Unsupervised Video Object Segmentation with Online Adversarial Self-Tuning

Tiankang Su[1] Huihui Song[1*] Dong Liu[2] Bo Liu[3] Qingshan Liu[4]
[1]B-DAT and CICAEET, Nanjing University of Information Science and Technology, Nanjing, China
[2]Netflix Inc, Los Gatos, CA, 95032, USA
[3]Walmart Global Tech, Sunnyvale, CA, 94086, USA
[4]School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China

Abstract

The existing unsupervised video object segmentation methods depend heavily on the segmentation model trained offline on a labeled training video set, and cannot well generalize to the test videos from a different domain with possible distribution shifts. We propose to perform online finetuning on the pre-trained segmentation model to adapt to any ad-hoc videos at the test time. To achieve this, we design an offline semi-supervised adversarial training process, which leverages the unlabeled video frames to improve the model generalizability while aligning the features of the labeled video frames with the features of the unlabeled video frames. With the trained segmentation model, we further conduct an online self-supervised adversarial finetuning, in which a teacher model and a student model are first initialized with the pre-trained segmentation model weights, and the pseudo label produced by the teacher model is used to supervise the student model in an adversarial learning framework. Through online finetuning, the student model is progressively updated according to the emerging patterns in each test video, which significantly reduces the test-time domain gap. We integrate our offline training and online finetuning in a unified framework for unsupervised video object segmentation and dub our method Online Adversarial Self-Tuning (OAST). The experiments show that our method outperforms the state-of-the-arts with significant gains on the popular video object segmentation datasets.

1. Introduction

Video Object Segmentation (VOS) aims to track the moving objects in a video sequence with an accurate segmentation mask. The existing VOS works can be categorized into two paradigms based on whether the prior knowledge is provided at the test time. One is the *Semi-supervised VOS* (SVOS), where a model is trained on the training set, and at the test time is provided with the ground truth mask on the first frame as prior to segment the objects in the subsequent frames. The other is called *Unsupervised VOS*[1] (UVOS), where no ground-truth mask is provided at the test time and there is no prior to leverage to segment the target.

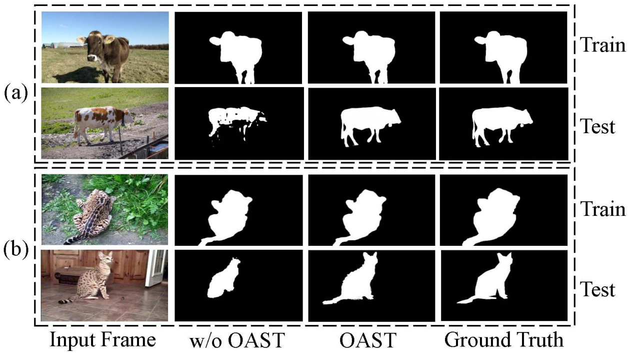

Figure 1. Examples of "*visual discrepancy*" (row (a)-(b)) in UVOS, where "train" and "test" indicate where the example image is drawn from. Our OAST method produces more precise segmentation masks (third column) comparing to the method without OAST (second column).

We focus on UVOS since it requires no prior input and is closer to the real-world applications. The existing UVOS works train a model with a labeled video set, and then directly apply it to the unlabeled videos at the test time [51, 7, 40]. Without any prior as input, the inference has to completely depend on the trained model. This has proven to be effective when the test data are drawn from the same domain as the training data [68, 65, 46]. However, the inference result can become degraded when the test data originate from a different domain which is a common case under the zero-shot setting. The main scenario that can cause severe domain shifts in the test data is called "*visual*

[*] Corresponding author. Email:songhuihui@nuist.edu.cn.
This work is supported in part by National Key Research and Development Program of China under Grant No. 2018AAA0100400, in part by the NSFC under Grant Nos. 61825601, U21B2044, 62276141, 61872189, in part by the Postgraduate Research&Practice Innovation Program of Jiangsu Province under Grant Nos. KYCX23_1367.

[1] It is also referred to as "zero-shot VOS" or "primary object segmentation" in the literature.

discrepancy". Compared to the training data, the test data may be captured in different environments and manifest significant inter-class ambiguity and intra-class variance. Figure 1(a) shows a cow object from the training and test data respectively. Due to their appearance difference, the model is confused to perform good segmentation. On the contrary, Figure 1(b) shows a leopard object for training and a cat object for test. They have highly similar visual appearances even belonging to different categories. As seen, the segmentation result is not satisfactory.

Therefore, there is a demand to bridge such visual discrepancy in UVOS. Motivated by the online finetuning method commonly used in SVOS [54, 6, 28, 2], we propose to perform online test-time finetuning in UVOS to account for the ad-hoc test videos. Specifically, we start from a pre-trained VOS model, and progressively update it according to the emerging visual patterns in the test video. The updates and predictions are performed online, during which the model only has access to the current test video without having access to the full test data or any training data.

There are two challenges in devising an online finetuning method for UVOS. The first is how to reduce the overfitting of the VOS model to the labeled training data, and make it easier to generalize to the test data in new domain. To this end, we propose to a semi-supervised training strategy by adding arbitrary unlabeled videos into model training. It takes a labeled video frame and any unlabeled video frame (without using test data and with a distribution shift from the training data) as input, and trains a discriminator to distinguish the predicted intermediate feature map of the labeled frame from that of the unlabeled frame. The discriminator is trained in an adversarial manner such that the two kinds of feature map become indistinguishable. This training process aligns the features of the labeled video frames to the features of the unlabeled frames, which helps relieve the visual discrepancy to the future test data, and improves the generalizability of the trained VOS model.

The second challenge lies in that we have no supervision on the test video to enable online finetuning. We therefore propose to tackle the task in a self-supervised manner. With the trained VOS model, we initialize a teacher model and a student model with the model weights. Motivated by the fact that the mean teacher prediction tends to be more accurate than the individual model prediction [49, 56], given a video frame at the test time, we perform data augmentation over it and use the augmentation-averaged prediction from the teacher model as the pseudo ground-truth segmentation mask. Meanwhile, we feed the raw frame into the student model and get a predicted mask. These two masks are then fed into a discriminator, which employs an adversarial loss to minimize the difference of these two masks such that they become indistinguishable. In this process, the parameters of the student model are updated to make similar predictions as the mean teacher prediction. Once the training is done, the weights of the teacher model are updated by the weights of the student model using exponential moving average to account for the emerging patterns in the test data.

