# 3D Pose Based Feedback For Physical Exercises Supplementary Material

## 1 Supplementary Material

### 1.1 Video

The video, which can be accessed on our project page https://senakicir.github.io/projects/exercise_feedback gives an overview of our paper by introducing our problem, explaining our methodology and presenting our results.

### 1.2 Further Ablation Studies

**Ablation study on TMP module.** The architecture of our classification branch is highly inspired by the frame-level module architecture proposed by Zhang *et al.* (SGN) [1]. The set-up used by SGN is a spatial MaxPooling (SMP) layer, followed by two convolutional layers and a temporal MaxPooling layer (TMP). Our classification branch omits the TMP layer, because we have found it to not be useful for our case. We suspect that the reason for this is we do not have temporal data as input to our networks, instead we have DCT coefficients. In Table 1 we present an ablation study on using a TMP layer. The results obtained using a TMP layer are worse in terms of both classification accuracy and correction success.

| | Classification Accuracy(%) | Classification Loss | Correction Accuracy(%) | Correction Loss |
|---|---|---|---|---|
| Ours with TMP | 76.1 | 1.62 | 91.5 | 3.35 |
| Ours (w/o TMP) | **90.9** | 1.30 | **94.2** | 1.45 |

**Table 1. Ablation study on the TMP layer.** We show the results of our framework trained using a TMP module and without using a TMP module (ours). We note that the TMP module only plays a role in the classification branch, however this still affects the performance of correction also, as the two branches are trained as part of a single network. We find that the results are worse for both classification accuracy and correction success when the TMP module is included.

**Ablation study on the smoothness loss.** We have trained our network with different weights for the smoothness loss term, denoted as $w_{\text{smooth}}$. The results are reported in Table 2. We find that $w_{\text{smooth}} = 1e-3$ gives the best results in terms of the correction success. However the results obtained when $w_{\text{smooth}} = 0$ gives slightly better results for classification accuracy.

| | Classification Accuracy(%) | Correction Success(%) |
|---|---|---|
| $w_{\text{smooth}} = 1e-1$ | 81.8 | 72.7 |
| $w_{\text{smooth}} = 1e-2$ | 81.8 | 85.9 |
| $w_{\text{smooth}} = 1e-3$ **(Ours)** | 90.9 | **94.2** |
| $w_{\text{smooth}} = 1e-5$ | 81.8 | 85.0 |
| $w_{\text{smooth}} = 0$ | **93.4** | 87.5 |

**Table 2. Ablation study on the smoothness loss.** We present the results of our network when trained with different weights for the smoothness loss. We find that setting the weight of the smoothness loss to $1e-3$ gives the best results in terms correction success. The classification accuracy is higher when smoothness loss is not used.

### 1.3 Classification on the NTU RGB+D Dataset

**NTU RGB+D Dataset.** We use the NTU RGB+D [2], a widely-used dataset for evaluating the performance of action classification networks [2,3,1]. We use cross-subject division to split the training set and test set according to the person ID. A total of $40$ subjects were divided into a training set of $17$ subjects, a validation set of $3$ subjects and a test set of $20$ subjects.

NTU RGB+D contains $56,880$ action samples. As stated in the official dataset release, 302 samples in the dataset dataset have missing or incomplete skeleton data. In addition, some of the actions involve two people interacting with each other, which do not match the expected input to the model, and the video frame lengths are not uniform across sequences. To overcome these challenges, we have pre-processed this dataset.

In order to remove noisy data, for the missing or incomplete skeleton data action sequences, we used the official list of missing data indices. As they only represent $0.53\%$ of the total data volume, these lossy data are directly removed from the dataset.

For some of the sequences, there are two-subjects in a single frame. For such sequences we consider the pose coordinates of only one of them. Since the two people motions involve movements which are mirror-symmetrical (e.g. A55 Hugging, A59 Walking towards, etc...), we find this method to be sufficient.

