Adversarial Cross-Domain Action Recognition with Co-Attention

Boxiao Pan,[*] Zhangjie Cao,[*] Ehsan Adeli, Juan Carlos Niebles
Stanford University
{bxpan, caozj, eadeli, jniebles}@cs.stanford.edu

Abstract

Action recognition has been a widely studied topic with a heavy focus on supervised learning involving sufficient labeled videos. However, the problem of cross-domain action recognition, where training and testing videos are drawn from different underlying distributions, remains largely underexplored. Previous methods directly employ techniques for cross-domain image recognition, which tend to suffer from the severe temporal misalignment problem. This paper proposes a Temporal Co-attention Network (TCoN), which matches the distributions of temporally aligned action features between source and target domains using a novel cross-domain co-attention mechanism. Experimental results on three cross-domain action recognition datasets demonstrate that TCoN improves both previous single-domain and cross-domain methods significantly under the cross-domain setting.

Introduction

Action recognition has long been studied in the computer vision community because of its wide range of applications in sports (Soomro and Zamir 2014), healthcare (Ogbuabor and La 2018), and surveillance systems (Ranasinghe, Al Machot, and Mayr 2016). Recently, motivated by the success of deep convolution networks in tasks on still images, such as image recognition (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016) and object detection (Girshick 2015; Ren et al. 2015), various deep architectures (Wang et al. 2016; Tran et al. 2015) have been proposed for video action recognition. When large amounts of labeled videos are available, deep learning methods can achieve state-of-the-art performance on several benchmarks (Kay et al. 2017; Kuehne et al. 2011; Soomro, Zamir, and Shah 2012).

Although current action recognition approaches achieve promising results, they mostly assume that the testing data follows the same distribution as the training data. Indeed, the performance of these models degenerate significantly when applied to datasets with different distributions due to domain shift. This greatly limits the application of current action recognition models. An example of domain shift is illustrated in Fig. 1, in which the source video is from a movie while the target video depicts a real-world scene. The challenge of domain shift motivates the problem of *cross-domain* action recognition, where we have a target domain that consists of unlabeled videos and a source domain that consists of labeled videos. The source domain is related to the target domain but is drawn from a different distribution. Our goal is to leverage the source domain to boost the performance of action recognition models on the target domain.

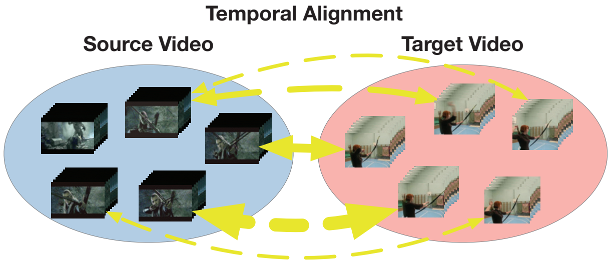


Figure 1: Illustration of our proposed temporal co-attention mechanism. Since different video segments represent distinct action stages, we propose to align segments that have similar semantic meanings. Key segment pairs that are more similar (denoted by thicker arrows) are assigned higher co-attention weights, thus contributing more during the cross-domain alignment stage. Here we show two samples with the action *Archery* from HMDB51 (left) and UCF101 (right).

The problem of cross-domain learning, also known as domain adaptation, has been explored on still image applications such as image recognition (Long et al. 2015; Ganin et al. 2017), object detection (Chen et al. 2018), and semantic segmentation (Hoffman et al. 2017). For still image problems, source and target domains differ mostly on appearance, and typical methods minimize a distribution distance within a latent feature space. However, for action recognition, source and target actions also differ temporally. For example, actions may appear at different time steps or last for different lengths in different domains. Thus, cross-domain action recognition requires matching action feature distributions between domains both spatially and temporally. Current action recognition methods typically generate features per-frame (Wang et al. 2016) or per-segment (Tran

[*] Equal contribution

et al. 2015). Previous cross-domain action recognition methods directly match segment feature distributions (Jamal et al. 2018) or with weights based on an attention mechanism (Chen et al. 2019). However, segment features can only represent parts of the action and may even be irrelevant to the action (e.g., background frames). Naively matching segment feature distributions would ignore the temporal order of segments and can introduce noisy matchings with background segments, which may be sub-optimal.

To address this challenge, we propose a Temporal Co-attention Network (TCoN). We first select segments that are critical for cross-domain action recognition by investigating temporal attention, which is a widely used technique in action recognition (Sharma, Kiros, and Salakhutdinov 2015; Girdhar and Ramanan 2017) that helps the model focus on segments more related to the action. However, the vanilla attention mechanism fails when applied to the cross-domain setting. This is because many key segments are domain-specific, as they might not exist in one domain although they do in the other. Thus when calculating the attention score for a segment, apart from its self-importance, whether it matches with those in the other domain should also be taken into consideration. Only segments that are action-informative and also common in both domains should be paid close attention to. This motivates our design a novel cross-domain co-attention module, which calculates attention scores for a segment based on both its action informativeness as well as cross-domain similarity.

We further design a new matching approach by forming "target-aligned source features" for target videos, which are derived from source features but aligned with target features temporally. Concatenating such target-aligned source segment features in temporal order naturally forms action features for source videos that are temporally aligned with target videos. Then, we match the distributions of the concatenated target-aligned source features with the concatenated target features to achieve cross-domain adaptation. Experimental results show that TCoN outperforms previous methods on several cross-domain action recognition datasets.

