Politecnico di Torino

Transformers for Egocentric Action Recognition

Group 3

S301231 Andrés Cárdenas
S300164 Edgar Gaytán
S303452 Luca Scalenghe

MLDL
A.A 2021/2022

Introduction

Introduction

Our work can be summarized in 2 parts:

1. Reproduction and discussion of the main approaches that have been proposed for the egocentric action recognition task.
2. A method to combine said approaches with the transformer architecture and MCC Loss.

Preamble: Egocentric Action Recognition

Challenges:

- Fine-grained actions (e.g: open bottle)
- Short action span (<1 s)
- Actions inside long videos

Opportunities:

- Exploitable Temporal Relationships
- Ordered Frames

Video Modalities

Feature Extraction

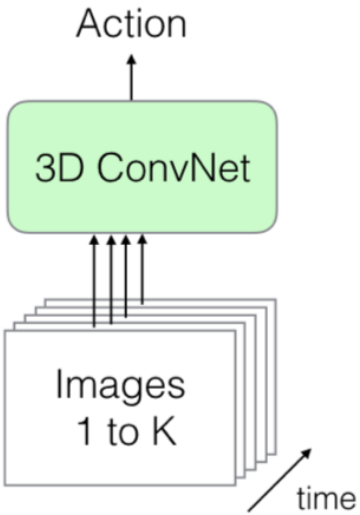
2D ConvNets Enhanced (TSM):

- Less Parameters, easier to train
- Less computationally expensive.

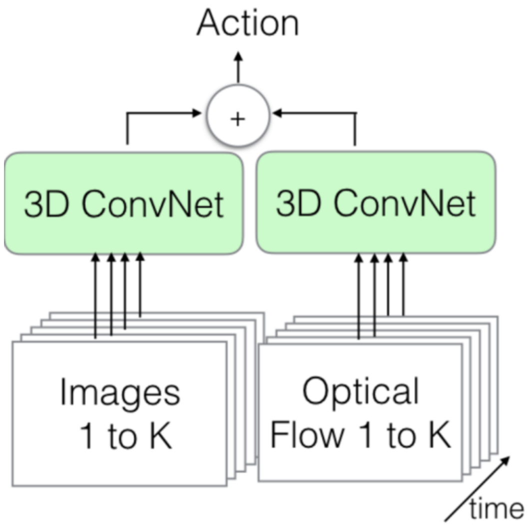
3D ConvNets (I3D):

- Many parameters, harder to train
- Convolutions of dense frames are computationally expensive.

I3D

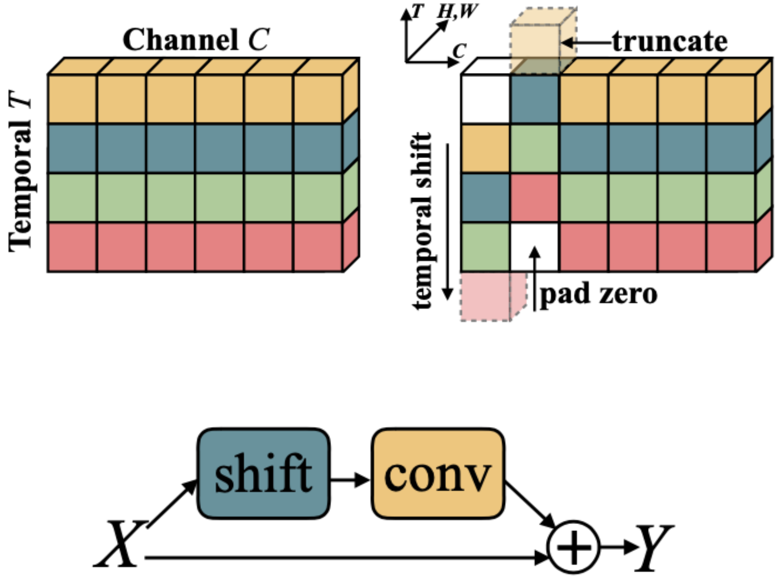


Single-stream convolutional neural network



Two-stream convolutional neural network

TSM



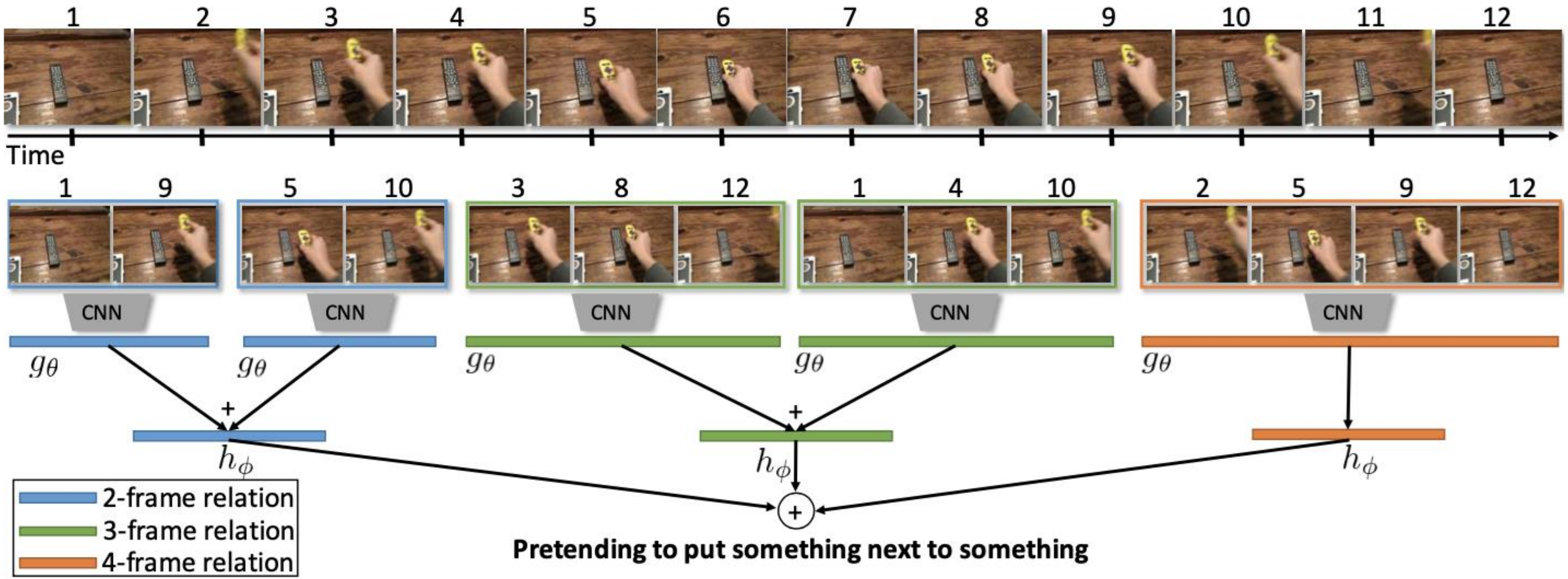
(b) Residual TSM.

