Context-based Interpretable Spatio-Temporal Graph Convolutional Network for Human Motion Forecasting
Supplementary Material

1. Additional Experiments

1.1. Experimental results

More visualizations of applying the model CIST-GCN in pose sequences from H3.6M are shown in Fig. 7, the input (from 0 to 9 frames) and output poses are split by a vertical red line. Classes presented here are, "sitting-down", "purchases", "directions", "discussion", "posing", and "smoking" respectively. To quantify these movements, we used the relative angle variation to show cycles in pose motion which were calculated for all motion classes. We calculate the angle for a single joint using data from 2 frames, consistently keeping frame 0 as a reference. The equation is presented in Eq. 1. As we can see, some actions have "spontaneous movements" (defined in the main paper) while others are cyclic movements.

$$\vec{\theta} = \frac{\vec{x}_0 \cdot \vec{x}_t}{|\vec{x}_0||\vec{x}_t|} \tag{1}$$

Additionally, the comparison of architectures mentioned in the paper is complemented with Fig. 5. This shows again that our model is lightweight compared to state-of-the-art architectures while still having a comparable performance in the MPJPE metric.

2. Interpretability results

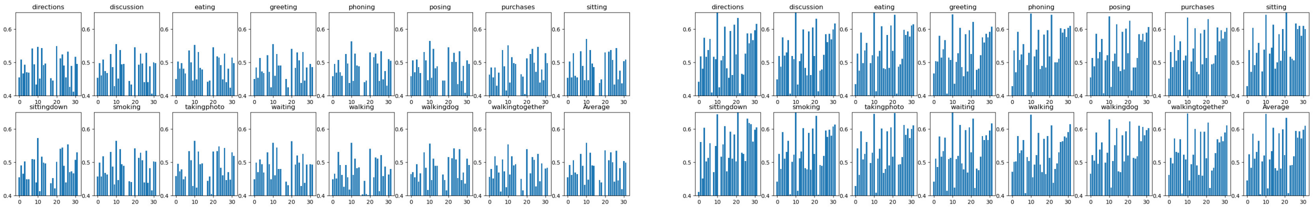
2.1. Feature importance vectors

Complementary to the visual t-SNE representation presented in the main paper, Fig. 3 shows input and output displacement representations obtained from the model and reduced to only 2 dimensions via t-SNE. Fig. 3a represents lesser cluster-like visualization, but Fig. 3b resulted in a better representation comparable to the interpretable variables from the model.

In order to enhance the clarification, we conducted a similar experiment as in GAGCN i.e. we plot the internal average weights for H3.6M obtained by the gating network from the input and output DST-GCN in Fig. 1 and 2 respectively. We also plot all motion classes and use all samples similar to the main paper but excluding MPJPE weighting. We observe weights are visually separated for similar action classes, which also aligns with the t-SNE plot using all weighting vectors (mentioned in the main paper). In both plots, we observe a similar pattern for similar actions for both input and output sequences. This reflects, that the model suitably fits enough to obtain a certain accuracy level although small differences were found in similar input sequences too. Also, we understand that both architectures are not directly comparable making this comparison slightly unfair.

