# LSTA$^3$N - Domain Adaptation for Egocentric Action Recognition

Lorenzo Bergadano
*Politecnico di Torino*
s304415@studenti.polito.it

Matteo Matteotti
*Politecnico di Torino*
s294552@studenti.polito.it

Paolo Rizzo
*Politecnico di Torino*
paolo.rizzo@studenti.polito.it

***Abstract*—In this report, two well-known solutions have been simultaneously implemented for the first time in order to mitigate two of the biggest challenges which come with egocentric videos when performing action recognition. TA$^3$N, which has achieved the state of the art for the domain shift problem in the temporal dimension, has been enriched with LSTA, a recurrent unit which focuses on features from relevant spatial regions while tracking the attention maps smoothly across time.**

**The code is available at: https://github.com/polrizzo/LSTA-3N**

## I. Introduction

Compared to their third-person counterpart, egocentric videos bring new challenges to the task of action recognition, mainly related to the limited variety of scenarios provided by the available datasets and to the unconstrained nature of the video shot. As a result, the definition of a model able to focus on regions of interest which is also robust to environment bias is particularly challenging. To simultaneously address both challenges, one needs to come up with a model that is able to perform:

- *Domain Adaptation* (**DA**): to train the model on a labeled source domain and apply it to an unlabeled target domain;
- *Attention*: to prune the network search and to avoid computing features from irrelevant regions.

To this end, we have implemented two well-known solutions together for the first time: Temporal Attentive Adversarial Adaptation Network (TA$^3$N) [1], which has achieved state-of-the-art performance when addressing the domain shift problem in the temporal dimension, and Long Short-Term Attention (LSTA) [2], an RNN model which tracks discriminative areas and locates active objects.

## II. Related Work

### A. Action Recognition

Two opposite successful schools of thought have been proposed to deal with any action recognition task. 3D CNN based methods usually achieve better performance than their bidimensional counterpart but at the cost of an increased model's complexity. In I3D [3], a 3D convolution for spatial and motion streams is deployed, followed by a late fusion of modalities. 2D CNNs, like TSN [5] or TBN [6] are computationally cheap but are not able to model any temporal relationships since they process one frame at the time.

Some methods have been proposed to achieve 3D-CNN performance at 2D complexity. TSM [4] shifts the channels along the temporal dimension, facilitating information exchange between neighbouring frames. Temporal Relation Network (TRN) [7] enables temporal relational reasoning in neural networks whilst learning and describing temporal relations at multiple time scales.

In general, however, the performance of action recognition from videos is still not comparable to the advances made in object recognition from still images. One of the biggest limitations is that the strategies developed for image classification would require fine-grained frame-level annotations [11], [12], which are practically unfeasible.

The traditional challenges are further enhanced when dealing with egocentric videos, due to a strongly limited variety of scenarios provided by the available datasets and the strong *ego-motion*, typically caused by the sharp movements of the camera's wearer.

### B. Attention

Attention was proposed to help the networks identifying relevant features coming from regions of interest. Suthakaran *et al.* proposed ego-rnn [13], the first network endowed with attention to perform egocentric activity recognition. Though the method achieved state-of-the-art performances, it generates the attention maps independently at each frame, with the risk of focusing on completely different regions in two consecutive frames. Long Short-Term Attention (LSTA) [2] addressed this shortcoming, adding a temporal attention mechanism which enables a smooth track of the attention maps across time, generating them sequentially rather than frame by frame.

### C. Domain Adaptation

Domain adaptation (DA) generalises the learner across different domains by mitigating the domain shift problem. Existing works explore both Supervised [8], [9] and Unsupervised Domain Adaptation (UDA). In [1] the authors have applied Deep UDA to videos through the so-called Temporal Attentive Adversarial Adaptation Network (TA$^3$N), which bridges the gap between source and target domain by adversarially learning a domain-invariant representation of them.