In summary, our main contributions are as follows: (1) We design a novel cross-domain co-attention module to concentrate the model on key segments shared by both domains, extending the traditional self-attention to cross-domain co-attention; (2) We propose a novel matching mechanism to enable distribution matching on temporal-aligned features; We also conduct experiments on three challenging benchmark datasets, and the results suggest that the proposed TCoN achieves state-of-the-art performance.

Related Work

Video Action Recognition. With the success of deep Convolutional Neural Networks (CNNs) on image recognition (Krizhevsky, Sutskever, and Hinton 2012), many deep architectures have been proposed to tackle action recognition from videos (Feichtenhofer, Pinz, and Zisserman 2016; Wang et al. 2016; Tran et al. 2015; Carreira and Zisserman 2017; Zhou et al. 2018; Lin, Gan, and Han 2018). One branch of work is based on 2D CNNs. For instance, Two-Stream Network (Feichtenhofer, Pinz, and Zisserman 2016) utilizes an additional optical flow stream to better leverage temporal information. Temporal Segment Network (Wang et al. 2016) proposes a sparse sampling approach to remove redundant information. Temporal Relation Network (Zhou et al. 2018) further presents Temporal Relation Pooling to model frame relations at multiple temporal scales. Temporal Shifting Module (Lin, Gan, and Han 2018) shifts feature channels along the temporal dimension to model temporal information efficiently. Another branch involves using 3D CNNs that learn spatio-temporal features. C3D (Tran et al. 2015) directly extends the 2D convolution operation to 3D. I3D (Carreira and Zisserman 2017) leverages pre-trained 2D CNNs such as ImageNet pre-trained Inception V1 (Szegedy et al. 2015) by inflating 2D convolutional filters into 3D. However, all these works suffer from the spatial and temporal distribution gap between domains, which imposes challenges for cross-domain action recognition.

Domain Adaption. Domain Adaptation aims to solve the cross-domain learning problem. In computer vision, previous work mostly focuses on still images. These methods fall into three categories. The first category focuses on minimizing distribution distances between source and target domains. DAN (Long et al. 2015) minimizes the MMD distance between feature distributions. DANN (Ganin et al. 2017) and CDAN (Long et al. 2018) minimize the Jensen-Shannon divergence between feature distributions with adversarial learning (Goodfellow et al. 2014). The second category exploits techniques on semi-supervised learning. RTN (Long et al. 2016) exploits entropy minimization while Asym-Tri (Saito, Ushiku, and Harada 2017) uses pseudo labels. The third category groups all the image translation methods. Hoffman *et al*. (Hoffman et al. 2017) and Murez *et al*. (Murez et al. 2018) translate source labeled images to the target domain to enable supervised learning there. In this work, we adapt the first two categories of methods to videos as there is no sophisticated video translation method yet.

Cross-Domain Action Recognition. Despite the success of previous domain adaptation methods, cross-domain action recognition remains largely unexplored. Bian *et al*. (Bian, Tao, and Rui 2012) is the first to tackle this problem, which learns bag-of-words features to represent target videos and then regularizes the target topic model by aligning topic pairs across domains. However, their method requires partial target labeled data. Tang *et al*. (Tang et al. 2016) learns a projection matrix for each domain to map all features into a common latent space. Liu *et al*. (Liu et al. 2019) employs more domains under the assumption that domains are bijective and trains the classifier with a multi-task loss. However, these methods assume deep video-level features available, which are not yet mature enough. Another work (Jamal et al. 2018) employs the popular GAN-based image domain adaptation approach to match segment features directly. Very recently, TA^{3}N (Chen et al. 2019) proposes to attentively adapt segments that contribute the most to the overall domain shift by leveraging the entropy of domain label predictor. However, both deep models suffer from temporal misalignment between domains since they only match segment features.

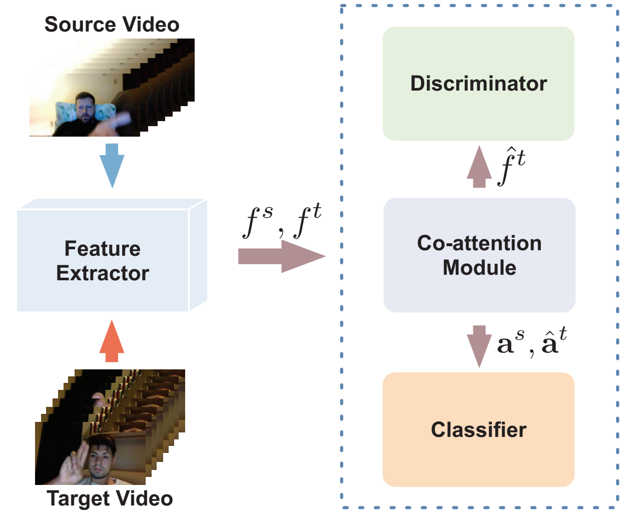


Figure 2: Our proposed TCoN framework. We first generate source and target features f^s and f^t with a feature extractor G_f. The co-attention module uses f^s and f^t to generate ground-truth attention scores for source video ($\mathbf{a}^s$) and predicted score for target video ($\hat{\mathbf{a}}^t$), which the classifier uses to make predictions. At the same time, the co-attention module generates target-aligned source features $\hat{f}^t$, which the discriminator receives to enable temporally aligned distribution matching (see Figs. 3 and 4 for details).

