Appendix

This appendix contains the following sections:

- Section A: Efficiency analysis.
- Section B: Impact of different data augmentation for TTT (*c.f.* Section 4.2).
- Section C: TTT with depth supervision from depth predictor.
- Section D: Further study on Naive TTT (TTT-N) strategy (*c.f.* Section 4.2).
- Section E: Impact of the depth prediction loss weight λ (*c.f.* Section 4.1).
- Section F: Sampling strategy for densely annotated videos: DAVIS-16 [31] and SegTrackV2 [19].
- Section G: Extending DATTT to existing ZSVOS method.
- Section H: Additional visual results (*c.f.* Figure 6).

A. Efficiency analysis

When utilizing Mit-b1 [52] as our backbone, the parameters in each component are distributed as follows: 26.30M for the backbone, 0.53M for the segmentation head, 0.53M for the depth head, and 1.05M for the depth-aware modulation layer. Consequently, we introduce only a minimal number of additional parameters to the baseline model. The baseline model, consisting of the backbone with the segmentation head, requires 1.4 hours per epoch for training and 8 milliseconds per frame for inference. In contrast, our full model, which includes the baseline, the depth head, and the depth-aware modulation layer, demands 1.5 hours per epoch for training, 9 milliseconds per frame for inference, and 49 milliseconds per frame for TTT. The power consumption on the GPU is recorded as 230W for training, 80W for inference, and 200W for TTT, indicating that the additional computation cost to the basic training and inference processes is minimal. While the incorporation of TTT adds to the computational time, it is an inherent drawback of the technique, and we strive to mitigate its impact. It is worth noting that our proposed strategy is significantly faster and more effective than the naive strategy (*c.f.* Figure 5).

B. Impact of different data augmentation for TTT

We apply random horizontal flipping, resizing, cropping, and photometric distortion for data augmentation. In detail, photometric distortion includes random brightness, contrast, saturation, and hue. We further ablate the effect of each type of augmentation on it. Table F shows that the proposed method works well when removing any one kind of data augmentation, which indicates that our success does not come from any particular trick. Each type of augmentation doesn't affect the model significantly since it is used to create a pair of samples for consistent depth map optimization. Therefore, the proposed TTT strategy is the key to success.

	DAVIS-16	FBMS	Long.
-	77.1	73.7	65.2
w/o resize	+0.5	+3.2	+7.7
w/o crop	+0.4	+2.5	+7.5
w/o flip	+0.4	+3.1	+7.2
w/o brightness	+0.4	+2.9	+7.3
w/o contrast	+0.7	+2.8	+7.3
w/o saturation	+0.3	+2.9	+7.7
w/o hue	+0.4	+2.8	+6.4
full	+0.4	+3.2	+7.9

Table F. **Impact of different data augmentation for TTT in DAVIS-16 [31], FBMS [28], Long-Videos [20] datasets.** $\mathcal{J}$ is reported for comparison.

Depth Extractors	Depth Supervision	DAVIS-16	FBMS	Long.
Monodepth2 [12]	-	77.1	73.7	65.2
	Consistent Depth	+0.4	+3.2	+7.9
	Pseudo Depth	−1.4	+0.5	+2.9
LiteMono [57]	-	76.8	79.0	68.1
	Consistent Depth	+2.0	+1.5	+6.3
	Pseudo Depth	+0.7	−0.1	−2.6
ZoeDepth [2]	-	79.9	76.4	64.0
	Consistent Depth	+0.5	+4.7	+9.5
	Pseudo Depth	−0.4	+0.5	+1.9

Table G. **TTT with depth supervision from depth predictor in DAVIS-16 [31], FBMS [28], Long-Videos [20] datasets.** $\mathcal{J}$ is reported for comparison. Results that the dropped after TTT are masked as red. '*Consistent Depth*' denotes self-supervised learning via consistent depth map prediction. '*Pseudo Depth*' denotes supervised learning via pseudo depth map supervision.

C. TTT with depth supervision from depth predictor

We also experiment with pre-calculated depth maps for test-time training. This means that instead of generating two batches of images to minimize the distance between their depth maps ('*Consistent Depth*' in Table G), we predict one batch of depth maps and calculate the error with the pseudo depth maps ('*Pseudo Depth*' in Table G). This process is similar to the training-time training stage (*c.f.* Section 4.1), but without the binary cross entropy for segmentation. As shown in Table G, it brings less improvement and sometimes fails. This can be due to the model being explicitly required to learn depth estimation and damaging its ability to segmentation.