Unlike other methods which mainly focused on domain discrepancy along the spatial dimension, TA$^3$N addressed the domain shift problem in the temporal direction, aligning those

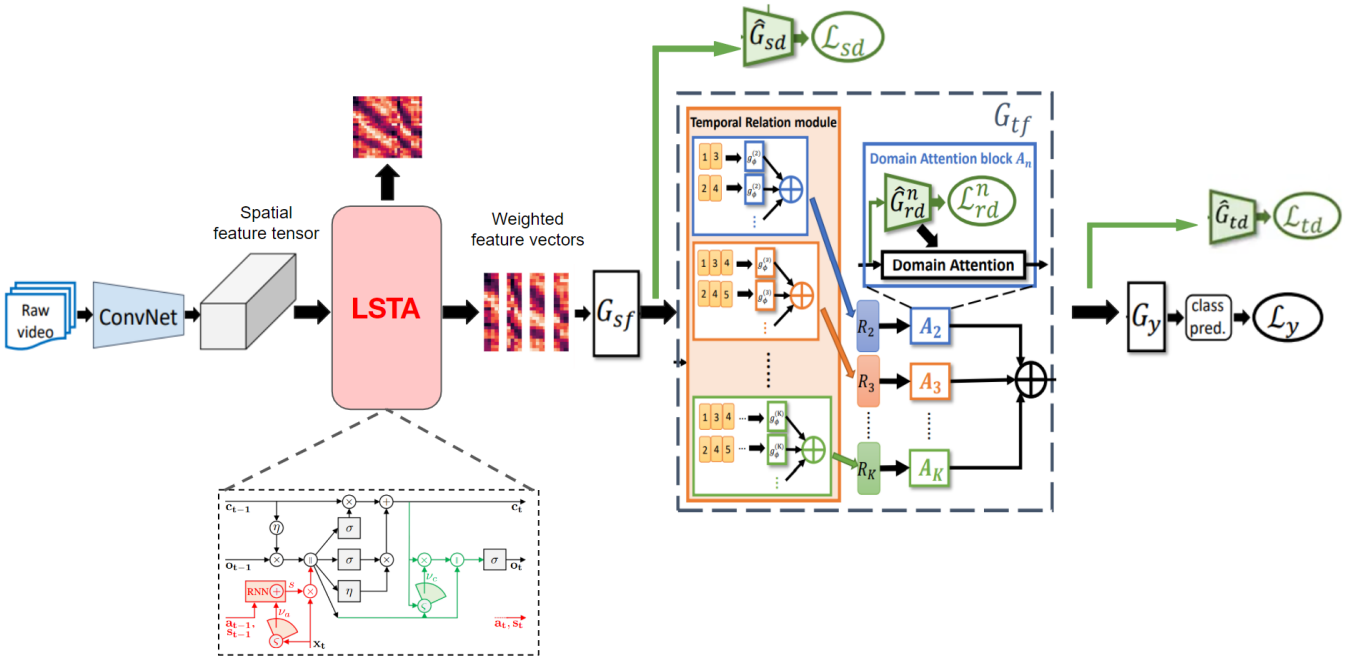


Fig. 1: Our proposed method, LSTA$^3$N. The architecture requires spatial feature tensors ($B \times C \times N \times W \times H$) which are fed through LSTA [2]. This module keeps an internal state for attention, so that the attention maps are smoothly tracked across time. LSTA returns attentively-weighted feature vectors, which are then sent to TA$^3$N [1]. Putting together LSTA with TA$^3$N implies that attention is carried out on both the spatial and the temporal dimension.

temporal features which have a high contribution to the overall domain shift.

## III. The Proposed Method

As said before, TA$^3$N [1] already implements an attention mechanism, which focuses on aligning the temporal features which show the largest domain discrepancy. This amounts to the optimisation of an attentive entropy loss function, which contributes to the overall TA$^3$N loss function.

As highlighted in Section IV, the inclusion of the domain attention mechanisms significantly improves the overall accuracy. Yet the performance is still quite low. In our view, in the context of egocentric videos, it becomes crucial to develop a model able to discriminate between which elements to give attention to and which to ignore.

To this end, we have introduced **LSTA$^3$N**, which pairs the attentive adaption strategy proposed by TA$^3$N with an LSTA module put at its very beginning. For each frame, a spatial attention map is generated and used to weigh the extracted feature vectors. These are then fed through the TA$^3$N architecture. In this way, attention is computed on two different levels:

- spatially, driving the network's focus towards the most active regions;
- temporally, adjusting the adaptation based on each specific temporal feature's domain discrepancy.

