TSTM: Temporal Segmentation for Task-relevant Mask in Visual Reinforcement Learning Generalization

Supplementary Material

In this supplementary material, we provide additional descriptions and experimental results for TSTM. We first present the temporal segmentation architecture. Then, we give a detailed description of the TSTM algorithm with the complete learning pipeline and pseudocode. We also report experiments on Robotic Manipulation with extended comparisons and learning curves. Furthermore, we investigate GT-free alternatives for mask acquisition and provide fair comparisons under the same supervision. We then analyze the training and inference overhead, followed by additional comparisons with the PPO baseline. Finally, we provide extensive ablation studies on hyperparameters, temporal window length, and implementation details to facilitate reproducibility.

A. Temporal Segmentation Architecture

As shown in Figure 3, the temporal segmentation model adopts a temporal segmentation network architecture that integrates an encoder, a temporal module, and a decoder. The encoder uses three convolutional layers and two pooling layers to extract frame-wise spatial features. The temporal module with two ConvLSTM layers then models temporal dependencies across observations. The decoder reconstructs per-frame segmentation masks via two convolutional and two upsampling layers. The teacher S_{T} and student S_{S} share the same number of layers, but the student is more lightweight, as each layer contains fewer parameters than that of the teacher.

B. Detailed Description of TSTM Algorithm

The overall pseudocode pipeline, summarized in Algorithm 1, consists of three stages: temporal segmentation training, invariant-representation RL training, and testing. In the temporal segmentation stage, the teacher network is first optimized with its supervised loss. After convergence, it is frozen and used to guide the lightweight student network via knowledge distillation, enabling the student to acquire the teacher's intermediate representations. The trained student model then serves as the temporal segmentation module in all subsequent stages. During RL training, both the original and augmented observation sequences are processed by the segmentation module, encoded into latent representations, and projected into a compact feature space. The encoder and projector are jointly optimized with an invariant-representation objective that enforces consistency across augmented views while preserving feature diversity and reducing redundancy. Meanwhile, the SAC actor and critic are updated with their respective learning objectives. At test time, the observation sequence is segmented and encoded into a state representation, which is then provided to the policy for action selection and environment interaction.

C. Experiments on Robotic Manipulation

Experimental Environment. For Robotic Manipulation, we evaluate on the Reach task, where a robotic arm is required to reach a specified target position. It consists of one training and five test environments that differ in visual attributes such as object appearance.

Comparison with State-of-the-Art Methods. Table C1 and Figure C1 present the results of our method and other baselines on the Reach task of Robotic Manipulation. As shown in Table C1, TSTM achieves the highest final return on most test environments, including Test 1 (32.5 ± 1), Test 2 (27.3 ± 5), Test 4 (32.5 ± 1), and Test 5 (32.3 ± 1), while also obtaining the best overall average test return of 31.4. On the training environment and Test 3, its performance is comparable to other methods. Figure C1 illustrates the average test return curves over training steps. As shown in Figure C1, TSTM reaches comparable or higher returns with fewer training steps on most tasks (Test 1, Test 2, Test 4, and Test 5), while ultimately achieving the best final return. The results in Table C1 and Figure C1 demonstrate that our method outperforms other visual RL approaches in most environments.

D. Ablation Studies on Hyperparameters

To validate the effectiveness of the key hyperparameter values in our loss function (Eq. (15)), we conducted ablation experiments with different hyperparameter settings, as shown in Table D1. The results demonstrate that the selected values of $\lambda = 2.5, \mu = 1.25, \rho = 0.005$ achieve 851 ± 19 on *video easy* and 741 ± 22 on *video hard* in the representative Cartpole Swingup task, ranking highest under both settings. Considering the strong cross-difficulty performance with a single configuration and avoiding per-setting tuning, we adopt these hyperparameters. The results in Table D1 confirm the effectiveness of the chosen hyperparameters.

E. Implementation Details

Computational Resources. All experiments are conducted on a GPU server equipped with four NVIDIA GeForce RTX 4090 GPUs (24 GB memory each). The key hyper-

Table C1. Generalization performance on the Reach task of Robotic Manipulation under one training environment and five test environments. The best score for each task is shown in **bold** while the second best score is underlined.

