Comparison of Deep Learning Approaches for Protective Behaviour Detection Under Class Imbalance from MoCap and EMG data

Karim Radouane, Andon Tchechmedjiev, Binbin Xu, Sebastien Harispe

Comparison of Deep Learning Approaches for Protective Behaviour Detection Under Class Imbalance from MoCap and EMG data

Karim Radouane, Andon Tchechmedjiev, Binbin Xu and Sébastien Harispe
EuroMov Digital Health in Motion Univ Montpellier, IMT Mines Ales, Ales, France
{karim.radouane, andon.tchechmedjiev, binbin.xu, sebastien.harispe}@mines-ales.fr

***Abstract*—The AffecMove challenge organised in the context of the H2020 EnTimeMent project offers three tasks of movement classification in realistic settings and use-cases. Our team, from the EuroMov DHM laboratory participated in Task 1, for protective behaviour (against pain) detection from motion capture data and EMG, in patients suffering from pain-inducing muskuloskeletal disorders. We implemented two simple baseline systems, one LSTM system with pre-training (NTU-60) and a Transformer. We also adapted PA-ResGCN a Graph Convolutional Network for skeleton-based action classification showing state-of-the-art (SOTA) performance to protective behaviour detection, augmented with strategies to handle class-imbalance. For PA-ResGCN-N51 we explored naïve fusion strategies with an EMG-only convolutional neural network that didn't improve the overall performance. Unsurprisingly, the best performing system was PA-ResGCN-N51 (w/o EMG) with a F_1 score of 53.36% on the test set for the minority class (MCC 0.4247). The Transformer baseline (MoCap + EMG) came second at 41.05% F_1 test performance (MCC 0.3523) and the LSTM baseline third at 31.16% F_1 (MCC 0.1763). On the validation set the LSTM showed performance comparable to PA-ResGCN, we hypothesize that the LSTM over-fitted on the validation set that wasn't very representative of the train/test distribution.**

***Index Terms*—AffectMove Task 1, Multimodal behaviour classification, Deep learning architectures.**

I. Introduction

The 2021 AffectMove challenge [1] organised in the context of the EnTimeMent H2020 project and collocated, as a workshop, with the ACII 20201 conference, offered the opportunity of participating in three challenge tasks that involve the detection of characteristics of human motion in multimodal and unimodal settings:

1) **Protective Behaviour Detection based on Multimodal Body Movement Data.** This first task provided participants with Motion capture (MoCap) + Electromyography (EMG) data from the EMOPAIN [2] for protective behaviour detection (which is a response to pain while performing the actions) across several subjects and tasks, with fixed three-second recordings.
2) **Detection of Reflective Thinking based on Body Movement Data.** The second task provided participants with MoCap data only for reflective thinking classification across several subjects and tasks, with variable recording lengths and uneven sampling.
3) **Detection of Lightness and Fragility in Dance Movement based on Multimodal Data.** The third tasked involved the multimodal classification of movement from video and accelerometers to detect frailty in dance moves.

Task 1 focuses mainly on protective behaviour detection in response to pain in subjects suffering from disorders due to chronic musculoskeletal pain. Such disorders, including the typical example of low-back-pain, although it may appear to be a mundane non-systemic disease of little concern, in reality it affects a significant part of the population. Chronic musculoskeletal pain causes significant disruptions of quality of life, but is also a leading cause of medical leave in professionals. Musculoskeltal disorder and the resulting pain are a significant burden on national healthcare systems. Although there are general principles in their treatment, few objective criteria exist to evaluate the recuperation of patients or their level of pain (strong psychosomatic component). Protective behaviour in response to pain in particular, hinders recovery by accentuating non-use, and also constitutes an objective marker in gauging the real level of pain (as opposed to self-reported scales). Being able to detect protective behaviour automatically from sensor data, can provide a valuable tool to reform clinical protocols in the treatment of chronic musculoskeletal disorders.

In the dataset of Task 1, each subject and experimental sequence pair represented one record and was materialized as a single file. The MoCap data was encoded in the first 51 columns of the data file (17 joints $\times$ 3 spatial coordinates) and the EMG data (two electrodes on the upper fibres of trapezius muscles, two on the lumbar paraspinal muscles) was encoded in the next 4 columns, followed by the action identifier. Each instance was composed of 180 frames (60fps) and the EMG signal was prepossessed (signal envelope), down-sampled and aligned with MoCap frames.

Predicting the protective behaviour requires devising systems that either rely on MoCap solely or multimodal systems considering both MoCap and EMG.

Possible machine learning approaches include those from conventional machine learning and deep learning. Relevant model architectures mainly stem from computer vision and particularly from skeleton-based action recognition systems. In this work, we choose to focus on deep learning approaches,

This work was granted access to the HPC resources of IDRIS under the allocation 2021-AD011011309R1 made by GENCI.

