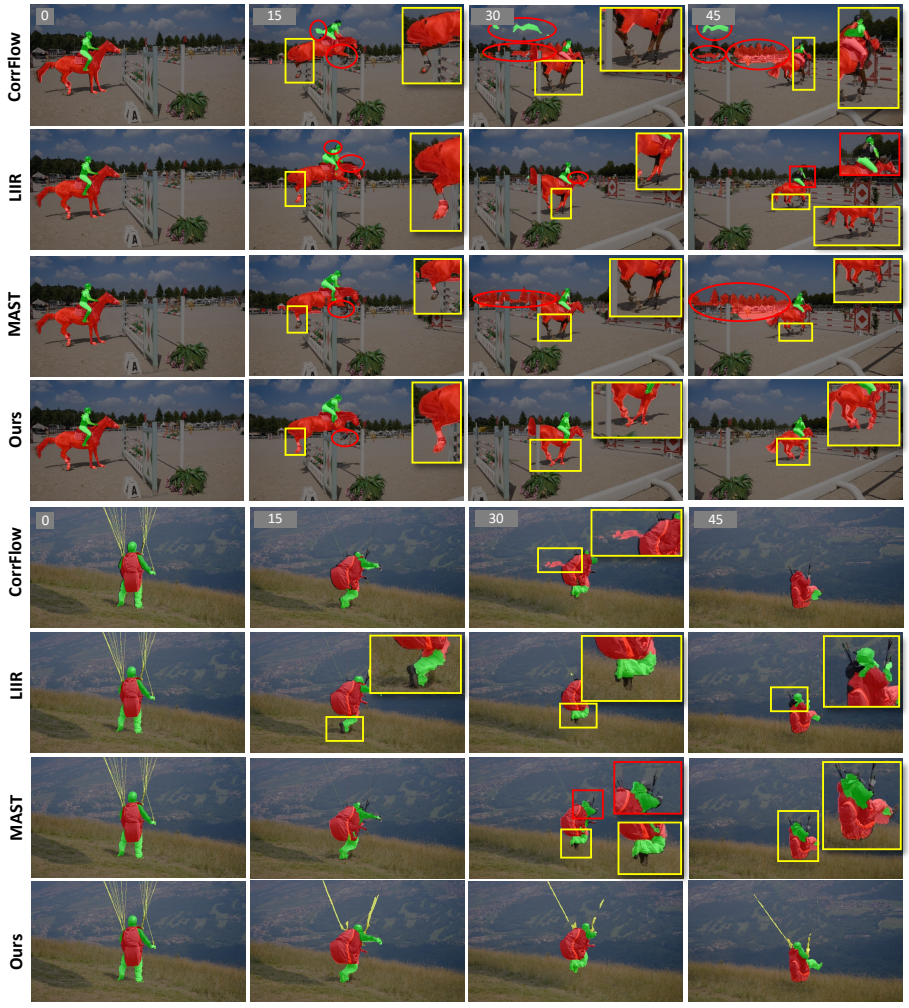


Fig. 12: Qualitative comparison of our method with existing ones on DAVIS2017 Val dataset. As shown, our method clearly provides better segmentation results, compared with existing ones.

References

1. Ainslie, J., Ontanon, S., Alberti, C., Cvicek, V., Fisher, Z., Pham, P., Ravula, A., Sanghai, S., Wang, Q., Yang, L.: ETC: Encoding long and structured inputs in transformers. arXiv preprint arXiv:2004.08483 (2020) 5
2. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 221–230 (2017) 3

