Shuffle and Attend: Video Domain Adaptation

Jinwoo Choi[1⋆], Gaurav Sharma[2], Samuel Schulter[2], and Jia-Bin Huang[1]

[1] Virginia Tech, Blacksburg, VA 24060, USA
{jinchoi,jbhuang}@vt.edu
[2] NEC Labs America, San Jose, CA 95110, USA

In this supplementary material, we provide additional implementation details and experimental results to complement the main paper.

1 Dataset

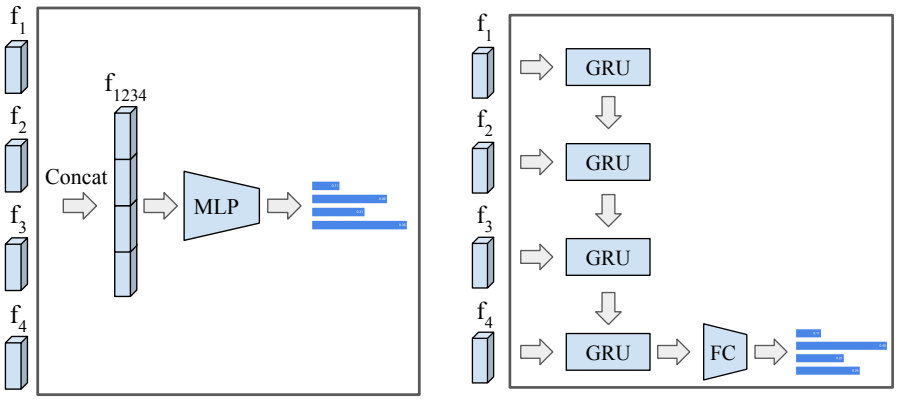
1.1 Summary of the datasets

Table 1: Video domain adaptation datasets summary.

	UCF	HMDB	Kinetics	NEC-Drone
Length (sec.)	1-33	1-33	1-10	1-22
Spatial resolution	320×240	varies × 240	varies	1920×1080
Frame rate	25	30	varies	30
# of classes	12	12	7	7
# of training videos	1,438	840	9,955	560
# of validation videos	571	360	742	206
# of test videos	-	-	-	228
# of training frames	276,148	84,883	2,415,462	75,901
# of validation frames	107,223	34,023	181,878	29,224
# of test frames	-	-	-	29,742
Domain gap	UCF→HMDB: 14.7%p HMDB→UCF: 8.0%p		Kinetics→Drone: 64.5%p	

We present the summary of the datasets used in this work in Table 1. In addition to the other information, we add a domain gap row between datasets by measuring classification performance difference between the supervised source only I3D (lower bound) and the supervised target I3D (upper bound) in the last row of Table 1. Kinetics→Drone has a domain gap of 64.5% while UCF→HMDB has 14.7%p and HMDB→UCF has 8.0%p. The domain gap difference suggests that Kinetics→Drone is a more challenging setting than UCF→HMDB and HMDB→UCF.

⋆ Part of this work was done when Jinwoo Choi was an intern at NEC Labs America.



(a) MLP implementation (b) GRU implementation

Fig. 1: **Implementations of clip attention network Φ.** Best viewed with zoom and color.

2 Implementation Details

Attention module. There can be multiple valid choices for the architecture of the attention module $\Phi(\cdot)$, e.g., a standard feed-forward network which takes concatenation of the clip features as input, or a recurrent network that consumes the clip features one by one. We explore two specific choices: (1) MLP and (2) gated recurrent unit (GRU) network, as shown in Figure 1. Let us assume the number of clips per video $N = 4$ for brevity. *The MLP-based attention module* takes four clips as input. Then we concatenate the four clips (temporally ordered as shown in Figure 1 (a)) and pass it through 4 fully connected layers. The four fully connected layers consist of 3 layers with 1024 hidden units each, and four units in the final fully connected layer. The output of the MLP is the length-4 vector indicating that which clip is more important (discriminative) and which clip is less important. *The GRU-based attention module* also takes four clips as input. Then we pass each clip in temporal order to the GRU with 1024 hidden units, as shown in Figure 1 (b). We pass the output feature of GRU to a fully connected layer to get the length-4 vector indicating that which clip is more important and which clip is less important.

Baselines. We implement competitive baseline video domain adaptation methods by extending the two state-of-the-art image-based domain adaptation methods DANN [3], and ADDA [7] as shown in Figure 2. The major differences between DANN and ADDA are two-fold: (1) The DANN method shares the feature backbone parameters between source and target encoders. The ADDA method, on the other hand, does not share the network parameters. (2) The DANN method uses a gradient reversal layer or inverted GAN loss for adversarial training of domain classifiers. We extend DANN and ADDA by replacing the feature vector input to the domain classifier from image feature to clip feature. We sample a $L = 16$-frames long clip from each video and pass it through a clip feature

Table 2: Results on UCF↔Olympics.