978-1-6654-0021-3/21/$31.00

building at first the baselines with simple methods (LSTM [3], [4] and Transformer [5]) and then exploring one state-of-the-art (SOTA) system of action recognition based on graph convolutional networks [6], which leverage the specificity of skeleton data (skeleton topology, relative distances, velocities). Each of the systems was developed by a different member of the team independently and then integrated in a common evaluation setting, in order to perform comparable model selection.

In this paper, we give a synthetic account of related work, followed by a description of the three systems (design, model selection, evaluation) and then a meta-analysis of the results with regard to the overall objective and the final scores of our systems on the test set of the challenge.

II. Related Work

The common baseline neural network model to treat time series type data is generally based on LSTM (Long short-term memory) architectures and skeleton-based action recognition is no exception. In the EMOPAIN Challenge 2020 [3], that is based on the same EMOPAIN dataset [2] as the current AffectMove task (with different labels), participants (also in [4]) used a stacked LSTM model as baseline built with three LSTM layers (32 hidden units) followed by a dropout layer with probability of 0.5. This model gave an average accuracy of 0.828 on *test* set (MOCAP and EMG data combined).

More generally, LSTMs when applied to movement classification in larger benchmarks such as NTU-60 or NTU-120 [7] have either, been augmented to take into account spatiotemporal information, showing moderate performance (STA-LSTM at 64% accuracy [8]), or with attention mechanisms, showing close to SOTA performance (AGC-LSTM at 95% accuracy [9]).

Besides approaches based on recurrent neural networks, Transformers have shown tremendous potential, initially in Natural Language Processing, but in the past year the model has been adapted and applied to other types of modalities, from speech, to computer vision, to multi-modal systems. There are now numerous Transformer adaptations on temporal, spatial or spatio-temporal data, including architectures specific to Motion Capture and/or pose estimation data. A few notable examples include:

- the Variational Autoencoder Transfomer (Transformer VAE [10]) that allows learning a representation of parameters of human motion (MoCap), pre-trained on an action-conditioned movement generation task;
- the Spatial Temporal Transformer Network (ST-TA [11]), is also a Transformer specifically adapted to MoCap time-series, by a fusion of a temporal attention model (in the standard formulation of temporal self-attention) and of a spatial transformer that extends the more typical graph convolutional network approach with an attention mechanism.

While Transformer VAE isn't technically an action recognition system, it can, and has been evaluated as such (on a small subset of NTU-13). Although it doesn't show SOTA performance on-par with systems specialized for action recognition, the architecture is extremely innovative and could be eventually harnessed to produce multi-task representations of movement similar to what has transformed Natural Language Processing since 2018. ST-TR on the other hand shows SOTA performance (96.1% accuracy on NTU-60), but the "temporal transformer" is actually implemented as a convolutional neural network with KQV (Key-Query-Value) attention on top, and the spatial transformer is in fact a Graph Convolutional Network with added attention. As such, ST-TR is much more similar to the more common GCN architectures than it is with the Transformer, besides the use of attention, which in itself isn't specific to transformers and has existed long before the inception of Transformers.

Speaking of GCNs, they constitute the other major family of approaches for skeleton-based action-recognition and the current SOTA system among GCNs, PA-ResGCN [6], is almost *ex aequo* with ST-TR at 96.1%, which probably doesn't constitute a significant difference. ResGCN combines the 3D coordinates with spatial skeleton structure (relative distances between joints) and with velocity gradients. Additionally, part-wise attention is used to inform the contribution of 5 manually defined body parts (sub-graphs) to the overall classification. We shall not further describe the state of the art of GCNs for skeleton based action recognition, but Song et al. [6] already present a very comprehensive literature review and comparison. GCNs have already been applied to the EMOPAIN dataset as well [12].

Based on the observations, we chose to implement a vanilla Transformer baseline without adaptations to evaluate its potential for Task 1. We also set-out to adapt and apply the current non-Transformer SOTA system for skeleton-based action recognition, PA-ResGCN to gauge its performance in the *real-life setting* offered by Task 1.

This review doesn't address the multi-modal integration of EMG, although our Transformer baseline significantly benefited from EMG through the most naïve form of early fusion: concatenation of EMG features with the 3D joint coordinates. The PA-ResGCN system doesn't allow such a use of EMG, and the time allotted for the challenge didn't allow us to explore effective late fusion strategies either. We did make attempts on simple late fusion, through the concatenation of a fully connected layer from the PA-ResGCN output (-N51) with that of a simple CNN effective at EMG classification with a common training objective. However, these attempts were unsuccessful (lower performance than without fusion). However, attempts for multimodal fusion of EMG and MoCap on EMOPAIN, have shown some success, particularly with more sophisticated fusions mechanisms [4], [13], indicating that this may be a path forward for further improvement. Another aspect of interest to exploiting the Task 1 data set, is the handling of significant class imbalance (Table I), we won't explore the literature exhaustively, but we employed a dynamic weighting scheme from the literature [14] for the binary cross entropy loss, which proved *capital* in obtaining good convergence during training. We also experimented with

using a Mathew's Correlation Coefficient loss [15], which is robust to class imbalance and matched the metric used in Task 1. Although it was also successful in countering the class imbalance, it's numerically less numerical stable.