2.2. Feature Maps

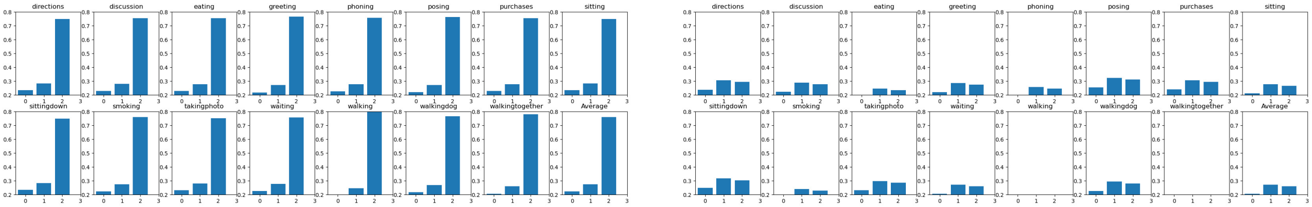
In Fig. 4, we show detailed per-layer average activation maps of the temporal and spatial adjacency matrices (acting as relation matrices) for the samples depicted in Fig. 2 from the main paper. As we see, normalization is from 0 to 1 and is located on the right while action categories are located on the left. "Walking" actions (first two rows) are more similar to the hardest case of the "eating" action (last row) than the easiest case (third row). This is because the hardest case of "eating" also included "walking" motion, similar results were found for similar actions when these are cyclic. This implies that feature maps contain more details to deduce the type of motion which can often be informative when actions cover at least two classes of motion. However, comprehending how the input displacements are transformed into output displacements is challenging, impeding the association of specific movements to particular interpretations. In order to present more dataset cases, we present in Fig. 6 the average of the input relation maps alongside the output relation maps for different action categories with their respective variation of the angle-based movement. We observe the following behavior: to begin with, the values in both rows and columns display sparsity, hindering an exhaustive analysis even within the output map. Secondly, due to the model having learned the sequence displacements, the values within the "dsgn-in" layers undergo changes between each layer as intermediate displacements are translated into output displacements. Also, understanding the "dsgn-out" matrices is challenging due to their composition of displace-



(a)

(b)

Figure 1. Normalized average gating weights from the input DST-GCN for (a) spatial and (b) temporal domains.



(a)

(b)

Figure 2. Normalized average gating weights from the output DST-GCN for (a) spatial and (b) temporal domains.

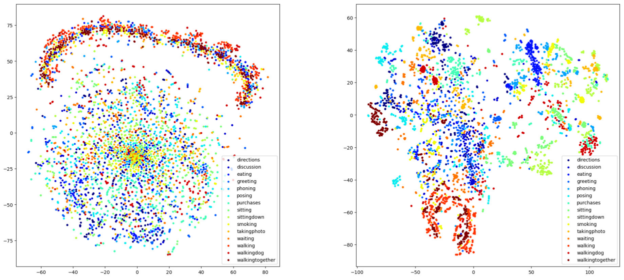


Figure 3. t-sne representation of the test set using (a) input and (b) output displacements. MPJPE values are represented by scatter size.

ment relations spanning the 25 output frames. However, there are discernible patterns, particularly in cyclical or simply stationary movements. This helps in cases where the movements contain more predictable patterns, and in situations where the prediction contains confusing patterns these feature maps can offer a certain level of uncertainty that can be used to question the prediction of the model. This is observed in situations where the poses contain no relevant information in the first nine frames and only the last frame begins to have a large variation in movement.

Furthermore, we observed that interpreting feature maps becomes more challenging, and may deviate from typical patterns when MPJPE values are much larger than the test set average. More experiments are required to formulate a hypothesis to explain why the interpretation layers are inconsistent when MPJPE values are very large.

Table 1. Average MPJPE over all frames on Human3.6M dataset is computed for the architecture $M32$. The architecture M32 mentioned in the paper obtained 66.7 global average MPJPE.

Model	augs	vels&accs	APTCN	DGCN	GN	CN	MPJPE
M32	✓	✓					68.0
			✓	✓	✓		72.2
			✓	✓			72.4
	✓		✓	✓			69.4
	✓		✓	✓	✓	✓	69.3
		✓			✓		68.2
		✓		✓	✓	✓	68.6
		✓	✓		✓		67.8
		✓	✓	✓	✓	✓	68.4
	✓	✓	✓		✓		67.9
	✓	✓		✓	✓	✓	67.5
	✓	✓			✓		67.4
	✓	✓	✓				67.4
	✓	✓	✓		✓	✓	66.9
	✓	✓	✓	✓	✓		66.9
	✓	✓	✓	✓	✓	✓	**66.7**

3. Implementation details

During training, we set 50 epochs for training except in ExPI where we set 100 epochs. Also, we set an initial learning rate of 0.01, a Warm-Up of 100 iterations, a learning rate schedule of 0.8 every 3000, 10000, and 1100 iterations for H3.6M, AMASS, and ExPI respectively, batch sizes of 128, 256, and 32 for H3.6M, AMASS and ExPI, dropout of 0.1. Furthermore, data augmentation was used as a regularization method using mainly 3D transformations such as mirror and rotation on 2 axes and small translation and