Temporal Aggregation

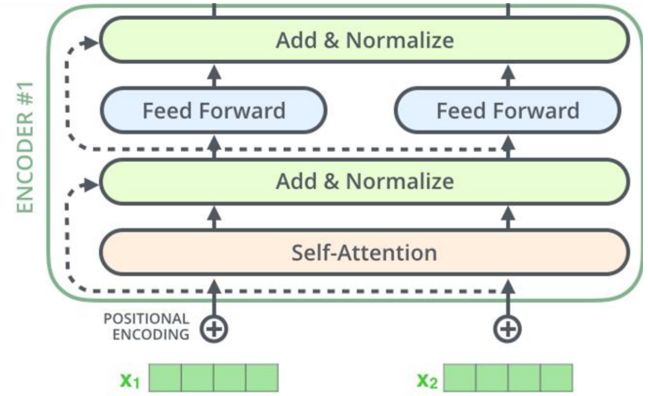
$$\hat{y} = G_y^{avg}(AvgPool(X))$$

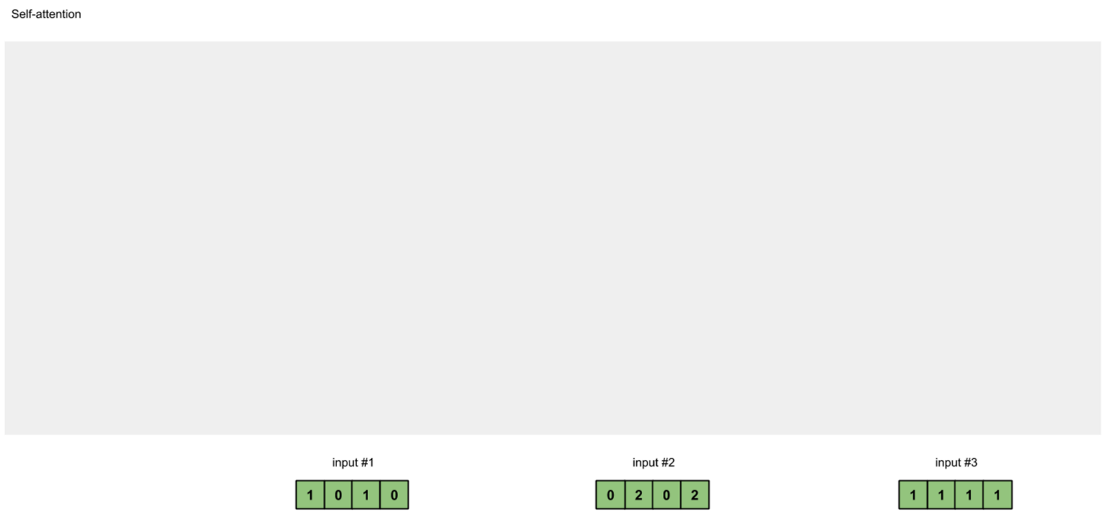
$$\hat{y} = G_y^{trn}(G_{tf}^{trn}(X))$$

TRN

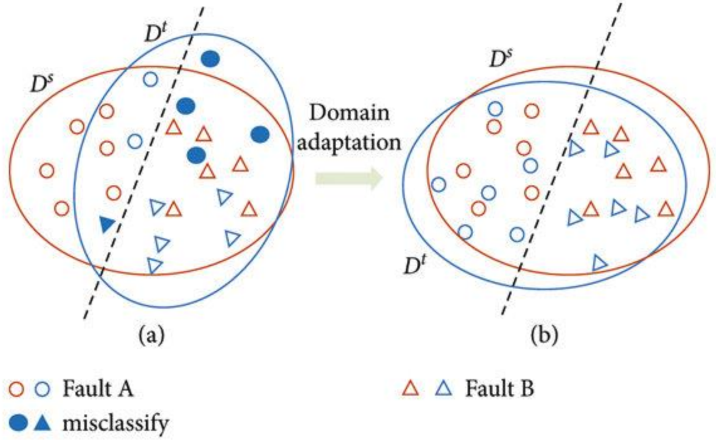

1 2 3 4 5 6 7 8 9 10 11 12
Time
1 9
5 10
3 8 12
1 4 10
2 5 9 12
CNN
CNN
CNN
CNN
CNN
g_θ
g_θ
g_θ
g_θ
g_θ
+
+
h_ϕ
h_ϕ
h_ϕ
+
2-frame relation
3-frame relation
4-frame relation
Pretending to put something next to something

Transformers and Self-attention mechanism

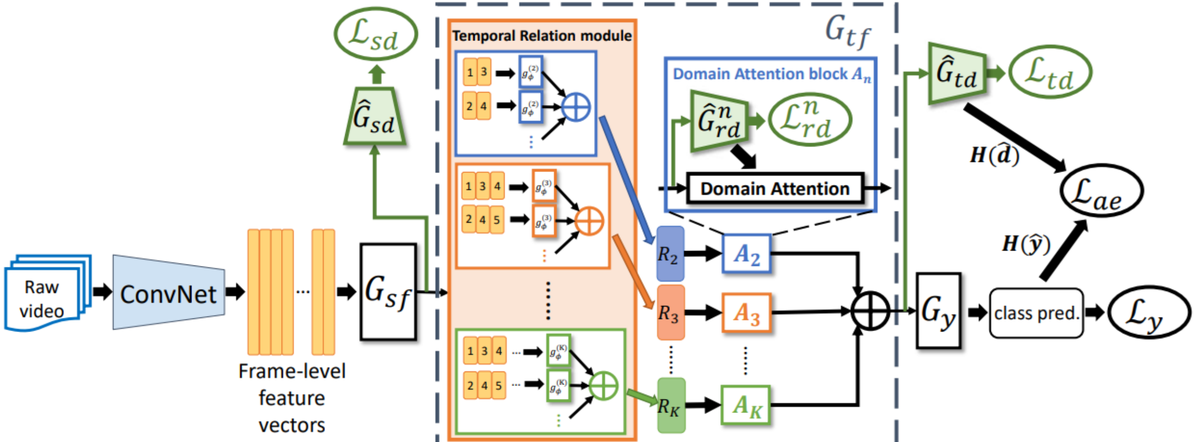




Domain Adaptation



TA3N (Temporal Attentive Adversarial Adaptation Network)



