Action Segmentation with Mixed Temporal Domain Adaptation

Min-Hung Chen\textsuperscript{1*} Baopu Li\textsuperscript{2} Yingze Bao\textsuperscript{2} Ghassan AlRegib\textsuperscript{1}
\textsuperscript{1}Georgia Institute of Technology \textsuperscript{2}Baidu USA
cmhungsteve@gatech.edu \{baopuli, baoyingze\}@baidu.com alregib@gatech.edu

Abstract

The main progress for action segmentation comes from densely-annotated data for fully-supervised learning. Since manual annotation for frame-level actions is time-consuming and challenging, we propose to exploit auxiliary unlabeled videos, which are much easier to obtain, by shaping this problem as a domain adaptation (DA) problem. Although various DA techniques have been proposed in recent years, most of them have been developed only for the spatial direction. Therefore, we propose Mixed Temporal Domain Adaptation (MTDA) to jointly align frame- and video-level embedded feature spaces across domains, and further integrate with the domain attention mechanism to focus on aligning the frame-level features with higher domain discrepancy, leading to more effective domain adaptation. Finally, we evaluate our proposed methods on three challenging datasets (GTEA, 50Salads, and Breakfast), and validate that MTDA outperforms the current state-of-the-art methods on all three datasets by large margins (e.g. 6.4\% gain on F1@50 and 6.8\% gain on the edit score for GTEA).

1. Introduction

Action segmentation is of significant importance for a wide range of applications, including video surveillance and analysis of daily human activities. Given a video, the goal is to simultaneously segment the video by time and predict each segment with a corresponding action category. While video classification has shown great progress given the recent success of deep neural networks [40, 26, 25], temporally locating and recognizing action segments in long untrimmed videos is still challenging.

Action segmentation approaches can be factorized into extracting low-level features using convolutional neural networks and applying high-level temporal models. Encouraged by the advances in speech synthesis [28], recent approaches rely on temporal convolutions to capture long-range dependencies across frames using a hierarchy of temporal convolutional filters [17, 4, 19, 7].

Despite the success of these temporal models, the performance gains come from densely-annotated data for fully-supervised learning. Since manually annotating precise frame-by-frame actions is time-consuming and challenging, these methods are not easy to extend to larger scale for real-world applications.

In this paper, we regard action segmentation as a domain adaptation (DA) problem with the transductive set-
ting [29, 3] given the observation that the main challenge is the distributional discrepancy caused by spatio-temporal variations across domains. For example, different people (also noted as subjects) may perform the same action with different styles in terms of spatial locations and temporal duration. The variations in the background environment also contribute to the overall domain discrepancy. To solve this problem, we propose to diminish the domain discrepancy by utilizing auxiliary unlabeled videos, which are much easier to obtain.

Videos can suffer from domain discrepancy along both the spatial and temporal directions, bringing the need of alignment for embedded feature spaces along both directions [2]. However, most DA approaches have been developed only for images and not videos [22, 24, 10, 11, 21, 20, 33]. Therefore, we propose Mixed Temporal Domain Adaptation (MTDA) to jointly align frame- and video-level embedded feature spaces across domains, as shown in Figure 1. We further integrate with the domain attention mechanism to focus on aligning the frame-level features with higher domain discrepancy, leading to more effective domain adaptation. To support our claims, we evaluate our approaches on three datasets with high spatio-temporal domain discrepancy: Georgia Tech Egocentric Activities (GTEA) [8], 50Salads [35], and the Breakfast dataset [14], and achieve new state-of-the-art performance on all three datasets. Since our approach can adapt a model trained in one environment to new environments using only unlabeled videos without additional manual annotation, it is applicable to large-scale real-world scenarios, such as video surveillance.

In summary, our contributions are three-fold:

1. **Local Temporal Domain Adaptation**: We propose an effective adversarial-based DA method to learn domain-invariant frame-level features. To the best of our knowledge, this is the first work to utilize unlabeled videos as auxiliary data to diminish spatio-temporal variations for action segmentation.

2. **Mixed Temporal Domain Adaptation (MTDA)**: We jointly align local and global embedded feature spaces across domains by integrating additional DA mechanism which aligns the video-level feature spaces. Furthermore, we integrate the domain attention mechanism to aggregate domain-specific frames to form global video representations, leading to more effective domain adaptation.

