Supplementary Material: Deep Domain Adaptation in Action Space

Arshad Jamal[1]
arshad@iitk.ac.in
Vinay P Namboodiri[2]
vinaypn@iitk.ac.in
Dipti Deodhare[1]
dipti@cair.drdo.in
KS Venkatesh[2]
venkats@iitk.ac.in

[1] Centre for AI & Robotics
Bangalore, India

[2] Indian Institute of Technology,
Kanpur, India

1 Overview of Supplementary Material

In the main text, both the proposed approaches have been described in detail. The methods have been extensively evaluated using three sets of multi-domain action datasets. In the main text, due to space constraint, only the results for **UO** (UCF50 and Olympic Sports) and **KMS** (KTH, MSR Action II and Sonycam) datasets were presented and discussed and the additional results have been included in the supplementary material. We start with the algorithmic details of Action Modeling on Latent Space (AMLS) approach. In the next section, we describe the *Symmetrized KL Divergence* measure and then discuss the KMS and UO datasets with few example images. In Section 5, we describe our third dataset collection (**HU**) comprising five common classes of HMDB51 and UCF50 and present its domain adaptation results. In the next section, we discuss the qualitative analysis of the results for the **HU** dataset and present some of the negative examples observed in our experiments. Finally, in Section 7, we discuss the hyper-parameters and compare some of the results for their different choices.

2 AMLS Algorithm

There are two main steps in the proposed algorithm: (i) use incremental subspace learning method [6] to find the subspace representation for the sequence of target domain points. (ii) use Geodesic Flow Kernel or Subspace Alignment method to perform a sequence of adaptation and classify each action clip using SVM. In the paper, we have used Sequential Karhunen-Loeve Method (SKLM) [5] for the subspace learning. The pseudo code of the approach is given in Algorithm 1.

Algorithm 1: Psudo Code for Action Modeling on Latent Subspace (AMLS) for Unsupervised Domain Adaptation.

```
input: Subspace dimension $d$
Data: Source features $\mathbf{V}_{\mathcal{S}} \in \mathbb{R}^{N \times D}$, source labels $\mathbf{y_S} \in [1, C]$ and target data
      $\mathbf{V}_{\mathcal{T}} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_M\}$, where $\mathbf{v}_i \in \mathbb{R}^{M_i \times D}$, $M_i$ is the number of C3D features
      corresponding to the $i^{\text{th}}$ clip across all target videos and $\mathbf{M}$ is the maximum
      number of clips.
Result: Predicted target clip labels $\mathbf{y_T}$
$\mathbf{S} \leftarrow$PCA($\mathbf{V}_{\mathcal{S}}$, $d$)
$\mathbf{T}_0 \leftarrow \mathbf{S}$
for $m \leftarrow 1$ to $\mathbf{M}$ do
    $\mathbf{T}_m \leftarrow$SKLM($\mathbf{T}_{m-1}, \mathbf{v}_m$);
    $\mathbf{G}_m \leftarrow$ GFK or SA($\mathbf{S}, \mathbf{y_S}, \mathbf{T}_{m-1}, \mathbf{T}_m, d$);
    $\mathbf{y}_m \leftarrow$ kNN or SVM($\mathbf{G}_m, \mathbf{V}_{\mathcal{S}}, \mathbf{v}_m$);
end
$\mathbf{y_T} \leftarrow \{\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_M}\}$
```