Temporal Co-attention Network (TCoN)

Suppose we have a source domain consisting of N_s labeled videos $\mathcal{D}^s = \{V^s_i, y_i\}_{i=1}^{N_s}$ and a target domain consisting of N_t unlabeled videos $\mathcal{D}^t = \{V^t_{i'}\}_{i'=1}^{N_t}$. The two domains are drawn from two different underlying distributions p_s and p_t, but they are related and share the same label space. We will also provide analysis for the case where they do not share the same space in later section. The goal of cross-domain action recognition is to design an adaptation mechanism to transfer the recognition model learned from the source domain to the target domain with a low classification risk.

In addition to the appearance gap similar to the image case, there are two other main challenges that are specific to cross-domain action recognition. First, not all frames are useful under the cross-domain setting. Non-key frames contain noisy background information unrelated to the action, and even key frames can exhibit different cues for different domains. Second, current action recognition networks cannot generate holistic action features for the entire action in the video. Instead, they produce features of segments. Since segments are not temporally aligned between videos, it is hard to construct features for the entire action. To attack these two challenges, we design a co-attention module to focus on segments that contain important cues and are shared by both domains, which addresses the first challenge. We further leverage the co-attention module to generate temporally target-aligned source features. This enables distribution matching on temporally aligned features between domains, thus addressing the second challenge.

Architecture

Fig. 2 shows the overall architecture of TCoN. During training, given a source and target video pair, we first partition each source video into K_s segments and each target video into K_t segments uniformly. We use v^s_{ij} to denote the j-th segment in the i-th source video, and $v^t_{i'j'}$ is similarly defined for the target video. Then, we generate features f^*_{ij} ($* \in (s,t)$) for each segment with a feature extractor G_f, which is a 2D or 3D CNN, i.e., $f^*_{ij} = G_f(v^*_{ij})$. Next, we use the co-attention module to calculate the co-attention matrix between this source and target video pair, from which we further derive the source and target attention score vectors, as well as the target-aligned source feature $\hat{f}^t$. Then, the discriminator module performs distribution matching given the source, target, and target-aligned source features. Finally, a shared classifier G_y accepts the source and target features together with their attention scores, to predict the labels $\hat{y}^*_i$. For label prediction, we follow the standard practice, where we first predict per-segment labels and then weighted-sum segment predictions by attention scores.

Cross-Domain Co-Attention

Co-attention was originally used in NLP to capture the interactions between questions and documents to boost the performance of question answering models (Xiong, Zhong, and Socher 2016). Motivated by this, we propose a novel cross-domain co-attention mechanism to capture correlations between videos from two domains. To the best of our knowledge, this is the first time that co-attention is explored under the cross-domain action recognition setting.

The goal of the co-attention module is to model relations between source and target video pairs. As shown in Fig. 3, given a pair of source and target videos V^s_i and $V^t_{i'}$, and their segment features $f^s_{ij}|_{j=1}^{K_s}$ and $f^t_{i'j'}|_{j'=1}^{K_t}$, we use k to index video pairs, i.e., $pair(k) = (i, i')$, which denotes that video pair k is composed of the source video V^s_i and the target video $V^t_{i'}$. We first calculate a self-attention score vector $\mathbf{a}^{ss}_i$ and $\mathbf{a}^{tt}_{i'}$ for each video:

$$a^{ss}_{ij} = \frac{1}{K_s - 1} \sum_{\bar{j} \neq j} \langle f^s_{i\bar{j}}, f^s_{ij} \rangle, \tag{1}$$

$$a^{tt}_{i'j'} = \frac{1}{K_t - 1} \sum_{\bar{j}' \neq j'} \langle f^t_{i'\bar{j}'}, f^t_{i'j'} \rangle, \tag{2}$$

where a^{ss}_{ij} and $a^{tt}_{i'j'}$ are the j-th and j'-th element of $\mathbf{a}^{ss}_i$ and $\mathbf{a}^{tt}_{i'}$, respectively. $\langle \cdot, \cdot \rangle$ denotes the inner-product operation. Self-attention score vectors measure the intra-domain importance of a segment within a video. After we obtain these self-attention score vectors, we then derive each (j, j')-th element of the cross-domain similarity matrix A^{st}_k as $a^{st}_{jj'} = \langle f^s_{ij}, f^t_{i'j'} \rangle$. Note that for clarity, we drop the pair index k for $a^{st}_{jj'}$. Each element $a^{st}_{jj'}$ measures the cross-domain similarity between segment pair v^s_{ij} and $v^t_{i'j'}$. Finally, we calculate the cross-domain co-attention score matrix A^{co}_k by

$$A^{co}_k = \left(\mathbf{a}^{ss}_i (\mathbf{a}^{tt}_{i'})^T \right) \odot A^{st}_k, \tag{3}$$