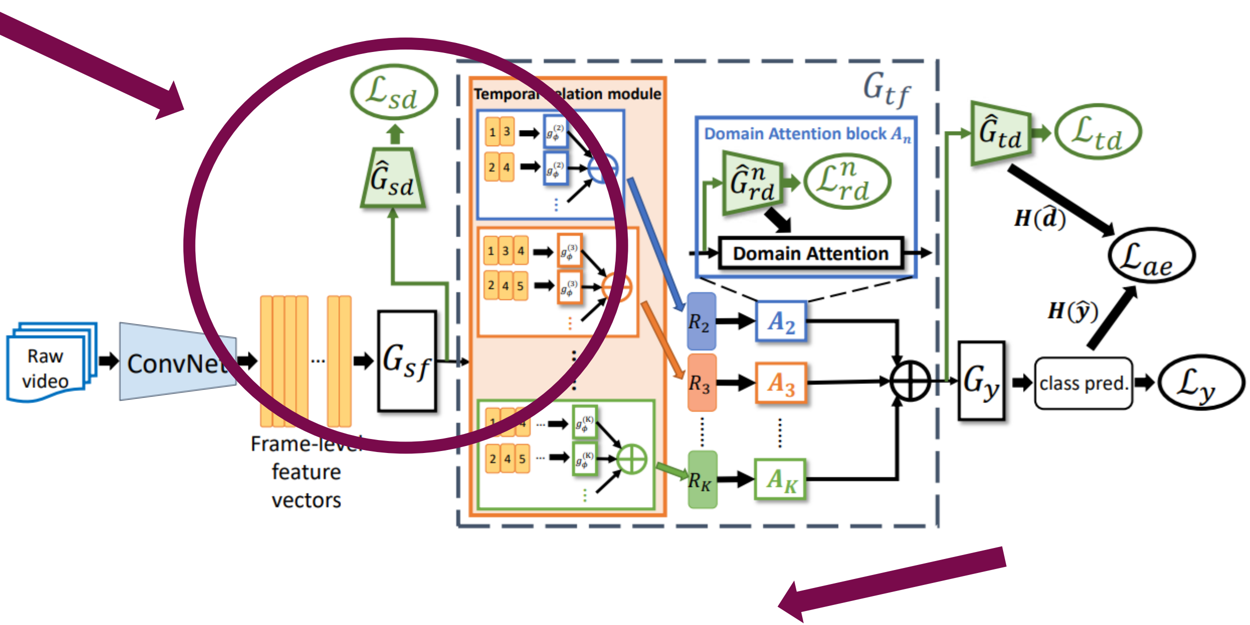
$$X \in \mathbb{R}^{n_f \times n_c}$$

$$G_{sf} : \mathbb{R}^{n_f \times n_c} \rightarrow \mathbb{R}^{n_h \times n_c}$$

$$G^{trn}_{tf} : \mathbb{R}^{n_h \times n_c} \rightarrow \mathbb{R}^{n_r}$$

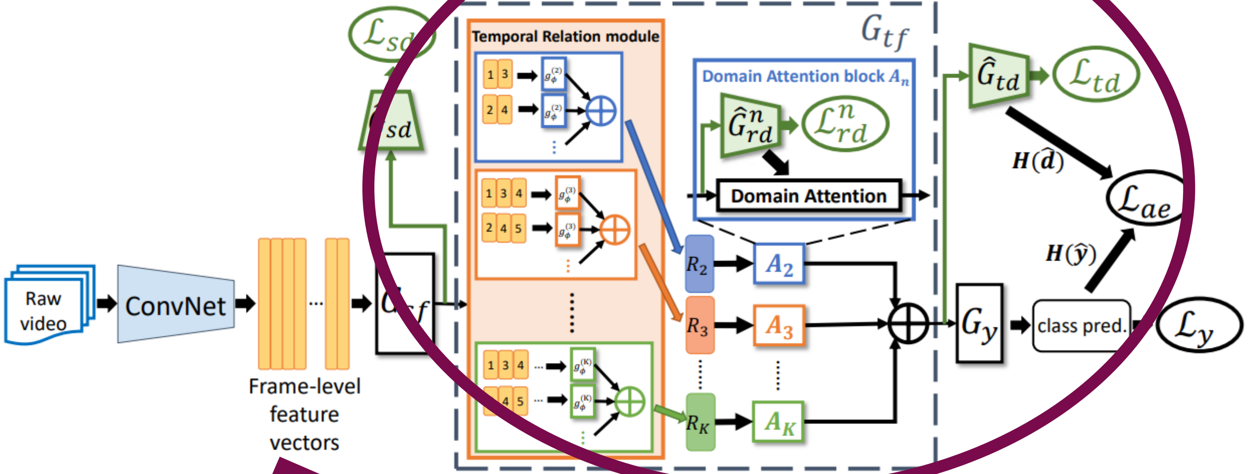
$$\hat{y} = G^{trn}_{y}(G^{trn}_{tf}(G_{sf}(X)))$$

