Depth-aware Test-Time Training for Zero-shot Video Object Segmentation

Weihuang Liu[1]   Xi Shen[2]   Haolun Li[1]   Xiuli Bi[3]   Bo Liu[3]   Chi-Man Pun[1,*]   Xiaodong Cun[4,*]

[1] University of Macau   [2] Intellindust

[3] Chongqing University of Posts and Telecommunications   [4] Tencent AI Lab

Abstract

Zero-shot Video Object Segmentation (ZSVOS) aims at segmenting the primary moving object without any human annotations. Mainstream solutions mainly focus on learning a single model on large-scale video datasets, which struggle to generalize to unseen videos. In this work, we introduce a test-time training (TTT) strategy to address the problem. Our key insight is to enforce the model to predict consistent depth during the TTT process. In detail, we first train a single network to perform both segmentation and depth prediction tasks. This can be effectively learned with our specifically designed depth modulation layer. Then, for the TTT process, the model is updated by predicting consistent depth maps for the same frame under different data augmentations. In addition, we explore different TTT weight updating strategies. Our empirical results suggest that the momentum-based weight initialization and looping-based training scheme lead to more stable improvements. Experiments show that the proposed method achieves clear improvements on ZSVOS. Our proposed video TTT strategy provides significant superiority over state-of-the-art TTT methods. Our code is available at: https://nifangbaage.github.io/DATTT/.

1. Introduction

Zero-shot video object segmentation (ZSVOS) is a fundamental task in computer vision, aiming to accurately segment the primary moving object within a video. The term "zero-shot" means no human guidance is provided during the inference, which is different from one-shot video object segmentation (OSVOS), where the annotation of the first frame is provided. The problem is of great importance due to its wide applications in video understanding [24], video surveillance [15], video editing [34, 58], *etc*.

However, this task remains challenging, as the model is required to learn category-agnostic features to detect moving objects. Despite being trained on the most widely used public dataset ($\sim$3.5K videos), the deployed models often
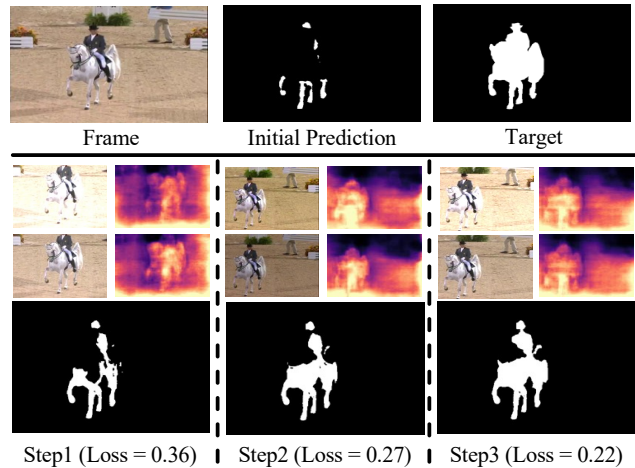
*Corresponding Author

Figure 1. **Key idea of our Depth-aware Test-Time Training.** During the test-time training, the model is required to predict consistent depth maps for the same video frame under different data augmentation (2nd row). The model is progressively updated and provides more precise mask prediction (3rd row).

fail when confronted with real-world scenarios. The above issue might be attributed to the lack of large-scale training sets. However, collecting and annotating large-scale video datasets is costly. As an alternative solution, test-time training (TTT) has emerged as a promising approach, which serves as the main focus of this work. TTT consists of conducting training on each test video. Therefore, the model is expected to automatically adapt to new scenarios.

Leading studies [11, 22, 39] on TTT are mainly focused on image recognition and have showcased that a well-designed self-supervised objective function enables the model to adapt to new distributions, thereby achieving an improved fit for individual test samples. These approaches are then extended to videos for OSVOS [1] and video semantic segmentation [47, 49], and demonstrated to be effective on unusual weather scenarios (snow, fog, rain, *etc*.). Different from these works, our key idea in this work (see Figure 1) is to leverage depth information on video TTT. Our motivation comes from the fact that the primary object should be close to the camera and thus has a relatively small depth, *i.e.* the

depth map should contain informative signals to segment primary moving objects.

In this study, we propose a novel framework named Depth-aware Test-Time Training (DATTT) for ZSVOS. Similar to other video TTT frameworks [1, 47, 49], our DATTT is a two-stage training framework. In the first stage, we learn a single model that jointly predicts the mask of the moving object as well as the depth map of the entire image. To achieve this, the model is built upon the commonly used two-stream ZSVOS model by incorporating a depth decoder that utilizes image features to predict the depth map. Therefore, the model has a shared image encoder, a flow encoder, and different decoder heads for each task. We also find that the interaction between these two tasks through a depth modulation layer enables better performance when performing TTT. In the second TTT stage, given an input video, the model is required to predict consistent depth maps under different data augmentations for each frame. Through optimizing the consistency loss, the image encoder part of the model is updated, leading to adaptive predictions on the mask head. Note that the depth supervision in the first stage comes from monocular depth predictors [2, 12, 57], thus it is free but assumed to be noisy.

Experimentally, we evaluate our DATTT on five widely-used ZSVOS datasets: DAVIS-16 [31], FBMS [28], Long-Videos [20], MCL [17], and SegTrackV2 [19]. Our empirical results suggest that both the first stage training and the TTT training benefit from the additional depth information introduced in the network. The improvement is clear and consistent across different depth predictors and the proposed depth modulation layer also provides important performance gain for TTT. We also explore different TTT strategies and ultimately find that the momentum-based weight initialization and looping-based training scheme lead to more consistent improvement. In terms of fair comparison to competitive TTT approaches [11, 36, 39, 48], our DATTT enables more stable improvement for TTT and manages to achieve significantly better performance. Although very different from state-of-the-art ZSVOS approaches, our DATTT still provides competitive performance, demonstrating the effectiveness of conducting TTT during inference.