Method	TTT-N	DAVIS-16		FBMS		Long.		MCL		STV2	
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Baseline	-	75.9	77.5	75.1	76.5	63.9	67.5	57.3	70.8	61.5	70.4
TENT [48]	✓	−0.2	−0.2	+0.2	+0.3	+0.4	+0.3	+0.7	+0.3	−0.1	−0.1
BN [36]	✓	−0.1	−0.1	**+0.4**	**+0.6**	+0.7	+0.4	+1.0	+0.8	+0.1	+0.1
TTT-Rot [39]	-	75.3	76.2	75.4	77.2	59.1	62.8	57.7	70.5	66.4	73.3
	✓	−0.1	−0.1	+0.1	+0.3	+0.8	+1.1	+1.0	+0.9	−0.4	−0.1
TTT-MAE [11]	-	73.5	74.1	74.6	75.7	64.4	67.5	55.7	66.8	62.2	70.2
	✓	−0.2	−0.1	−0.1	−0.1	+0.3	+0.2	+0.1	+0.2	+0.1	+0.1
Ours	-	77.1	78.4	73.7	75.8	65.2	68.0	53.5	66.2	61.5	69.2
	✓	**+0.3**	**+0.3**	+0.1	+0.3	**+1.3**	**+1.5**	**+1.9**	**+1.5**	**+1.0**	**+1.2**

Table H. **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**.

D. Further study on Naive TTT (TTT-N) strategy

We compare with other TTT methods [11, 36, 39, 48] following the proposed TTT-LTV strategy (*c.f.* Section 4.2) and show the result in Table 4 in the main paper. Here, we further compare with them following the naive image-based TTT strategy (*c.f.* TTT-N in Section 4.2). As shown in Table H, other TTT methods can not obtain consistent improvement in different datasets following the TTT-N strategy, which is the same as in the TTT-LTV strategy. Although the improvement of our method is not as obvious as that in the TTT-LTV strategy, it is more stable than others. It demonstrates that depth-aware test-time training is necessary in ZSVOS.

E. Impact of the depth prediction loss weight λ

We use a hyper-parameter λ to balance the two losses as described in Equation 2. We choose different λ and find it is important in learning depth-aware features. As shown in Table I, a larger λ allows the model to learn stronger 3D knowledge during the training-time training, which leads to better results when the model is directly applied to the test videos. However, the well-trained image encoder cannot benefit from the proposed self-supervised task consistently at test time. Finally, we choose $\lambda = 0.1$ since it performs well both with and without TTT.

λ	TTT	DAVIS-16	FBMS	Long.
1	-	77.5	75.4	65.5
	✓	76.7	72.4	67.4
0.1	-	77.1	73.7	65.2
	✓	**77.5**	76.9	**73.1**
0.01	-	76.1	73.8	64.6
	✓	76.7	**77.9**	72.6

Table I. **Impact of the depth prediction loss weight λ in DAVIS-16 [31], FBMS [28], Long-Videos [20] datasets.** $\mathcal{J}$ is reported for comparison. Results that the dropped after TTT are masked as red. The best result is marked as **bold**.

Clip Nums	DAVIS-16	STV2
-	77.1	61.5
1	−3.1	+2.8
5	−0.1	+3.1
10	+0.4	+4.4
20	+0.5	+2.6

Table J. **Sampling strategy for densely annotated videos: DAVIS-16 [31] and SegTrackV2 [19] datasets.** $\mathcal{J}$ is reported for comparison. Results that the dropped after TTT are masked as red.

F. Sampling strategy for densely annotated videos: DAVIS-16 [31] and SegTrackV2 [19]

DAVIS-16 [31] and SegTrackV2 [19] are densely annotated, while FBMS [28], Long-Videos [20], and MCL [17] are annotated once every few frames. We perform TTT frame-by-frame on FBMS, Long-Videos, and MCL. As for DAVIS-16 and SegTrackV2, we first divide the video frames into 10 video clips, which means that the interval of consecutive frames in each clip is 10, and then perform TTT by sampling a single frame from each clip. As shown in Table J, the performance may drop when performing TTT without the sampling strategy. Conducting the sampling strategy allows model training from more diverse input which helps to combat overfitting. Similarly, Figure 5 in the main paper shows that training too many epochs in the same frame may drop the performance in sparsely annotated video.

	DAVIS-16	FBMS	Long.
Baseline	77.8	73.6	64.6
+ Ours Architecture	78.6	74.1	65.8
+ Ours TTT Strategy	**78.8**	**78.2**	**71.7**

Table K. **Using HFAN [29] as the baseline model on DAVIS-16 [31], FBMS [28], Long-Videos [20] for TTT.** $\mathcal{J}$ is reported for comparison.