The loss of LSTA$^3$N is naturally retrieved by summing the cross-entropy loss of LSTA with the overall loss of TA$^3$N, defined by the authors as:

$$\mathcal{L} = \frac{1}{N_S}\sum_{i=1}^{N_S}\mathcal{L}_y^i + \frac{1}{N_{S\cup T}}\sum_{i=1}^{N_{S\cup T}}\gamma\mathcal{L}_{ae}^i - \frac{1}{N_{S\cup T}}\sum_{i=1}^{N_{S\cup T}}(\lambda^s\mathcal{L}_{sd}^i + \lambda^r\mathcal{L}_{rd}^i + \lambda^t\mathcal{L}_{td}^i)$$

where $\lambda^s$, $\lambda^r$ and $\lambda^t$ are the trade-off weights for each domain loss. $\gamma$ is the weighting for the attentive entropy loss.

Another source of novelty of our proposition is that well-known methods have been used in a new guise: LSTA has never been used for domain adaptation and TA$^3$N has been tested by its authors on coarse-grained actions only, as brought up by Munro and Damen [10]. As part of this work, we have used TA$^3$N in order to classify fine-grained egocentric actions.

## IV. TA$^3$N - Experiments and Results

### A. Implementation details

The project has been developed using the split of EPIC-Kitchens for DA proposed in MM-SADA [7]. To form the domains, the three largest kitchens have been selected. These are P08, P01 and P22, referred to as D1, D2 and D3 respectively. Some statistics for each domain are shown in Table I.

| | Number of action segments | | |
|---|---|---|---|
| | *D1* | *D2* | *D3* |
| **Training set** | 1543 | 2495 | 3897 |
| **Test set** | 435 | 750 | 974 |

TABLE I: Number of per-domain action segments, divided between training and test set.

Following the indications of the paper, the performances have been analysed on the 8 largest action classes: 'put', 'take', 'open', 'close', 'wash', 'cut', 'mix', and 'pour', which form 80% of the training action segments for these domains. The per-domain class distribution is depicted in Figure 2.

In this work, we have focused on appearance and motion only, using the publicly available frames. Though other modalities have been proved to improve the overall accuracy [14], RGB and Flow frames are widely considered the two most common modalities in the cross-domain action recognition task. As shown by [10], the motion modality is more domain-invariant than RGB, resulting in a better interpretation of the background-irrelevant information. On the other hand, appearance information coming from RGB frames can identify semantically meaningful information under different camera setups.

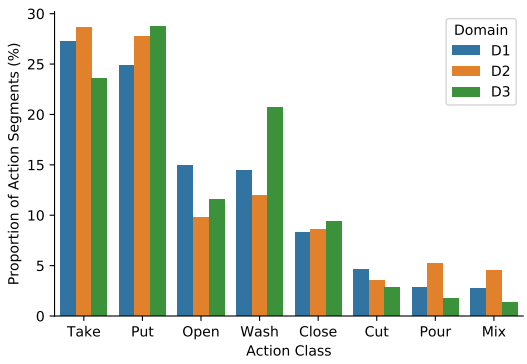


Fig. 2: Per-domain class distribution for the selected classes

### *B. Familiarisation with the architectures and the sampling strategy*

For the first step, action recognition was performed using pre-trained model weights for both RGB and optical flow frames. The compared architectures are TSM [4] and I3D [3].

TSM consists of a ResNet-50 pretrained on ImageNet for image recognition, which models the temporal dimension by shifting the features across time. After splitting each video sequence into windows of 5 clips each, the network has been fed with 5 frames *uniformly sampled* from each clip. The extracted features have 2048 dimensions.

I3D consists of a BNInception pre-trained on Kinetics. A *dense* sampling strategy has been deployed. Similarly to I3D, the video has been split into windows of 5 clips. Within each clip, the network convolutes over a window of 16 consecutive frames. A final average pooling layer returns the final output of the network, which has 1024 dimensions.

The results are summarised in II. The two most important aspects are: the systematic outperformance of TSM relative to I3D; the remarkable results achieved using flow frames, showing even better performance than RGB when TSM is applied to *D1*. Though the former may not be a surprise (as shown already by the authors of TSM themselves), the latter may be. The explanation is instead apparent. Though appearance information is crucial when performing every recognition task, flow frames better capture the actions' progression in time and substantially contribute to the prediction of the verb classes.