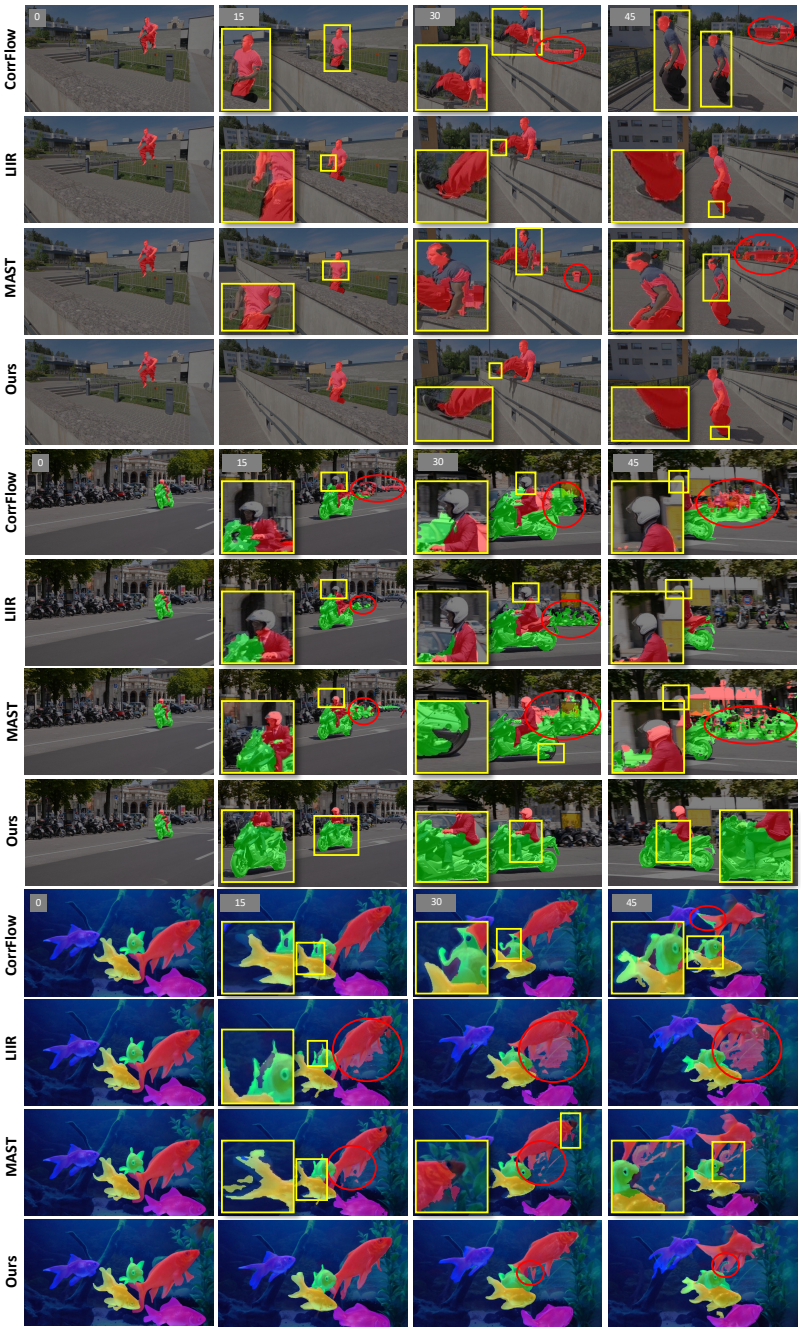


Fig. 13: Results on challenging cases from DAVIS2017 Val dataset. Compared with other methods, our method still shows more advances in fast motion (1st - 4th rows), cluttered background (5th - 8th rows), and high deformation (9th - 12th rows).

Fig. 14: Limitations of our method (results are generated from DAVIS2017 Val dataset). Our method fails to segment objects under intra-target scale variations (1st row), out-of-plane rotation (2nd and 3rd rows).

3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229 (2020) 5
4. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: International Conference on Machine Learning. pp. 1691–1703 (2020) 5
5. Chen, Y., Wang, S., Liu, J., Xu, X., de Hoog, F., Huang, Z.: Improved feature distillation via projector ensemble. In: Neural Information Processing Systems. pp. 12084–12095 (2022) 5, 9, 11, 12, 22
6. Cheng, H.K., Chung, J., Tai, Y.W., Tang, C.K.: Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8890–8899 (2020) 10
7. Cheng, H.K., Schwing, A.G.: Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In: European Conference on Computer Vision. pp. 640–658 (2022) 4
8. Cheng, H.K., Tai, Y.W., Tang, C.K.: Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5559–5568 (2021) 4
9. Cheng, T., Wang, X., Chen, S., Zhang, Q., Liu, W.: Boxteacher: Exploring high-quality pseudo labels for weakly supervised instance segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3145–3154 (2023) 5
10. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: IEEE/CVF International Conference on Computer Vision. pp. 764–773 (2017) 3, 7
11. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12124–12134 (2022) 5
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representation (2021) 2, 5