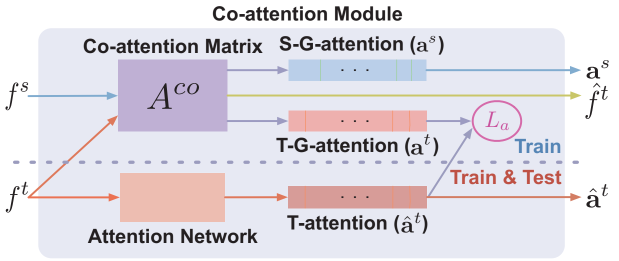


Figure 3: Structure of the co-attention module. It first calculates the co-attention matrix A^{co} from source and target features f^s and f^t. Then, source (S-G-attention) and target (T-G-attention) ground-truth attention $\mathbf{a}^s$ and $\mathbf{a}^t$ as well as target-aligned source features $\hat{f}^t$ are derived from A^{co}. $\mathbf{a}^t$ is used to train the target attention network, which predicts target attention (T-attention) $\hat{\mathbf{a}}^t$ for test time use.

where $\odot$ represents element-wise multiplication. As can be seen from this process, in the co-attention matrix A_k^{co}, only the elements corresponding to those segment pairs v_{ij}^s and $v_{i'j'}^t$ with high a_{ij}^{ss}, $a_{i'j'}^{tt}$ and $a_{jj'}^{st}$ would be assigned high values, which means that only key segment pairs that are also common for both domains will be paid high attention to. Thus, this co-attention matrix effectively reflects the correlations between source and target video pairs. Also, key segments (those with only high a_{ij}^{ss} or $a_{i'j'}^{tt}$) or common segments (those with only high $a_{jj'}^{st}$) are not ignored, but paid less attention to. Only segments that are neither important nor common are discarded as they are essentially noise and do not help with the task.

For each segment, we derive attention scores from the above co-attention matrix by averaging its co-attention scores with all segments in the other video. We generate the ground-truth attention $\mathbf{a}_i^s$ and $\mathbf{a}_{i'}^t$ for V_i^s and $V_{i'}^t$ as follows,

$$\mathbf{a}_i^s = \frac{1}{M_i^s} \sum_{i \in pair(k)} \sum_{row} A_k^{co}, \tag{4}$$

$$\mathbf{a}_{i'}^t = \frac{1}{M_{i'}^t} \sum_{i' \in pair(k)} \sum_{column} A_k^{co}, \tag{5}$$

where $\sum_{row}$ and $\sum_{column}$ are summation with respect to rows and columns, respectively. M_i^s is the number of related pairs to V_i^s and $M_{i'}^t$ is defined similarly. All the attention vectors sum to 1. Since we should not assume access to source videos during testing time, we further use an attention network G_a, which is a fully-connected network, to predict attention scores for target videos:

$$\hat{a}_{i'j'}^t = G_a(f_{i'j'}^t), \tag{6}$$

where $\hat{a}_{i'j'}^t$ is the j'-th element of the predicted attention. We further calculate the loss for the attention network with supervision from the ground-truth attention:

$$C_a = \frac{1}{N_t} \sum_{i'=1}^{N_t} L_a(\hat{\mathbf{a}}_{i'}^t, \mathbf{a}_{i'}^t), \tag{7}$$

where L_a is the regression loss. With the source and target attention, the final classification loss is thus:

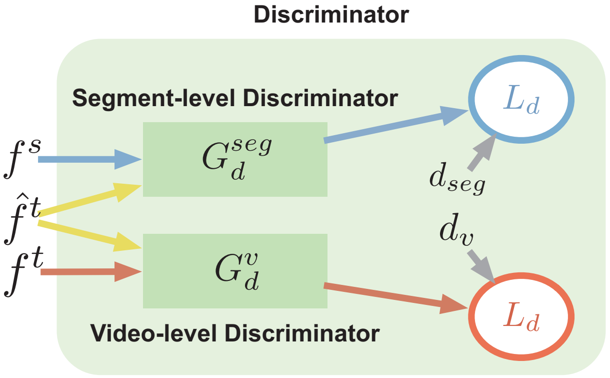


Figure 4: Structure of the discriminator. Source f^s and target-aligned source $\hat{f}^t$ segment features are input to the segment-level discriminator to ensure that $\hat{f}^t$ does follow the source feature distribution. Target f^t and target-aligned source $\hat{f}^t$ video features are input to the video-level discriminator to enable temporally aligned distribution matching.

$$\begin{aligned} C_y = & \frac{1}{N_s} \sum_{i}^{N_s} L_y\Big(\sum_{j=1}^{K_s} a_{ij}^s G_y(f_{ij}^s), y_i^s\Big) \\ & + \frac{1}{N_t} \sum_{i'}^{N_t} L_y\Big(\sum_{j'=1}^{K_t} \hat{a}_{i'j'}^t G_y(f_{i'j'}^t), y_{i'}^t\Big), \end{aligned} \tag{8}$$

where L_y is the cross-entropy loss for classification. We train the classifier using source videos with ground-truth labels. Similar to previous work (Saito, Ushiku, and Harada 2017), we also use target videos that have high label prediction confidence as training data, where the predicted pseudo-labels serve as the supervision. This helps preserve the λ term in the error bound derived in Theorem 1 in (Ben-David et al. 2007). Note that the total number of source and target video pairs is quadratic to the size of dataset, which is very large. For efficiency and co-attention with higher quality, we only calculate co-attention for video pairs with similar semantic information, i.e., video pairs with similar label prediction probability (which acts as a soft label).