A two-stream late fusion was then performed, and the overall increase in the accuracy can be appreciated in Table III. This reveals that the contribution of either modality is indeed complementary.

| | | Accuracy (%) | | | | | |
|---|---|---|---|---|---|---|---|
| **Network** | **Sampling** | **RGB** | | | **Flow** | | |
| | | *D1* | *D2* | *D3* | *D1* | *D2* | *D3* |
| **I3D** | **5 clips 16 frames** | 45.29 | 55.70 | 57.91 | 44.14 | 45.67 | 44.15 |
| **TSM** | **5 clips 5 frames** | 54.02 | 65.38 | 67.45 | 58.39 | 59.56 | 63.86 |

TABLE II: Impact of each modality on either architecture.

| | | Accuracy (%) | | |
|---|---|---|---|---|
| **Network** | **Sampling** | **RGB + Flow** | | |
| | | *D1* | *D2* | *D3* |
| **I3D** | **5 frames 16 clips** | 48.74 | 56.25 | 56.88 |
| **TSM** | **5 frames 5 clips** | 58.39 | 73.55 | 64.37 |

TABLE III: Impact of fusing modalities on either architecture.

### *C. Temporal aggregation*

To improve the perfomance, two different temporal aggregation strategies have been explored.

- *Average Pooling (AvgPool)*: simple pooling strategy based on average on the temporal dimension.
- *Temporal Relational Network (TRN)* [7]: temporal aggregation method which enables temporal relational reasoning in neural networks for videos

To guarantee consistency in the results, the same hyperparameters were applied. Particularly important in this phase is the choice of the learning rate. The best tested configuration is found by setting the initial learning rate to 0.01. The results are summarised in Table IV.

Not surprisingly, TRN applied to TSM constantly outperforms AvgPool applied to I3D, even though the number of considered frames is less than a third. This is because TRN learns the casual relations between sparse frames, which turns out to be more efficient than sampling dense frames, convolving them and later applying a simple average operation. Amongst the tested configurations, the deployment of the temporal relational module within the TSM architecture leads

to the best results, showing once more how crucial the correct modeling of the temporal dimension is.

This also explains the sharp increase in the overall accuracy when considering RGB frames, as causation is better encoded through appearance information rather than motion information.

| | | Accuracy (%) | | | | | |
|---|---|---|---|---|---|---|---|
| **Net** | **Strategy** | **RGB** | | | **Flow** | | |
| | | *D1* | *D2* | *D3* | *D1* | *D2* | *D3* |
| **I3D** | **AvgPool** | 52.18 | 61.07 | 63.86 | 53.33 | 64.13 | 62.16 |
| **TSM** | **TRN** | 58.62 | 72.93 | 72.29 | 58.62 | 69.60 | 65.91 |

TABLE IV: Performance after having applied temporal aggregation.

### *D. Domain Adaptation*

Though different Domain Adaptation techniques have been deployed in the context of Egocentric Action Recognition, TA$^3$N is one of the few which has attempted deep Unsupervised Domain Adaption (UDA), improving the domain alignment by attending to the temporal dynamics of videos. For this work, TA$^3$N has been implemented on all the possible combinations of domain shifts, with the following workflow:

- we first applied the model trained on the source domain directly to the target domain, without any modification (*Source Only* row in Table VIII).
- we then gradually added the *Adversarial Discriminators* to the spatial, temporal and relational features. This leads to three different types of discriminators $\hat{G}_{sd}$, $\hat{G}_{td}$ and $\hat{G}_{rd}$ respectively.
- we then considered them all together. In so doing, the Temporal Adversarial Adaptation Network (TA$^2$N) was obtained.
- TA$^2$N was eventually extended with domain attention. In so doing, the Temporal Attentive Adversarial Adaptation Network (TA$^3$N) was obtained.

In all these steps, both *AvgPool* and *TRN* have been utilised as frame aggregators. The results are summarised in Table VIII. For each domain shift, the *Target Only* row are the results achieved without any domain adaptation, as per Table IV. It can be regarded as the upper bound which every Domain Adaptation technique should tend to. Analogously, the *Source Only* row shall be regarded as the theoretical lower bound.

To make the results more robust, we have considered the best accuracy starting from $30\%$ of the total epochs.

In our opinion, the only way to assess the impact brought by each adversarial discriminator and by the domain attention is by keeping all the other hyper-parameters fixed. Not only should this hold across one single domain shift, but across every domain shift. Our choice to favour a fairer comparison comes at the cost of poorer performance (the largest single accuracy was below $40\%$), as the hyper-parameter tuning is performed just once to optimise the global performance rather than being shift-specific and maximising each single accuracy.