Different motion sequences can have different lengths. In order to counter different video frame rates, SGN tries to segment the entire skeleton sequence into 20 clips equally, and randomly select one frame from each clip to have a new sequence of 20 frames. However, in order to maintain consistency between the NTU dataset and the EC3D data, we modify the inputs when feeding our data to our model by finding their top $25$ DCT coefficients. By doing so, for all input sequences we have $25$ DCT coeffiecients and do not have to worry about the differences in sequence lengths. We then normalize, centralize and rotate the dataset as we have done with EC3D.

**Classification results on NTU RGB+D Dataset.** We present our classification branch's results on the NTU RGB+D dataset in Table 3 to compare our model's performance to SOTA action classification methods. Although we do not achieve SOTA performance, our performance is comparable to the results of many of these methods, showing that it is a reliable, lightweight method for action classification on other mainstream datasets. We note that our goal is not to achieve SOTA action classification,

but to achieve high enough accuracy to give reliable feedback and to also evaluate the results of the action correction branch.

| Methods | Classification Accuracy (%) |
|---|---|
| HBRNN-L [4] | 59.1 |
| Part-Aware LSTM [2] | 62.9 |
| ST-LSTM + Trust Gate [5] | 69.2 |
| STA-LSTM [6] | 73.4 |
| GCA-LSTM [7] | 74.4 |
| DPRL+GCNN [8] | 83.5 |
| HCN [3] | 86.5 |
| AS-GCN [9] | 86.8 |
| VA-CNN [10] | 88.7 |
| SGN [1] | **89.0** |
| Ours | 70.1 |

**Table 3. The classification branch's results on the NTU RGB+D dataset.** While we do not achieve SOTA performance, we outperform HBRNN-L [4], Part-Aware LSTM [2], and ST-LSTM + Trust Gate [5] . Our results are comparable to those of STA-LSTM [6] and GCA-LSTM [7]. We conclude that our network is able to achieve acceptable performance on larger, more mainstream datasets as well.

### 1.4 Further Qualitative Results

We have evaluated the behaviour of our framework on inputs that are already correct and we present qualitative results in Figure 1. We find that since the input sequences are already correct, the framework's adjustments are very minor.

### 1.5 Detailed Quantitative Results

We present detailed quantitative results of our model in Tables 4 and 5. Table 4 presents a confusion matrix of the results of our classification branch. We can clearly see which actions are confused with which ones. We find that the "correct squat" is once confused with a "front bent squat" and the "correct plank" is confused once with a "hunch back" plank. It is surprising to see that the "front bent squat" is confused with mistakes from other types of exercises, namely a "not low enough lunge" and "correct plank". This is a clear failure case and improvements to the model should focus on eliminating such mistakes.

Table 5 presents the results of the correction branch. We can see how the results of the correction branch are classified by the classification branch. For instance, for the category of "not low enough lunge" sequences, 6 of them are successfully corrected whereas 4 are still classified as incorrect, giving a 60% correction success for this instruction.

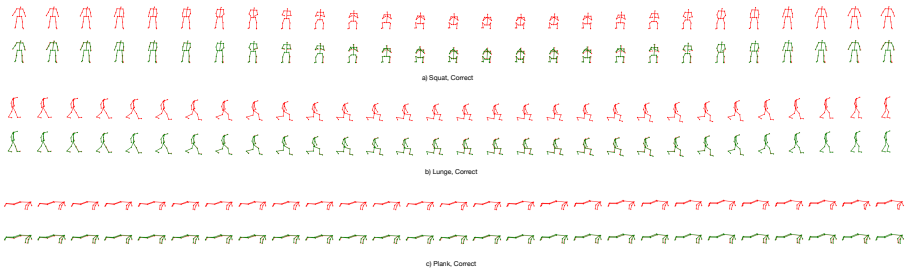


**Fig. 1. Qualitative results from our framework of correcting already "correct" motions.** We input motions and corrected output motions from categories a) squats, b) lunges, c) planks. We present the input sequences (red) in the top row. The corrected sequences (green) overlaid on top of the input sequences (red) are presented in the bottom row. We find that since the input sequences are already correct, the adjustments made by our framework are very minor. This figure is best viewed in color and zoomed in on a screen.