Spatial Features Alignment



$$\hat{y} = G_y^{trn}(G_{tf}^{trn}(G_{sf}(X)))$$

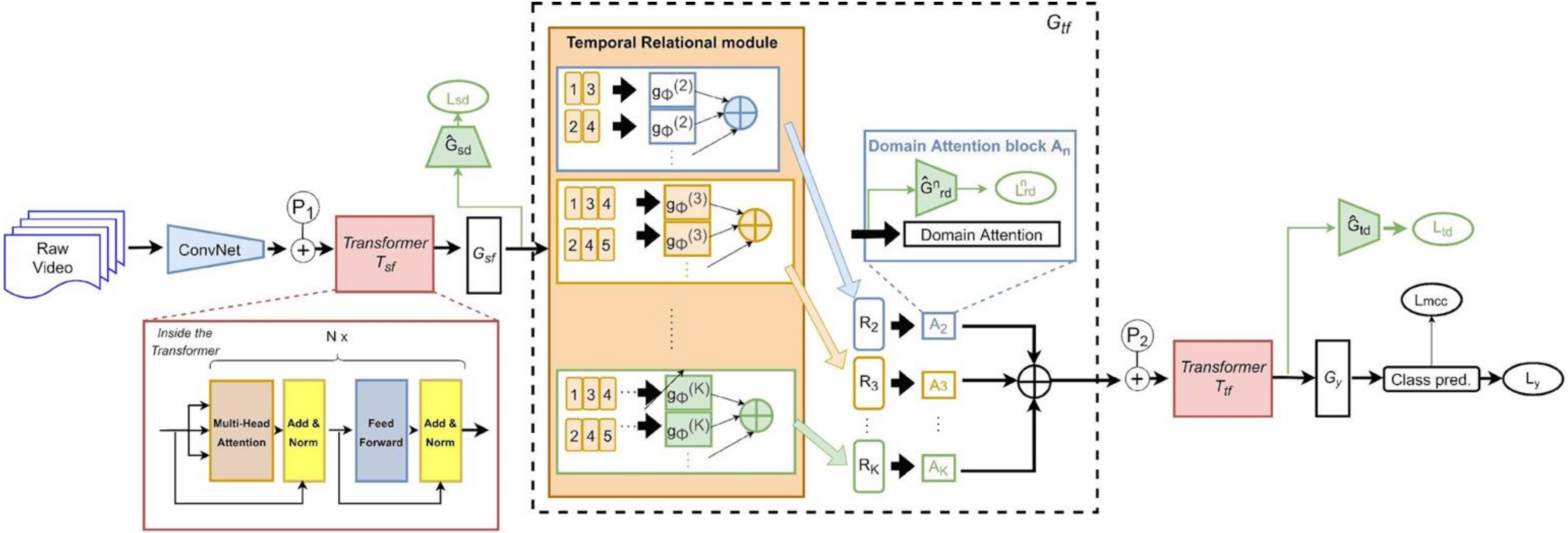
Temporal feature alignment



$$\hat{y} = G_y^{trn}(G_{tf}^{trn}(G_{sf}(X)))$$

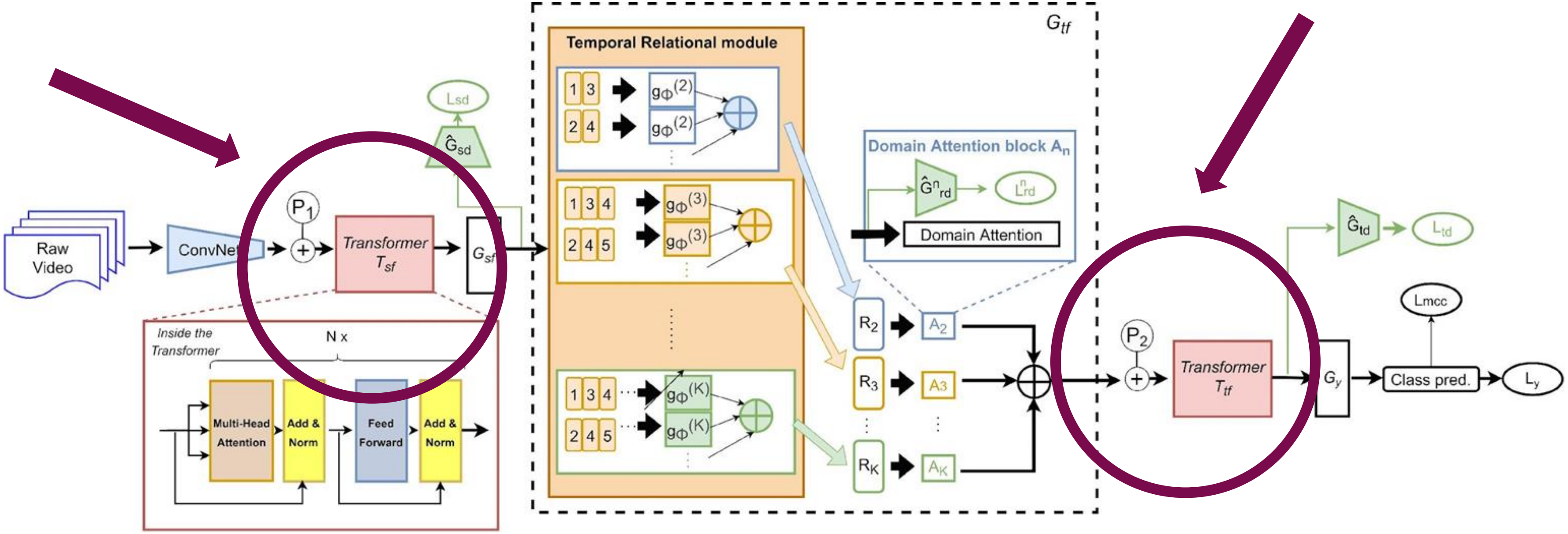
How to improve the original architecture?