Temporal Adaptation

Fig. 4 illustrates the video-level discriminator we employ to match the distributions of target-aligned source and target video features, and the segment-level discriminator we use to match target-aligned source and source segment features.

For each pair k of video V_i^s and $V_{i'}^t$, the target-aligned source segment feature $\hat{f}_{kj'}^t|_{j'=1}^{K_t}$ is calculated as follows,

$$\hat{f}_{kj'}^t = \sum_{j=1}^{K_s} (A_k^{co})_{jj'} f_{ij}^s, \tag{9}$$

where $(A_k^{co})_{jj'}$ is the (j, j')-th element of A_k^{co}. Note that after we get A_k^{co}, we further normalize each column of it with a softmax function to keep the norm of target-aligned source features same as source features. Each target-aligned source segment feature is thus derived by weighted-summing source segment features, where the weight is the co-attention score between the target segment feature and each source segment feature. Thus, each target-aligned source segment feature preserves the semantic meaning of the corresponding target segment and falls on the source distribution. We concatenate segment features $\hat{f}_{kj'}^t|_{j'=1}^{K_t}$ as $\hat{F}_k^t$ and $f_{i'j'}^t|_{j'=1}^{K_t}$ as $F_{i'}^t$ in temporal order, which then naturally form as action features. Furthermore, $F_{i'}^t$ and all corresponding $\hat{F}_k^t$ are strictly temporally aligned since segments at the same time step express the same semantic meaning.

Then, we can derive the domain adversarial loss to match video distributions of target-aligned source features and target features. To further ensure the target-aligned source features to fall in the source feature space, we also match the segment distributions of target-aligned source features and source features. The loss for the video-level C_{dv} and segment-level C_{ds} discriminators are defined as follows,

$$C_{dv} = \frac{1}{N_t}\sum_{i'} L_d(G_d^v(F_{i'}^t), d_v) + \frac{1}{N_{st}}\sum_{k} L_d(G_d^v(\hat{F}_k^t), d_v), \tag{10}$$

$$\begin{aligned} C_{ds} = & \frac{1}{N_s K_s}\sum_{i,j} L_d(G_d^{seg}(f_{ij}^s), d_{seg}) \\ & + \frac{1}{N_{st} K_t}\sum_{k,j'} L_d(G_d^{seg}(\hat{f}_{kj'}^t), d_{seg}), \end{aligned} \tag{11}$$

where N_{st} is the number of all pairs and L_d is the binary cross-entropy loss. The video domain label d_v is 0 for target-aligned source features and 1 for target features. The segment domain label d_{seg} is 0 for target-aligned source segment features and 1 for target segment features.

Optimization

We perform optimization in the adversarial learning manner (Ganin et al. 2017). We use θ_f, θ_a, θ_y, θ_d^v and θ_d^{seg} to denote the parameters for G_f, G_a, G_y, G_d^v and G_d^{seg} and optimize:

$$\min_{\theta_f,\theta_a,\theta_y} C_y + \lambda_a C_a - \lambda_d(C_{dv} - C_{ds}), \tag{12}$$

$$\min_{\theta_d^v,\theta_d^{seg}} C_{dv} + C_{ds}, \tag{13}$$

where λ_a and λ_d are trade-off hyper-parameters.

With the proposed Temporal Co-attention Network that contains an attentive classifier as well as a video- and a segment-level adversarial networks, we can simultaneously align the source and target video distributions while also minimize the classification error within both source and target domains, thus address the cross-domain action recognition problem in an effective way.

Experiments

Most prior work was done on small-scale datasets such as (UCF101-HMDB51)$_1$ (Tang et al. 2016), (UCF101-HMDB51)$_2$ (Chen et al. 2019) and UCF50-Olympic_Sports (Jamal et al. 2018). For fair comparison with these works, we also evaluate our proposed method on these datasets. Moreover, we construct a large-scale cross-domain dataset, namely Jester (S)-Jester (T) (S for source, T for target), and further conduct experiments there. For (UCF101-HMDB51)$_1$, (UCF101- HMDB51)$_2$ and UCF50-Olympic_Sports, we follow the prior works to construct the datasets by selecting the same action classes in two domains, while for Jester, we merge sub-actions into super-actions and split half of sub-actions into each domain. Please refer to the supplementary material for full details.

There are different types and extent of domain gap present in different datasets. For (UCF101-HMDB51)$_1$, (UCF101-HMDB51)$_2$ and UCF50-Olympic_Sports, the domain gap is caused by appearance, lighting, camera viewpoint, etc., but not the action. Whereas for Jester, the gap arises from different action dynamics instead of other factors (since data samples from the same dataset but different sub-actions constitute a single super-action class). Hence, models trained on Jester suffer more from the temporal misalignment problem. Together with being at a larger scale, Jester is considered to be much harder than the other datasets.

We compare **TCoN** with single-domain methods (**TSN** & **C3D** & **TRN**) pre-trained on the source dataset, several cross-domain action recognition methods including a shallow learning method **CMFGLR** (Tang et al. 2016), deep learning methods **DAAA** (Jamal et al. 2018) and **TA^{3}N** (Chen et al. 2019), as well as a hybrid model which directly applies state-of-the-art domain adaptation method **CDAN** (Long et al. 2018) to videos. We mainly use TSN (Wang et al. 2016) as our backbone, but for a fair comparison with the prior work, we also conduct experiments using C3D (Tran et al. 2015) and TRN (Zhou et al. 2018).