The grid search space is reported in Table V and the best combination is shown in bold face. Not surprisingly, the game-changing parameters are $\gamma$, *i.e.*, the weight for the attentive entropy loss, $\beta_s$, *i.e.* the trade-off weight for the spatial domain loss, $\beta_t$, *i.e.*, the trade-off weight for the temporal domain loss, and $\beta_r$, *i.e.*, the one for the relational domain loss.

| **Hyper-parameters** | **Values** |
|---|---|
| Initial learning rate | **0.0003**, 0.001, 0.003, 0.1 |
| Learning rate (lr) decay | **DANN** [16], (lr $\times$ 0.1) every 10 epoch |
| $\gamma$ | **0.0003**, 0.03, 0.3 |
| $\beta_s$ | 0.25, **0.5**, 0.75, 1 |
| $\beta_t$ | 0.25, 0.5, **0.75**, 1 |
| $\beta_g$ | 0.25, 0.5, **0.75**, 1 |
| Optimiser | **SGD**, Adam |

TABLE V: Grid search space.

A key role is played by the learning rate configurations. Unlike Table IV, a small initial learning rate ($3 \times 10^{-4}$) and the learning-rate adaptive strategy showed in DANN [16] were needed. Other tested choices usually ended up converging to poor results ($\approx 28\%$ of accuracy) quite fast. To balance the small learning rate out, the number of training epochs was set to 100. For other hyperparameters, such as momentum and weight decay of the SGD, some small deviations from the values suggested by the authors of TA$^3$N have been analysed, but they lead to little-to-no change.

Table VI averages all the measures of accuracy of Table VIII, putting across the average impact of all TA$^3$N components. The results yielded by our best-tested working configuration are largely in keeping with what expected.

Integrating the adversarial discriminators generally improves the results. For both AvgPool and TRN, the relational discriminator $\hat{G}_{rd}$ and the temporal discriminator $\hat{G}_{td}$ single-handedly outperforms the inclusion of the mere spatial discriminator $\hat{G}_{sd}$. AvgPool benefit more from the alignment and the learning of the temporal dynamics brought by $\hat{G}_{rd}$, whereas $\hat{G}_{td}$ shows better performance when implemented in the TRN architecture. This comes as no surprise, as the Temporal Relational Module encodes the temporal dynamics in a way that a simple average cannot capture.

The core of TA$^3$N goes beyond the aggregation of its component, and assesses how they interact with each other in the bigger scheme of things. Indeed, by combining all three discriminators, TA$^2$N improves the performance for both architectures ($+0.95\%$ for TRN, $+0.94\%$ for AvgPool). But the substantial gain occurs when TA$^3$N is implemented, *i.e.*, when domain attention is applied ($+1.19\%$ and $+0.95\%$, respectively).

Though figures suggest AvgPool attained better performance than TRN, the latter always needed fewer iterations before reaching convergence. This could be seen in Table VII, which summarises the partial results of Tables VIII up to the 50$^{\text{th}}$ epoch. It is apparent how TRN reached close-to-convergence results after just half of the epochs.

This also creates the possibility that TRN could have benefited from an architecture-specific learning rate tuning.

However, in view of our decision of choosing consistency and reproducibility over performance, this solution's impact would have propagated through the whole table, ultimately worsening other performances.

### *E. Motivation for our improvement*

When applying domain adaptation, we have come across a couple of drawbacks:

- a huge sensitivity of the performance to the hyper-parameters choice and the need of a really small learning rate;
- overall poor performance, as the best accuracy is just below $40\%$.

In TA$^3$N, attention is performed when aligning the temporal dynamics, attending to those clips which are more similar across domains. However, TA$^3$N does not come with a built-in spatial attention module. Consequently, the model's attention can be easily driven away, especially when dealing with the unconstrained nature of the EPIC-KITCHENS. First-person non-scripted videos recorded in real kitchens lead to articulated scenes, and a simple model can easily be misled by insignificant elements. Thus, the classification becomes particularly involved.