3. **Experiments and Analyses**: We evaluate on three challenging real-world datasets and outperform all the previous state-of-the-art methods. We also perform analysis and ablation study on different design choices to identify key contributions of each component.

### 2. Related Works

In this section, we review the most recent work for action segmentation including the fully- and weakly-supervised setting. We also review the most related domain adaptation work for images and videos.

**Action Segmentation.** Encouraged by the advances in speech synthesis [28], recent approaches rely on temporal convolutions to capture long-range dependencies across frames using a hierarchy of temporal convolutional filters [17, 4, 19, 7]. ED-TCN [17] follows an encoder-decoder architecture with a temporal convolution and pooling in the encoder, and upsampling followed by deconvolution in the decoder. TricorNet [4] replaces the convolutional decoder in the ED-TCN with a bi-directional LSTM (Bi-LSTM). TDRN [19] builds on top of ED-TCN [17] and use deformable convolutions instead of the normal convolution and add a residual stream to the encoder-decoder model. MS-TCN [7] stacks multiple stages of temporal convolutional network (TCN) where each TCN consists of multiple temporal convolutional layers performing acausal dilated 1D convolution. With the multi-stage architecture, each stage takes an initial prediction from the previous stage and refines it. We build our approach on top of MS-TCN, focusing on developing methods to effectively exploit unlabeled videos instead of modifying the architecture.

**Domain Adaptation.** Most recent DA approaches are based on deep learning architectures designed for addressing the domain shift problems given the fact that the deep CNN features without any DA method outperform traditional DA methods using hand-crafted features [6]. Most DA methods follow the two-branch (source and target) architecture, and aim to find a common feature space between the source and target domains. The models are therefore optimized with a combination of classification and domain losses [3].

One of the main classes of methods used is Discrepancy-based DA, whose metrics are designed to measure the distance between source and target feature distributions, including variations of maximum mean discrepancy (MMD) [22, 23, 45, 44, 24] and the CORAL loss [37]. By diminishing the distance of distributions, discrepancy-based DA methods reduce the gap across domains. Another common method, Adversarial-based DA, adopts a similar concept as GANs [12] by integrating domain discriminators into the architectures. Through the adversarial objectives, the discriminators are optimized to classify different domains, while the feature extractors are optimized in the opposite direction. ADDA [39] uses an inverted label GAN loss to split the optimization into two parts: one for the discriminator and the other for the generator. In contrast, the gradient reversal layer (GRL) is adopted in some works [10, 11, 46] to invert the gradients so that the discriminator and generator are optimized simultaneously. Re-
cently, TADA [41] adopts the attention mechanism to adapt the transferable regions and images. Differently, we design the attention mechanism for spatio-temporal domains, aiming to attend to the important parts of temporal dynamics for domain adaptation.

**Domain Adaptation for Actions.** Unlike image-based DA, video-based DA is still an under-explored area. A few works focus on small-scale video DA with only few overlapping categories [36, 43, 13]. [36] improves the domain generalizability by decreasing the effect of the background. [43] maps source and target features to a common feature space using shallow neural networks. AMLS [13] adapts pre-extracted C3D [38] features on a Grassmann manifold obtained using PCA. However, the datasets used in the above works are too small to have enough domain shift to evaluate DA performance. Recently, Chen et al. [2] propose two larger cross-domain datasets for action recognition and the state-of-the-art approach TA^3N. However, these works focus only on the classification task while we concentrate on the more challenging temporal segmentation task.

### 3. Technical Approach

We first introduce our baseline model which is the current state-of-the-art approach for action segmentation, MS-TCN [7] (Section 3.1). And then we describe how we incorporate unlabeled video to align frame-level feature spaces (Section 3.2), and present our proposed method with the attention-based video-level domain adaptation (Section 3.3).