To summarize, our main contributions are as follows:

- We introduce the Depth-aware Test-Time Training (DATTT) for zero-shot video object segmentation (ZSVOS). To the best of our knowledge, for the first time, demonstrating that performing TTT with consistent depth constraint brings significant improvement.
- We propose a depth modulation layer which enables interaction between the depth prediction head and mask prediction head and has shown to be effective for TTT process.
- Our DATTT achieves competitive performances compared to state-of-the-art approaches on ZSVOS, demonstrating

the effectiveness of performing TTT during inference.

2. Related Work

Zero-shot Video Object Segmentation. Zero-shot video object segmentation (ZSVOS) is a task aimed at segmenting the primary moving objects without requiring any annotations during inference. Traditionally, heuristic algorithms, including background subtraction [8, 63], object proposal [18, 55], and point trajectories [3, 27], have been commonly employed to address ZSVOS. With the rapid advancements in deep learning, neural networks have emerged as the most popular technique for ZSVOS [42, 50, 62]. To exploit the temporal information in the video sequence, early works [41, 45, 51] have developed recurrent neural network (RNN)-based models to leverage the correlations between successive frames. More recent research [6, 29, 30, 54, 56] has focused on incorporating motion information, resulting in significant performance improvements. Off-the-shelf optical flow estimation methods [37, 40] are utilized to extract motion cues, which are then combined with appearance information in a two-stream model. For example, Yang et al. [54] propose an attentive multi-modality collaboration network that integrates appearance and motion information using a co-attention mechanism. This multi-modality feature fusion suppresses misleading information and emphasizes the relevant foreground features. Since the optical flow which represents motion information for all pixels often fails to align well with the primary objects, Pei et al. [29] introduce a hierarchical feature alignment network that aligns appearance and motion features hierarchically using distinct modules.

Depth-based Object Segmentation. The depth map obtained by depth sensors offers valuable geometric insights for scene understanding. Multi-modal features extracted from RGB images and depth maps provide complementary information in both appearance and spatial position. Depth map has been shown to be beneficial in salient object detection [5, 9, 32, 33, 38, 59, 61] since it provides discriminative information in spatial structure. Chen et al. [5] propose a complementarity-aware fusion module to exploit cross-modal information. Zhao et al. [59] propose a contrast-enhanced network to bridge the RGB features and depth features and measure the contrast between salient and non-salient regions. Liu et al. [21] fuse the multi-modal information via spatial attention. The integration of depth information as input may hinder the practical application. To address this, Piao et al. [33] propose a depth distiller that transfers the depth knowledge to the RGB stream during training and only RGB input is necessary for testing. To the best of our knowledge, the use of depth maps in video object segmentation remains to be explored.

Test-Time Training. Previous works on test-time training [11, 22, 36, 39, 43, 44, 48] have demonstrated that fine-

tuning the pre-trained model for individual instances enables better adaptation to each specific instance. Wang *et al.* [48] minimize test entropy to adapt the normalization layers. Schneider *et al.* [36] replace the trained statics of the normalization layer by estimating from the test sample. Sun *et al.* [39] develop a Y-shape model containing the backbone, main head, and auxiliary head. During testing, they fine-tune the backbone by predicting the rotation degree using the auxiliary head. TTT-MAE [11] reconstructs the test image using the masked autoencoder to adapt the model to a new test distribution. Test-time training is also used in some one-shot video object segmentation (OSVOS) methods [4, 7, 46]. These kinds of methods further retrain the pre-trained model on a frame with annotation and then test it on the entire video sequence. Caelles *et al.* [4] propose the first online training-based OSVOS method. They first finetune the pre-trained model on the video with the first frame annotation, then test the entire video sequence using the new weights. Since the appearance of the object changes over time, Voigtlaender *et al.* [46] propose an online adaptation scheme. For each frame, they generate pseudo labels using the estimated masks and a threshold to adapt the model into the current frame. Ci *et al.* [7] pre-scan the whole video and generate pseudo labels, then retrain the model based on these labels. Some works [1, 47] discuss some existing image test-time training methods in handling challenging videos with human corruption. A concurrent work [49] extends TTT-MAE to video streams. The model is initialized from the previous model and trained on a set of available frames.

3. Preliminaries

Test-time Training. Test-time training (TTT) [11, 22, 39] aims to adapt the pre-trained model to a new test distribution with a well-designed objective function without supervision. A commonly used TTT network includes a shared encoder $\mathcal{E}$ and two decoders for the main task $\mathcal{D}_{main}$ and self-supervised task $\mathcal{D}_{ssl}$. Typical TTT framework involves two-stage training. In the first stage, the network is trained with the main loss $\mathcal{L}_{\text{main}}$ and self-supervised loss $\mathcal{L}_{\text{ssl}}$:

$$\min_{\mathcal{E}, \mathcal{D}_{main}, \mathcal{D}_{ssl}} \mathcal{L}_{\text{main}} + \lambda \mathcal{L}_{\text{ssl}}, \quad (1)$$

where λ is a hyper-parameter to balance the two components.

In the second stage, which is referred as test-time training (TTT), for each individual input, the encoder $\mathcal{E}$ will be fine-tuned according to the self-supervised objective: $\min_{\mathcal{E}} \mathcal{L}_{\text{ssl}}$.

Another natural choice is to fine-tune also the $\mathcal{D}_{ssl}$. Empirically, the difference is negligible between only fine-tuning $\mathcal{E}$ and fine-tuning the extra $\mathcal{D}_{ssl}$ [39].

Zero-shot Video Object Segmentation. Zero-shot video object segmentation (ZSVOS) aims to localize the moving object in the video without any guidance during inference. Current ZSVOS models [6, 29, 35] consists of an image

encoder for visual feature extraction, a flow encoder for motion information, and a decoder to get the mask prediction. Given a video frame and its optical flow map, the image and flow encoder extract the multi-scale image and flow features, respectively. The aggregation of the image and flow features is used in the decoder to decode the object mask. The ground truth object mask is used to supervise the model via the binary cross-entropy loss, which serves as the main loss $\mathcal{L}_{\text{main}}$ if considering TTT on ZSVOS.