III. Methods

A. Dataset and Protocol

As described earlier, the data set provides a sequence of 180 3D skeleton joints (17×3) along with 4 channels of EMG signal downsampled and synchronized. The dataset is labeled with actions and protective behaviour annotations, but the labels are only provided to participants for the training and validation subsets. Table I presents some general statistics about training and validation data. One important characteristic is the strong class imbalance and the significantly smaller size of the validation set. Additionally, since there are several repetitions by several subjects (a joint inter/intra subjects acquisition design), it is important to keep subject sessions atomically linked together if the data set is shuffled prior to training a system, as otherwise significant bias would be induced.

TABLE I
Characteristics and distribution of classes across the train and validation sets.

Training set			Validation set		
Protective	**Count**	**Percent**	**Protective**	**Count**	**Percent**
0	4522	77.60%	0	1580	85.68%
1	1305	22.40%	1	264	14.32%

We also observed, in one of the baselines for the EMOPAIN dataset [3], that the scores obtained on the *validation* set are surprisingly low (accuracy 0.4636, and F_1 0.4811 [4]) for a binary classification problem, while the performance on the training set was higher, and comparable to that obtained on the test set. These discordant scores between *validation* and *test* sets suggest that these two datasets are hypothetically highly different in feature space (they represent different distributions of classes and subjects). It appears that in the AffectMove challenge in 2021, a similar data splitting rule was likely applied in preparing the data sets, given the observed distribution (Table I), as models with good performance on the *training* set systematically showed less interesting performance on the *validation* set in a consistent manner. This raises the question of how representative the validation set is compared to the final tests set, as tuning hyper parameters on a non-representative distribution would likely lead to a system that significantly under-performs on the test set.

In this work, we chose to use the *validation* set to fine-tune the systems *and* to produce the final prediction, although in hindsight, a retraining on both the training and validation sets to produce the final predictions could have been more appropriate for some of the models, particularly the LSTM which suffered greatly for significant overfitting on the validation set, as we will describe later.

B. Transformer-based architectures

Several *simple* Transformer models [5] have been tested in our experiments. These models rely on the very popular self-attention mechanism to construct an embedding representation of the input signal, that will next be used to perform the final binary classification. We introduce the general design approach considered as well as details about the Transformer architecture we have adopted.

1) General Architecture: We consider a general seq-to-one approach i.e. for a given sequence of input values indexed by the temporal dimension (MoCap and corresponding EMG data), the model produces a single 2D output corresponding to the probability distribution of observed protective behaviour (probability that a protective behaviour is observed or not).

The several steps considered are (i) iterative computation of timestep embeddings using transformer blocks based on self-attention, (ii) instance embedding averaging the timestep embeddings, (iii) linear transformation of the instance embedding into $\mathbb{R}^2$, (iv) softmax to obtain the final probability distribution over the two classes for protective behaviour.

In the following we index by $t \in [\![1, 180]\!]$ the different timesteps composing a given input to classify.[1] For a timestep t, we also denote respectively the 4×1D EMG input values and the 17×3D MoCap input values by $x^t_{EMG} \in \mathbb{R}^4$, and $x^t_{MOC} \in \mathbb{R}^{51}$ respectively. In this section $x^t = (x^t_{EMG}, x^t_{MOC})$ is the concatenation of the two vectors x^t_{EMG} and x^t_{MOC} at time t ($x^t \in \mathbb{R}^{55}$). The input provided to the transformer model was $X = [x^{(1)}, \dots, x^{(180)}]^\intercal \in \mathbb{R}^{55 \times 180}$. No additional efforts have been made to help the model (i) extract specific characteristics of the MOCAP and EMG data (e.g. by applying specific preprocessing steps), (ii) explicitly distinguish between the 17 3D coordinates corresponding to the MoCap data.

The general transformer architecture considered in our experiments can mainly be seen as an encoder stacking n transformer blocks that are trainable mappings, i.e. a transformer block $i \in [\![1, n]\!]$ is a mapping $B^{(i)} : \mathbb{R}^{d \times 180} \to \mathbb{R}^{d \times 180}$, i.e. the stacking is the composition $B^{(n)}(\dots(B^{(1)}(X)))$. A linear transformation is first applied element-wise to each of the timestep entries of $X \in \mathbb{R}^{55 \times 180}$ to obtain the $\mathbb{R}^{d \times 180}$ input that will be given to the first transformer block, i.e. the same linear transformation from $\mathbb{R}^{55}$ to $\mathbb{R}^d$ is used to project each $x^t, t \in [\![1, 180]\!]$ into $\mathbb{R}^d$ (d is the embedding size and an hyperparameter of our system).

Each transformer block applies standard treatments such as : (i) standard Multi-Head self-Attention (MHA), (ii) normalization prior and after MHA, (iii) a feedforward network applied element-wise using 2 fully connected layers based on ReLu activation function. The Query, Key, and Value linear transformations of each transformer block is a transformation from $\mathbb{R}^d$ to $\mathbb{R}^d$ with each of the h head processing an input in $\mathbb{R}^{d/h}$ (with configurations defined such as $d \mod h = 0$).