Transformer Introduction



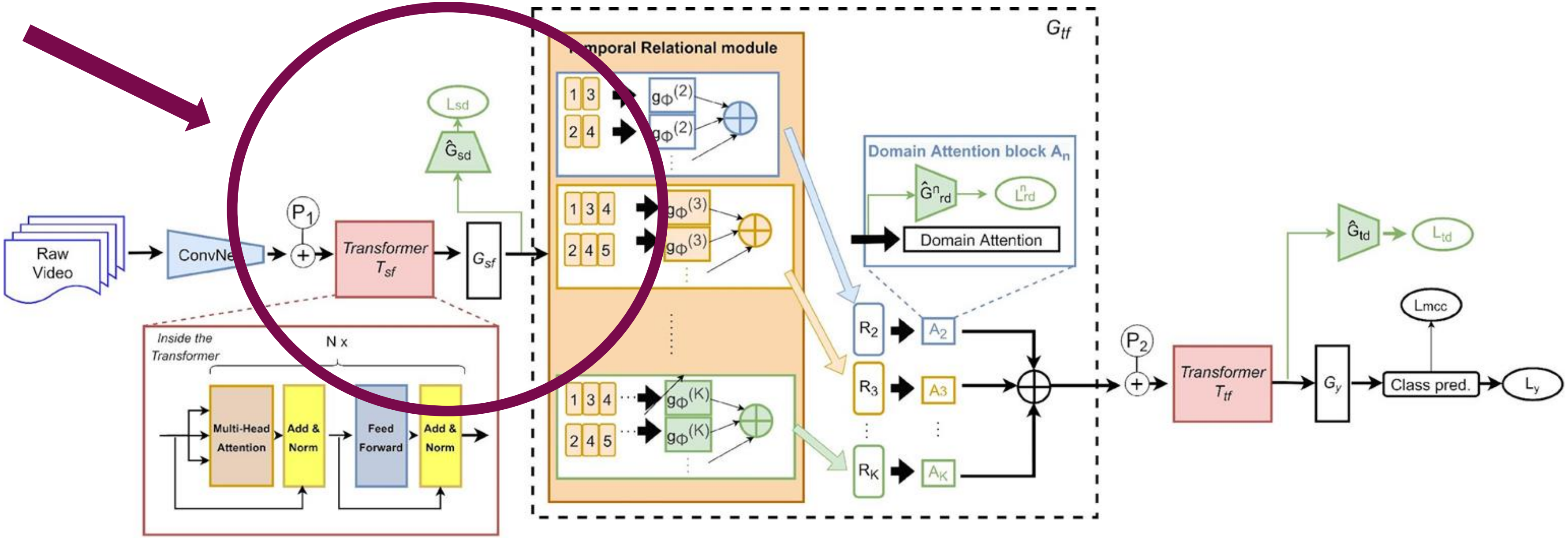
$$\hat{y} = G_y^{trn}(T_{tf}(G_{tf}^{trn}(G_{sf}(T_{sf}(X + P_1))) + P_2))$$

Transformer Introduction



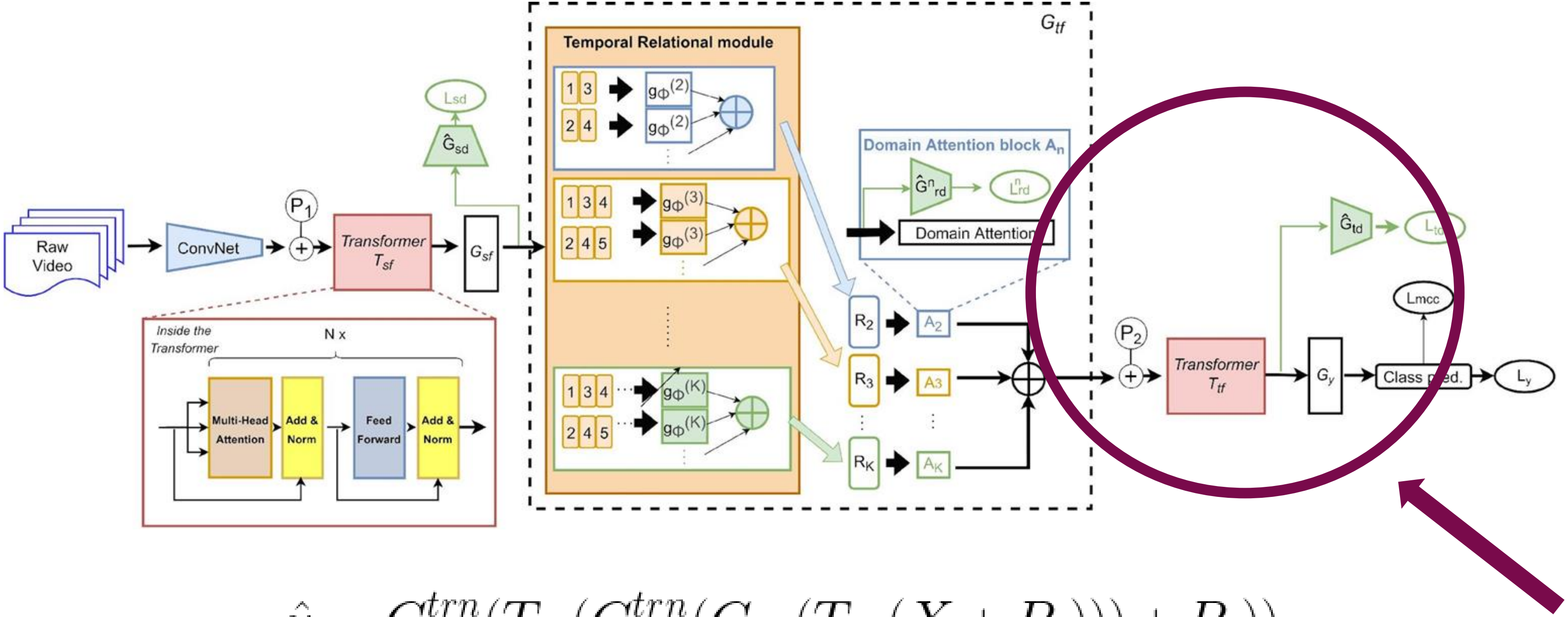
$$\hat{y} = G_y^{trn}(T_{tf}(G_{tf}^{trn}(G_{sf}(T_{sf}(X + P_1))) + P_2))$$

Spatial Alignment (Frame Self-Attention)