3 Symmetrized KL Divergence (SKLD)

Let $\mathbf{V}_{\mathcal{S}} \in \mathbb{R}^{D \times N_S}$, $\mathbf{V}_{\mathcal{T}} \in \mathbb{R}^{D \times N_T}$ be the features for the source and target datasets and $\mathbf{S}$, $\mathbf{T} \in \mathbb{R}^{D \times d}$ be the basis of the two subspaces learned from them. The *SKLD* between the source and target domain, as introduced in [3], is defined as:

$$SKLD(\mathcal{S}, \mathcal{T}) = \frac{1}{d^*} \sum_{i}^{d*} \theta_i \left\{ KL(\mathcal{S}_i || \mathcal{T}_i) + KL(\mathcal{T}_i || \mathcal{S}_i) \right\} \tag{1}$$

where d^* is the optimal dimensionality of the subspace and θ_i is the i^{th} principal angle. $\mathcal{S}_i$ and $\mathcal{T}_i$ are two one-dimensional distributions of $\mathbf{V}'_{\mathcal{S}}\mathbf{s}_i$ and $\mathbf{V}'_{\mathcal{T}}\mathbf{t}_i$ respectively. $\mathbf{s}_i$ and $\mathbf{t}_i$ are the i^{th} basis vector of the source and target points on the subspace.

The principle angles θ_i between two subspaces are efficiently computed using the SVD of matrix $\mathbf{S}'\mathbf{T} = \mathbf{U}\Gamma\mathbf{V}'$ and they are $\theta_i = arccos(\gamma_i)$, where γ_i is the i^{th} singular value in the diagonal matrix Γ. The principle vectors $\mathbf{s}_i = (\mathbf{SU})_{.,i}$ and $\mathbf{t}_i = (\mathbf{TV})_{.,i}$ are i^{th} columns of the product matrix.

The Symmetrized KL Divergence, as defined in Eq (1), represents the dissimilarity between the distribution of the two domains and it is computed by approximating the domain distribution with one dimensional Gaussians. If we normalize the features to have zero mean, we only need to compute the variances of the two distributions, which is defined as,

$$\sigma_{i\mathcal{S}}^2 = \frac{1}{N_S} \mathbf{s}'_i \mathbf{V}'_{\mathcal{S}} \mathbf{V}_{\mathcal{S}} \mathbf{s}_i, \quad \sigma_{i\mathcal{T}}^2 = \frac{1}{N_T} \mathbf{t}'_i \mathbf{V}'_{\mathcal{T}} \mathbf{V}_{\mathcal{T}} \mathbf{t}_i \tag{2}$$

In this case of approximate Gaussian distribution, the SKLD is computed in close-form as,

$$SKLD(\mathcal{S}, \mathcal{T}) = \frac{1}{d^*} \sum_{i}^{d^*} \theta_i \left\{ \frac{1}{2} \frac{\sigma_{i\mathcal{S}}^2}{\sigma_{i\mathcal{T}}^2} + \frac{1}{2} \frac{\sigma_{i\mathcal{T}}^2}{\sigma_{i\mathcal{S}}^2} - 1 \right\} \tag{3}$$

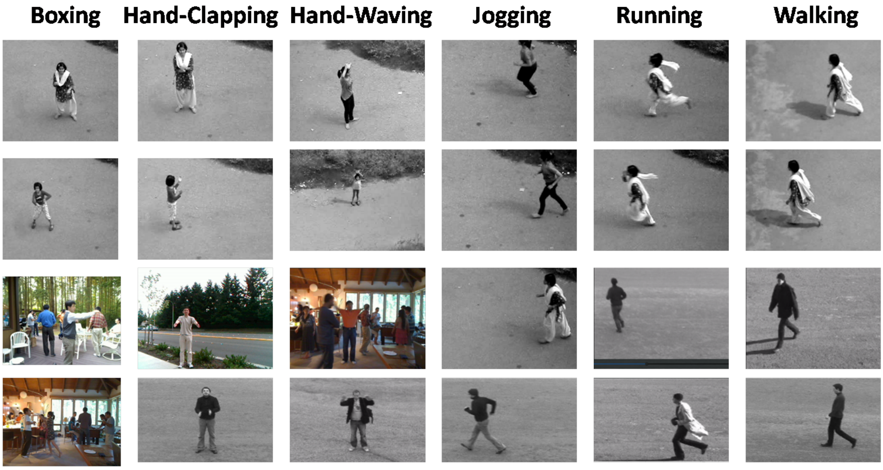

Figure 1: Examples images from action videos of six classes from KMS dataset. The first two rows have images from the SonyCam dataset and the last two rows have images from all three datasets.