#### 3.1. Baseline Model: MS-TCN

The basic component of our baseline model is single-stage temporal convolutional network (SS-TCN), as shown in the left part of Figure 2. SS-TCN consists of multiple temporal convolutional layers performing acausal dilated 1D convolution. Dilated convolution is used to increase the temporal receptive field exponentially without the need to increase the number of parameters, which can prevent the model from over-fitting the training data. Motivated by the success of multi-stage architectures [42, 27], several SS-TCNs are stacked to form the multi-stage TCN (MS-TCN). Each stage takes the prediction from the previous stage and utilizes the multi-layer temporal convolution feature generator \( G_f \) to obtain the frame-level features \( f = \{ f_1, f_2, \ldots, f_T \} \), and then converts them into the frame-level predictions \( \hat{y} = \{ \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T \} \) by a fully-connected layer \( G_y \).

The overall prediction loss function for each stage is a combination of a classification loss and a smoothing loss, which can be expressed as follows:

\[
\mathcal{L}_y = \mathcal{L}_{cls} + \alpha \mathcal{L}_{T-MSE} \tag{1}
\]

where \( \mathcal{L}_{cls} \) is a cross-entropy loss, \( \mathcal{L}_{T-MSE} \) is a truncated mean squared error used to reduce the difference between adjacent frame-level prediction to improve the smoothness, and \( \alpha \) is the trade-off weight for the smoothness loss. To train the complete model, we minimize the sum of the losses over all stages.

#### 3.2. Local Temporal Domain Adaptation

Despite the progress of MS-TCN on action segmentation, there is still a large room for improvement. The main challenge is the distributional discrepancy caused by spatio-temporal variations across domains. For example, different subjects may perform the same action completely different due to personalized spatio-temporal styles. Therefore, the problem becomes generalizing the model across domains. In this paper, we propose to reduce the domain discrepancy by performing unsupervised DA with auxiliary unlabeled videos.

Encouraged by the success of adversarial-based DA approaches [10, 11], for each stage, we feed the frame-level features \( f \) to an additional shallow binary classifiers, called the local domain classifiers \( G_{ld} \), to discriminate whether the data is from the source or target domain. Before back-propagating the gradients to the main model, a gradient reversal layer (GRL) is inserted between \( G_{ld} \) and the main model to invert the gradient, as shown in Figure 2. During adversarial training, \( G_f \) is learned by maximizing the domain discrimination loss \( \mathcal{L}_{ld} \), while \( G_{ld} \) is learned by minimizing \( \mathcal{L}_y \) with the domain label \( d \). Therefore, the feature generator \( G_f \) will be optimized to gradually align the feature distributions between the two domains. In this paper, we note the adversarial local domain classifier \( G_{ld} \) as the combination of a GRL and a domain classifier \( G_{ld} \), and investigate the integration of \( G_{ld} \) for different stages. From our experiments, the best performance happens when \( G_{ld} \) is
are integrated into middle stages. For more details, please see Section 4.5.

The overall loss function becomes a combination of the baseline prediction loss $L_y$ and the local domain loss $L_{ld}$, which can be expressed as follows:

$$L = \sum_{y} N_y \cdot L_y - \sum_{f} \beta_l L_{ld}$$  \hspace{1cm} (2)$$

$$L_{ld} = \frac{1}{T} \sum_{j=1}^{T} L_{ld}(G_{ld}(f_j), d_j)$$  \hspace{1cm} (3)$$

where $N_y$ is the total stage number, $N_s$ is the number of selected stages, and $T$ is the number of frames from each video. $L_{ld}$ is a binary cross entropy loss function, and $\beta_l$ is the trade-off weight for local domain loss $L_{ld}$. 

### 3.3. Mixed Temporal Domain Adaptation

One main drawback of integrating DA into local frame-level features $f$ is that the video-level feature space is still not fully aligned. Although $f$ is learned using the context and dependencies from neighbor frames, the temporal receptive field still cannot guarantee to cover the whole video length. Furthermore, aligning video-level feature spaces also helps to generate domain-adaptive frame-level predictions for action segmentation. Therefore, we propose to jointly align local frame-level feature spaces and global video-level feature spaces, as shown in Figure 5.

**Global Temporal Domain Adaptation.** To achieve this goal, we first aggregate $f = \{f_1, f_2, ..., f_T\}$ using temporal pooling to form video-level feature $V$. Since each feature $f_k$ captures context in different time by temporal convolution, $V$ still contain temporal information despite the naive temporal pooling method. After obtaining $V$, we add another domain classifier (noted as global domain classifier $G_{gd}$) to explicitly align the embedded feature spaces of video-level features.