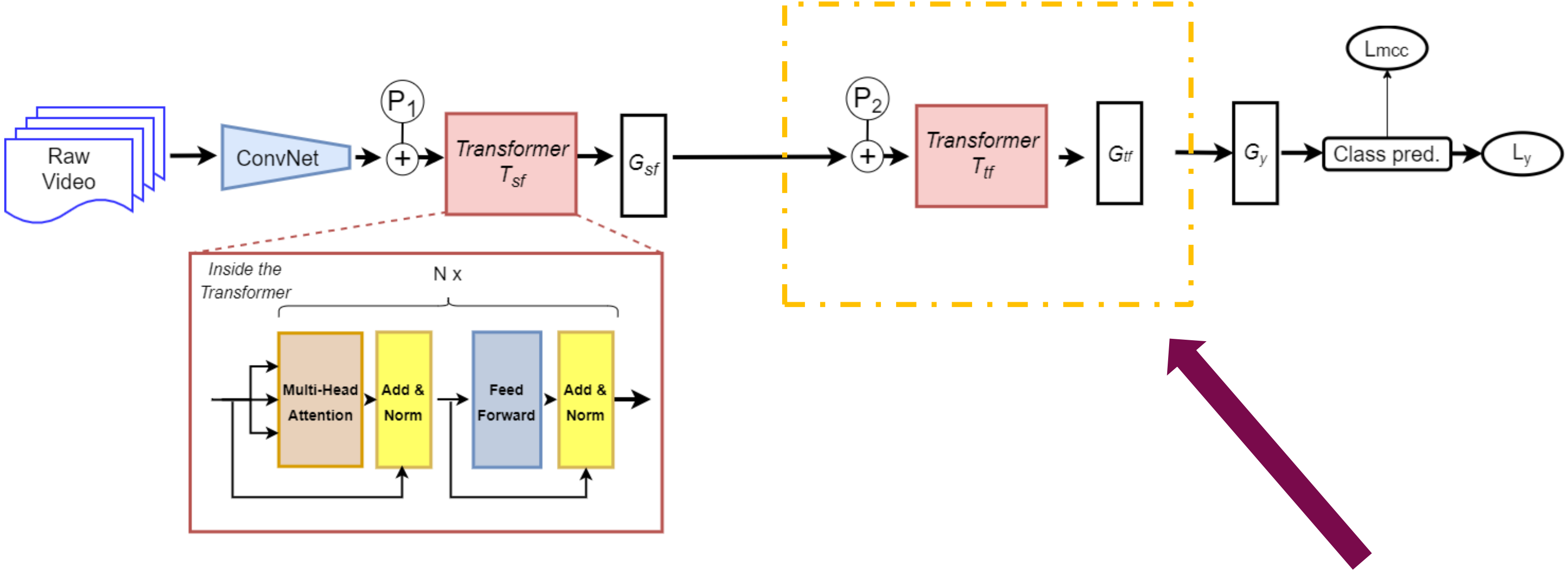
$$\hat{y} = G_y^{trn}(T_{tf}(G_{tf}^{trn}(G_{sf}(T_{sf}(X + P_1))) + P_2))$$

Temporal Alignment (Relation Self-Attention)



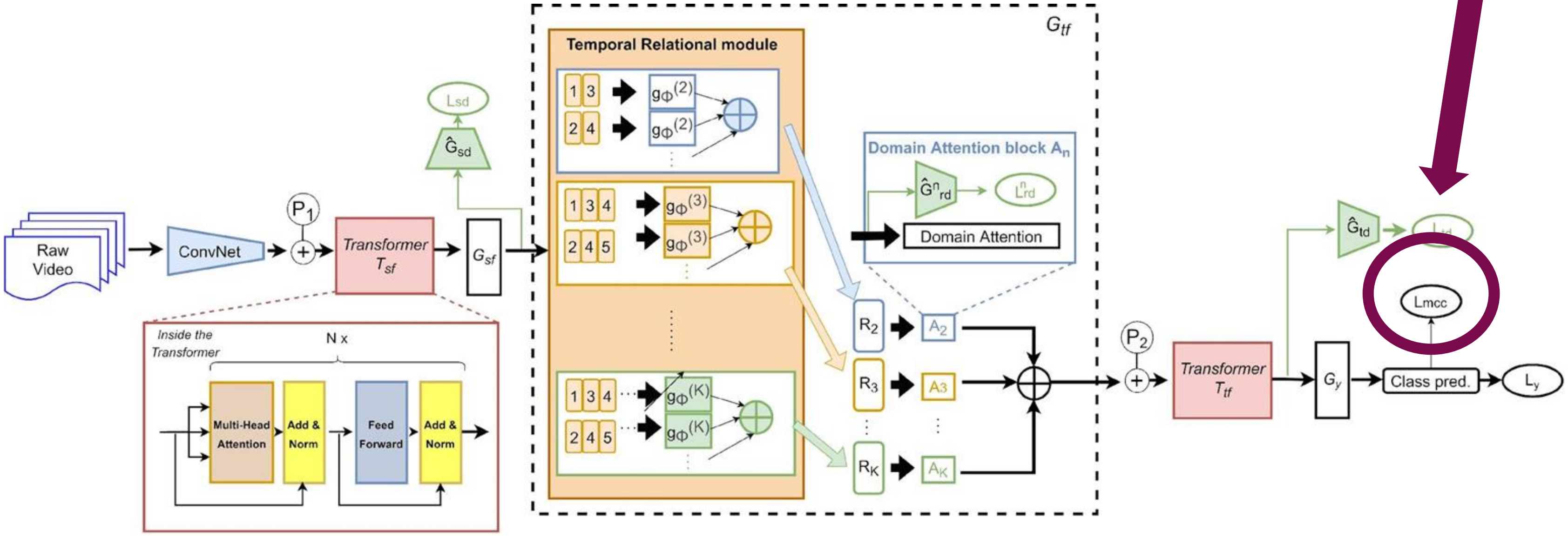
$$\hat{y} = G_y^{trn}(T_{tf}(G_{tf}^{trn}(G_{sf}(T_{sf}(X + P_1))) + P_2))$$