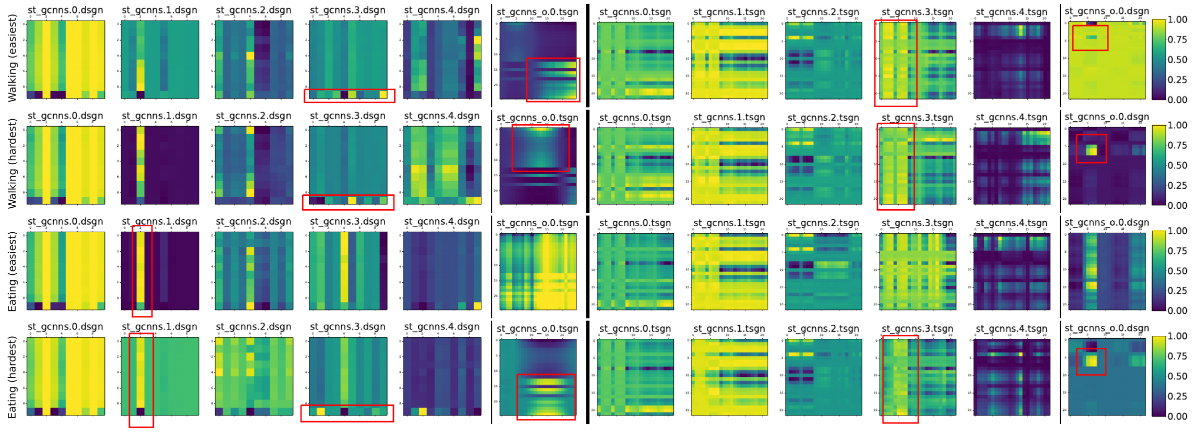


Figure 4. Normalized (0-1) and per-layer average adjacency matrices extracted from the CIST-GCN architecture in the spatial (left) and temporal (right) domains. The last section from both sides has the output adjacency matrices. The interpretations shown belong to the model fed with the samples depicted in Fig. 2 from the main paper.

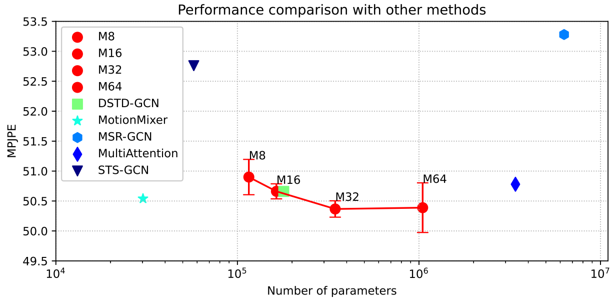


Figure 5. CIST-GCN is compared in performance vs number of parameters with state-of-the-art architectures.

scaling on the 3-axis. Nonetheless, it was observed that similar performance was achieved also with 30 epochs and a learning rate decay of 0.1 every 10 epochs. For ExPI, 50 epochs were enough for comparable performance. It is worth mentioning that input frames used for the three standard datasets were set to 10 whereas the ExPI dataset used 50 input frames as the standard experiment settings. We use Conv+BN+PReLU blocks in the whole architecture. More details are shown on our publicly available code [1].

4. Ablation studies

We also performed experiments to show the importance of each learning feature added to the training or architecture. In Tab. 1 the performance effect of the addition of every feature in our architecture and learning process is shown. We analyzed the addition of augmentations (augs) as a training strategy, and velocities (vels), and accelerations (accs) as feature aggregation. The addition of Dynamic GCN (DGCN), CN, GN, and APTCN is considered in the architecture exploration. For simplicity, the global average of MPJPE over all 25 output frames is used. Firstly, we can clearly see the effect of augmentation in the MPJPE metric. Secondly, individual modules do not provide a large improvement however, we do obtain interpretability from every module we added. Finally, extra feature aggregation added a small improvement, as supported by other works and as explained in our work.