Therefore, the global domain loss $L_{gd}$ is added into the overall loss, which can be expressed as follows:

$$L = \sum_{y} N_y \cdot L_y - \sum_{f} (\beta_l L_{ld} + \beta_g L_{gd})$$  \hspace{1cm} (4)$$

$$L_{gd} = L_{gd}(G_{gd}(G_{td}(f)), d)$$  \hspace{1cm} (5)$$

where $L_{gd}$ is also a binary cross entropy loss function, and $\beta_g$ is the trade-off weight for global domain loss $L_{gd}$.

**Domain Attention Mechanism.** Although aligning video-level feature spaces across domains benefits action segmentation, not all the frame-level features are equally important to align. In order to effectively align overall temporal dynamics, we want to focus more on aligning the frame-level features which have larger domain discrepancy. Therefore, we assign larger attention weights to those features, as shown in Figure 3.

Hence, we integrate each stage with the domain attention mechanism, as shown in Figure 4, which utilizes the entropy criterion to generate the domain attention value for each frame-level feature as below:

$$w_j = 1 - H(\hat{d}_j)$$  \hspace{1cm} (6)$$

where $\hat{d}_j$ is the domain prediction from $G_{td}$. $H(p) = -\sum_k p_k \cdot \log(p_k)$ is the entropy function to measure uncertainty. $w_j$ increases when $H(\hat{d}_j)$ decreases, which means the domains can be distinguished well. We also add a residual connection for more stable optimization. Finally, we aggregate the attended frame-level features with temporal pooling to generate the video-level feature $h$, which is noted
as domain attentive temporal pooling (DATP) and can be expressed as:

$$h = \frac{1}{T} \sum_{j=1}^{T} (w_j + 1) \cdot f_j \quad (7)$$

In addition, we add the minimum entropy regularization to refine the classifier adaptation. However, we only want to minimize the entropy for the videos that are similar across domains. Therefore, we attend to the videos which have low domain discrepancy, so that we can focus more on minimizing the entropy for these videos. The attentive entropy loss $\mathcal{L}_{ae}$ can be expressed as follows:

$$\mathcal{L}_{ae} = \frac{1}{T} \sum_{j=1}^{T} (1 + \hat{d}_j) \cdot H(\hat{y}_j) \quad (8)$$

where $\hat{d}$ and $\hat{y}$ is the output of $\hat{G}_{ld}$ and $\hat{G}_{yd}$, respectively. We also adopt the residual connection for stability.

By adding Equation (8) into Equation (4), and replacing $G_{tf}(f)$ with $h$ by Equation (7), the overall loss of our final proposed Mixed Temporal Domain Adaptation (MTDA), as shown in Figure 5, can be expressed as follows:

$$\mathcal{L} = \sum_{i=1}^{N_s} \mathcal{L}_y - \sum_{i=1}^{N_s} (\beta_1 \mathcal{L}_{ld} + \beta_2 \mathcal{L}_{gd} - \mu \mathcal{L}_{ae}) \quad (9)$$

where $\mu$ is the weight for the attentive entropy loss.

4. Experiments

To evaluate how our approaches can diminish spatial-temporal discrepancy for action segmentation, we choose three challenging datasets: Georgia Tech Egocentric Activities (GTEA) [8], 50Salads [35], and the Breakfast dataset [14], which separate the training and testing sets by different people (noted as subjects), resulting in high spatio-temporal variations. By following the transductive setting for DA, “Source” refers to the original training set, and “Target” refers to the testing set without labels. With these three datasets, we show how our approaches adapt the same actions across different people by decreasing the spatio-temporal variations across videos.
4.1. Datasets

The GTEA dataset contains 28 videos including 7 kitchen activities performed by 4 subjects. All the videos were recorded by a camera that is mounted on the actors head. There are totally 11 action classes including background. On average, each video is around one minute long with 20 action instances. We use 4-fold cross-validation for evaluation by leaving one subject out.

The 50Salads dataset contains 50 videos for salad preparation activities performed by 25 subjects. There are totally 17 action classes. On average, each video contains 20 action instances and is 6.4 minutes long. For evaluation, we apply 5-fold cross-validation by leaving five subjects out.