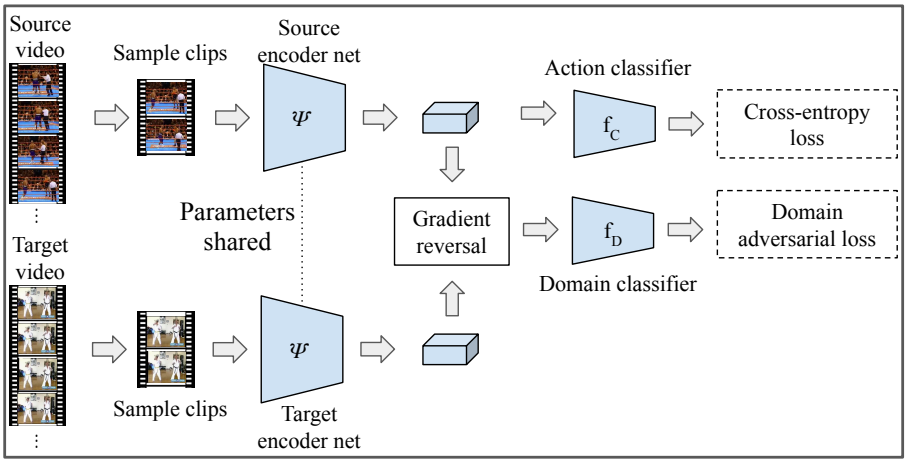
Method	Encoder	UCF→Olympics	Olympics→UCF
TA^{3}N [2]	ResNet-101-based TRN	98.1	91.9
TCoN [6]	ResNet-101-based TRN	96.8	96.7
SAVA (ours)	I3D	98.1	96.7

encoder (e.g., I3D [1] in this work). We feed the clip features from source and target videos to the domain classifier.

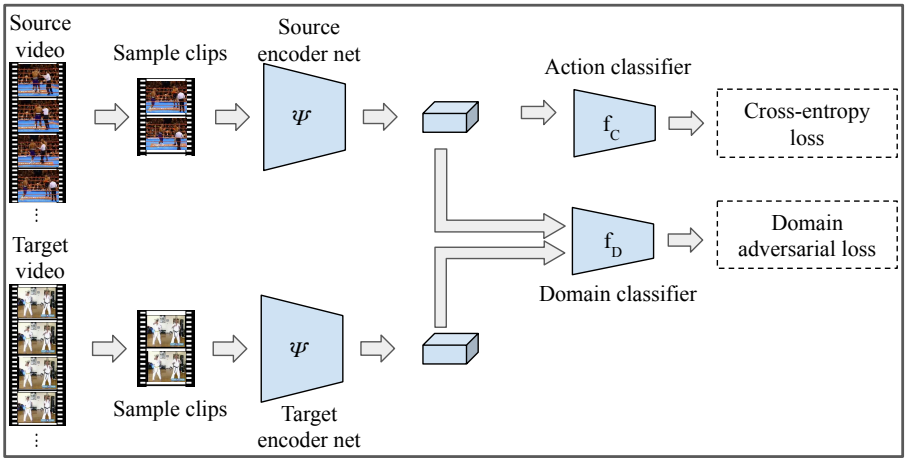
I3D-based TA^{3}N. The original TA^{3}N builds upon the TRN using the 2D ResNet-101 [4] feature backbone. We replace the 2D ResNet-101 backbone with the I3D backbone. Given a video, we densely slide a temporal window with a temporal stride of 1 to sample 16 frames long clips. For a frame k in the video, the temporal window consists of frames from $k-7$ to $k+8$. We zero-pad the beginning and the end of the input video. We then extract I3D features by feeding the sliding windows to the I3D feature backbone. As a result, we obtain a $1,024$ dimensional feature vector for every frame. We then follow the remaining steps as in the original TA^{3}N method.

3 Comparison on UCF↔Olympics

We also provide a further comparison of SAVA with TCoN and TA^{3}N on the UCF↔Olympics dataset [5] in Table 2. SAVA achieves competitive or better performance over all the cases.



(a) Overview of DANN for video



(b) Overview of ADDA for video

Fig. 2: **Overview of the DANN and ADDA extended for video.** Best viewed with zoom and color.

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
2. Chen, M., Xue, H., Cai, D.: Domain adaptation for semantic segmentation with maximum squares loss. In: ICCV (2019)
3. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML (2015)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
5. Jamal, A., Namboodiri, V.P., Deodhare, D., Venkatesh, K.: Deep domain adaptation in action space. In: BMVC (2018)
6. Pan, B., Cao, Z., Adeli, E., Niebles, J.C.: Adversarial cross-domain action recognition with co-attention. In: AAAI (2020)
7. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)