Training Details

For TSN, C3D and TRN, we train on source and test on target directly. For shallow learning methods, we use deep features from the source pre-trained model as input. For the hybrid model, we apply domain discriminator in CDAN to segment features and the consensus domain discriminator output of all segments as final output. For DAAA and TA^{3}N, we use their original training strategy. For TCoN, since the target attention network is not well-trained at the beginning, we use uniform attention at the first few iterations and plug it in when the loss of the attention network is lower than a certain threshold. To train TCoN more efficiently, we only calculate co-attention for segment pairs within mini-batchs.

We implement TCoN with the PyTorch framework (Paszke et al. 2017). We use the Adam optimizer (Kingma and Ba 2014) and set the batch size to 64. For TSN and TRN-based models, we adopt the BN-Inception (Ioffe and Szegedy 2015) backbone pre-trained on ImageNet (Deng et al. 2009). The learning rate is initialized to 0.0003 and decreases by $\frac{1}{10}$ every 30 epochs. We adopt the same data augmentation technique as in (Wang et al. 2016). For C3D-based models, we strictly follow the settings in (Jamal et al. 2018) and use the same base model (Tran et al. 2015) pre-trained on Sports-1M dataset (Karpathy et al. 2014). We initialize the learning rate for the feature extractor to 0.001 while 0.01 for classifier since it is trained from scratch. The trade-off parameter λ_d is increased gradually from 0 to 1 as in DANN (Ganin and Lempitsky 2014). For the number of segments, We do grid search for each dataset in *[1, minimum*

Table 1: Accuracy (%) of TcoN and compared methods on three datasets based on the TSN backbone. R + F denotes the results obtained by combining predictions from RGB and Flow models, which are calculated by averaging the logits (before softmax) from RGB and Flow models and then selecting the class with the highest entry in the obtained logits.

Method	(HMDB51 → UCF101)$_1$			UCF50 → Olympic_Sports			Olympic_Sports → UCF50			Jester (S) → Jester (T)		
	RGB	Flow	R + F	RGB	Flow	R + F	RGB	Flow	R + F	RGB	Flow	R + F
TSN (Wang et al. 2016)	82.10	76.86	83.11	80.00	81.82	81.75	76.67	73.34	74.47	51.70	49.89	50.56
CMFGLR (Tang et al. 2016)	85.14	78.45	84.85	81.06	79.64	80.23	77.43	77.05	78.89	52.52	54.34	53.36
DAAA (Jamal et al. 2018)	88.36	89.93	91.31	88.37	88.16	89.01	86.25	87.00	87.93	56.45	55.92	57.63
CDAN (Long et al. 2018)	90.09	90.96	91.86	90.65	90.46	91.77	90.08	90.13	90.57	58.33	55.09	59.30
TCoN (ours)	**93.01**	**96.07**	**96.78**	**93.91**	**95.46**	**95.77**	**91.65**	**93.77**	**94.12**	**61.78**	**71.11**	**72.24**

Table 2: Accuracy (%) of TCoN and DAAA on two tasks with UCF50-Olympic_Sports dataset based on C3D backbone.

Method	UCF50 → Olympic_Sports			Olympic_Sports → UCF50		
	RGB	Flow	R + F	RGB	Flow	R + F
C3D (Tran et al. 2015)	82.13 ±0.52	81.12 ±0.85	83.05 ±0.65	83.16 ±0.75	81.02 ±0.97	83.79 ±0.82
DAAA (Jamal et al. 2018)	91.60 ±0.18	89.16 ±0.26	91.37 ±0.22	89.96 ±0.35	89.11 ±0.47	90.32 ±0.43
TCoN (ours)	**94.73** ±0.12	**96.03**±0.15	**95.92** ±0.14	**92.88** ±0.28	**94.25** ±0.22	**94.77** ±0.25

Table 3: Accuracy (%) of TCoN and TA^{3}N (Chen et al. 2019) based on TRN backbone using only RGB input (U: UCF50 / UCF101, O: Olympic_Sports, H: HMDB51).

Method	U → O	O → U	(U → H)$_2$	(H → U)$_2$	J(S) → J(T)
TA^{3}N	**98.15**	92.92	78.33	81.79	60.11
TCoN	96.82	**96.79**	**87.24**	**89.06**	**62.53**

Table 4: Accuracy (%) of TCoN compared with baselines when non-overlapping classes exist between domains.

Method	(HMDB51 → UCF101)$_{\text{all}}$
TSN (Wang et al. 2016)	66.81
TRN (Zhou et al. 2018)	68.07
DAAA (Jamal et al. 2018)	71.45
TCoN (ours)	**75.23**

Table 5: Accuracy (%) of TCoN and its variants.

Method	Jester (S) → Jester (T)		
	RGB	Flow	R + F
TCoN - SAdNet	61.23	68.23	71.13
TCoN - TAdNet	58.76	64.56	65.48
TCoN - CoAttn	57.25	56.93	57.95
TCoN - Attn	59.03	62.74	63.13
TCoN	**61.78**	**71.11**	**72.24**

video length] on a validation set. Please refer to supplementary material for the actual numbers.