The Breakfast dataset is the largest among the three datasets with 1712 videos for breakfast preparation activities performed by 52 subjects. The videos were recorded in 18 different kitchens with 48 action classes where each video contains 6 action instances on average and is around 2.7 minutes long. For evaluation, we utilize the standard 4-fold cross-validation by leaving 13 subjects out.

4.2. Evaluation Metrics

For all the three datasets, we use the following evaluation metrics as in [17]: frame-wise accuracy (Acc), segmental edit score, and segmental F1 score at the IoU threshold k\%, denoted as F1@k (k = \{10, 25, 50\}).

While frame-wise accuracy is one of the most common evaluation metrics for action segmentation, it does not take into account the temporal dependencies of the prediction, causing large qualitative differences with similar frame-wise accuracy. In addition, long action classes have higher impact on this metric than shorter action classes, making this metric not able to reflect over-segmentation errors. To address the above limitations, the segmental edit score penalizes over-segmentation by measuring the ordering of predicted action segments independent of slight temporal shifts. Finally, another suitable metric segmental F1 score (F1@k) becomes popular recently since it is found that the score numbers better indicate the qualitative segmentation results. F1@k also penalizes over-segmentation errors while ignoring minor temporal shifts between the predictions and ground truth. F1@k is determined by the total number of actions but not depends on the duration of each action instance, which is similar to mean average precision (mAP) with intersection-over-union (IoU) overlap criteria.

4.3. Implementation

Our implementation is based on the PyTorch [30] framework. We extract I3D [1] features for the video frames and use these features as input to our model. The video frame rates are the same as [7]. For fair comparison, we adopt the same architecture design choices of MS-TCN [7] as our baseline model. The whole model consists of four stages where each stage contains ten dilated convolution layers. We set the number of filters to 64 in all the layers of the model and the filter size is 3. For optimization, we use the Adam optimizer and the batch size equals to 1. Since the target data size is smaller than the source data, each target data is loaded randomly multiple times in each epoch during training. For the weighting of loss functions, we follow the common strategy as [10, 11] to gradually increase \(\beta_g\) from 0 to 1. The weighting \(\alpha\) for smoothness loss is 0.15 as in [7] and \(\mu\) is chosen as \(1 \times 10^{-4}\) via the grid-search.

<table>
<thead>
<tr>
<th></th>
<th>F1@{10, 25, 50}</th>
<th>Edit</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GTEA</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source only (MS-TCN)</td>
<td>85.8 83.4 69.8</td>
<td>79.0</td>
<td>76.3</td>
</tr>
<tr>
<td>DA (L)</td>
<td>89.6 87.9 74.4</td>
<td>84.5</td>
<td>80.1</td>
</tr>
<tr>
<td>DA (L + G)</td>
<td>90.0 88.6 74.9</td>
<td>85.6</td>
<td>79.6</td>
</tr>
<tr>
<td>DA (L + G + A)</td>
<td>90.5 88.4 76.2</td>
<td>85.8</td>
<td>80.0</td>
</tr>
<tr>
<td><strong>50Salads</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source only (MS-TCN)</td>
<td>76.3 74.0 64.5</td>
<td>67.9</td>
<td>80.7</td>
</tr>
<tr>
<td>DA (L)</td>
<td>79.2 77.8 70.3</td>
<td>72.0</td>
<td>82.8</td>
</tr>
<tr>
<td>DA (L + G)</td>
<td>80.2 78.4 70.6</td>
<td>73.4</td>
<td>82.2</td>
</tr>
<tr>
<td>DA (L + G + A)</td>
<td>82.0 80.1 72.5</td>
<td>75.2</td>
<td>83.2</td>
</tr>
<tr>
<td><strong>Breakfast</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source only (MS-TCN)</td>
<td>52.6 48.1 37.9</td>
<td>61.7</td>
<td>66.3</td>
</tr>
<tr>
<td>DA (L)</td>
<td>72.8 67.8 55.1</td>
<td>71.7</td>
<td>70.3</td>
</tr>
<tr>
<td>DA (L + G)</td>
<td>72.6 66.9 54.3</td>
<td>72.6</td>
<td>69.2</td>
</tr>
<tr>
<td>DA (L + G + A)</td>
<td>74.2 68.6 56.5</td>
<td>73.6</td>
<td>71.0</td>
</tr>
</tbody>
</table>

Table 1: The experimental results for all our approaches on GTEA, 50Salads, and the Breakfast dataset (L: local temporal DA, G: global temporal DA without domain attention, A: domain attention mechanism).