4. Depth-aware Test-time Training for ZSVOS

In this section, we present our Depth-aware Test-time Training (DATTT) for ZSVOS. Our entire framework is illustrated in Figure 2. Our DATTT is designed following TTT-Rot [39] and TTT-MAE [11]. During the first stage training, which is denoted as training-time training, DATTT is trained on a large-scale dataset to perform two tasks jointly: primary moving object segmentation and depth estimation, which are realized by the main task decoder $\mathcal{D}_m$ and the depth decoder $\mathcal{D}_d$ respectively. During the TTT, the model is required to produce consistent depth maps between two augmented samples. The error is used to update the image encoder $\mathcal{E}_v$ to better understand the current scene, thus is expected to give better mask prediction.

We start with training-time training in Section 4.1 to introduce our first stage training. We then detail our TTT on videos in Section 4.2.

4.1. Training-time Training

Objective function. Given an input frame v and flow f, we first extract their features through an image encoder $\mathcal{E}_v$ and the flow encoder $\mathcal{E}_f$, as illustrated in Figure 2 (a). The encoded image and flow features are aggregated to predict the primary moving object through the mask decoder $\mathcal{D}_m$. In this work, we leverage a simple summation as the aggregation of image and flow features. The depth decoder $\mathcal{D}_d$ estimates the depth map using the image features. We use the depth maps obtained by off-the-shelf monocular depth estimation methods [2, 12, 57] as pseudo ground truth d. The impact of different monocular depth estimation methods are provided in Section 5.3.

Denoting the ground-truth mask as m. The total objective function can be formulated as:

$$\mathcal{L}_{\text{bce}}(\mathcal{D}_m(\mathcal{E}_v(v) + \mathcal{E}_f(f)),) + \lambda \mathcal{L}_{\text{depth}}(\mathcal{D}_d(\mathcal{E}_v(v)), d) \quad (2)$$

where λ is the hyper-parameter to balance the two losses. $\mathcal{L}_{\text{depth}}$ is the standard scale-invariant log loss [10] for depth estimation following [2].

Depth-aware Modulation Layer. We introduce the depth-aware modulation layer, which is illustrated in Figure 3. The basic idea is to enable features in the mask decoder to receive information from features in the depth decoder.
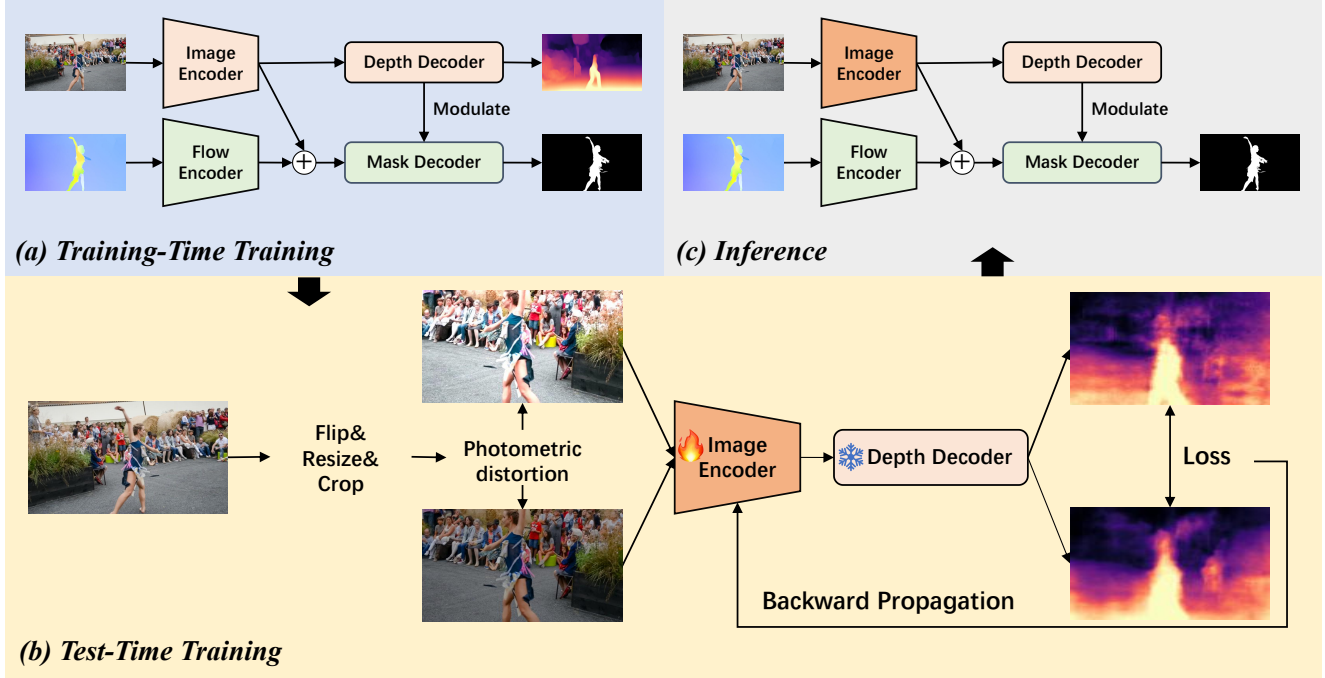
Figure 2. **Overview of the proposed Depth-aware Test-Time Training.** We add a depth decoder to commonly used two-stream ZSVOS architecture to learn 3D knowledge. The model is first trained on large-scale datasets for object segmentation and depth estimation. Then, for each test video, we employ photometric distortion-based data augmentation to the frames. The error between the predicted depth maps is backward to update the image encoder. Finally, the new model is applied to infer the object.
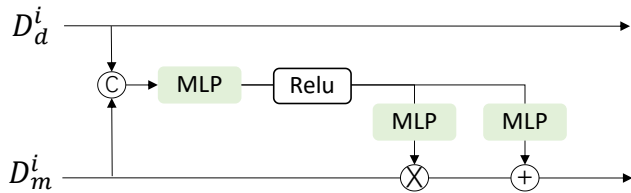

Figure 3. **The proposed depth-aware modulation layer**. At each scale i, we generate the modulation parameter by the depth feature $\mathcal{D}_d^i$ and the object feature $\mathcal{D}_m^i$ to modulate $\mathcal{D}_m^i$.