We design our method based on the training and finetuning strategies discussed above and dub it *Online Adversarial Self-Tuning* (OAST) due to its adversarial nature during training and testing. We conduct extensive experiments over five popular benchmark datasets and demonstrate that our OAST method achieves the state-of-the-art performance with significant gains. Our main contributions include: (1) an online test-time finetuning method for UVOS, which dynamically updates the VOS model to adapt into the test data in a new domain, and, to our best knowledge, is the first online finetuning method for UVOS, (2) an offline semi-supervised adversarial training method to improve the generalization ability of the VOS model, (3) an online self-supervised adversarial fintuning method for test-time adaptation to account for each new test video.

2. Related Work

Online Finetuning in Semi-Supervised VOS. The existing SVOS works focus on modeling the spatial-temporal object dynamics in video [51, 53, 62] or fusing complementary signals via multimodal learning [6, 14, 61]. In SVOS, a model is first trained offline on the training data to learn a general concept of objects. Then given a video at the test time, the model is finetuned on the given ground-truth mask of the first frame, and then applied to segment the rest of the frames in the video. Such a finetuning step is critical for SVOS performance [42, 3, 25, 24, 33]. However, due to the lack of any supervision in UVOS, there is no mechanism to perform finetuning, we motivates us for this research.

Unsupervised VOS. The popular UVOS methods operate on the RGB frames of a video, and model their high-order relations [57, 31], pixel-wise correspondence [64, 23], or long-range dependencies [9, 31, 46]. Some recent efforts incorporate the motion cues as additional signals to infer the object mask [68, 65, 46, 40]. LVO [51] trains a two-stream network, taking an RGB frame and the optical flow into a ConvGRU module to infer the segmentation mask. TransportNet [65] uses the optimal structural matching to align the frame features and optical flow features to suppress the distracting noisy signals in each modality. MATNet [68] proposes a motion-attentive transition to fuse the motion and appearance features. RTNet [46] identifies the primary objects in a video by correlating the intra-frame contrast, the motion cues and temporal coherence of recurring objects. These methods do not have a mechanism to perform online test-time finetuning to account for the emerging patterns in the test videos as our work does.

Domain Adaptation. The semi-supervised adversarial training in our method bears assembly with the Unsuper-

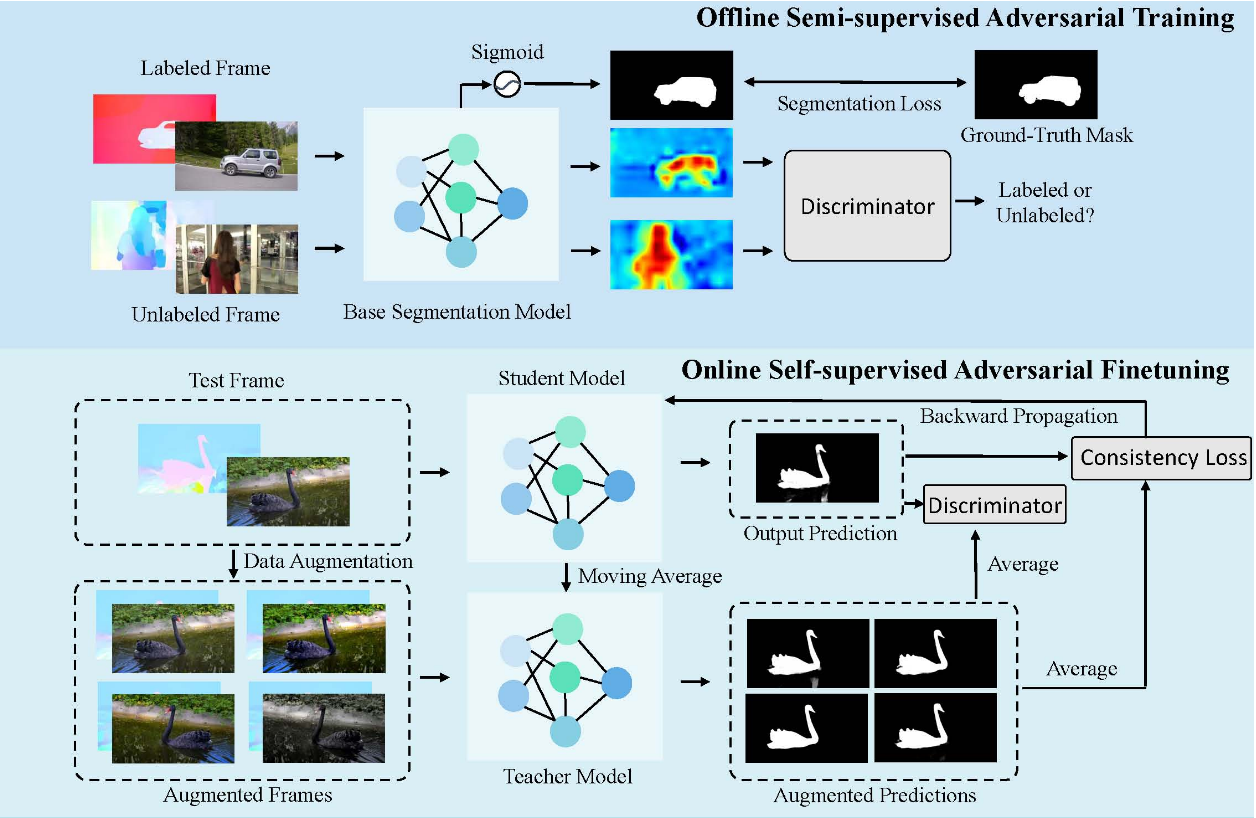

Figure 2. Overview of our OAST method. In the *offline semi-supervised adversarial training* stage, given a labeled video frame and any unlabeled video frame (without using test data and together with their optical flow map) as input, our base segmentation model outputs two segmentation masks for each. A discriminator is trained in an adversarial manner such that the feature maps from the labeled and unlabeled frames become distinguishable. In the *online self-supervised finetuning* stage, a teacher model and a student model are initialized with the trained base model weights. Given a test frame (together with its optical flow map), we use the teacher model to generate an augmentation-averaged prediction as the pseudo label, while using the student model to generate a predicted map on the raw frame. The two masks are fed into a discriminator to minimize the gap between the teacher and student model. The student model is updated online by adversarial learning over each test frame, and the teacher model is updated by the student model weights using exponential moving average.

vised Domain Adaptation (UDA) [35, 69, 22, 10, 29], which is a task of transferring knowledge from the labeled source domain to the unlabeled target domain to improve model performance on the target domain. During training, UDA methods align the feature distributions between the two domains typically with adversarial training [12, 52, 41]. This setup requires a pre-defined unlabeled target domain for alignment. On the contrary, we adopt any ad-hoc unlabeled videos, and the purpose is to reduce the risk of overfitting the VOS model to the labeled training data, and improve the model generalizability. The unlabeled videos play the role of regularization in training, and they do not need to overlap with the domain of the test videos.