Since standard Transformers are permutation invariant, we implemented optional positional embeddings a trainable linear

[1] $[\![i, j]\!]$ denotes the set of integer between i and j, both included.

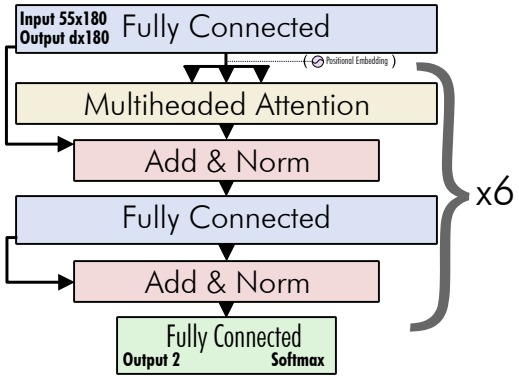

Fig. 1. The transformer baseline architecture.

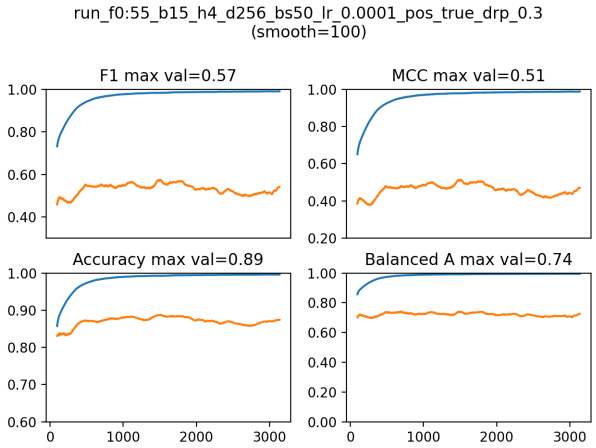

Fig. 2. Evolution of several performance metrics (F_1, MCC, Accuracy and Balanced Accuracy) during training for a specific tested Transformer architecture (blue training set, orange validation set) - values have been smooth averaging the 100 surrounding values of each point. The architecture used all 55 features, 15 blocks, 4 heads, embedding sizes of 256, used positional embedding; training has been made using a batch size of 50 and a learning rate of 10^{-4}.

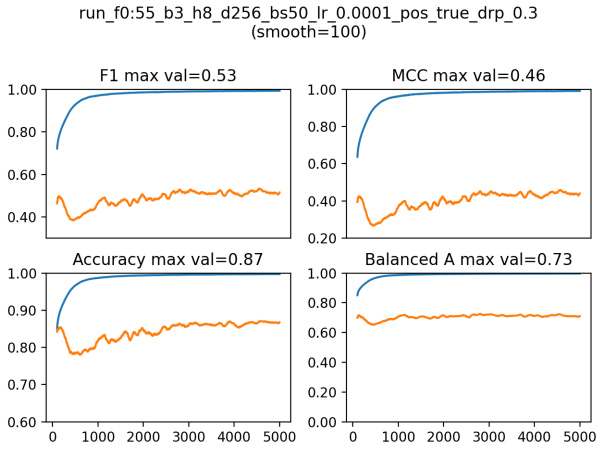

Fig. 3. Similar to Figure 2 with different hyperparameters: 55 features, 3 blocks, 8 heads, embedding sizes of 256, used positional embedding; training has been made using a batch size of 50 and a learning rate of 10^{-4}.

projection applied to each timestep (linear transformation from $[\![1, n]\!]$ to $\mathbb{R}^d$). When we activate positional embeddings, the embedding is summed to the inputs prior to applying the first transformer block.

The final instance embedding has been defined as the average of the output of block $B^{(n)}$ which can be seen as the final timestep embeddings. A simple linear projection is then applied to obtain a $\mathbb{R}^2$ projection on which a softmax is applied to obtain the probability distribution referring to the protective behaviour. Figure 1 illustrates the model.

2) Specific design choices: In order to constrain the architecture which relies on a large number of parameters, we have considered shared trainable parameters among the transformers blocks (i.e. same parameters among each $B^{(i)}, i \in [\![1, n]\!]$). Considering an architecture with n transformer blocks, the encoder part of the model in charge of the processing from the input data (optionally augmented by the positional embeddings) to the timesteps embeddings can therefore be seen has a repeated processing of the same transformer block. In addition, no residual connection (skipping connection) has been used.

3) Training: We added several dropout layers during training to avoid overfitting. The loss function we consider is a weighted Binary Cross entropy (weights computed at batch level), optimized with Adam.