Task	Environment	SAC	SODA	SVEA	SGQN	MaDi	SimGRL	TSTM (Ours)
Reach	Train	-20.6 ± 31	32.1 ± 1	32.2 ± 1	32.3 ± 1	32.1 ± 1	**33.2 ± 2**	33.1 ± 2
	Test 1	-24.3 ± 15	-14.3 ± 19	-25.6 ± 8	-4.1 ± 28	2.7 ± 21	1.3 ± 54	**32.5 ± 1**
	Test 2	-19.1 ± 17	-16.8 ± 27	-13.8 ± 41	15.1 ± 33	8.3 ± 20	-16.2 ± 52	**27.3 ± 5**
	Test 3	-37.2 ± 6	-37.5 ± 18	-9.3 ± 33	26.5 ± 7	13.9 ± 14	**32.6 ± 1**	32.5 ± 1
	Test 4	-17.0 ± 23	-14.7 ± 38	-14.4 ± 40	17.2 ± 18	-11.2 ± 29	31.6 ± 2	**32.5 ± 1**
	Test 5	-29.9 ± 10	-57.8 ± 18	-42.9 ± 20	-35.5 ± 16	-29.0 ± 20	15.8 ± 28	**32.3 ± 1**
	Test Average	-25.5	-28.2	-21.2	3.8	-3.1	13.0	**31.4**

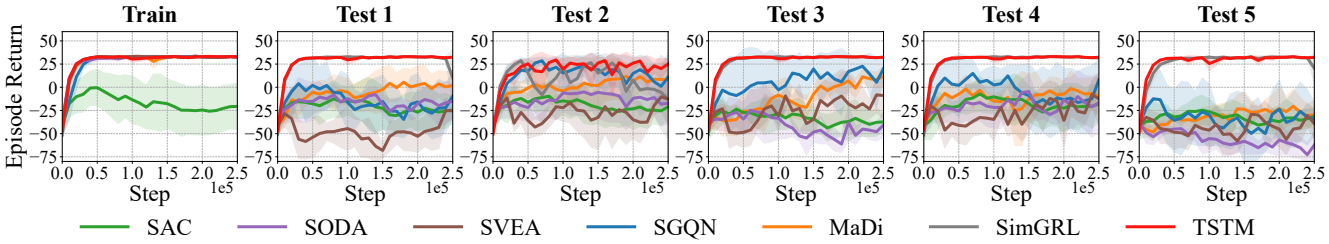

Figure C1. Learning curves on the Reach task of Robotic Manipulation under 6 different settings (one training and five test environments).

Table D1. The hyperparameter ablation results of our method on Cartpole Swingup task under the *video easy* and *video hard* settings. Experiments are conducted over three random seeds. The best score for each task is shown in **bold** while the second best score is underlined.

	Method	Cartpole Swingup
Video Easy	$\lambda = 2.5, \mu = 1.25, \rho = 0.005$	**851±19**
	$\lambda = 1.25, \mu = 2.5, \rho = 0.005$	817±74
	$\lambda = 1.25, \mu = 1.25, \rho = 0.1$	803±57
	$\lambda = 1.25, \mu = 0, \rho = 0$	815±73
	$\lambda = 1.25, \mu = 0, \rho = 0.05$	826±25
	$\lambda = 0, \mu = 1.25, \rho = 0.05$	825±24
	$\lambda = 0, \mu = 1.25, \rho = 0$	765±117
Video Hard	$\lambda = 2.5, \mu = 1.25, \rho = 0.005$	**741±22**
	$\lambda = 1.25, \mu = 2.5, \rho = 0.005$	698±49
	$\lambda = 1.25, \mu = 1.25, \rho = 0.1$	710±58
	$\lambda = 1.25, \mu = 0, \rho = 0$	693±75
	$\lambda = 1.25, \mu = 0, \rho = 0.05$	708±19
	$\lambda = 0, \mu = 1.25, \rho = 0.05$	724±20
	$\lambda = 0, \mu = 1.25, \rho = 0$	660±83

parameters and network configurations are summarized in Table E1.

Data Augmentation. Figure 2 illustrates that data augmentation is applied only during training to improve policy robustness to visual perturbations by overlaying standard Place365 images on raw visual observations. All modules in Figure 2 are used during training, while only the segmentation network, encoder, and actor are retained at test time.

F. GT Masks and GT-Free Alternatives

Figure F1 presents two GT-free strategies for training the temporal segmentation model, including optical flow-based self-supervised pseudo-labels (*Optical Flow*) and direct transfer from DAVIS (*Transfer*). Ground-truth masks can be obtained at no additional cost via the simulator interface; nevertheless, the GT-free results remain comparable to the original TSTM, indicating improved real-world transferability.