Denoting the i-th scale of the feature from the depth decoder and the mask decoder as $\mathcal{D}_d^i$ and $\mathcal{D}_m^i$ respectively, the depth-aware modulation layer is represented as $\mathcal{M}$ which serves to update $\mathcal{D}_m^i$:

$$\mathcal{D}_m^i = \mathcal{M}(\mathcal{D}_m^i, \mathcal{D}_d^i) \tag{3}$$

where $\mathcal{M}$ is composed of standard operators, such as MLP, Relu, concatenation, dot product, and summation, which can be seen in Figure 3.

4.2. Test-time Training on Videos

Depth-aware TTT. Given a test video with T frames $V = \{v_t | t \in [1, 2, ..., T]\}$, we conduct TTT to update the image encoder $\mathcal{E}_v$ by optimizing consistent depth map between a single frame under two data augmentation (Figure 2 (b)). The updated image encoder $\mathcal{E}_v$ is expected to be beneficial for the mask prediction (Figure 2 (c)).

Precisely, for the i-th frame v_i, we obtain two augmented images v_i^1 and v_i^2 by applying different data augmentation on v_i, which includes: random horizontal flip, resize, crop, and photometric distortion. Then for TTT, we seek to optimize:

$$\mathcal{L}_{\text{depth}}(\mathcal{D}_d(\mathcal{E}_v(v_i^1)), \mathcal{D}_d(\mathcal{E}_v(v_i^2))) \tag{4}$$

Note that we keep the $\mathcal{D}_d$ frozen and only fine-tune the image encoder $\mathcal{E}_v$, this is consistent with [39]. We also find that training $\mathcal{D}_d$ and $\mathcal{E}_v$ together cannot bring improvement. **Naive TTT strategy (TTT-N).** For each video, we train its own image encoder for each frame. This is a naive image test-time training strategy as Azimi *et al*. [1] (Figure 4 (b)). It treats the video as individual frames and adapts each frame by initializing the model with pre-trained weights. Although the model is adapted to each frame during testing, this strategy does not obtain additional benefits from the available frames in the video.

However, video is composed of a series of highly correlated images. The spatio-temporal correspondence in the video might boost the test-time training in the video data. Hence we introduce two effective strategies for video TTT: Momentum-based Weight Initialization (TTT-MWI) and
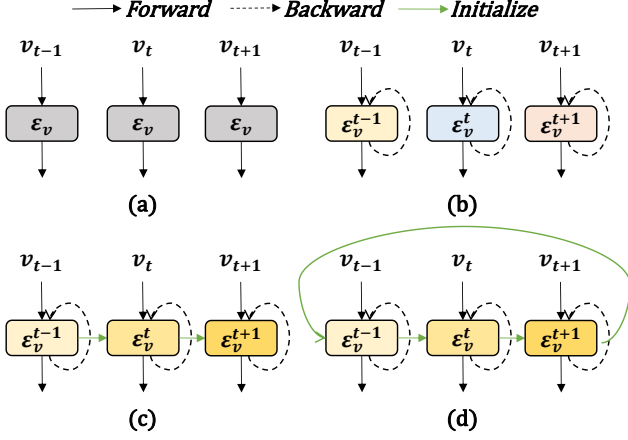
Figure 4. **A glance at different frameworks for ZSVOS described in Section 4.2.** (a) The previous ZSVOS methods directly apply the trained model to infer the test video. (b) Image-based test-time training methods (TTT-N) fine-tune the model on each individual frame. (c) Video test-time training by momentum-based weight initialization (TTT-MWI) trains the model based on past models. (d) Video test-time training by looping through the video (TTT-LTV) benefits from the global information.

Loop Through the Video (TTT-LTV). Note that the study of different strategies are provided in Section 5.3.

Momentum-based Weight Initialization (TTT-MWI). Since the scenes in consecutive frames in a video are highly similar, the model that has been finetuned on the past frame is more suitable to initialize than the pre-trained model when tuning in the current frame. Therefore, we adapt the model to the video following the temporal order, where the parameters of the image encoder $\mathcal{E}_v^t$ for t-th frame v_t is initialized by the previous $\mathcal{E}_v^{t-1}$ instead of the original weight $\mathcal{E}_v$, which can be seen from Figure 4 (c). In this way, the model is initialized by a better weight which has been adapted to the current scene based on past frames. The same strategy is also demonstrated to be efficient in the related work [47]. However, they only discuss TTT in online video streaming, here we further explore TTT in offline video.

Loop Through the Video (TTT-LTV). The model is trained on each arrival frame for several epochs in the online setting. The model benefits from retaining information in past frames by momentum-based weight initialization since the past information is helpful. In some offline settings (such as video editing), the entire video is available. To exploit more information in the entire video, we suggest performing video TTT by looping through the video rather than frame-by-frame. Instead of training several epochs in the current frame and then moving to the next frame, the model is adapted at each frame once and loops for several epochs in the video (Figure 4 (d)). The scene knowledge is accumulated epoch-by-epoch and then serves as the past and future knowledge for the model for the current frame to train in a global view.

5. Experiments

5.1. Datasets and Evaluation Metrics

We evaluate the proposed method on five widely-used datasets, including DAVIS-16 [31], FBMS [28], Long-Videos [20], MCL [17], and SegTrackV2 [19]. DAVIS-16 [31] contains a total of 50 videos with pixel-level annotations for each frame, including 30 videos for training and 20 videos for validation. FBMS [28] consists of 29 training videos and 30 testing videos, with only 720 annotated frames. Long-Videos [20] contains 3 long videos with each over 1,500 frames. MCL [17] is composed of 9 videos in the low-resolution. SegTrackV2 [20] involves 14 videos of fast motion and object deformation. Youtube-VOS [53] is used to train the model, which is a large-scale dataset including 3,471 videos. Region similarity $\mathcal{J}$ and boundary accuracy $\mathcal{F}$ are reported for evaluation. $\mathcal{J}$ is defined as:

$$\mathcal{J} = \left| \frac{m_{gt} \cap m_{pred}}{m_{gt} \cup m_{pred}} \right|, \quad (5)$$

where m_{gt} and m_{pred} are the ground truth mask and predicted mask, respectively. $\mathcal{F}$ can be calculated as:

$$\mathcal{F} = \frac{2 \times p \times r}{p + r}, \quad (6)$$

where $p = \left| \frac{m_{gt} \cap m_{pred}}{m_{pred}} \right|$ and $r = \left| \frac{m_{gt} \cap m_{pred}}{m_{gt}} \right|$.