G. Extending DATTT to existing ZSVOS method

Our approach operates independently of other ZSVOS methods. For instance, we utilize HFAN [29] as the baseline model for our DATTT approach. HFAN incorporates additional feature alignment modules for both appearance and motion features. As demonstrated in Table K, our proposed depth-aware architecture and TTT strategy each yield noticeable enhancements.

H. Additional visual results

We provide more visual results similar to Figure 6 (main paper) in Figure G. The pre-trained model struggles to handle these videos at first, and then clear improvements are observed after performing TTT. The proposed method works well in both single-object and multi-object scenarios.

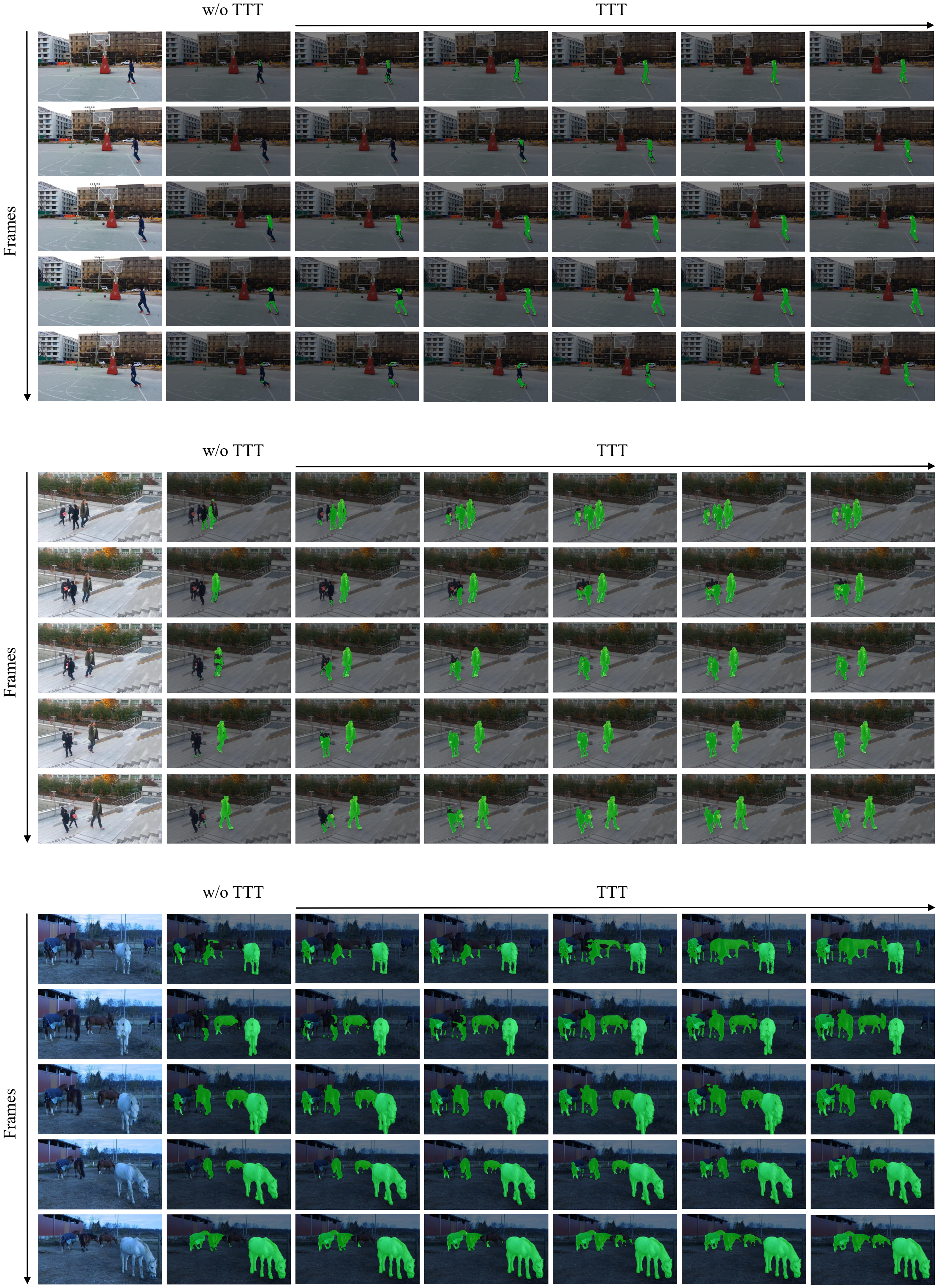

Figure G. **Additional visual results.** The background in the results is dimmed for better visualization.