4.4. Experimental Results

We first compare with the baseline model MS-TCN [7] to see how our approaches effectively utilize the unlabeled videos for action segmentation. “Source only” means the model is trained only with source labeled videos. And then we compare the proposed approach to the state-of-the-art methods on all three datasets.

**Local Temporal Domain Adaptation.** By integrating domain classifiers with frame-level features \(f\), the results on all three datasets with respect to all the metrics are improved significantly, as shown in the row “DA (L)” in Table 1. For example, on the GTEA dataset, our approach outperforms the baseline by 4.6% for F1@50, 5.5% for the edit score and 3.8% for the frame-wise accuracy. Although “DA (L)” mainly works on the frame-level features, they are learned using the context from neighbor frames, so they still contain temporal information, which is critical to diminish the temporal variations for actions across domains.

**Mixed Temporal Domain Adaptation.** Despite the improvement from local temporal DA, the temporal recep-
tive fields of frame-level features are still not guaranteed to cover the whole video length. Therefore, we aggregate frame-level features to generate a video-level feature for each video and apply additional domain classifier on it. However, aggregating frames by temporal pooling without considering the importance of each frame does not ensure better performance, especially for the Breakfast dataset, which contains much higher domain discrepancy than the other two. The F1 score and frame-wise accuracy both have slightly worse results, as shown in the row “DA (L + G)” in Table 1. Therefore, we apply the domain attention mechanism to aggregate frames more effectively, leading to better global temporal DA performance. For example, on the Breakfast dataset, “DA (L + G + A)” outperforms “DA (L)” by 1.4% for F1@50, 1.9% for the edit score and 0.7% for the frame-wise accuracy, as shown in Table 1.

Our final method “DA (L + G + A)”, which is also MTDA, outperforms the baseline by large margins (e.g. 6.4% for F1@50, 6.8% for the edit score and 3.7% for the frame-wise accuracy on GTEA; 8.0% for F1@50, 7.3% for the edit score and 2.5% for the frame-wise accuracy on 50Salads), as demonstrated in Table 1.

Comparison with the State-of-the-Art. Here we compare the proposed MTDA to the state-of-the-art methods, and MTDA outperforms all the previous methods on the three datasets with respect to three evaluation metrics: F1 score, edit distance, and frame-wise accuracy, as shown in Table 2.

For the GTEA dataset, the authors of MS-TCN [7] also fine-tune the I3D features to improve the performance (e.g. from 85.8% to 87.5% for F1@10). Our MTDA outperforms the fine-tuned MS-TCN even without any fine-tuning process since we learn the temporal features more effectively from unlabeled videos, which is more important for action segmentation.

cGAN [9] utilizes supplementary modalities including depth maps and optical flow with an auxiliary network. cGAN outperforms MS-TCN in terms of the F1 score and edit score on the 50Salads dataset. Our MTDA outperforms cGAN, indicating more effective way to utilizes auxiliary data.

For the Breakfast dataset, the authors of MS-TCN [7] also use the improved dense trajectories (IDT) features, which encode only motion information and outperform the I3D features since the encoded spatial information is not the critical factor for the Breakfast dataset. Our MTDA outperforms the IDT-version of MS-TCN by a large margin with the same I3D features. This shows that our DATP module effectively aggregate frames by considering the temporal structure for action segmentation.