13. Duke, B., Ahmed, A., Wolf, C., Aarabi, P., Taylor, G.W.: SSTVOS: sparse spatiotemporal transformers for video object segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5912–5921 (2021) 2
14. Gao, M., Zheng, F., Yu, J.J.Q., Shan, C., Ding, G., Han, J.: Deep learning for video object segmentation: a review. Artificial Intelligence Review **56**(1), 457–531 (2023) 1, 2
15. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with hilbert-schmidt norms. In: International Conference on Algorithmic Learning Theory. pp. 63–77 (2005) 9
16. Heo, M., Hwang, S., Oh, S.W., Lee, J., Kim, S.J.: VITA: video instance segmentation via object token association. In: Neural Information Processing Systems (2022) 2
17. Hu, P., Wang, G., Kong, X., Kuen, J., Tan, Y.P.: Motion-guided cascaded refinement network for video object segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1400–1409 (2018) 3
18. Huang, T., You, S., Wang, F., Qian, C., Xu, C.: Knowledge distillation from a stronger teacher. In: Neural Information Processing Systems. pp. 33716–33727 (2022) 5, 9, 11, 12, 22
19. Jeon, S., Min, D., Kim, S., Sohn, K.: Mining better samples for contrastive learning of temporal correspondence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1034–1044 (2021) 4, 13
20. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning. pp. 3519–3529 (2019) 6, 9, 13
21. Lai, Z., Xie, W.: Self-supervised learning for video correspondence flow. In: BMVC (2019) 4, 13, 14, 22
22. Lai, Z., Lu, E., Xie, W.: Mast: A memory-augmented self-supervised tracker. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6479–6488 (2020) 4, 13, 14, 22
23. Li, L., Wang, W., Zhou, T., Li, J., Yang, Y.: Unified mask embedding and correspondence learning for self-supervised video segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18706–18716 (2023) 4, 12, 13
24. Li, L., Zhou, T., Wang, W., Yang, L., Li, J., Yang, Y.: Locality-aware inter-and intra-video reconstruction for self-supervised correspondence learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8719–8730 (2022) 4, 13, 14, 22
25. Li, X., Wei, T., Chen, Y.P., Tai, Y.W., Tang, C.K.: Fss-1000: A 1000-class dataset for few-shot segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2869–2878 (2020) 10
26. Li, X., Loy, C.C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: European Conference on Computer Vision. pp. 90–105 (2018) 4
27. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2117–2125 (2017) 17
28. Liu, D., Kan, M., Shan, S., Chen, X.: Function-consistent feature distillation. International Conference on Learning Representations (2023) 9
29. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021) 5

30. Miao, J., Wei, Y., Yang, Y.: Memory aggregation networks for efficient interactive video object segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10363–10372 (2020) 2
31. Miles, R., Yucel, M.K., Manganelli, B., Saà-Garriga, A.: MobileVOS: Real-time video object segmentation contrastive learning meets knowledge distillation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10480–10490 (2023) 2, 3, 5, 11, 12, 22
32. Pan, X., Ye, T., Xia, Z., Song, S., Huang, G.: Slide-transformer: Hierarchical vision transformer with local self-attention. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2082–2091 (2023) 5
33. Park, D.Y., Cha, M.H., Kim, D., Han, B., et al.: Learning student-friendly teacher networks for knowledge distillation. In: Neural Information Processing Systems. pp. 13292–13303 (2021) 5
34. Park, K., Woo, S., Oh, S.W., Kweon, I.S., Lee, J.Y.: Per-clip video object segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1352–1361 (2022) 2
35. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2663–2672 (2017) 3
36. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 724–732 (2016) 10, 11
37. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017) 10
38. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017) 9
39. Richter, M.L., Pal, C.: Receptive field refinement for convolutional neural networks reliably improves predictive performance. arXiv preprint arXiv:2211.14487 (2022) 2
40. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended cssd. IEEE transactions on pattern analysis and machine intelligence **38**(4), 717–729 (2015) 10
41. Son, J.: Contrastive learning for space-time correspondence via self-cycle consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14679–14688 (2022) 4, 13
42. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: IEEE/CVF International Conference on Computer Vision. pp. 7262–7272 (2021) 5
43. Tsai, Y.H., Yang, M.H., Black, M.J.: Video segmentation via object flow. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3899–3908 (2016) 2
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 5998–6008 (2017) 5, 6
45. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 136–145 (2017) 10

46. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4794–4803 (2022) 5, 7, 9
47. Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., Girshick, R.: Early convolutions help transformers see better. In: Neural Information Processing Systems. pp. 30392–30400 (2021) 5
48. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018) 10
49. Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. In: Neural Information Processing Systems. pp. 2491–2502 (2021) 2, 4, 21
50. Yang, Z., Yang, Y.: Decoupling features in hierarchical propagation for video object segmentation. In: Neural Information Processing Systems (2022) 4, 6, 7, 10, 13, 14, 15
51. Yoo, J.S., Lee, H., Jung, S.W.: Hierarchical spatiotemporal transformers for video object segmentation. arXiv preprint arXiv:2307.08263 (2023) 2
52. Yu, Y., Yuan, J., Mittal, G., Fuxin, L., Chen, M.: Batman: Bilateral attention transformer in motion-appearance neighboring space for video object segmentation. In: European Conference on Computer Vision. pp. 612–629 (2022) 2
53. Zeng, Y., Zhang, P., Zhang, J., Lin, Z., Lu, H.: Towards high-resolution salient object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7234–7243 (2019) 10
54. Zhang, Y., Li, L., Wang, W., Xie, R., Song, L., Zhang, W.: Boosting video object segmentation via space-time correspondence learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2246–2256 (2023) 2
55. Zong, M., Qiu, Z., Ma, X., Yang, K., Liu, C., Hou, J., Yi, S., Ouyang, W.: Better teacher better student: Dynamic prior knowledge for knowledge distillation. In: International Conference on Learning Representations (2022) 5