Transformer as temporal aggregator



$$\hat{y} = G_y^{avg}(AvgPool(T_{tf}(G_{sf}(T_{sf}(X + P_1))) + P_2))$$

MCC Loss



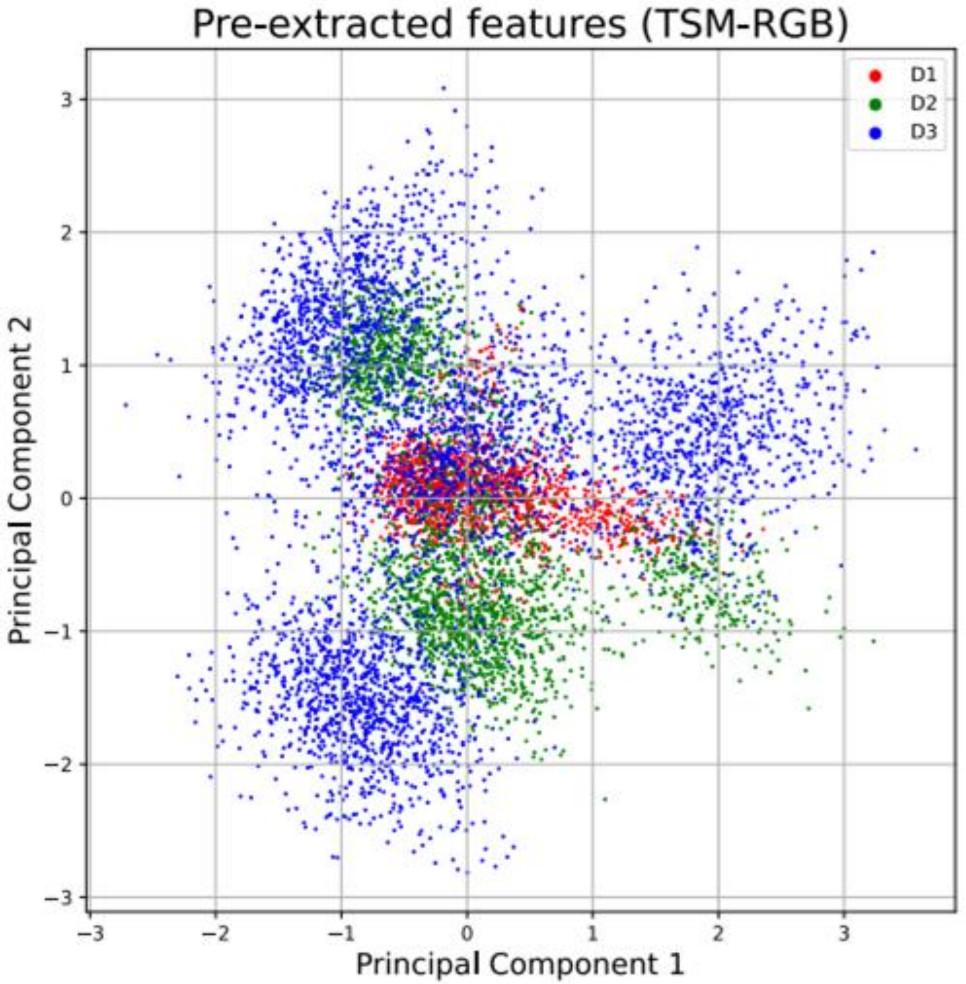
Raw Video
ConvNet
Transformer T_{sf}
G_{sf}
Lsd
$\hat{G}_{sd}$
Inside the Transformer
N x
Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
G_{tf}
Temporal Relational module
Domain Attention block A_n
Domain Attention
Transformer T_{tf}
G_y
$\hat{G}_{td}$
Lmcc
Class pred.
L_y

EPIC-KITCHENS

- One of the largest datasets available for egocentric action recognition
- 32 participants in 4 cities
- 55h of video recorded
- 11.5M frames
- 39.6K action segments

Dataset

- **Participants P01, P08, P22**
- **PCA of extracted features from RGB flow using TSM network**
- **Clear difference between the features of different participants**

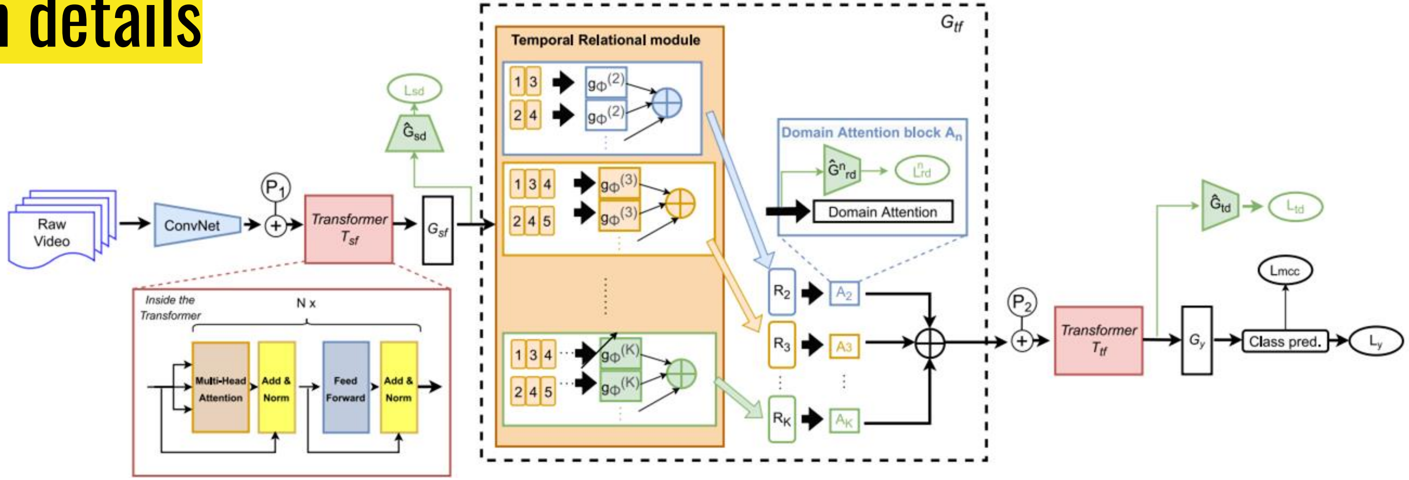


Experiments

Network	Accuracy RGB (%)	Accuracy Optical Flow (%)	Accuracy RGB+Flow(%)
I3D	53.70	57.77	59.76
TSM	70.06	69.71	75.43
I3D + AveragePool	62.57	66.57	x
TSM + TRN	71.98	68.86	x

- **Results using pre-trained architectures for I3D and TSM**
- **Combined flows shows higher overall accuracy, consistent with the two-stream hypothesis**
- **Using temporal aggregation strategies increased performance**