We believe that the overall performance can substantially benefit from avoiding computing irrelevant features and from focusing on the most active image regions.

| **Temporal Module** | **AvgPool** | | **TRN** | |
|---|---|---|---|---|
| | **Accuracy (%)** | **Gain** | **Accuracy (%)** | **Gain** |
| Source Only | 34.26 | - | 34.31 | - |
| $\hat{G}_{sd}$ | 34.38 | **+0.12** | 34.40 | **+0.09** |
| $\hat{G}_{td}$ | 34.38 | **+0.12** | 34.48 | **+0.17** |
| $\hat{G}_{rd}$ | 34.44 | **+0.18** | 34.44 | **+0.12** |
| All $\hat{G}_d$ (TA$^2$N) | 35.22 | **+0.96** | 35.25 | **+0.94** |
| All $\hat{G}_d$ + domain attention (TA$^3$N) | 35.45 | **+1.19** | 35.26 | **+0.95** |

TABLE VI: Comparison between the average contribution of each component of TA$^3$N.

| **Temporal Module** | **AvgPool** | **TRN** |
|---|---|---|
| **Results after 50 epochs** | **Accuracy (%)** | **Accuracy (%)** |
| Source Only | 32.48 | 33.82 |
| $\hat{G}_{sd}$ | 32.57 | 33.94 |
| $\hat{G}_{td}$ | 32.52 | 34.67 |
| $\hat{G}_{rd}$ | 32.63 | 33.74 |
| All $\hat{G}_d$ (TA$^2$N) | 33.27 | 34.75 |
| All $\hat{G}_{sd}$ + domain attention (TA$^3$N) | 33.87 | 34.59 |

TABLE VII: Comparison between the (partial) average contribution of each component of TA$^3$N after the 50$^{\text{th}}$ epoch. At this point, *TRN* systematically outperforms *AvgPool*.

## V. LSTA$^3$N

In light of the above, we have paired the attentive attention with a proper attention strategy aimed at focusing on the most active regions of each frame. In our vision, coupling the two attention layers should increase the overall performance.

In this variation of TA$^3$N, an LSTA module is placed right before the actual TA$^3$N architecture. It inputs the spatial feature tensors coming from the last convolutional layer block of a TSM and outputs the attentively-weighted feature vectors. These are then fed into TA$^3$N after being passed through a temporal relational module for aggregation.

Not only is this the first time LSTA is used for domain adaptation, but most attempts which made use of attention to address the problem are based on still images, [17], [18], [19]. To the best of our knowledge, this is the first attempt on egocentric videos.

### *A. Feasibility study*

We first conducted a feasibility study on the results yielded by the LSTA network [2] when employed as a frame aggregator inside the TSM architecture, similarly to what has been done in Section IV-B. Table IX compares our method's performance after 200 epochs with what has been seen in Table II. The extremely encouraging results motivated us in pursuing **LSTA$^3$N**.

### *B. The results on $S \rightarrow S$*

Similarly to the process highlighted in Section IV-C, we evaluated the performance of LSTA$^3$N on the same domain-domain framework. This can be accounted as the theoretical upper bound towards which every domain adaptation technique should tend to. The results are shown in Table X, and compared to the ones shown in Table IV. Though there is a slight decrease compared to TSM+TRN, our claim is that attention will indeed help domain adaptation by driving the network towards the most active regions of the different domains, which should be domain-invariant and activity-specific.

### *C. The results on $S \rightarrow T$*

We applied LSTA$^3$N to the $6$ different domain shifts and compared the performance achieved by TA$^3$N alone, as seen in Table VIII. Table XI set the results side by side.

To guarantee a fair comparison, we stuck as much as possible to the hyper-parameter choice carried out whilst applying TA$^3$N. The only two changes we made regarded the attentive entropy loss weight $\gamma$ and the learning rate. The former was increased to $0.03$, whilst the latter was increased to $0.01$, similarly to the one used in Section IV-C.

Though one could argue that these numbers could be the real game-changers, it should be noticed that they were both considered when applying TA$^3$N as well (Table V), but similar results were never achieved before. Furthermore, the possibility of using such a large learning rate is a direct consequence of the engineered spatial attention. Being focused on the most active regions, it is less likely that the network ends up being