4 Example Images from the Datasets

In Fig 1, few example images of the action videos of six classes in the **KMS** dataset have been shown. In this multi-domain dataset collection, the KTH has only grayscale images making the adaptation to color images extremely challenging. The 3D-CNN is fine-tuned using the Training set of the KTH data. Similarly, in Fig 2, few example images from the six common classes of the UCF50 and Olympic Sports dataset has been given.

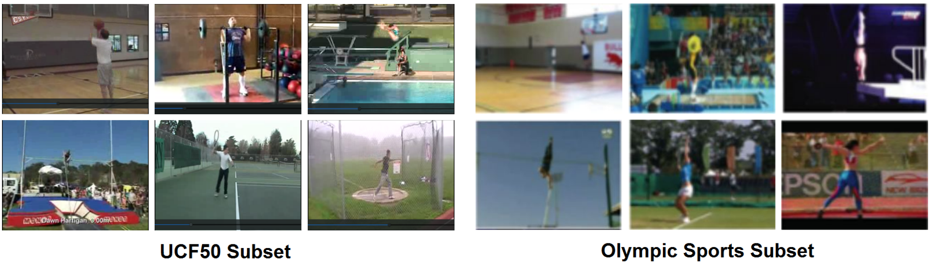

Figure 2: Examples images from action videos of six common classes from UO dataset.

5 Experiments with UCF50 and HMDB51 Datasets

In the third series of experiments, we use five common classes of UCF50 and HMDB51 datasets (denoted by **U** for UCF50 subset and **H** for HMDB51 subset). The classes are - *GolfSwing, Basketball, Biking, HorseRiding* and *PullUps*. For the UCF50 subset, we use 70%-30% train-test split suggested in [7], which results into 360 – 140 train/test action videos for training and testing. For the HMDB51 subset, we use the splits described in [4],

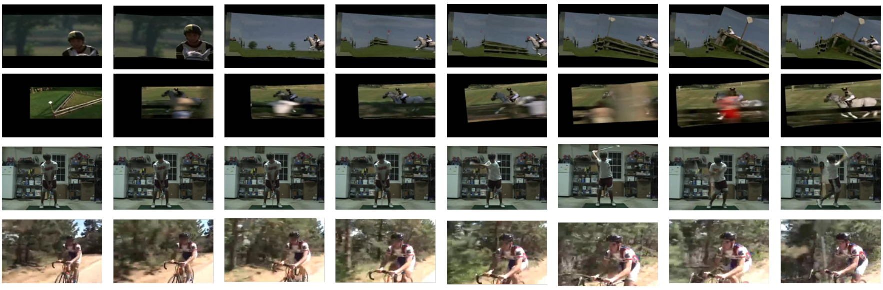

Figure 3: Few negative examples for which all methods failed to correctly predict the class of action video. First frame from each 16-frame clip in the action video has been shown.

Table 1: Action Classification Accuracy (%) using the 3D-CNN features for the five common classes of the HU dataset (**H**:HMDB51 and **U**:UCF50). The pre-trained network was fine-tuned independently using subsets of HMDB51 and UCF50 datasets, resulting in two fine-tuned networks. The first two cols indicate results for the network that was fine-tuned using HMDB51 dataset and the last two cols are for the network fine-tuned using UCF50 dataset. The 4096-dim, *fc7* features have been used for the experiment. Our approaches outperform the baseline methods in three out of four cases.