5.2. Implementation Details

All the experiments are performed on a single NVIDIA A40 GPU. Random horizontal flipping, resizing, cropping, and photometric distortion are used for data augmentation. The input images are resized into 512×512. The model is pre-trained on the Youtube-VOS dataset for 10 epochs with setting $\lambda = 0.1$. The ablation of λ is provided in the appendix, Section E. During test-time training, we train the model for 10 epochs in each test video. The mini-batch size is set to 8. The model is optimized by the Adam optimizer with a learning rate of $6e^{-5}$ and $1e^{-5}$ for training-time training and test-time training. We choose Mit-b1 [52] and Swin-Tiny [23] as the image encoder and flow encoder. The depth decoder and segmentation decoder are implemented with the lightweight decoder in SegFormer [52]. RAFT [40] is used to extract the optical flow map. Monodepth2 [12], LiteMono [57], and ZoeDepth [2] are used to obtain the depth map. We use Mit-b1 as the backbone, MonoDepth2 as the depth extractor, and TTT-LTV as the TTT strategy in our default setting.

5.3. Analysis and Ablation Studies

Impact of Architecture Design and Depth Quality. We first verify the proposed depth-aware decoder. The baseline is set to the commonly used two-stream model. As shown
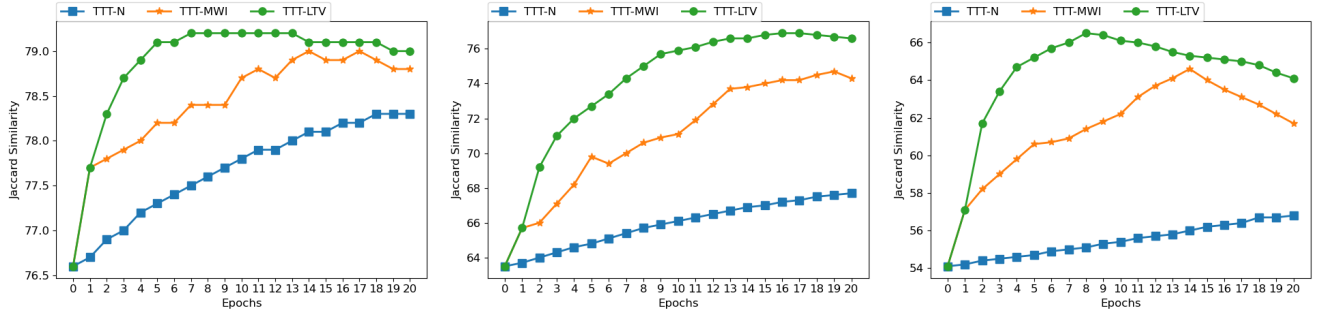
Figure 5. **The performance varies with the number of training epochs on FBMS [28], Long-Videos [20], MCL [17] datasets.** The proposed strategy (TTT-LTV introduced in Section 4.2) requires less time for the model to adapt to the target video on the three datasets and achieves better results.

Backbone	$\mathcal{D}_d$	Mod.	TTT	DAVIS-16	FBMS	Long.
Mit-b1 [52]	-	-	-	75.9	75.1	63.9
	✓	-	-	77.0	77.5	62.8
	✓	-	✓	77.2	**78.0**	70.5
	✓	✓	-	77.1	73.7	65.2
	✓	✓	✓	**77.5**	76.9	**73.1**
Swin-T [23]	-	-	-	77.8	74.1	65.7
	✓	-	-	78.7	74.5	67.0
	✓	-	✓	78.8	75.0	72.1
	✓	✓	-	79.0	76.6	63.5
	✓	✓	✓	**79.2**	**79.2**	**75.9**

Table 1. **Ablation study for the proposed depth-aware decoder on DAVIS-16 [31], FBMS [28], and Long-Videos [20] datasets.** $\mathcal{J}$ is reported for comparison. Taking depth as additional supervision ($\mathcal{D}_d$) boosts performance, and the modulation layer (Mod.) obtains a more significant improvement during TTT.

Depth Extractors	TTT	DAVIS-16	FBMS	Long.
w/o depth	-	75.9	75.1	63.9
MonoDepth2 [12]	-	77.1	73.7	65.2
	✓	+0.4	+3.2	+7.9
LiteMono [57]	-	76.8	79.0	68.1
	✓	+2.0	+1.5	+6.3
ZoeDepth [2]	-	79.9	76.4	64.0
	✓	+0.5	+4.7	+9.5

Table 2. **Ablation study using different depth estimation methods on DAVIS-16 [31], FBMS [28], and Long-Videos [20] datasets.** $\mathcal{J}$ is reported for comparison. DATTT shows consistent improvements using different depth estimation methods.

in Table 1, our method is better than the baseline under two different backbones. The results are intuitive since depth information is beneficial to segment the primary object. In addition, the depth modulation layer is more effective in both training-time training and test-time training. This can be attributed to the updated depth features further facilitating object segmentation via feature modulation in the decoders.

We also experiment with the depth maps obtained by different depth estimation methods as supervision. Table 2 shows that the proposed DATTT gains consistent improvement across different methods. It demonstrates that utilizing 3D information in the given video to finetune the model is effective for ZSVOS.

Test-Time Training Strategy. We discuss different test-time training schemes proposed in Section 4.2. As shown in Table 3, they obtain consistent improvement in several datasets compared to directly testing on the video, which proves that the depth-aware test-time training is effective in ZSVOS. However, the performances are significantly varying in different strategies. First, treating the video as a whole rather than individual frames greatly improves the performance of video test-time training. Utilizing the parameters of the previous frame without resetting the parameter to the pre-trained keeps the model remembering the past scenes. It becomes easier to find the primary object because of the temporal smoothness. Additionally, it is useful to make the model iteratively train on the video in the offline setting. As the iteration goes on, both past and future information are available. The frames in the front of the video can be further refined after snooping future information.