Implementation details



- **For all components: lr1 = 0.001 and dropout 0.8**
- **For the transformer: lr2 = 0.0001, single encoding layer with 2 heads, dropout of 0.1**
- **GRLs use hyper-parameter β = 1 for all the domain classifiers**
- **MCC uses a temperature value of T = 2 and weighting value μ = 1**

Results -TA3N

Components	Aggregation Method	TA3N	Frame Self-Attention	Relational Self-Attention	Frame + Relational Self-Attention	Frame + Relational + MCC
Source	Average Pooling	36.05	38.19	37.30	38.43	**38.74***
	TRN Pooling	36.34	34.50	**38.61**	37.46	37.47*
Grd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	35.92	39.32	**39.41**	38.44	38.02
Gsd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	36.34	38.57	**39.23**	38.27	37.59
Gtd	Average Pooling	36.05	**38.33**	37.81	38.10	37.99
	TRN Pooling	36.75	38.73	**39.45**	38.43	38.69
All Gd	Average Pooling	36.29	38.99	**39.76**	38.84	38.07
	TRN Pooling	36.63	**38.87**	38.86	38.63	38.57
All Gd + Domain Attention	Average Pooling	37.22	**39.32**	39.18	38.78	37.47
	TRN Pooling	$\boxed{37.47}$	38.66	39.28	38.60	**39.04**

TA3N without any modification obtains the best result using TRN as temporal aggregator with the domain attention mechanism and all the domain classifiers

Results - Domain Classifiers

Components	Aggregation Method	TA3N	Frame Self-Attention	Relational Self-Attention	Frame + Relational Self-Attention	Frame + Relational + MCC
Source	Average Pooling	36.05	38.19	37.30	38.43	**38.74***
	TRN Pooling	36.34	34.50	**38.61**	37.46	37.47*
Grd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	35.92	39.32	**39.41**	38.44	38.02
Gsd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	36.34	38.57	**39.23**	38.27	37.59
Gtd	Average Pooling	36.05	**38.33**	37.81	38.10	37.99
	TRN Pooling	36.75	38.73	**39.45**	38.43	38.69
All Gd	Average Pooling	36.29	38.99	**39.76**	38.84	38.07
	TRN Pooling	36.63	**38.87**	38.86	38.63	38.57
All Gd + Domain Attention	Average Pooling	37.22	**39.32**	39.18	38.78	37.47
	TRN Pooling	37.47	38.66	39.28	38.60	**39.04**

Introduction of the adversarial task is beneficial.

Peak performance when the three classifiers are used simultaneously

Results - Transformers

Components	Aggregation Method	TA3N	Frame Self-Attention	Relational Self-Attention	Frame + Relational Self-Attention	Frame + Relational + MCC
Source	Average Pooling	36.05	38.19	37.30	38.43	**38.74***
	TRN Pooling	36.34	34.50	**38.61**	37.46	37.47*
Grd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	35.92	39.32	**39.41**	38.44	38.02
Gsd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	36.34	38.57	**39.23**	38.27	37.59
Gtd	Average Pooling	36.05	**38.33**	37.81	38.10	37.99
	TRN Pooling	36.75	38.73	**39.45**	38.43	38.69
All Gd	Average Pooling	36.29	38.99	**39.76**	38.84	38.07
	TRN Pooling	36.63	**38.87**	38.86	38.63	38.57
All Gd + Domain Attention	Average Pooling	37.22	**39.32**	39.18	38.78	37.47
	TRN Pooling	37.47	38.66	39.28	38.60	**39.04**

Transformers proved to be beneficial bringing increments of around 2%

Results - Best configuration

Components	Aggregation Method	TA3N	Frame Self-Attention	Relational Self-Attention	Frame + Relational Self-Attention	Frame + Relational + MCC
Source	Average Pooling	36.05	38.19	37.30	38.43	**38.74***
	TRN Pooling	36.34	34.50	**38.61**	37.46	37.47*
Grd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	35.92	39.32	**39.41**	38.44	38.02
Gsd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	36.34	38.57	**39.23**	38.27	37.59
Gtd	Average Pooling	36.05	**38.33**	37.81	38.10	37.99
	TRN Pooling	36.75	38.73	**39.45**	38.43	38.69
All Gd	Average Pooling	36.29	38.99	$\boxed{\textbf{39.76}}$	38.84	38.07
	TRN Pooling	36.63	**38.87**	38.86	38.63	38.57
All Gd + Domain Attention	Average Pooling	37.22	**39.32**	39.18	38.78	37.47
	TRN Pooling	37.47	38.66	39.28	38.60	**39.04**

Transformer Ttf with all domain classifiers and average pooling is best performing

This result also confirms what previous research already stated that temporal alignment is more important then spatial alignment

Results - MCC loss

Components	Aggregation Method	TA3N	Frame Self-Attention	Relational Self-Attention	Frame + Relational Self-Attention	Frame + Relational + MCC
Source	Average Pooling	36.05	38.19	37.30	38.43	**38.74***
	TRN Pooling	36.34	34.50	**38.61**	37.46	37.47*
Grd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	35.92	39.32	**39.41**	38.44	38.02
Gsd	Average Pooling	36.05	38.86	39.13	**39.23**	37.81
	TRN Pooling	36.34	38.57	**39.23**	38.27	37.59
Gtd	Average Pooling	36.05	**38.33**	37.81	38.10	37.99
	TRN Pooling	36.75	38.73	**39.45**	38.43	38.69
All Gd	Average Pooling	36.29	38.99	**39.76**	38.84	38.07
	TRN Pooling	36.63	**38.87**	38.86	38.63	38.57
All Gd + Domain Attention	Average Pooling	37.22	**39.32**	39.18	38.78	37.47
	TRN Pooling	37.47	38.66	39.28	38.60	**39.04**

Adding MCC loss was a partial success

We think that with further exploration this would be a successful approach

Conclusion

- Showed the main approaches adopted in this field with special focus on the domain adaptation task

- We presented a promising improvement using transformers for spatio-temporal alignment

- Open questions:
  - Integration of domain adaptation using other modalities (audio…)
  - Use of deeper transformers without incurring into excessive complexity

Thanks for your attention