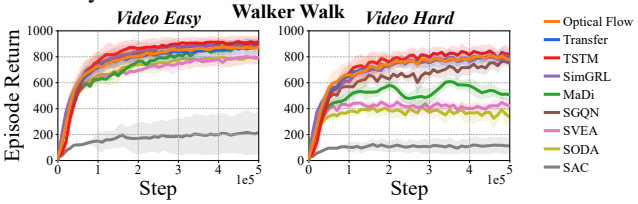

Figure F1. GT-free training strategies: optical-flow pseudo labels and transfer from DAVIS, both achieving comparable performance to the original TSTM.

G. Fair Comparison

Table G1 shows that under the same GT supervision, TSTM outperforms MaDi with GT auxiliary loss in terms of return, demonstrating that the performance improvements arise from our temporal module.

Table G1. Comparison with MaDi under the same GT supervision.

	Video Easy		Video Hard	
Task	MaDi+GT	TSTM	MaDi+GT	TSTM
Walker Walk	807 ± 83	**912 ± 42**	497 ± 153	**821 ± 36**
Cartpole Swingup	807 ± 18	**851 ± 19**	564 ± 77	**741 ± 22**

Hyperparameter	Value
Observation image size	$84 \times 84 \times 3$
Observation sequence length k	5
Action repeat in SAC	4 (Walker walk, Walker stand, Ball in cup), 8 (Cartpole), 2 (Finger spin)
Discount factor γ	0.99
Episode length	1,000 (DMC) / 50 (Robotic)
Number of frames	500,000 (DMC) / 250,000 (Robotic)
Replay buffer size	500,000 (DMC) / 250,000 (Robotic)
Optimizer for Actor and Critic	Adam (lr $= 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$)
Optimizer for α in SAC	Adam (lr $= 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)
Batch size	128
Target Q networks update frequency	2
Data augmentation	Random shift and overlay [21]
Actor update frequency	2
Teacher ConvLSTM hidden dim	256
Student ConvLSTM hidden dim	32
Projector hidden dimension	2048
Segmentation loss weights ν	0.5
Distillation weight β	0.7
Distillation temperature ζ	4.0
Dice constant for numerical stability ϵ	10^{-5}
Policy consistency weight ς	2.0
INV variance threshold Γ	1.0
INV stability constant η	10^{-4}
INV coefficients λ	2.5
INV coefficients μ	1.25
INV coefficients ρ	0.005

Table E1. Hyperparameters in our methods (DMC-GB and Robotic Manipulation).

H. Training/Inference Overhead

Training/inference overhead (*FPS and wall-clock time*) is summarized in Table H1. TSTM achieves competitive speed, particularly at inference, relative to most SOTA end-to-end visual RL methods.

Table H1. FPS and wall-clock time on Walker Walk.

Walker Walk	SAC	SODA	SVEA	SGQN	MaDi	SimGRL	TSTM
Training FPS	17.6	9.0	14.0	8.8	5.4	11.5	6.5
Training Time (h)	7.9	15.4	9.9	15.8	25.7	12.1	21.4
Inference FPS	49.6	44.6	51.2	50.4	50.4	51.2	46.6
Inference Time (h)	6.3	7.0	6.1	6.2	6.2	6.1	6.7

I. PPO Baseline

We include PPO as a baseline and report *wall-clock time* in Table I1. SAC outperforms PPO, and TSTM-SAC substantially exceeds TSTM-PPO. These performance gains justify the choice of SAC as the baseline, with only a modest increase in time cost.

Table I1. Comparison with PPO baseline and wall-clock time on Walker Walk.

Walker Walk	PPO	TSTM-PPO	SAC	TSTM-SAC
Video Easy	204±118	**210±89**	245±165	**912±42**
Video Hard	93±39	**209±65**	122±47	**821±36**
Total Wall-Clock Time (h)	9.6	11.1	14.2	28.1

J. Temporal Window Length

Figure J1 shows that the ablation over k supports the choice $k = 5$ used in our paper.

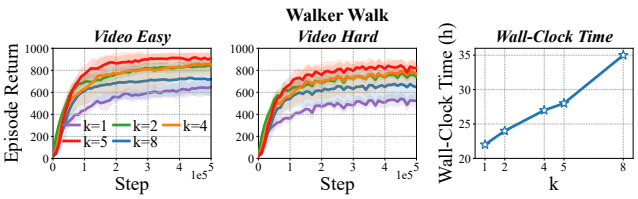

Figure J1. Ablation on temporal window length k and corresponding wall-clock time.