| S → T | D1 → D2 | | D1 → D3 | | D2 → D1 | | D2 → D3 | | D3 → D1 | | D3 → D2 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Temporal module** | **AvgPool** | **TRN** | **AvgPool** | **TRN** | **AvgPool** | **TRN** | **AvgPool** | **TRN** | **AvgPool** | **Trn** | **AvgPool** | **Trn** |
| *Target Only* | 61.07 | 72.93 | 63.86 | 72.29 | 52.18 | 58.62 | 62.86 | 72.29 | 52.18 | 58.62 | 61.07 | 72.93 |
| *Source Only* | 34.47 | 35.07 | 32.51 | 33.71 | 35.90 | 33.95 | 33.35 | 35.39 | 33.76 | 33.83 | 35.55 | 33.90 |
| $\hat{G}_{sd}$ | 34.63 | 35.67 | 32.61 | 33.64 | 35.90 | 33.57 | 33.36 | 35.03 | 33.70 | 33.31 | 36.07 | 35.19 |
| $\hat{G}_{td}$ | 34.67 | 34.39 | 32.56 | 33.00 | 35.90 | 36.68 | 33.49 | 34.94 | 33.44 | 34.73 | 36.19 | 36.83 |
| $\hat{G}_{rd}$ | / | 32.54 | / | 33.26 | / | 34.54 | / | 34.47 | / | 33.96 | / | 37.91 |
| All $\hat{G}_d$ (TA$^2$N) | 34.54 | 35.07 | 32.82 | 33.46 | 35.84 | 36.16 | 35.31 | 32.79 | 33.64 | 34.67 | 39.23 | 39.32 |
| All $\hat{G}_{sd}$ + domain attention (TA$^3$N) | 34.63 | 34.19 | 33.61 | 34.28 | 35.96 | 36.16 | 34.72 | 31.76 | 34.80 | 35.19 | 38.99 | 39.96 |

TABLE VIII: Best accuracy (%) for each domain shift and each component, after 100 epochs. For reference, also the theoretical upper bound (Target Only) is reported.

| | Accuracy (%) | | |
|---|---|---|---|
| **Network** | | | |
| | *D1* | *D2* | *D3* |
| **I3D** | 45.29 | 55.70 | 57.91 |
| **TSM** | 42.53 | 65.38 | 67.45 |
| **LSTA** | 54.48 | 73.06 | 73.51 |

TABLE IX: Feasibility study when using LSTA as frame aggregator in the TSM architecture. Results based on RGB frames, only.

| | Accuracy (%) | | |
|---|---|---|---|
| **Network** | | | |
| | *D1* | *D2* | *D3* |
| **I3D - AvgPool** | 52.18 | 61.07 | 63.86 |
| **TSM - TRN** | 58.62 | 72.93 | 72.29 |
| **LSTA$^3$N** | 56.23 | 70.00 | 70.43 |

TABLE X: Results achieved when LSTA$^3$N is applied to the same domain-domain framework. Results based on RGB frames, only.

| | D1 → D2 | D1 → D3 | D2 → D1 |
|---|---|---|---|
| **TA$^3$N** | 34.19 | 34.28 | 36.16 |
| **LSTA$^3$N** | **34.22** | **33.74** | **37.27** |
| | | | |
| | **D2 → D3** | **D3 → D1** | **D3 → D2** |
| **TA$^3$N** | 31.76 | 35.19 | 39.96 |
| **LSTA$^3$N** | **33.89** | **32.53** | **40.37** |

TABLE XI: Results achieved when LSTA$^3$N is used for cross-domain action recognition.

misled by insignificant elements even with a large learning rate.

LSTA$^3$N also requires the tuning of the LSTA's learning rate. The best tested configuration was found by setting it to a fairly small number (0.0005) and constantly halving it throughout the last 5 epochs.

Similarly to TA$^3$N, we considered the best accuracy after 30% of the total number of epochs, set to 50 in this case.

Amongst the 6 domain shifts, we have observed a significant improvement in the domain shifts stemming from $D2$ ($+1.11\%$ towards $D1$ and $+2.13\%$ towards $D3$, in terms of accuracy) whereas comparable results when $D2$ is the target.

For the remaining two shifts, the performance we achieved was not as good. Whilst the shift $D1 \rightarrow D3$ shows a small decrease ($-0.27\%$), the decrease attained when training on $D3$ and testing on $D1$ reached $2.66\%$. Adding up, the tested LSTA$^3$N increases the overall accuracy by $0.48\%$.

### D. Main limitations

Though the overall performance is satisfactory, our method comes with a few limitations. Unlike TA$^3$N which only requires feature vectors, LSTA$^3$N needs spatial feature tensors of shape $B \times C \times N \times H \times W$, which are clearly memory intensive. For example, the $D1 \rightarrow D3$ shift required the allocation of 36GB in the RAM. The model's computational complexity is increased, due to the generation of a per-frame attention map.