Test-Time Adaptation. Test-Time adaptation, also referred to as source-free domain adaptation, aims to adapt the model to the target domain without requiring any data from the source domain. One popular trend is to finetune the source model without explicitly performing domain alignment. TENT [55] adapts a pre-trained model by adjusting the trainable parameters in BatchNorm layers via entropy minimization. SHOT [26] freezes the source model and learns the target-specific feature representation to implicitly align the target to the source. [36] uses a diversity regularizer together with an input transformation module to improve adaptation stability. TTT [48] trains an additional auxiliary rotation prediction head to adapt to the target data distribution. T3A [15] adjusts the last classification layer during the inference time based on the pseudo-prototype representations learned from the online unlabeled data and the base classifier trained in source domain. Most existing works require re-training the source model to support test-time adaptation, and therefore is unrealistic for real-world online applications. To relieve this, CoTTA [56] is recently proposed to adapt a pre-trained source model to continually changing target data. It is a self-supervised framework where a mean teacher model is used to generate high-quality pseudo labels used for training the student model with a label consistency loss. We are motivated by this method, but the difference is that we introduce adversarial learning to better align test video frame to the pre-trained model at the test time, which is tailored to the VOS task and proven to outperform CoTTA under the zero-shot setting.

3. Online Adversarial Self-Tuning for UVOS

3.1. Overview

Figure 2 illustrates our OAST method, which consists of an *offline semi-supervised adversarial training* and an *online self-supervised adversarial finetuning* stage.

At the training stage, given a labeled and an unlabeled video frame, we employ a base segmentation model to predict a segmentation mask for each frame. We minimize the segmentation loss between the labeled frame and its ground-truth mask. Besides, we take the two predicted feature maps, and feed them into a discriminator for adversarial training. the feature maps from the labeled and unlabeled frames are enforced to be indistinguishable, which reduces the domain gap between the labeled and unlabeled data, and increases the generalizability of the base model.

At the finetuning stage, we initialize a teacher and a student model with the pre-trained model weights. Given a video frame at the test time, we perform data augmentation and adopt the augmentation-averaged prediction from the teacher model as the pseudo ground-truth mask. Concurrently, the frame is fed into the student model to produce a predicted mask. The two masks are then sent into a discriminator and an adversarial training process is conducted to minimize their gap. In this process, the weights of the student model are updated to account for the new test data, and the weights of the teacher model are updated by the weights of the student model using exponential moving average.

3.2. Base Segmentation Model

Motivated by the success of the two-stream network in VOS [68, 46, 65], our base segmentation model takes both appearance and motion cues as input. Given a video frame $\mathbf{I}_a \in \mathbb{R}^{H\times W\times 3}$ and its optical flow map $\mathbf{I}_m \in \mathbb{R}^{H\times W\times 3}$, where H and W are the height and width, we concatenate them together as a hybrid 4D tensor input $\mathbf{I}_{am} = [\mathbf{I}_a; \mathbf{I}_m] \in \mathbb{R}^{2\times H\times W\times 3}$. Our model is a U-shape network consisting of an encoder and a decoder. The encoder is adapted from the MobileViT [34] by inflating the 2D convolution layers into the 3D convolution layers similar to I3D [4] to meet the requirement of our input. The decoder is a symmetric architecture as the encoder in which the down-scaling stages are replaced by the up-scaling stages and the skip connections are used to link the corresponding stages together. We use the pre-trained MobileViT to initialize the encoder and randomly initialize the decoder. The whole model was finetuned end-to-end with the training data. The details of the network architecture can be found in the supplemental material. At the end of the network, a 1×1 convolution layer followed by upscaling and a sigmoid function is applied on the feature, producing a predicted mask $\mathbf{P} \in [0, 1]^{H\times W}$.

3.3. Offline Semi-Supervised Adversarial Training

Given a labeled video frame $\mathbf{I}^l_{am}$ and a unlabeled video frame $\mathbf{I}^u_{am}$, where we have a ground-truth segmentation mask $\mathbf{G}^l \in \{0, 1\}^{H\times W}$ for the labeled frame as supervision. We apply our base segmentation model as a generator to produce two feature maps $\mathbf{F}^l$, $\mathbf{F}^u \in \mathbb{R}^{H\times W\times C}$ (by upscaling the feature maps output from the last layer of the backbone), as well as a segmentation mask for the labeled frame only. We first employ the classic VOS segmentation loss including Cross-Entropy (CE) loss and Intersection-over-Union (IoU) loss over the predicted segmentation mask and the ground-truth mask, formulated as,

$$\mathcal{L}_{seg} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{IoU}, \tag{1}$$

where $\mathcal{L}_{CE} = -\sum_{i,j} \mathbf{G}^l_{ij} \log \mathbf{P}^l_{ij}$, $\mathcal{L}_{IoU} = 1 - \frac{1}{HW}\sum_{ij} \min(\mathbf{G}^l_{ij}, \mathbf{P}^l_{ij}) / \max(\mathbf{G}^l_{ij}, \mathbf{P}^l_{ij})$, and λ is a tradeoff parameter.

We choose a light-weight network as the discriminator. It contains four convolution layers, in which each layer reduces the input resolution by 2 and every two layers are connected by a BatchNorm and a ReLU layer. The details of the discriminator architecture can be found in the supplemental material. The discriminator takes a feature map of size $H \times W$ as input and outputs a scalar probability value indicating the likelihood of the input feature map being labeled. In our semi-supervised training, we feed $\mathbf{F}^l$ and $\mathbf{F}^u$ into the discriminator and yield probability values f^l, $f^u \in [0, 1]$ as outputs.