4.5. Ablation Study and Analysis

Integration of G_{ld} and Stages. Since we use Multi-stage TCN [7] as our baseline model and develop our approaches upon it, it raises a question: How to effectively perform DA by integrating the domain classifiers to a multi-stage architecture? Our architecture contains four stages as in [7]. First, we integrate G_{ld} into one stage and the results are demonstrated in Table 3. The results show that the best performance happens when G_{ld} is integrated into middle stages, such as S2 or S3. S1 is not a good choice for DA

<table>
<thead>
<tr>
<th>GTEA</th>
<th>F1@{10, 25, 50}</th>
<th>Edit</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-CNN [18]</td>
<td>58.7 54.4 41.9</td>
<td>49.1</td>
<td>60.6</td>
</tr>
<tr>
<td>Bi-LSTM [34]</td>
<td>66.5 59.0 43.6</td>
<td>-</td>
<td>55.5</td>
</tr>
<tr>
<td>ED-TCN [17]</td>
<td>72.2 69.3 56.0</td>
<td>-</td>
<td>64.0</td>
</tr>
<tr>
<td>TricorNet [4]</td>
<td>76.0 71.1 59.2</td>
<td>-</td>
<td>64.8</td>
</tr>
<tr>
<td>TDRN [19]</td>
<td>79.2 74.4 62.7</td>
<td>74.1</td>
<td>70.1</td>
</tr>
<tr>
<td>cGAN [9]</td>
<td>80.1 77.9 69.1</td>
<td>78.1</td>
<td>78.5</td>
</tr>
<tr>
<td>MS-TCN [7]</td>
<td>85.8 83.4 69.8</td>
<td>79.0</td>
<td>76.3</td>
</tr>
<tr>
<td>MS-TCN (FT) [7]</td>
<td>87.5 85.4 74.6</td>
<td>81.4</td>
<td>79.2</td>
</tr>
<tr>
<td><strong>MTDA</strong></td>
<td><strong>90.5</strong> <strong>88.4</strong> <strong>76.2</strong></td>
<td><strong>85.8</strong></td>
<td><strong>80.0</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>50Salads</th>
<th>F1@{10, 25, 50}</th>
<th>Edit</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDT+LM [31]</td>
<td>44.4 38.9 27.8</td>
<td>45.8</td>
<td>48.7</td>
</tr>
<tr>
<td>Bi-LSTM [34]</td>
<td>62.6 58.3 47.0</td>
<td>55.6</td>
<td>55.7</td>
</tr>
<tr>
<td>ST-CNN [18]</td>
<td>55.9 49.6 37.1</td>
<td>45.9</td>
<td>59.4</td>
</tr>
<tr>
<td>ED-TCN [17]</td>
<td>68.0 63.9 52.6</td>
<td>59.8</td>
<td>64.7</td>
</tr>
<tr>
<td>TricorNet [4]</td>
<td>70.1 67.2 56.6</td>
<td>62.8</td>
<td>67.5</td>
</tr>
<tr>
<td>TDRN [19]</td>
<td>72.9 68.5 57.2</td>
<td>66.0</td>
<td>68.1</td>
</tr>
<tr>
<td>MS-TCN [7]</td>
<td>76.3 74.0 64.5</td>
<td>67.9</td>
<td>80.7</td>
</tr>
<tr>
<td>cGAN [9]</td>
<td>80.1 78.7 71.1</td>
<td>76.9</td>
<td>74.5</td>
</tr>
<tr>
<td><strong>MTDA</strong></td>
<td><strong>82.0</strong> <strong>80.1</strong> <strong>72.5</strong></td>
<td><strong>75.2</strong></td>
<td><strong>83.2</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Breakfast</th>
<th>F1@{10, 25, 50}</th>
<th>Edit</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>ED-TCN [17]</td>
<td>- - -</td>
<td>-</td>
<td>43.3</td>
</tr>
<tr>
<td>HTK [16]</td>
<td>- - -</td>
<td>-</td>
<td>50.7</td>
</tr>
<tr>
<td>TCFPN [5]</td>
<td>- - -</td>
<td>-</td>
<td>52.0</td>
</tr>
<tr>
<td>HTK (64) [15]</td>
<td>- - -</td>
<td>-</td>
<td>56.3</td>
</tr>
<tr>
<td>GRU [32]</td>
<td>- - -</td>
<td>-</td>
<td>60.6</td>
</tr>
<tr>
<td>MS-TCN [7]</td>
<td>52.6 48.1 37.9</td>
<td>61.7</td>
<td>66.3</td>
</tr>
<tr>
<td>MS-TCN (IDT) [7]</td>
<td>58.2 52.9 40.8</td>
<td>61.4</td>
<td>65.1</td>
</tr>
<tr>
<td><strong>MTDA</strong></td>
<td><strong>74.2</strong> <strong>68.6</strong> <strong>56.5</strong></td>
<td><strong>73.6</strong></td>
<td><strong>71.0</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source only</th>
<th>F1@{10, 25, 50}</th>
<th>Edit</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only</td>
<td>58.5 83.4 69.8</td>
<td>79.0</td>
<td>76.3</td>
</tr>
<tr>
<td>{S1}</td>
<td>88.6 86.2 73.6</td>
<td>84.2</td>
<td>78.7</td>
</tr>
<tr>
<td>{S2}</td>
<td>89.1 87.2 \textbf{74.4}</td>
<td>84.3</td>
<td>79.1</td>
</tr>
<tr>
<td>{S3}</td>
<td>89.2 87.3 72.3</td>
<td>83.8</td>
<td>78.9</td>
</tr>
<tr>
<td>{S4}</td>
<td>88.1 86.4 73.0</td>
<td>83.0</td>
<td>78.8</td>
</tr>
<tr>
<td>{S1, S2}</td>
<td>89.0 85.8 73.5</td>
<td>\textbf{84.8}</td>
<td>79.5</td>
</tr>
<tr>
<td>{S2, S3}</td>
<td>\textbf{89.6} \textbf{87.9} \textbf{74.4}</td>
<td>84.5</td>
<td>\textbf{80.1}</td>
</tr>
<tr>
<td>{S3, S4}</td>
<td>88.3 86.8 73.9</td>
<td>83.6</td>
<td>78.6</td>
</tr>
</tbody>
</table>