We train several Transformer models based on the aforementioned architecture using different hyperparameters (number of blocks, number of heads, size of internal projections d, using positional embeddings or not, batch size, the learning rate). We then select 21 models that maximizing the sum of the F_1 measure obtained on both given training and validation sets considering all training epochs – these models have been trained up to 4k epochs. F_1 sums of the selected models range from 1.63 to 1.60. Figure 2 and 3 show the results obtained in training two architecture configurations from which some of the 21 final models have been selected (smoothed with 10-point moving average). Note that several selected models share the same hyperparameter configurations but have been obtained at different training epochs.

4) Prediction: We then use a voting strategy over the predictions of the 21 models. For a given test input, the prediction is output maximising the probability of the output class. The final class considered for a given test input is defined using majority voting of all 21 predictions made by the 21 models. Figure 4 presents the distribution of the percentage of votes for majority classes. We observe that the 21 selected models fully agree (same prediction) in 76.4% of the test cases. A majority agreement of less than 90% of the models can be observed in 11.3% of the test cases.

C. LSTM

Here, we used two bi-directional LSTM layers model with batch normalization and dropout. Considering that LSTM type models in the two works [3], [4] showed relatively poor

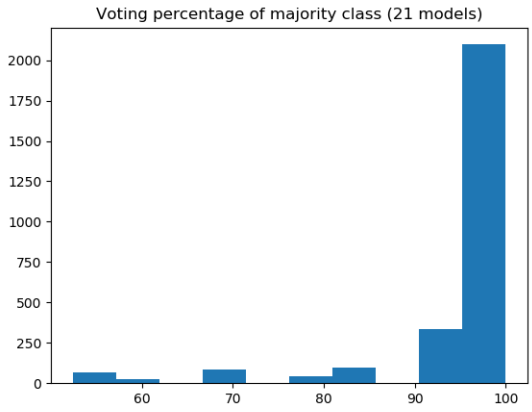

Fig. 4. Distribution of the percentages of votes given to the selected class (majority voting).

performance, we introduced a similar external human action skeleton NTU RGB+D dataset [7] to pre-train this model, then perform fine-tuning with the AffectMove challenge dataset. The NTU RGB+D dataset is one of the largest 3D Human Activity data sets, containing 60 different classes including daily, mutual, and health-related actions. We use only the skeleton sequences dataset extracted from $56,880$ videos samples. Since NTU RGB+D features recordings with 25 body joints instead of the 17 in the EUROPAIN dataset and the AffectMove dataset, only the corresponding 17 joints are kept. The skeleton sequences are also zero-padded or truncated to 180 frames in agreement with the EUROPAIN/AffectMove sequence length.

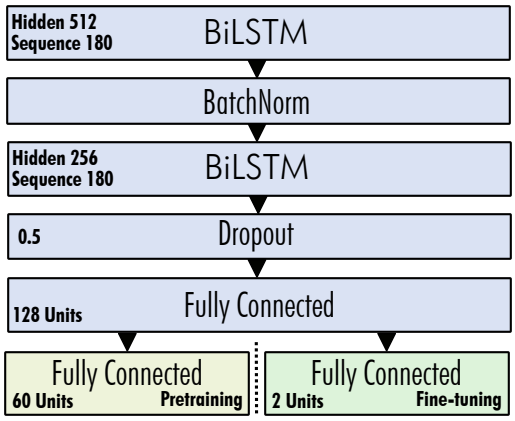

Fig. 5. Baseline BiLSTM model structure for pre-training (left) and fine-tuning on protective action prediction.

Only the MoCap data is used in this baseline model as pre-training couldn't include EMG data. The input of the model is thus 51×180 (17 sequences of 3D-joints). All data is normalized with z-score in which the main mean and standard deviation used for the filtering are estimated from the pre-training set. Since the pre-training dataset is quite large and complex, the number of hidden units are set relatively high (512 units for the 1st layer, 256 units for the 2nd layer). The learning-rate is fixed at 0.01 in the whole training and final testing process. No voting is applied, only the model that gave the best performance on the *validation* set is chosen to predict with unlabeled *test* set. This means that the final model tends to be "*biased*" towards the *validation* set whether there's over-fitting or not, which creates the risk of a collapse of performance under the test set.

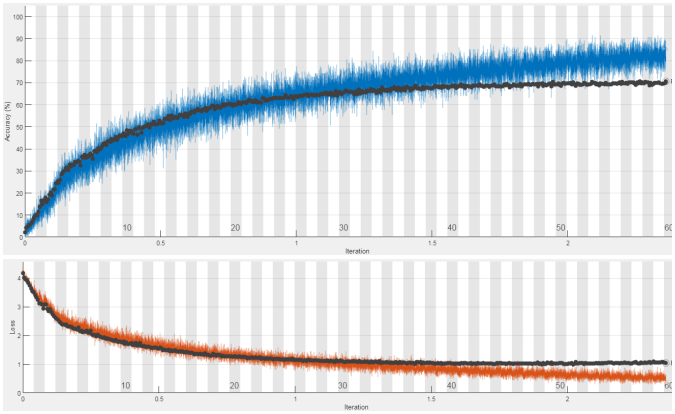

Fig. 6. Pre-training process on NTU RGB+D dataset. NTU RGB+D dataset is randomly split into training 90% and validation 10% sets in which the classes are stratified. The stopping validation accuracy at 57 epochs on the 60 classes is 70.69%.