With the discriminator output, we further apply an adversarial loss as follows,

$$\mathcal{L}_{domain} = \log(f^l) + \log(1 - f^u), \tag{2}$$

where the discriminator tries to enforce $f^l \to 1$, $f^u \to 0$, and the adversarial training tries to confuse the discriminator until the origins of the two input feature maps become indistinguishable. This essentially enforces the model to align the labeled frames with the unlabeled frames in the feature space, which relaxes the model from the training videos and better generalizes to new test videos.

Based on the aforementioned losses, our total training loss is defined as follows,

$$\mathcal{L} = \mathcal{L}_{seg} + \lambda_d\mathcal{L}_{domain}, \tag{3}$$

where λ_d is the tradeoff parameters, and we jointly train our feature map generator and the discriminator in an adversarial min-max fashion.

3.4. Online Self-Supervised Adversarial Finetuning

The challenge in online finetuning for UVOS lies in that we do not have any supervision on the test video. To resolve this, we propose a self-supervised tuning framework. We first take the weights of our base model and use them to initialize a teacher model and a student model. Motivated by [49] that the augmentation-averaged prediction usually provides a more accurate result than the individual model prediction, we take the averaged prediction from the teacher model as the pseudo-label to train the student model.

Given a video frame and its optical flow map $\mathbf{I}_{am} \in \mathbb{R}^{2\times H\times W\times 3}$ at the test time, we perform data augmentation including horizontal flip, color enhancement, multi-scale, and feed the augmented samples into the teacher model. We take the outputs from the teacher model and average them as the pseudo ground-truth mask $\mathbf{P}_t \in [0,1]^{H\times W}$. The student model is then optimized by the cross-entropy loss between the student and teacher predictions as,

$$\mathcal{L}_{CE}(\theta_s) = -\sum_{i,j} \mathbf{P}_{t,ij} \log \mathbf{P}_{s,ij}, \tag{4}$$

where θ_s is the model weights of the student model and $\mathbf{P}_s \in [0,1]^{H\times W}$ is the segmentation confidence map from the student model.

We further employ adversarial learning to align $\mathbf{P}_t$ and $\mathbf{P}_s$ to make them indistinguishable and improve the prediction quality. To this end, we design a similar light-weight discriminator in Section 3.3 for our online learning. The difference is the final logits are sent into a 1×1 convolution layer followed by upsampling and a sigmoid function to produce an output confidence map. Details of the model architecture can be found in the supplemental material. Given $\mathbf{P}_t$ and $\mathbf{P}_s$, we denote the discriminator outputs as $\tilde{\mathbf{P}}_t, \tilde{\mathbf{P}}_s \in [0,1]^{H\times W}$. Then an adversarial loss is defined as follows,

$$\mathcal{L}_{adv}(\theta_s) = \sum_{i,j} \log(\tilde{\mathbf{P}}_{t,ij}) + \log(1 - \tilde{\mathbf{P}}_{s,ij}), \tag{5}$$

where the discriminator tries to distinguish $\tilde{\mathbf{P}}_{t,ij}, \tilde{\mathbf{P}}_{s,ij}$, and the adversarial learning makes them indistinguishable.

Therefore, the overall loss for finetuning the student model is as follows,

$$\mathcal{L}(\theta_s) = \mathcal{L}_{CE}(\theta_s) + \lambda_a \mathcal{L}_{adv}(\theta_s), \tag{6}$$

where λ_a is a trade-off parameter.

It is difficult for the teacher model to effectively guide the student model if the weights of the teacher model are never updated. Therefore, we employ the exponential moving average method to update the weights of the teacher model θ_t using the weights of the student model θ_s,

$$\theta_t = \alpha\theta_t + (1-\alpha)\theta_s, \tag{7}$$

where α is a smoothing factor. This update takes place once the student model is online updated for every new test video frame. At each step, the student model is used to predict the segmentation map on the test frame.

4. Experiments

4.1. Implementation Details

The network architectures in our method can be found in the supplemental material. Our experiments follow the common practices as in [65, 68]. The training set consists of three parts: (a) all training data in DAVIS16 [43], which contains 30 videos and $2,000$ frames, (b) $10,000$ frames from Youtube-VOS [62], a subset of the training set of Youtube-VOS by sampling one frame every ten frames in each video, making sure its scale be similar as other data sets, (c) the test data of Youtube-objects [44] (independent of the test data in each evaluation dataset), which contains 126 videos and over $20,000$ frames. We use (a) and (b) as the labeled data while using (c) as the unlabeled data to train our model. All frames are resized to $384 \times 640 \times 3$, and the optical flow is estimated by RAFT [50]. The model is trained with the AdamW optimizer [30] with an initial learning rate of $1\text{e}-4$. The batch size and weight decay are set as 2 and $1\text{e}-2$, and the augmentation strategies including random horizontal flip, random cropping and random rotation covering a range of degrees $(-10, 10)$ are applied during training. The tradeoff parameters λ and λ_d are respectively set as 1 and $1\text{e}-4$ based on cross-validation. At the online test-time finetuning stage, we set the learning rate of the optimizer as $1\text{e}-5$, and set the coefficient α of the exponential moving average and the tradeoff parameter λ_a to 0.999 and $1\text{e}-4$ based on cross-validation. The data augmentation at this stage includes random horizontal flip, color enhancement and multi-scale resizing. Given a test frame, 3 augmented frames from the above three augmentation methods, together with the original frame itself, are used to infer augmentation-averaged prediction (after re-aligning the outputs from the original and the augmented frames). The offline training process takes 55 epochs, and the online fine-tuning takes 10 epochs. The proposed method is implemented in the framework of Pytorch 1.10.1 with two NVIDIA 2080TI GPUs and directly produces the binary segmentation mask without any post-processing technique.

4.2. Datasets and Evaluation Metrics

Our experiments are performed over five benchmark datasets commonly used for UVOS. Note that only the training data in DAVIS16 set is used for training our model, while the other data sets are all out-of-domain data for evaluating the online adaptation performance. Following the common practice in existing UVOS works [40, 68, 46], we only use part of Youtube-VOS in training, but do not use it for test evaluation, given that this data set is only used as a testbed for semi-supervised VOS [39]. Again, most evaluation data sets here are independent of the training data, which is a perfect setup for the zero-shot VOS.