Table 3: The experimental results for the integration of G_{ld} and different stages of MS-TCN [7] on the GTEA dataset. \{S_n\} means adding G_{ld} to the nth stage. The stages with smaller n are closer to inputs.
because of two reasons: 1) \(S_1\) corresponds to low-level and transferable features with less discriminability where DA shows limited effects [22]. 2) \(S_1\) capture less temporal information from neighbor frames, which is critical for action segmentation. However, higher stages (e.g. \(S_4\)) are not always better. We conjecture that it is because higher stages are used to refine the prediction. They may affect the semantic structure of feature representations, which is important to DA. In our case, integrating \(G_{10}\) into \(S_2\) provides the best overall performance.

We also add multiple domain classifiers to adjacent stages. However, multi-stage DA does not always guarantee improved performance. For example, \(\{S_1, S_2\}\) has worse results than \(\{S_2\}\) in terms of F1@\{10, 25, 50\}. Since \(\{S_2\}\) and \(\{S_3\}\) provide the best single-stage DA performance, we use \(\{S_2, S_3\}\), which performs the best, as the final model for all our approaches in all the experiments.

**Qualitative results.** In addition to evaluating the quantitative performance using the above metrics, it is also common to evaluate the qualitative performance to ensure that the prediction results are aligned with human vision. Here we compare our approaches with the baseline model MS-TCN [7] and the ground truth, as shown in Figure 6. MS-TCN fails to predict open before the long pour action in the middle part of the video, and falsely predict pour before stir in the end of the video, as shown in the “Source only” row. With local and global temporal DA, our approach can detect all the actions happened in the video, as shown in the row “DA (L + G)”. Finally, with the domain attention mechanism, our proposed MTDA also removes the falsely predicted action pour. For more qualitative results, please refer to the supplementary material.

5. Conclusion and Future Work

In this paper, we consider action segmentation as a DA problem and reduce the domain discrepancy by performing unsupervised DA with auxiliary unlabeled videos. To diminish domain discrepancy for both the spatial and temporal directions, we propose Mixed Temporal Domain Adaptation (MTDA) to jointly align frame- and video-level embedded feature spaces across domains, and further integrate with the domain attention mechanism to focus on aligning the frame-level features with higher domain discrepancy, leading to more effective domain adaptation. The comprehensive experiment results validate that our approach outperforms all the previous state-of-the-art methods. Our approach can adapt models effectively by using auxiliary unlabeled videos, leading to further possible applications to large-scale problems, such as video surveillance and human activity analysis. For the future work, we would like to develop DA approaches with self-supervised learning to reduce the need of additional unlabeled videos.
References