The pre-training on NTU RGB+D reached a stable validation accuracy 70.69%. For the fine-tuning of the pre-trained model, the best model obtained showed an accuracy of 90%, F_1 at 0.6514 for protective behaviour class. On the validation set, it seemed that this model is interesting in that it appeared to outperform the more complex models (Transformer and GCN), however, as we'll see in the meta analysis, the phenomenon didn't persist in the final evaluation on the test set.

TABLE II
RE-TRAINING RESULTS WITH THE PRE-TRAINED MODEL ON *training* SET, EVALUATED ON *validation* SET (1844 SAMPLES), EARLY STOPPED AT THE END OF 3 EPOCHS

	Metrics	
	Prot. B. 0	**Prot. B. 1**
Specificity	0.6552	0.9413
Precision	0.9430	0.6477
Recall	0.9413	0.6552
F_1	0.9421	0.6514
MCC	0.5936	
Accuracy	0.9008	

D. Graph Convolutional Network

Graph Convolutional Networks for skeleton-based action recognition take into account the spatial and temporal characteristics specific to human skeleton data. Indeed, the human skeleton can be seen as a natural graph, and if we also consider the evolution over time of 3D joint coordinates, skeleton-pose data can be modeled as a temporal graph. Such a representation is efficient with regard to skeletal pose data and also in preserving the most important information and topology.

TABLE III
MODEL SELECTION RESULTS FOR PA-RESGCN. THE FIRST THREE LINES AND THE LAST LINE ARE BASED ON THE PA-RESGCN-N51 MODEL, WHILE LINE FOUR IS BASED ON THE PA-RESGCN-B19.

Model	Params.	EMG	Weighted. Loss	Optimizer	Class	F_1	P	R	Avg. F_1	Weighted Avg. F_1	MCC	Acc.	Balanced Acc.
N51	0.73m	✗	✗	SGD	1	0.54	0.55	0.52	0.73	0.87	0.46	0.87	0.73
					0	**0.93**	0.92	**0.93**					
N51	0.73m	✗	✓	SGD	1	**0.64**	**0.56**	**0.74**	**0.78**	**0.89**	**0.58**	**0.88**	**0.82**
					0	**0.93**	**0.95**	0.90					
N51	0.73m	✗	✓	Adam	1	0.50	0.42	0.63	0.70	0.83	0.41	0.82	0.74
					0	0.89	0.93	0.85					
B19	3.61m	✗	✓	SGD	1	0.56	0.47	0.69	0.73	0.86	0.48	0.84	0.78
					0	0.91	0.94	0.87					
N51 + CNN	0.86m	✓	✓	SGD	1	0.60	**0.56**	0.64	0.75	0.88	0.53	**0.88**	0.78
					0	**0.93**	0.94	0.91					

Architecture like ResGCN [6] also includes part-wise attention, which can be beneficial for protective behaviour detection, as some parts of the body are more significantly involved in the protective behaviour (the area impacted by pain, typically the lower neck or the lower back).

1) Architecture: For the GCN implementation we started from ResGCN in its `PA-ResGCN` variant [6], which encodes three main features extracted from skeleton poses: joints (like most other systems), velocities and bones. The use of these three features allows PA-ResGCN to reach strong SOTA results as highlighted in Section 2. We modify the official implementation of ResGCN [2] by adapting the data loader for the data format of AffectMove Task 1, notably the formulation of the graph for the GCN. The class imbalance in the dataset prevents PA-ResGCN from performing as well as the baseline models, which is why we propose a weighted reformulation of the binary cross-entropy loss that accounts for and compensates the imbalance during training.

Additionally, we propose a simple multimodal extension that performs a late fusion between PA-ResGCN(-N51) and a convolutional neural network for EMG classification.

For the latter, the signals are first filtered with an exponential moving average ($\alpha = 0.2$). The CNN architecture consists of one layer normalization per Batch to adjust scale of input EMG data with learnable parameters, followed by one 2D convolutional layer, batch pooling and two linear layers. PReLU activation is used after the convolutional layers and Tanh in between the two linear layers. This architecture serves to extract 128 EMG features, which are then concatenated to the 256 features extracted by PA-ResGCN-N51. Figure 7 illustrates the full multimodal architecture, please refer to Song et al. [6] for the details of the PA-ResGCN-N51 architecture and part-wise attention.