DAVIS16 consists of 50 videos with high-quality dense pixel-level annotations, including 30 training videos and 20 test videos. We evaluate on the test set, and employ the standard metrics for this dataset including the region similarity $\mathcal{J}$, the boundary accuracy $\mathcal{F}$ and the overall $\mathcal{J}\&\mathcal{F}$ score that is the average of $\mathcal{J}$ and $\mathcal{F}$ scores as the evaluation metrics. Besides, we also report the video salient object detection [16, 5] performance in terms of structure-measure (S_a), F-measure (F_m), and Mean Absolution Error (MAE), as in [5, 16, 45].

FBMS contains 59 videos, including 29 videos for training and 30 videos for test. We evaluate on the test set, and report the standard metrics for this dataset including the region similarity $\mathcal{J}$, as well as S_a, F_m and MAE.

DAVSOD is a challenging dataset for the video salient object detection task. It contains a total of 96 videos, including 61 videos for training and 35 videos for testing. We report results on the test data in terms of S_a, F_m and MAE.

MCL contains 9 videos with a diverse set of objects and backgrounds, varying in video length from 30 frames to 100 frames. The evaluation metrics S_a, F_m and MAE are used to evaluate the performance of different methods over the whole dataset.

ViSal is created for the video salient object detection task and it consists of 17 videos with a total of 193 precisely annotated frames. The whole dataset is used for evaluation in terms of S_a, F_m and MAE.

4.3. Comparison with the State-of-the-Arts

Table 1 lists the comparison between our OAST method and the state-of-the-art UVOS methods on DAVIS16 and FBMS datasets. Following the common practice in UVOS, we only report $\mathcal{J}$ on FBMS [65, 9, 68]. From the results, we have the following observations: (1) Our method achieves the best performance over all evaluation metrics on both datasets. This shows that the online test-time finetuning is an essential step for improving the UVOS performance. (2) Although our method achieves similar performance as HFAN [40] on the DAVIS16 dataset, it significantly outperforms the HFAN on the FBMS dataset with a large margin of 6.9% absolute improvement. This may attribute to the model overfit to the DAVIS16 training data and validates the value of adding unlabeled data for improving the model generalizability. (3) Our method beats the MATNet [68], COSNet [31], AnDiff [64] and DBSNet [9], which take advantage of post-processing techniques such as CRF [19] and instance pruning to ensure the segmentation quality. This indicates that the adversarial learning strategies in our method produce high quality segmentation maps comparable to the sophisticated post-processing.

Methods	DAVIS16			FBMS
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$
COSNet (CVPR19) [31]	80.0	80.5	79.4	75.6
AGNN (ICCV19) [57]	79.9	80.7	79.1	-
AGS (CVPR19) [59]	78.6	79.7	77.4	-
EpO+ (WACV20) [1]	78.1	80.6	75.5	-
MATNet (AAAI20) [68]	81.6	82.4	80.7	76.1
DFNet (ECCV20) [67]	82.6	83.4	81.8	-
3DC-Seg (BMVC20) [32]	84.5	84.3	84.7	71.5
FSNet (ICCV21) [16]	83.3	83.4	83.1	-
F2Net (AAAI21) [27]	83.7	83.1	84.4	77.5
TransportNet (ICCV21) [65]	84.8	84.5	85.0	78.7
AMCNet (ICCV21) [63]	84.6	84.5	84.6	76.5
RTNet (CVPR21) [46]	85.2	85.6	84.7	-
CFANet (WACV22) [5]	82.8	83.5	82.0	-
D2Conv3D (WACV22) [47]	86.0	85.5	86.5	-
IMPNet (AAAI22) [20]	85.6	84.5	86.7	77.5
DBSNet (ACMMM22) [9]	85.3	85.9	84.7	78.5
HFAN (ECCV2022) [40]	86.7	86.2	87.1	76.1
TMO (WACV2023) [7]	86.1	85.6	86.6	79.9
OAST (Ours)	87.0	86.6	87.4	83.0

Table 1. Results on DAVIS16 and FBMS test sets, and the numbers in “red”, “blue”, and “green” indicate the top three performance.

Table 2 shows the performance of different methods in terms of S_a, F_m and MAE over all five datasets. From the results, we can see that (1) Our OAST method surpasses the other methods by large margins and reaches new state-of-the-art results on FBMS [37], DAVSOD [8] and MCL [18]. This again verifies the importance of online finetuning for UVOS, (2) Our method shows promising results on the out-of-domain datasets including DAVSOD [8], ViSal [58] and MCL [18]. Note that none of these datasets are used in training, and they have significant visual and semantic discrepancies from the training data. This validates the adaptability of our method to the domain shift in the novel data.

Figure 3 visualizes the segmentation results on some test videos. It is interesting to see that the cat and the horse object (in the top three rows) can be perfectly segmented even their appearance and the background environment evolve significantly over time. In the last three rows of Figure 3, we show the worm, blackswan and cheetah objects that are unseen categories in the training data. As seen, our method is still able to precisely segment all of them owing to its online finetuning strategy.

4.4. Ablation Study

In this section, we run experiments to study how the individual component of our OAST method contributes to the overall performance. We conduct analysis over DAVIS16 and FMBS and adopt the standard metrics on these datasets to measure the performance.