The big discrepancy highlighted in the $D3 \rightarrow D1$ domain suggests that our method's performance is more volatile and requires a shift-specific hyper-parameter tuning to make sure the attention maps are correctly generated. Any inaccuracy would propagate through the network and would become a proper systematic bias which substantially worsens the overall cross-domain performance. An approach like ours, aimed at finding the hyperparameters which globally optimise the overall performance rather than each specific domain shift, may have yielded underestimated results, even though it strikes more balance among all domain shifts.

## VI. Conclusions and Future Work

We proposed LSTA$^3$N, an extension of TA$^3$N which applies attention also on the spatial dimension. We showed how the overall performance is positively affected by our method, attaining a $+0.48$ gain. Though it was natural to aim for a more significant improvement, it must be said that the $D3 \rightarrow D1$ shift single-handedly worsens the overall performance, perhaps requiring a specific tuning.

Future work, which goes beyond the aim of this project, will 1) assess the results when a shift-specific hyper-parameter tuning is performed and 2) focus on utilising more modalities, such as optical flow. In fact, as shown by [10], the motion modality is more domain-invariant than RGB, resulting in a better interpretation of the background-irrelevant information. This can further help the action of LSTA.

## REFERENCES


[1] Chen, M. H., Kira, Z., AlRegib, G., Yoo, J., Chen, R., Zheng, J. (2019). "Temporal attentive alignment for large-scale video domain adaptation,". In Proceedings of the IEEE/CVF International Conference on Computer Vision Sudhakaran, Swathikiran and Escalera, Sergio and Lanz, Oswald

[2] Sudhakaran, S., Escalera, S., Lanz, O. (2018). "Long Short-Term Attention for Egocentric Action Recognition", In CVPR, 2019.

[3] Carreira, J., Zisserman, A. (2017). "Quo vadis, action recognition? a new model and the kinetics dataset". In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

[4] Lin, J., Gan, C., Han, S. (2019). "Tsm: Temporal shift module for efficient video understanding". In Proceedings of the IEEE/CVF International Conference on Computer Vision.

[5] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L. V. (2016, October). "Temporal segment networks: Towards good practices for deep action recognition". In European conference on computer vision (pp. 20-36). Springer, Cham.

[6] Kazakos, E., Nagrani, A., Zisserman, A., Damen, D. (2019). "Epic-fusion: Audio-visual temporal binding for egocentric action recognition". In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5492-5501)

[7] Zhou, B., Andonian, A., Oliva, A., Torralba, A. (2018). "Temporal relational reasoning in videos". In Proceedings of the European conference on computer vision (ECCV)

[8] Motiian, S., Jones, Q., Iranmanes, S., Doretto, G. (2017), "Few-shot adversarial domain adaptation". In Advances in Neural Information Processing Systems

[9] Motiian, S., Piccirilli, M., Adjeroh, D. A., Doretto, G. (2017, October) "Unified deep supervised domain adaptation and generalization". In The IEEE International Conference on Computer Vision.

[10] Munro, J., Damen, D. (2020). "Multi-modal domain adaptation for fine-grained action recognition". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition

[11] Li, Y., Liu, M., Rehg, J. M. (2018). "In the eye of beholder: Joint learning of gaze and actions in first person video". In Proceedings of the IEEE/CVF International Conference on Computer Vision.

[12] Li, Y., Ye, Z., Rehg, J. M. (2015). "Delving into Egocentric Actions". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.

[13] Sudhakaran, S., Lanz, O. (2018). "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition". In Proceedings of the British Machine Vision Conference.

[14] Kazakos, E., Nagrani, A., Zisserman, A., Damen, D. (2019). "Epic-fusion: Audio-visual temporal binding for egocentric action recognition". In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5492-5501)

[15] Kingma, D., Lei Ba, J. (2014), "ADAM: A method for Stochastic Optimization". In proceedings of the International Conference on Representation.

[16] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. "Domain-adversarial training of neural networks". The Journal of Machine Learning Research.

[17] Li, J., Wang, H., Wu, K., Liu, C., Tan, J. (2022), "Cross-attention-map-based regularization for adversarial domain adaptation"

[18] Kang, G., Zheng, L., Yan, Y., Yang, Y., (2018), "Deep Adversarial Attention Alignment for Unsupervised Domain Adaptation: the Benefit of Target Expectation Maximization"

[19] Wang, X., Li, L., Ye, W., Long, M., Wang, J. (2019), "Transferable Attention for Domain Adaptation", In proceedings of the 33rd AAAI Conference on Artificial Intelligence