Experimental Results

The classification accuracies on three datasets using TSN-based TCoN are shown in Table 1. We can observe that the proposed TCoN outperforms all baselines on all datasets. In particular, TCoN improves previous methods with the largest margin on Jester, where temporal information is much more important, and temporal misalignment is more severe. Both CDAN and DAAA use segment features and minimize the Jensen-Shannon divergence of segment feature distributions between domains. The higher accuracy of TCoN demonstrates the importance of temporal alignment in distribution matching. We also notice that the Flow model consistently outperforms the RGB model for TCoN, indicating that TCoN well utilizes temporal information.

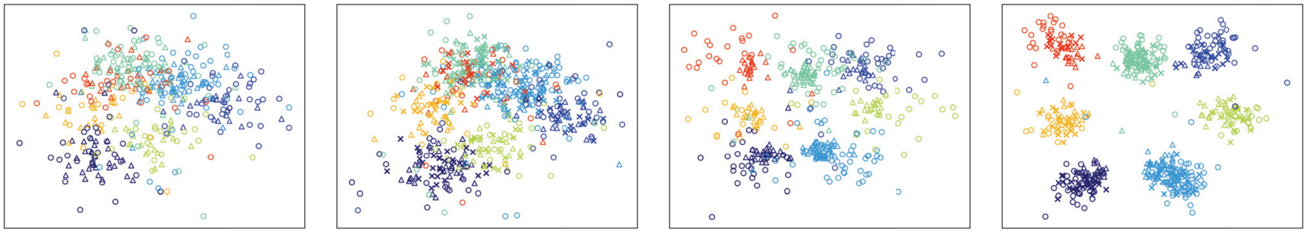
We also compare with DAAA (Jamal et al. 2018) under their experiment setting with the C3D backbone. From Table 2, we can observe that TCoN outperforms DAAA on both tasks, which further suggests the efficacy of the proposed co-attention and distribution matching mechanism.

Moreover, we compare with the state-of-the-art cross-domain action recognition method TA^{3}N (Chen et al. 2019) on two datasets they used, namely UCF50-Olympic_Sports and (UCF101-HMDB51)$_2$ (which is slightly different from (UCF101-HMDB51)$_1$ in 3 out of the 12 shared classes), as well as Jester. And we use the same backbone TRN (Zhou et al. 2018) and input modality (RGB) as theirs. The results from Table 3 show that on the datasets they used, TCoN outperforms TA^{3}N on 3 out of 4 tasks and on par with it on the other. Moreover, TCoN achieves better performance on Jester, which again corroborates that TCoN can well handle not only the appearance gap but also the action gap.

To test whether our model is robust when the two domains do not share the same action space during training, we conduct experiments on (HMDB51 → UCF101)$_{\text{all}}$, where we train TCoN on data from all classes in HMDB51, but only test on the shared classes between two domains (it is impossible to predict those non-overlapping target classes). The results from Table 4 show that TCoN still outperforms its baselines, suggesting the robustness of it in this case.

Analysis

Ablation Study. We compare TCoN with its four variants: (1) **TCoN - SAdNet** is the variant without the segment-level discriminator; (2) **TCoN - TAdNet** is the variant without using target-aligned source features but directly matching the source and concatenated target features with one discriminator; (3) **TCoN - CoAttn** is the variant with no



(a) DAAA segment features (b) TCoN segment features (c) DAAA video features (d) TCoN video features

Figure 5: t-SNE visualization of features from DAAA (Jamal et al. 2018) and our TCoN. The first two figures are segment features, while the last two are video features. Different colors represent different classes. △ stands for source features. ○ denotes target features. × represents target-aligned source features.

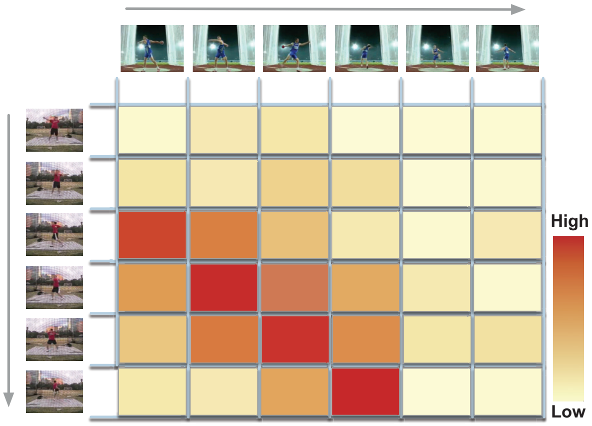


Figure 6: Co-attention matrix visualization for a video pair from UCF50 → Olympic_Sports.

co-attention computation. For the attentive classifier, we use self-attention instead; (4) **TCoN - AttnClassifier** is the variant where we directly average the classifier outputs from all segments instead of weighing them with attention scores generated from the co-attention matrix. The ablation study results on Jester (S) → Jester (T) are shown in Table 5, from which we can make the following observations: (1) TCoN outperforms TCoN-SAdNet on all modalities, demonstrating that the segment feature distribution for target-aligned source and source features are not exactly the same and a segment-level discriminator helps match the distributions; (2) TCoN outperforms TCoN-TAdNet by a large margin. This proves that target-aligned source features ease the temporal misalignment problem and improve distribution matching; (3) TCoN beats TCoN-CoAttn. This verifies the necessity of co-attention, which reflects both segment importance and similarity, in cross-domain action recognition; (4) TCoN outperforms TCoN-Attn, indicating that segments contribute to the prediction differently and it is crucial to focus on those informative ones.