Training Epoch. One crucial challenge in TTT is the additional time required for training. We investigate the impact of training epochs on performance. The results are presented in Figure 5. As the number of training epochs increases, the disparity between TTT-LTV and other schemes (TTT-N and TTT-MWI) becomes pronounced, indicating that the proposed approach requires less time to adapt to a given video. Furthermore, we observe that the optimal number of training epochs varies for different datasets. However, a suitable epoch (such as 10) yields satisfactory results across diverse datasets. We present the results obtained by our method in a video sequence in Figure 6. Note that more visual results are provided in the appendix, Figure G. Initially, the pre-trained model shows limited accuracy in detecting the walking peo-

Backbone	TTT Scheme	DAVIS-16		FBMS		Long.		MCL		STV2	
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Mit-b1 [52]	-	77.1	78.4	73.7	75.8	65.2	68.0	53.5	66.2	61.5	69.2
	TTT-N	+0.3	+0.3	+0.1	+0.3	+1.3	+1.5	+1.9	+1.5	+1.0	+1.2
	TTT-MWI	+0.4	**+0.5**	+2.3	+2.1	+7.2	+7.5	+7.6	+6.9	+2.0	+4.0
	TTT-LTV	**+0.4**	+0.4	**+3.2**	**+3.1**	**+7.9**	**+7.7**	**+8.4**	**+7.8**	**+4.4**	**+4.3**
Swin-T [23]	-	79.0	80.3	76.6	79.3	63.5	70.0	54.1	68.2	64.0	70.7
	TTT-N	+0.1	+0.2	+1.2	+1.0	+2.6	+1.7	+1.3	+1.4	+0.4	+0.4
	TTT-MWI	**+0.3**	+0.4	+2.2	+1.7	+7.6	+5.7	+8.1	+6.0	+1.4	+0.6
	TTT-LTV	+0.2	**+0.4**	**+2.6**	**+2.0**	**+12.4**	**+9.2**	**+12.0**	**+8.1**	**+1.5**	**+0.8**

Table 3. **Ablation study on the proposed test-time training scheme on DAVIS-16 [31], FBMS [28], Long-Videos [20], MCL [17], and SegTrackV2 [19] datasets.** The proposed strategy is effective for test-time training in video.
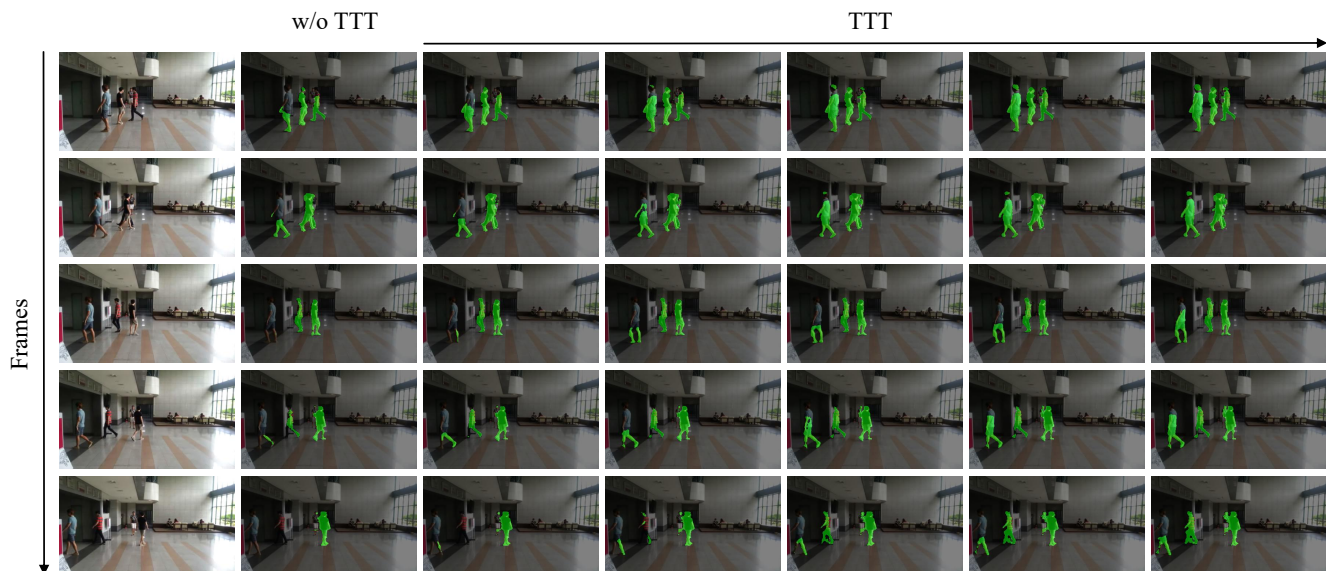

Figure 6. **Qualitative results of the proposed method.** The background in the results is dimmed for better visualization. The results obtained by the pre-trained model are less accurate and become better and better as TTT goes on.

ple in the video. As we apply depth-based TTT, we observe a progressive improvement in the subsequent results.

5.4. Comparison with the State-of-the-Art

We first compare with four prior methods for test-time training (TTT) in Table 4, including TENT [48], BN [36], TTT-Rot [39], and TTT-MAE [11]. In detail, we directly apply TTT to the baseline model for methods that do not incorporate an auxiliary head (TENT and BN). For methods that employ an auxiliary head (TTT-Rot, TTT-MAE, and ours), we train the respective baseline models and then perform TTT. All these methods follow the TTT strategy we have designed as described above.

It is observed that these methods do not yield substantial improvements in ZSVOS. The prior work [47] gets similar results, where these methods demonstrate promising performance in challenging weather conditions like rain or snow but fail to produce significant improvements in typical weather scenarios such as sunny. According to their analysis, the baseline model has achieved satisfactory performance by training on typical weather data, thereby limiting its potential for further enhancements. Instead, we attribute this limitation to no synthetic corruption in the target datasets, resulting in a minor domain shift between the training and test data. Severe domain shifts make the model easy to adapt to new samples via even solely updating the normalization statistics. Without clear domain shifts, the presence of task-specific cues becomes crucial to effectively guide TTT. In the context of ZSVOS, cues derived from 3D information prove to be advantageous, which explains the stable improvements exhibited by our proposed method when applied to test data.