Methods	DAVIS16			FBMS			DAVSOD			ViSal			MCL		
	$S_a \uparrow$	$F_m \uparrow$	$MAE \downarrow$	$S_a \uparrow$	$F_m \uparrow$	$MAE \downarrow$	$S_a \uparrow$	$F_m \uparrow$	$MAE \downarrow$	$S_a \uparrow$	$F_m \uparrow$	$MAE \downarrow$	$S_a \uparrow$	$F_m \uparrow$	$MAE \downarrow$
SSAV (CVPR2019) [8]	89.3	86.1	2.8	87.9	86.5	4.0	72.4	60.3	9.2	94.3	93.9	2.0	81.9	77.3	2.6
AnDiff (ICCV2019) [64]	-	80.8	4.4	-	81.2	6.4	-	-	-	-	90.4	3.0	-	-	-
F^{3}Net (AAAI2020) [60]	85.0	81.9	4.1	85.3	81.9	6.8	68.9	56.4	11.7	87.4	90.7	4.5	-	-	-
MINet (CVPR2020) [38]	86.1	83.5	3.9	84.9	81.7	6.7	70.4	58.2	10.3	90.3	91.1	4.1	73.0	62.3	3.8
GateNet (ECCV2020) [66]	86.9	84.6	3.6	85.7	83.2	6.5	70.1	57.8	10.4	92.1	92.8	3.9	-	-	-
PCSA (AAAI2020) [11]	90.2	88.0	2.2	86.6	83.1	4.1	74.1	65.5	8.6	94.6	94.0	1.7	75.4	68.3	3.8
DFNet (ECCV2020) [67]	-	89.9	1.8	-	83.3	5.4	-	-	-	-	92.7	1.7	-	-	-
3DC-Seg (BMVC2020) [32]	-	91.8	1.5	-	84.5	4.8	-	-	-	-	92.2	1.9	66.4	72.1	4.1
CASNet (TNNLS2020) [17]	87.3	86.0	3.2	85.6	86.3	5.6	69.4	-	8.9	82.0	84.7	2.9	-	-	-
FSNet (ICCV2021) [16]	92.0	90.7	2.0	89.0	88.8	4.1	77.3	68.5	7.2	-	-	-	86.4	82.1	2.3
ReuseVOS (CVPR2021) [39]	88.3	86.5	1.9	88.8	88.4	2.7	-	-	-	92.8	93.3	2.0	75.4	67.9	3.7
CFANet (WACV2022) [5]	91.8	90.9	1.5	90.9	91.5	2.6	75.3	66.2	8.3	-	-	-	-	-	-
DBSNet (ACMMM2022) [9]	92.4	91.4	1.4	88.2	88.5	3.8	77.8	68.8	7.6	93.1	92.8	2.0	86.8	83.0	2.0
HFAN (ECCV2022) [40]	93.4	92.9	0.9	87.5	84.9	3.3	75.3	68.0	7.0	94.1	93.5	1.1	83.4	78.8	1.8
TMO (WACV2023) [7]	92.8	92.0	0.9	88.6	88.2	3.1	76.7	70.8	7.2	94.2	94.7	1.0	84.2	84.5	1.9
OAST (Ours)	93.5	92.6	1.1	91.7	91.9	2.5	78.6	71.2	7.0	94.8	95.0	1.0	88.9	87.0	1.5

Table 2. Results on DAVIS16, FBMS, DAVSOD, ViSal and MCL datasets, in which the numbers in "red", "blue", and "green" color indicate the top three performance respectively.

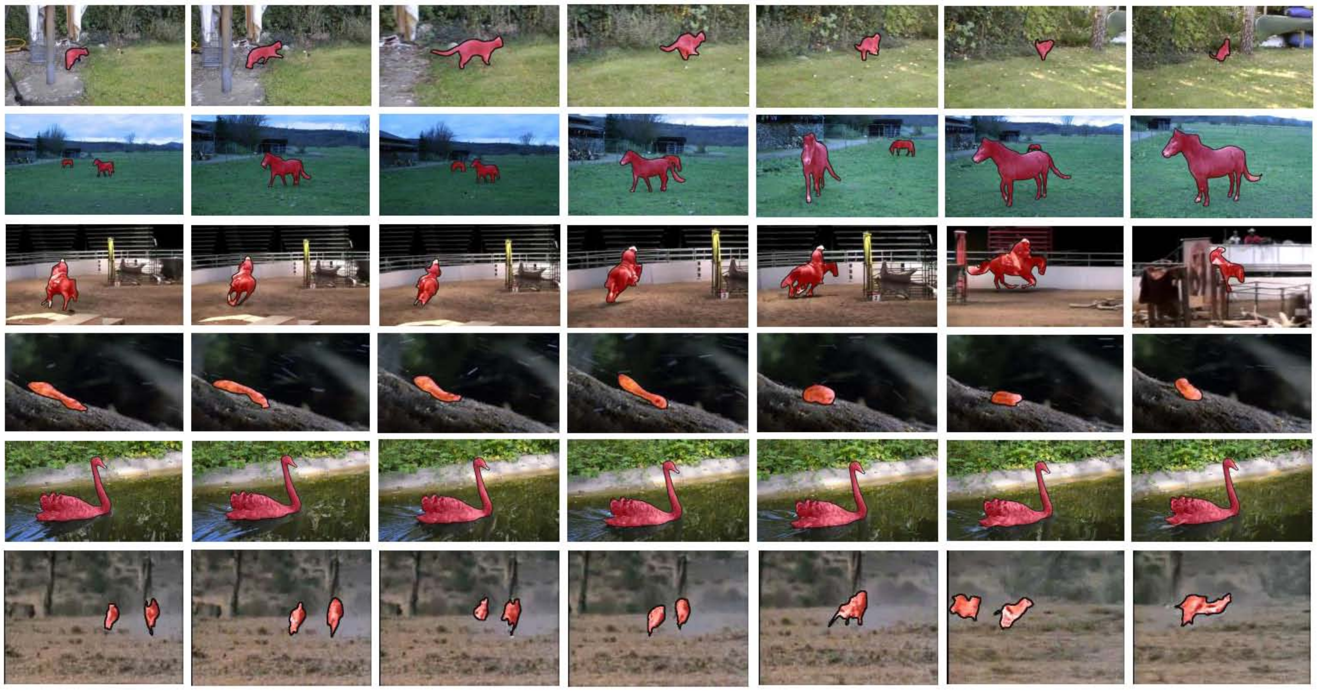

Figure 3. Exemplary segmentation results over test videos. The cat and horse objects in the first three rows can be precisely segmented even when their appearance and background evolve significantly over time. Meanwhile, the worm, blackswan and cheetah objects in the last three rows are also perfectly segmented even they are novel unseen categories from the training data.

Model	DAVIS16			FBMS
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$
Transformer Base Model	82.0	81.4	82.6	78.2
3D ResNet101 Base Model	80.9	80.5	81.3	76.1
Transformer Base Model w/ Offline Adv. Training	85.3	84.7	85.9	81.4
Transformer Base Model w/ Online Finetuning	84.0	83.4	84.5	80.5
OAST w/ 3D ResNet101	85.9	85.4	86.3	81.7
OAST w/ Transformer	87.0	86.6	87.4	83.0

Table 3. Results on DAVIS16 and FBMS test sets for different variants of the proposed OAST method, in which we try different base models between transformer and 3D ResNet101, and also evaluate the impact of each individual component in OAST.