2) Loss definition: We use Binary cross-entropy with logits along with an added weighing scheme to penalize the majority class. The weights are computed per batch like in Cui et al. [14], where we consider $\beta \to 1$, so that following the paper's notation, $E_n = n_{s,k}$. The loss l_k for the k^{th} batch is defined as :

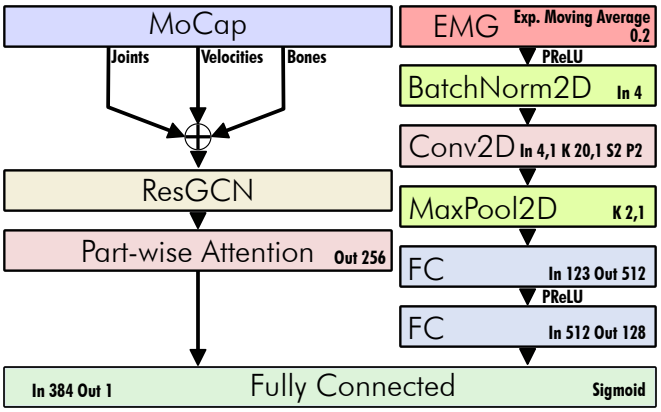

Fig. 7. Architecture of the multimodal PA-ResGCN-N51 + EMG CNN system. The final run only includes the left branch without EMG.

$$l_k = \sum_{(s,y_s)\in B_k} w_s[y_s log\sigma(x_s) + (1-y_s)\cdot log(1-\sigma(x_s))] \quad (1)$$

$$w_s = \frac{1}{n_{s,k}} \qquad n_{s,k} = \begin{cases} n_{-,k} + \epsilon, & \text{if } y_s = 0 \\ n_{+,k} + \epsilon, & \text{otherwise} \end{cases} \quad (2)$$

Where B_k is the set of training examples of the k^{th} batch, w_s is the weight per sample s where $n_{(-,k)}$ and $n_{(+,k)}$ are respectively the number of negative and positive samples present in the k^{th} batch, ϵ was set to 10^{-6} to avoid division by 0 errors.

3) Training & Optimization schedule: First, we use only the 3D MoCap data to train the PA-ResGCN, particularly the PA-ReGCN-N51 variant where N51 means there are 51 convolutional or FC layers within the model and where PA stands for part Attention. We experiment with two types of optimisation approaches: (i) Optimizer based on stochastic gradient descent (SGD) with a Nesterov momentum of 0.9, weight decay of 10^{-4} and with cosine scheduler (warm restarts [16]), similarly to Song et al. [6], (ii) an Adam optimizer with an initial learning rate of 10^{-3}. The SGD optimizer with a cosine scheduler led to better results in comparison with the

[2]https://github.com/yfsong0709/ResGCNv1

Adam optimizer (albeit slightly slower). Secondly, we jointly train the CNN for EMG measures and PA-ResGCN-N51 for 3D skeleton data for training the architecture to include information from EMG.

E. Model selection & validation

The model selection results are presented in Table III. We first evaluated the PA-ResGCN-N51 architecture at 0.77 million parameters in different Loss/optimizer settings, we then evaluated the best performing combination with the PA-ResGCN-B19 model with basic blocks (B19 stands for *19 basic blocks*, 3.61 million parameters) that achieved SOTA on NTU-60 and NTU-120. Finally, we evaluate the multimodal inclusion of EMG to PA-ResGCN-N51 (best performing model). The PA-ResGCN-N51 model configuration with a weighted loss and SGD gives the best performance with an F_1 score 0.64 in comparison with the same model with SGD and an unweighted loss (0.54 F_1, +10%). This improvement demonstrates the benefits introduced by the weighted loss. We can see that unweighted loss generally gives a slightly better precision (compared to the PA-ResGCN-B19 model), with a recall approximately similar to precision, while the weighted loss maximizes the recall and F_1 score. The Adam optimizer (on PA-ResGCN-N51) shows sub-par performance compared to SGD with the cosine scheduler. With the PA-ResGCN-B19 model for the best previous loss/optimizer combination, we observed lower overall performance. Contrarily to NTU-60 or NTU120, which are very large databases, our training data is comparatively smaller for Task 1 and we hypothesize that this model needs a lot more data to actually converge to something better. Likewise, the introduction of the EMG signals to the PA-ResGCN-N51 model degrades the results, as performance deteriorates by -6%. The reason for this decrease could be either a low signal-to-noise ratio in the filtered EMG signal envelope, or the cause of an ill-adapted fusion technique (single fully connected layer).

IV. Results and Discussion

If we consider the final ranking of our submitted runs on the test set of Task 1, unsurprisingly, the best performing system was PA-ResGCN-N51 (w/o EMG) with a F_1 score of 53.36% on the test set for the minority class (MCC 0.4247, Table IV line 3). The Transformer baseline (MoCap + EMG) came second at 41.05% F_1 test performance (MCC 0.3523, Table IV line 2) and the LSTM baseline third at 31.16% F_1 (MCC 0.1763, Table IV, line 1). While the Transformer had a comparable behaviour on the validation set and on the test set, the LSTM was actually on-par with PA-ResGCN-N51 on the validation set. On the test set the performance of the LSTM collapsed with a F_1 of only 0.3116. This seems to comfort the hypothesis that, the distribution of subjects in the *validation* set is quite different from the *training* or test *test* sets. Although, the objective of the task was to provide a realistic setting, the careful selection of a good validation set is paramount to producing models that generalize well regardless of the non-competitive framing. Another reason of the poor *test* performance of the LSTM could be that MoCap-only may be insufficient to predict if one behaviour is protective or not. Even though pre-training has been proved to be useful in many other application, since the *test* set labels are not disclosed yet, one cannot truly evaluate whether pre-training is beneficial or whether it hurts the generalizability of protective behaviour detection. Despite the poor behaviour of the LSTM architecture on the test corpus, pre-training did lead to a significant improvement on the validation set, and we can extrapolate that an architecture like ResNet in its larger form, could have benefited much more from pre-training on other datasets such as NTU, before fine-tuning on Task 1.