Then, in auxiliary head-based TTT approaches, adding auxiliary tasks can not improve the baseline performance consistently. For instance, when the shared image encoder is required to learn high-level semantic representations through image reconstruction, it frequently undermines its low-level

Backbone	Method	TTT	DAVIS-16		FBMS		Long.		MCL		STV2	
			$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Mit-b1 [52]	Baseline	-	75.9	77.5	75.1	76.5	63.9	67.5	57.3	70.8	61.5	70.4
	TENT [48]	✓	−0.5	−0.4	+0.4	+0.4	+0.6	+0.5	+1.0	+0.4	−0.3	−0.4
	BN [36]	✓	+0.3	+0.3	+0.9	+1.0	+0.9	+0.6	+1.4	+1.3	+0.3	+0.3
	TTT-Rot [39]	-	75.3	76.2	75.4	77.2	59.1	62.8	57.7	70.5	66.4	73.3
		✓	−0.4	−0.1	+0.4	+0.5	+2.6	+1.7	+4.0	+3.8	−2.4	−1.6
	TTT-MAE [11]	-	73.5	74.1	74.6	75.7	64.4	67.5	55.7	66.8	62.2	70.2
		✓	+0.4	+0.3	+0.5	+0.1	+0.9	+0.1	−1.5	−0.9	−0.7	−0.4
	Ours	-	77.1	78.4	73.7	75.8	65.2	68.0	53.5	66.2	61.5	69.2
		✓	**+0.4**	**+0.4**	**+3.2**	**+3.1**	**+7.9**	**+7.7**	**+8.4**	**+7.8**	**+4.4**	**+4.3**
Swin-T [23]	Baseline	-	77.8	79.0	74.1	77.7	65.7	71.2	47.3	61.2	62.6	70.2
	TENT [48]	✓	−0.4	−0.5	+0.3	+0.3	+0.3	+0.3	+0.6	+0.5	−0.5	−0.7
	BN [36]	✓	−0.2	−0.3	+0.6	+0.7	+0.6	+0.6	+1.0	+0.9	+0.4	+0.2
	TTT-Rot [39]	-	78.9	79.9	75.5	78.5	66.7	70.6	57.0	69.7	63.3	70.4
		✓	**+0.7**	**+0.6**	−1.0	−0.1	−1.8	−2.3	+0.6	+0.4	+0.4	+0.3
	TTT-MAE [11]	-	77.0	77.9	74.6	77.2	65.3	69.2	52.8	65.5	60.3	67.9
		✓	−0.1	−0.1	−0.3	−0.1	−0.8	−0.6	−0.9	−0.8	−0.6	−0.4
	Ours	-	79.0	80.3	76.6	79.3	63.5	70.0	54.1	68.2	64.0	70.7
		✓	+0.2	+0.4	**+2.6**	**+2.0**	**+12.4**	**+9.2**	**+12.0**	**+8.1**	**+1.5**	**+0.8**

Table 4. **Comparisons with state-of-the-art test-time training method on DAVIS-16 [31], FBMS [28], Long-Videos [20], MCL [17], and SegTrackV2 [19] datasets.** Results that the dropped after TTT are masked as red. The most significant improvement is marked as **bold**. The proposed method obtains stable improvements in diverse datasets.

		3DCSEG [26]	AGNN [50]	MATNet [60]	HFAN [29]	HCPN [30]	MED-VT [16]	Ours	Ours
		3D ResNet-152 [13]	Resnet-101 [14]	Resnet-101 [14]	Mit-b1 [52]	Resnet-101 [14]	Video-Swin-B [25]	Mit-b1 [52]	Swin-T [23]
DAVIS-16	$\mathcal{J}$	84.3	80.7	82.4	**86.2**	85.8	85.9	86.0	85.8
	$\mathcal{F}$	84.7	79.1	80.7	87.1	85.4	86.6	87.9	**88.5**
FBMS	$\mathcal{J}$	76.2	-	76.1	76.1	78.3	-	74.9	**78.8**
Long.	$\mathcal{J}$	34.2	68.3	66.4	74.9	-	-	75.6	**77.3**
	$\mathcal{F}$	33.1	68.6	69.3	76.1	-	-	77.1	**79.9**

Table 5. **Comparison with state-of-the-art ZSVOS methods on DAVIS-16 [31], FBMS [28], and Long-Videos [20] datasets.** The proposed method is superior to other ZSVOS methods that directly test on videos.

features that are essential for object segmentation. The success observed in rotation prediction can be attributed to the additional data augmentation introduced by varying degrees of rotation. These auxiliary tasks both exhibit limited impact when employed to TTT for ZSVOS, while our depth-based method shows notable improvements.

We also compared the proposed method with SOTA ZSVOS methods [16, 26, 29, 30, 50, 60]. These methods are usually pre-trained on large-scale datasets and then finetuned on the DAVIS-16 training set. For a fair comparison, we finetune the model that has been pre-trained on Youtube-VOS on the DAVIS-16 training set and perform TTT on the FBMS and Long-Videos dataset. As shown in Table 5, our method obtains the best performance in 2 of 3 datasets. Note that our method involves no bells and whistles blocks for feature fusion, information propagation, attention mechanism, *etc.*, suggesting performing TTT is effective for ZSVOS.

6. Conclusion

In this work, we introduce Depth-aware Test-Time Training for ZSVOS, which allows a pre-trained model better generalize to unseen scenarios. We propose a joint learning framework that simultaneously addresses object segmentation and depth estimation. During the inference, the consistent depth for the same frame under different data augmentations serves as the criterion for updating the model. Furthermore, different TTT strategies are explored. The experimental results demonstrate the effectiveness of our proposed approach in comparison to SOTA TTT approaches. We also achieve competitive performance compared to other ZSVOS methods.