Impact of transformer as base segmentation model. As shown in Section 3.2, we adopt the transformer-based 3D MobileViT as our base segmentation model. To verify its impact, we replace it with 3D ResNet101 [13], and get a CNN-based base segmentation model. Following the same strategy adapting MobileViT into our base model, we convert the 3D ResNet101 into an encoder-decoder architecture and append a 1×1 convolution to infer a segmentation map. We compare these two base models and report the results in rows 1-2 in Table 3. As seen, 3D ResNet101 drops the performance by 1.1% in terms of $\mathcal{J}\&\mathcal{F}$ on DAVIS16, and by 2.1% in terms of $\mathcal{J}$ on FBMS. Apparently, the transformer better models the long-range dependencies in the video content, leading to better results. The results in row 5 of Table 3

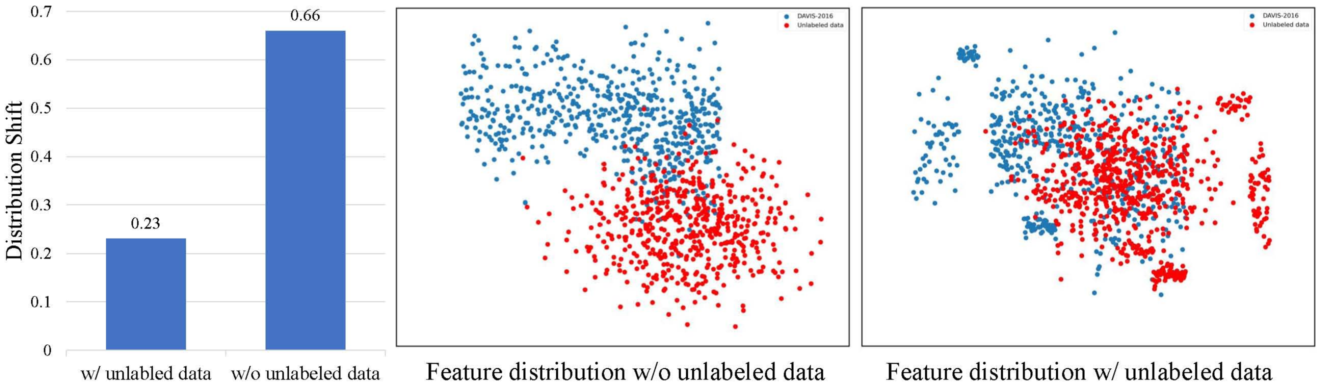

Figure 4. The first column shows the distribution shift metrics between the DAVIS16 and the unlabeled data with and without adding unlabeled data for offline adversarial training. The last two columns visualize the feature distribution with and without unlabeled data over DAVIS-2016 and unlabeled data.

show the result of our OAST method with 3D ResNet101 as the base model, which is worse than the result with transformer as shown in row 6.

Impact of offline adversarial training. On top of the transformer-based base segmentation model, we further add the offline adversarial training step as described in Section 3.3, and the results can be found in row 3 of Table 3. As seen, comparing to the base model (row 1), the offline adversarial training further improves the performance by 3.3% in terms of $\mathcal{J}\&\mathcal{F}$ on DAVIS16 and 3.2% in terms of $\mathcal{J}$ on FBMS. This verifies that the unlabeled data help alleviate model overfit to the labeled data and generalize better to the test data in new domain. Figure 4 displays distribution shift metrics (column 1) based on [21] and the intermediate feature distributions trained with (column 2) and without unlabeled data (column 3) over DAVIS-2016 and unlabeled data. As seen, the model trained with unlabeled data exhibits a distribution shift metric of 0.23 between DAVIS-2016 and the unlabeled data, significantly smaller than the model trained without unlabeled data. Furthermore, the feature distributions become much closer, indicating that adding unlabeled data via offline adversarial learning has effectively reduced the distribution shift.

Impact of online finetuning. Now we add online finetuning on top of our base segmentation model and show the result in row 4 of Table 3. As seen, comparing to the base model (row 1), the online finetuning improves the results by 2.0% in terms of $\mathcal{J}\&\mathcal{F}$ on DAVIS16 and 2.3% in terms of $\mathcal{J}$ on FBMS. This demonstrates the validity of our online finetuning strategy that could adapt to the new data distribution and ensure the model generalizability. Note that the performance of just adding online fine-tuning is worse than only adding unlabeled data for training. This is due to the fact that the model trained without unlabeled data tends to overfit to the labeled data, which confirms the necessity of offline adversarial training for online fine-tuning.

Model	DAVIS16			FBMS
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$
w/o adversarial learning	85.8	85.2	86.4	81.9
w/ adversarial learning	87.0	86.6	87.4	83.0

Table 4. Results on DAVIS16 and FBMS test sets with and without adversarial learning in OAST online finetuning.

Impact of all components in OAST. Adding all components together, our OAST method outperforms the base segmentation by 5.0% on DAVIS16 in terms of $\mathcal{J}\&\mathcal{F}$ and by 4.8% on FBMS in terms of $\mathcal{J}$ (row 6 vs row 1 in Table 3). It also outperforms all other variants in Table 3. This indicates that all of the proposed components are essential to the overall success of our OSAT method.

Impact of adversarial learning in OAST online finetuning. We also verify the impact of adversarial learning in the online finetuning described in Section 3.4, and the result can be seen in Table 4. As seen, without adversarial learning, the OAST performance drops by about 1% over all metrics on both datasets. This verifies the value of aligning the segmentation maps from the teacher and the student model to improve their consistency. It is worth noting that our method without adversarial learning boils down to the general online test-time adaptation method CoTTA [56], and adding adversarial learning makes our method more tailored to the UVOS task, leading to better performance.

5. Conclusion

We presented an online finetuning method for UVOS, which consists of an offline semi-supervised adversarial training and an online self-supervised online finetuning. At the training stage, we add the unlabeled videos to improve the model generalizability and use a light-weight discriminator to align the features of labeled and unlabeled training videos via adversarial learning. At the test stage, we initialize a teacher model and a student model, and use the augmentation-averaged prediction from the teacher to supervise the student model to adapt to the emerging patterns in the test videos, during which a discriminator is trained in an adversarial manner to minimize the gap between the predictions from the two models. Through online finetuning, the student model is trained by optimizing adversarial loss and consistency loss and the teacher model is updated with the student model weights via exponential moving average. Extensive experiments on five popular benchmark datasets have demonstrated the effectiveness of proposed method, substantially outperforming the state-of-the-art methods.