TABLE IV
Official results on Task 1 test set for the three submitted runs

System	F_1 C0	F_1 C1	MCC	Acc.
1. LSTM Baseline	85.83	31.16	0.17628	76.49
2. Transformer Baseline	89.96	41.05	0.35234	82.84
3. PA-ResGCN-N51	89.07	53.36	0.42471	82.28

V. Conclusion

In this work, our team has explored two baseline strategies (Transformer and LSTM) and a state of the art approach (PA-ResGCN) for the skeleton + EMG based protective behaviour prediction for Task 1 of the AffectMove challenge. The SOTA architecture of PA-ResGCN did indeed achieve the best performance both on the validation set and test set of the challenge compared to the baseline approaches without using the EMG signal. Although the fusion strategy we implemented to add EMG to PA-ResGCN-N51 was unsuccessful, it was trivial in nature due to the time constraints. More state of the art multi-modal fusion strategies may prove successful at integrating EMG and leading to increased performance (as observed with the Transformer). The effectiveness of PA-ResGCN remains relative, as the amount of training data for task 1 is on the smaller end of the scale. A more thorough error-analysis will allow for specific adaptations to PA-ResGCN that would lead to better results.

References

[1] T. Olugbade, R. Sagoleo, S. Chisio, N. Gold, A. C. C de Williams, B. de Gelder, A. Camurri, V. Gualtiero, and N. Bianchi-Berthouze, "The Affectmove 2021 Challenge - Affect Recognition from Naturalistic Movement Data," in *Proceedings of 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, 2021.

[2] M. S. H. Aung, S. Kaltwang, B. Romera-Paredes, B. Martinez, A. Singh, M. Cella, M. Valstar, H. Meng, A. Kemp, M. Shafizadeh, A. C. Elkins, N. Kanakam, A. de Rothschild, N. Tyler, P. J. Watson, A. C. de C Williams, M. Pantic, and N. Bianchi-Berthouze, "The Automatic Detection of Chronic Pain-Related Expression: Requirements, Challenges and the Multimodal EmoPain Dataset," *IEEE transactions on affective computing*, vol. 7, no. 4, pp. 435–451, 2016. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/30906508https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6430129/

[3] J. O. Egede, S. Song, T. A. Olugbade, C. Wang, A. Williams, H. Meng, M. Aung, N. D. Lane, M. Valstar, and N. Bianchi-Berthouze, "EMOPAIN challenge 2020: Multimodal pain evaluation from facial and bodily expressions," *arXiv preprint arXiv:2001.07739v3*, 2020. [Online]. Available: https://arxiv.org/abs/2001.07739v3

[4] C. Wang, T. A. Olugbade, A. Mathur, A. C. De C. Williams, N. D. Lane, and N. Bianchi-Berthouze, "Recurrent network based automatic detection of chronic pain protective behavior using mocap and semg data," in *Proceedings of the 23rd International Symposium on Wearable Computers*, ser. ISWC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 225–230. [Online]. Available: https://doi.org/10.1145/3341163.3347728

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[6] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1625–1633. [Online]. Available: https://doi.org/10.1145/3394171.3413802

[7] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1010–1019.

[8] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 816–833.

[9] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[10] M. Petrovich, M. J. Black, and G. Varol, "Action-conditioned 3d human motion synthesis with transformer vae," 2021.

[11] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," *Computer Vision and Image Understanding*, vol. 208-209, p. 103219, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314221000631

[12] C. Wang, Y. Gao, A. Mathur, A. C. De C. Williams, N. D. Lane, and N. Bianchi-Berthouze, "Leveraging activity recognition to enable protective behavior detection in continuous data," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 2, Jun. 2021. [Online]. Available: https://doi.org/10.1145/3463508

[13] C. Bao, Z. Fountas, T. Olugbade, and N. Bianchi-Berthouze, "Multimodal data fusion based on the global workspace theory," in *Proceedings of the 2020 International Conference on Multimodal Interaction*, ser. ICMI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 414–422. [Online]. Available: https://doi.org/10.1145/3382507.3418849

[14] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9260–9269.

[15] K. Abhishek and G. Hamarneh, "Matthews correlation coefficient loss for deep convolutional networks: Application to skin lesion segmentation," in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 2021, pp. 225–229.

[16] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *arXiv preprint cs.LG 1608.03983*, 2017.