References

[1] Fatemeh Azimi, Sebastian Palacio, Federico Raue, Jörn Hees, Luca Bertinetto, and Andreas Dengel. Self-supervised test-time adaptation on video data. In *WACV*, 2022. 1, 2, 3, 4

[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv*, 2023. 2, 3, 5, 6, 11

[3] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 2

[4] 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. 3

[5] Hao Chen and Youfu Li. Progressively complementarity-aware fusion network for rgb-d salient object detection. In *CVPR*, 2018. 2

[6] 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. 2, 3

[7] Hai Ci, Chunyu Wang, and Yizhou Wang. Video object segmentation by learning location-sensitive embeddings. In *ECCV*, 2018. 3

[8] Dubravko Culibrk, Oge Marques, Daniel Socek, Hari Kalva, and Borko Furht. Neural network approach to background modeling for video object segmentation. *Transactions on Neural Networks*, 2007. 2

[9] Xiaodong Cun and Chi-Man Pun. Defocus blur detection via depth distillation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 747–763. Springer, 2020. 2

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 3

[11] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. *NeurIPS*, 2022. 1, 2, 3, 7, 8, 12

[12] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 2, 3, 5, 6, 11

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 8

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[15] Ruibing Hou, Bingpeng Ma, Hong Chang, Xinqian Gu, Shiguang Shan, and Xilin Chen. Vrstc: Occlusion-free video person re-identification. In *CVPR*, 2019. 1

[16] Rezaul Karim, He Zhao, Richard P Wildes, and Mennatullah Siam. Med-vt: Multiscale encoder-decoder video transformer with application to object segmentation. In *CVPR*, 2023. 8

[17] Hansang Kim, Youngbae Kim, Jae-Young Sim, and Chang-Su Kim. Spatiotemporal saliency detection for video sequences based on random walk with restart. *IEEE TIP*, 2015. 2, 5, 6, 7, 8, 12

[18] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 2

[19] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 2, 5, 7, 8, 11, 12

[20] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In *NeurIPS*, 2020. 2, 5, 6, 7, 8, 11, 12, 13

[21] Nian Liu, Ni Zhang, and Junwei Han. Learning selective self-mutual attention for rgb-d saliency detection. In *CVPR*, 2020. 2

[22] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *NeurIPS*, 2021. 1, 2, 3

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 5, 6, 7, 8

[24] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *ICCV*, 2021. 1

[25] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022. 8

[26] 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. *BMVC*, 2020. 8

[27] Peter Ochs and Thomas Brox. Higher order motion models and spectral clustering. In *CVPR*, 2012. 2

[28] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 2013. 2, 5, 6, 7, 8, 11, 12, 13

[29] 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. 2, 3, 8, 13

[30] Gensheng Pei, Yazhou Yao, Fumin Shen, Dan Huang, Xingguo Huang, and Heng-Tao Shen. Hierarchical co-attention propagation network for zero-shot video object segmentation. *IEEE TIP*, 2023. 2, 8

[31] 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. 2, 5, 6, 7, 8, 11, 12, 13

[32] Yongri Piao, Wei Ji, Jingjing Li, Miao Zhang, and Huchuan Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *ICCV*, 2019. 2

[33] Yongri Piao, Zhengkun Rong, Miao Zhang, Weisong Ren, and Huchuan Lu. A2dele: Adaptive and attentive depth distiller for efficient rgb-d salient object detection. In *CVPR*, 2020. 2

[34] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv:2303.09535*, 2023. 1

[35] Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He. Reciprocal transformations for unsupervised video object segmentation. In *CVPR*, 2021. 3

[36] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *NIPS*, 2020. 2, 3, 7, 8, 12

[37] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 2

[38] Peng Sun, Wenhu Zhang, Huanyu Wang, Songyuan Li, and Xi Li. Deep rgb-d saliency detection with depth-sensitive attention and automatic multi-modal fusion. In *CVPR*, 2021. 2

[39] 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. 1, 2, 3, 4, 7, 8, 12

[40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 5

[41] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017. 2

[42] Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari. Learning to segment moving objects. *IJCV*, 2019. 2

[43] Alessio Tonioni, Oscar Rahnama, Thomas Joy, Luigi Di Stefano, Thalaiyasingam Ajanthan, and Philip HS Torr. Learning to adapt for stereo. In *CVPR*, 2019. 2

[44] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo. In *CVPR*, 2019. 2

[45] 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. 2

[46] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 3

[47] Riccardo Volpi, Pau De Jorge, Diane Larlus, and Gabriela Csurka. On the road to online adaptation for semantic image segmentation. In *CVPR*, 2022. 1, 2, 3, 5, 7

[48] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2020. 2, 3, 7, 8, 12

[49] Renhao Wang, Yu Sun, Yossi Gandelsman, Xinlei Chen, Alexei A Efros, and Xiaolong Wang. Test-time training on video streams. *arXiv*, 2023. 1, 2, 3

[50] 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. 2, 8

[51] 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. 2

[52] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 2021. 5, 6, 7, 8, 11

[53] 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. 5

[54] 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. 2

[55] Dong Zhang, Omar Javed, and Mubarak Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. 2

[56] Kaihua Zhang, Zicheng Zhao, Dong Liu, Qingshan Liu, and Bo Liu. Deep transport network for unsupervised video object segmentation. In *ICCV*, 2021. 2

[57] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *CVPR*, 2023. 2, 3, 5, 6, 11

[58] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation. In *CVPR*, 2023. 1

[59] Jia-Xing Zhao, Yang Cao, Deng-Ping Fan, Ming-Ming Cheng, Xuan-Yi Li, and Le Zhang. Contrast prior and fluid pyramid integration for rgbd salient object detection. In *CVPR*, 2019. 2

[60] 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. 8

[61] Tao Zhou, Huazhu Fu, Geng Chen, Yi Zhou, Deng-Ping Fan, and Ling Shao. Specificity-preserving rgb-d saliency detection. In *ICCV*, 2021. 2

[62] Tao Zhuo, Zhiyong Cheng, Peng Zhang, Yongkang Wong, and Mohan Kankanhalli. Unsupervised online video object segmentation with motion property understanding. *IEEE TIP*, 2019. 2

[63] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 2006. 2