Methods	$\mathbf{U_U \rightarrow H_U}$	$\mathbf{H_U \rightarrow U_U}$	$\mathbf{U_H \rightarrow H_H}$	$\mathbf{H_H \rightarrow U_H}$	**Avg Accuracy**
3D-CNN [8]	85.21	97.37	91.03	94.29	91.97
Baseline-S	86.82	**97.76**	92.23	92.74	92.39
Baseline-T	86.77	97.75	93.22	93.69	92.86
GFK_Action	89.33	96.76	93.72	94.46	93.57
AMLS_GFK (ours)	89.53	96.66	**95.9**	**95.36**	**94.36**
SA_Action	87.43	97.1	92.87	94.33	92.93
AMLS_SA (ours)	**90.25**	96.79	94.5	94.4	93.99

which results into a 350-150 video split for training and testing. For **HU** dataset collection, we solve four adaptation problems i.e. $\mathbf{U_U \rightarrow H_U}$, $\mathbf{H_U \rightarrow U_U}$, $\mathbf{U_H \rightarrow H_H}$ and $\mathbf{H_H \rightarrow U_H}$. The subscript **U** and **H** denotes the two networks trained using UCF50 and HMDB51 datasets.

As we did in two other experiments, we compare our approaches with five baseline methods, which are: **3D-CNN [8]**, **Baseline-S**, **Baseline-T**, Geodesic Flow Kernel (**GFK**) [3] and Subspace Alignment (**SA**) [1] methods. We use the one-vs-all SVM classifier to evaluate these methods. We start with the deep learning model trained for Sports 1M and then separately fine-tune it using the five common classes of the two datasets, resulting into two 3D-CNN models.

The results are shown in Table 1. The first two columns are for the network fine-tuned with UCF50 subset and the last two columns are for HMDB51 subset. It is evident that the two subspace based methods outperform the other baselines and our proposed approach further improves the results of these methods. Here, the results are for the subspace dimension of 100 (refer Section 7 for details).

6 Qualitative Analysis of the Results

In our experiments, we found that for few action videos, all the methods failed to correctly predict its class. Some of these examples from the **HU** datasets are shown in Fig 3. The findings were quite surprising as the performance for the **HU** dataset was generally very good. However, on seeing these videos, the reason of the failure was quite evident. In all these examples, the action is either not adequately visible or in 16-frame clip duration, there is no action being performed at all.

7 Discussion on Hyper Parameters

There are two important hyper-parameters, which can affect the performance of the algorithms. One of them is related to the SVM classifier, which are selected using the standard k-fold cross-validation technique. In all our experiments, we used $k = 6$. The other parameter is the dimension of the latent subspace. We experimented with the subspace dimensions in the range of $5 - 500$ and empirically found that the subspace dimension does affect the classification accuracy. In most of the cases, the accuracy initially improved with increase in the dimension and then it went down beyond certain dimensions. In all three experiments, we use the subspace dimension for which the average accuracy across all the adaptation problem is maximum. We report the results of the GFK and SA methods for this subspace dimension.

In the case of DAAA method, learning rate is one of the important hyper-parameter, which effect the training outcome. In this work, the learning rate was varied between $0.01 - 0.00001$ with the multiplication factor of $1/\sqrt{10}$. In the experiment, we found that the learning rate of 0.0001 worked well for the UO adaptation problems. So, we used the same learning rate for all the cases. In addition, domain confusion loss weight is another parameter, which could be varied. However, as suggested in [2], we used 0.1 for all the cases.

References

[1] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 2960–2967, 2013.

[2] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML 2015, Lille, France, 6-11 July 2015*, pages 1180–1189, 2015.

[3] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2066–2073, 2012.

[4] H. Kuhne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[5] Avraham Levy and Michael Lindenbaum. Efficient sequential karhunen-loeve basis extraction. In *ICCV*, page 739, 2001.

[6] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *Int. J. Computer. Vision*, 77(1-3):125–141, May 2008.

[7] Waqas Sultani and Imran Saleemi. Human action recognition across datasets by foreground-weighted histogram decomposition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 764–771, 2014.

[8] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4489–4497, 2015.