| Action /Instruction | | Confusion Matrix | | | | | | | | | | | Accuracy(%) | Average(%) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Squats | Correct | **9** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 90.0 | 89.4 | 90.9 |
| | Feet too wide | 0 | **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | | |
| | Knees inward | 0 | 0 | **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | | |
| | Not low enough | 0 | 0 | 0 | **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | | |
| | Front bent | 0 | 0 | 1 | 0 | **4** | 0 | 1 | 0 | 1 | 0 | 0 | 57.1 | | |
| Lunges | Correct | 0 | 0 | 0 | 0 | 0 | **8** | 0 | 4 | 0 | 0 | 0 | 66.7 | 88.9 | |
| | Not low enough | 0 | 0 | 0 | 0 | 0 | 0 | **10** | 0 | 0 | 0 | 0 | 100 | | |
| | Knee passes toe | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **10** | 0 | 0 | 0 | 100 | | |
| Planks | Correct | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **6** | 0 | 1 | 85.7 | 95.2 | |
| | Arched back | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **9** | 0 | 100 | | |
| | Hunch back | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **9** | 100 | | |

**Table 4. Detailed classification results for each exercise instruction category,** in the form of a confusion matrix.

| Action | Instruction | Correct | Incorrect | Successfully Corrected(%) | | Average(%) |
|---|---|---|---|---|---|---|
| Squats | Correct | 10 | 0 | 100 | 97.1 | 94.2 |
| | Feet too wide | 5 | 0 | 100 | | |
| | Knees inward | 5 | 0 | 100 | | |
| | Not low enough | 4 | 0 | 100 | | |
| | Front bent | 6 | 1 | 85.7 | | |
| Lunges | Correct | 12 | 0 | 100 | 83.3 | |
| | Not low enough | 6 | 4 | 60 | | |
| | Knee passes toe | 9 | 1 | 90 | | |
| Planks | Correct | 7 | 0 | 100 | 100 | |
| | Arched back | 9 | 0 | 100 | | |
| | Hunch back | 9 | 0 | 100 | | |

**Table 5. Detailed correction results on each exercise instruction category.** We depict how many output sequences are classified as "correct" and "incorrect". The "incorrect" class in this table is a grouping of all instructions that are not correct.

## References


1. Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., Zheng, N.: Semantics-guided neural networks for efficient skeleton-based human action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2020)
2. Shahroudy, A., Liu, J., Ng, T., Wang, G.: Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: Conference on Computer Vision and Pattern Recognition. (2016) 1010–1019
3. Li, Y., Yuan, L., Vasconcelos, N.: Co-Occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation. In: International Joint Conference on Artificial Intelligence. (2018)
4. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Conference on Computer Vision and Pattern Recognition. (2015)
5. Liu, J., a. Shahroudy, Xu, D., Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition. Volume 9907. (2016)
6. Song, S., Lan, C., Xing, J., Zeng, W., Liu, J.: An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In: AAAI Conference on Artificial Intelligence. (2017)
7. Liu, J., Wang, G., Hu, P., Duan, L., Kot, A.: Global context-aware attention lstm networks for 3d action recognition. In: Conference on Computer Vision and Pattern Recognition. (2017)
8. Tang, Y., Tian, Y., Lu, J., Li, P., Zhou, J.: Deep progressive reinforcement learning for skeleton-based action recognition. In: Conference on Computer Vision and Pattern Recognition. (2018)
9. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: Conference on Computer Vision and Pattern Recognition. (2019)
10. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive neural networks for high performance skeleton-based human action recognition. (2019)