Visualization of Co-Attention. We further visualize the co-attention matrix between a video pair on the UCF50 → Olympic_Sports dataset. The visualization is shown in Fig. 6, where the left video is from the source, and the top video is from the target. According to the co-attention matrix, we can observe that the first four frames of the target domain match the last four frames in the source domain, and the co-attention matrix assigns high values for these pairs. The first two frames of the source video show the person preparing his body for discus throw, which is not actually discus throw, thus are not considered as key frames. The last two frames of the target video are for the ending stage of the action, which are important but do not exist in the source video. This proves that our co-attention mechanism exactly focuses attention on segments containing key action parts that are similar across source and target domains.

Feature Visualization. We also plot the t-SNE embedding (Donahue et al. 2014) for both segment and video features for DAAA and TCoN on Jester in Fig. 5(a) - 5(d). For DAAA, we visualize the source (triangle) and target (circle) features. For TCoN, we visualize target-aligned source features (cross) as well. From Fig. 5(a) and 5(b), we can observe that the segment features from different classes (shown in different colors) are mixed together, which is expected since segments cannot represent the entire action. In TCoN, the distributions of source and target-aligned source segment features are indistinguishable, demonstrating the effectiveness of our segment-level discriminator. From Fig. 5(c) and 5(d), we can observe that for video features, TCoN has a better cluster structure than DAAA. In particular, in Fig. 5(d), those points representing target-aligned source features lie between source and target feature points, suggesting that they actually bridge source and target features together. This sheds light on how the proposed distribution matching mechanism draws the target action distribution closer to the source by leveraging the target-aligned source features.

Conclusion

In this paper, we propose TCoN to address cross-domain action recognition. We design a cross-domain co-attention mechanism, which guides the model to pay more attention to common key frames across domains. We further introduce a temporally aligned distribution matching technique that enables distribution matching of action features. Extensive experiments on three benchmark datasets verify that our proposed TCoN achieves state-of-the-art performance.

Acknowledgements. The authors would like to thank Panasonic, Oppo, and Tencent for the support.

References

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2007. Analysis of representations for domain adaptation. In *NIPS*.

Bian, W.; Tao, D.; and Rui, Y. 2012. Cross-domain human action recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*.

Carreira, J., and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*.

Chen, M.-H.; Kira, Z.; AlRegib, G.; Woo, J.; Chen, R.; and Zheng, J. 2019. Temporal attentive alignment for large-scale video domain adaptation. *arXiv preprint arXiv:1907.12743*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *CVPR*.

Ganin, Y., and Lempitsky, V. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Marchand, M.; and Lempitsky, V. 2017. Domain-adversarial training of neural networks. *JMLR*.

Girdhar, R., and Ramanan, D. 2017. Attentional pooling for action recognition. In *NIPS*.

Girshick, R. 2015. Fast r-cnn. In *ICCV*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A. A.; and Darrell, T. 2017. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jamal, A.; Namboodiri, V. P.; Deodhare, D.; and Venkatesh, K. S. 2018. Deep domain adaptation in action space. In *BMVC*.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. Hmdb: a large video database for human motion recognition. In *ICCV*. IEEE.

Lin, J.; Gan, C.; and Han, S. 2018. Temporal shift module for efficient video understanding. *arXiv preprint arXiv:1811.08383*.

Liu, A.-A.; Xu, N.; Nie, W.-Z.; Su, Y.-T.; and Zhang, Y.-D. 2019. Multi-domain and multi-task learning for human action recognition. *IEEE Transactions on Image Processing*.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. In *ICML*.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2016. Unsupervised domain adaptation with residual transfer networks. In *NIPS*.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. In *NIPS*.

Murez, Z.; Kolouri, S.; Kriegman, D.; Ramamoorthi, R.; and Kim, K. 2018. Image to image translation for domain adaptation. In *CVPR*.

Ogbuabor, G., and La, R. 2018. Human activity recognition for healthcare using smartphones. In *ICMLC*. ACM.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Ranasinghe, S.; Al Machot, F.; and Mayr, H. C. 2016. A review on applications of activity recognition systems with regard to performance and evaluation. *IJDSN*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Saito, K.; Ushiku, Y.; and Harada, T. 2017. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*. JMLR. org.

Sharma, S.; Kiros, R.; and Salakhutdinov, R. 2015. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*.

Soomro, K., and Zamir, A. R. 2014. Action recognition in realistic sports videos. In *Computer vision in sports*. Springer.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

Tang, J.; Jin, H.; Tan, S.; and Liang, D. 2016. Cross-domain action recognition via collective matrix factorization with graph laplacian regularization. *Image Vision Comput.* (P2).

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*. Springer.

Xiong, C.; Zhong, V.; and Socher, R. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.

Zhou, B.; Andonian, A.; Oliva, A.; and Torralba, A. 2018. Temporal relational reasoning in videos. In *ECCV*.