References

[1] Ijaz Akhter, Mohsen Ali, Muhammad Faisal, and Richard Hartley. Epo-net: Exploiting geometric constraints on dense trajectories for motion saliency. In *WACV*, 2020.

[2] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, 2018.

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[5] Yi-Wen Chen, Xiaojie Jin, Xiaohui Shen, and Ming-Hsuan Yang. Video salient object detection via contrastive features and attention modules. In *WACV*, 2022.

[6] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017.

[7] Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Chaewon Park, Donghyeong Kim, and Sangyoun Lee. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In *WACV*, 2023.

[8] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, 2019.

[9] Jiaqing Fan, Tiankang Su, Kaihua Zhang, and Qingshan Liu. Bidirectionally learning dense spatio-temporal feature propagation network for unsupervised video object segmentation. In *ACM MM*, 2022.

[10] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[11] Yuchao Gu, Lijuan Wang, Ziqin Wang, Yun Liu, Ming-Ming Cheng, and Shao-Ping Lu. Pyramid constrained self-attention network for fast video salient object detection. In *AAAI*, 2020.

[12] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Domain adaptive video segmentation via temporal consistency regularization. In *ICCV*, 2021.

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018.

[14] Ping Hu, Gang Wang, Xiangfei Kong, Jason Kuen, and Yap-Peng Tan. Motion-guided cascaded refinement network for video object segmentation. In *CVPR*, 2018.

[15] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *NIPS*, 2021.

[16] Ge-Peng Ji, Keren Fu, Zhe Wu, Deng-Ping Fan, Jianbing Shen, and Ling Shao. Full-duplex strategy for video object segmentation. In *ICCV*, 2021.

[17] Yuzhu Ji, Haijun Zhang, Zequn Jie, Lin Ma, and QM Jonathan Wu. Casnet: A cross-attention siamese network for video salient object detection. *TNNLS*, 2020.

[18] Hansang Kim, Youngbae Kim, Jae-Young Sim, and Chang-Su Kim. Spatiotemporal saliency detection for video sequences based on random walk with restart. *TIP*, 2015.

[19] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011.

[20] Youngjo Lee, Hongje Seong, and Euntai Kim. Iteratively selecting an easy reference frame makes unsupervised video object segmentation easier. In *AAAI*, 2022.

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

[22] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020.

[23] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C-C Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *CVPR*, 2018.

[24] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, 2018.

[25] Yuxi Li, Ning Xu, Jinlong Peng, John See, and Weiyao Lin. Delving into the cyclic mechanism in semi-supervised video object segmentation. *NIPS*, 2020.

[26] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

[27] Daizong Liu, Dongdong Yu, Changhu Wang, and Pan Zhou. F2net: Learning to focus on the foreground for unsupervised video object segmentation. In *AAAI*, 2021.

[28] Weide Liu, Guosheng Lin, Tianyi Zhang, and Zichuan Liu. Guided co-segmentation network for fast video object segmentation. *TCSVT*, 2020.

[29] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *NIPS*, 2016.

[30] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018.

[31] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019.

[32] Sabarinath Mahadevan, Ali Athar, Aljoša Ošep, Sebastian Hennen, Laura Leal-Taixé, and Bastian Leibe. Making a case for 3d convolutions for object segmentation in videos. In *BMVC*, 2020.

[33] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *TPAMI*, 2018.

[34] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2022.

[35] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *ECCV*, 2020.

[36] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambach, Evgeny Levinkov, Thomas Brox, and Jan Hendrik Metzen. Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999*, 2021.

[37] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 2013.

[38] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *CVPR*, 2020.

[39] Hyojin Park, Jayeon Yoo, Seohyeong Jeong, Ganesh Venkatesh, and Nojun Kwak. Learning dynamic network using a reuse gate function in semi-supervised video object segmentation. In *CVPR*, 2021.

[40] Gensheng Pei, Fumin Shen, Yazhou Yao, Guo-Sen Xie, Zhenmin Tang, and Jinhui Tang. Hierarchical feature alignment network for unsupervised video object segmentation. In *ECCV*, 2022.

[41] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, 2018.

[42] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017.

[43] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[44] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[45] Sucheng Ren, Chu Han, Xin Yang, Guoqiang Han, and Shengfeng He. Tenet: Triple excitation network for video salient object detection. In *ECCV*, 2020.

[46] Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He. Reciprocal transformations for unsupervised video object segmentation. In *CVPR*, 2021.

[47] Christian Schmidt, Ali Athar, Sabarinath Mahadevan, and Bastian Leibe. D2conv3d: Dynamic dilated convolutions for object segmentation in videos. In *WACV*, 2022.

[48] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.

[49] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NIPS*, 2017.

[50] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[51] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017.

[52] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[53] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, 2019.

[54] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017.

[55] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

[56] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, 2022.

[57] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, 2019.

[58] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. *TIP*, 2015.

[59] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven CH Hoi, and Haibin Ling. Learning unsupervised video object segmentation through visual attention. In *CVPR*, 2019.

[60] Jun Wei, Shuhui Wang, and Qingming Huang. F^{3}net: fusion, feedback and focus for salient object detection. In *AAAI*, 2020.

[61] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, 2018.

[62] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

[63] Shu Yang, Lu Zhang, Jinqing Qi, Huchuan Lu, Shuo Wang, and Xiaoxing Zhang. Learning motion-appearance co-attention for zero-shot video object segmentation. In *ICCV*, 2021.

[64] Zhao Yang, Qiang Wang, Luca Bertinetto, Weiming Hu, Song Bai, and Philip HS Torr. Anchor diffusion for unsupervised video object segmentation. In *ICCV*, 2019.

[65] Kaihua Zhang, Zicheng Zhao, Dong Liu, Qingshan Liu, and Bo Liu. Deep transport network for unsupervised video object segmentation. In *ICCV*, 2021.

[66] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, 2020.

[67] Mingmin Zhen, Shiwei Li, Lei Zhou, Jiaxiang Shang, Haoan Feng, Tian Fang, and Long Quan. Learning discriminative feature with crf for unsupervised video object segmentation. In *ECCV*, 2020.

[68] Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao. Motion-attentive transition for zero-shot video object segmentation. In *AAAI*, 2020.

[69] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.