[1] available code: qualityminds.cistgcn

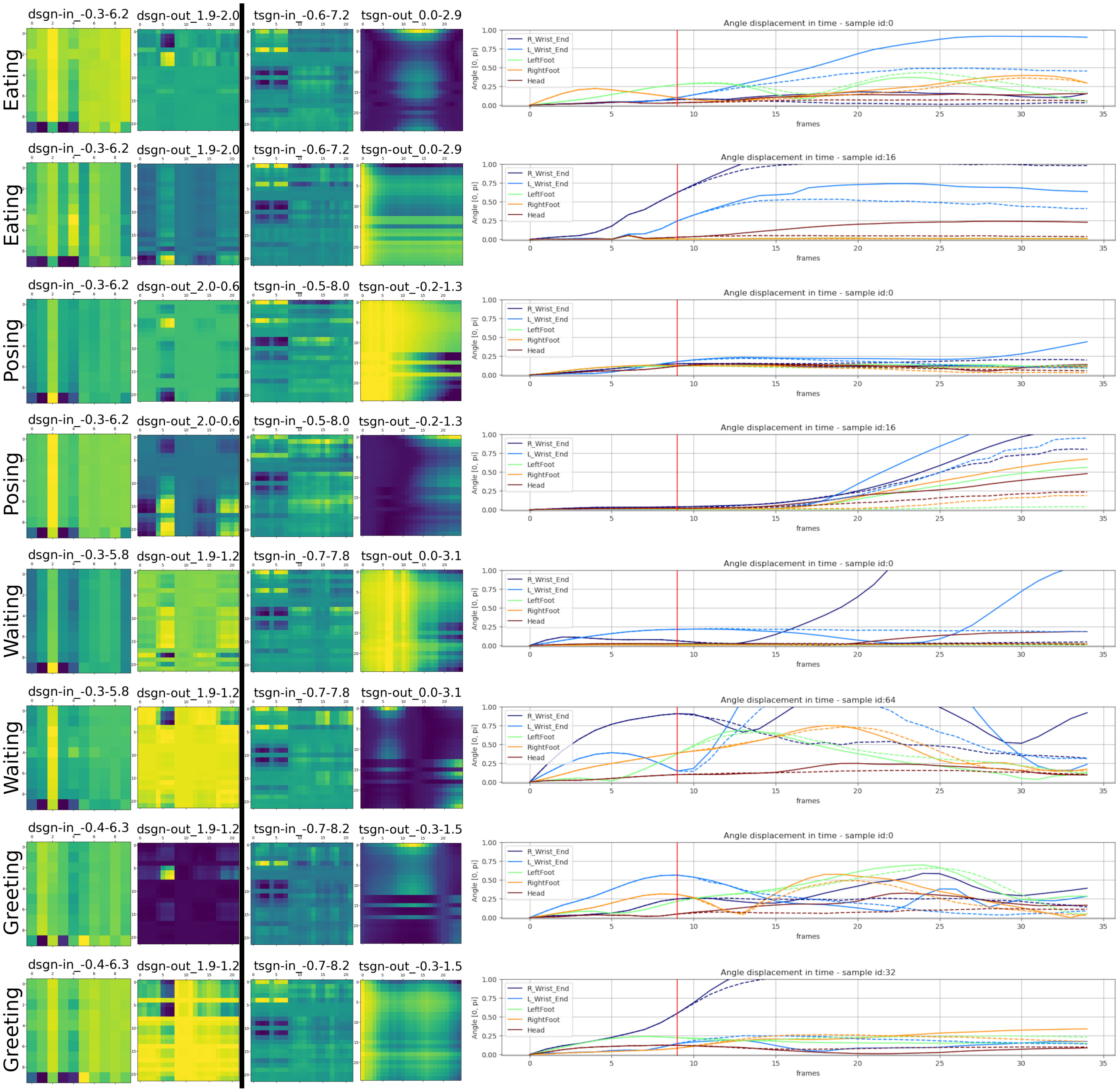
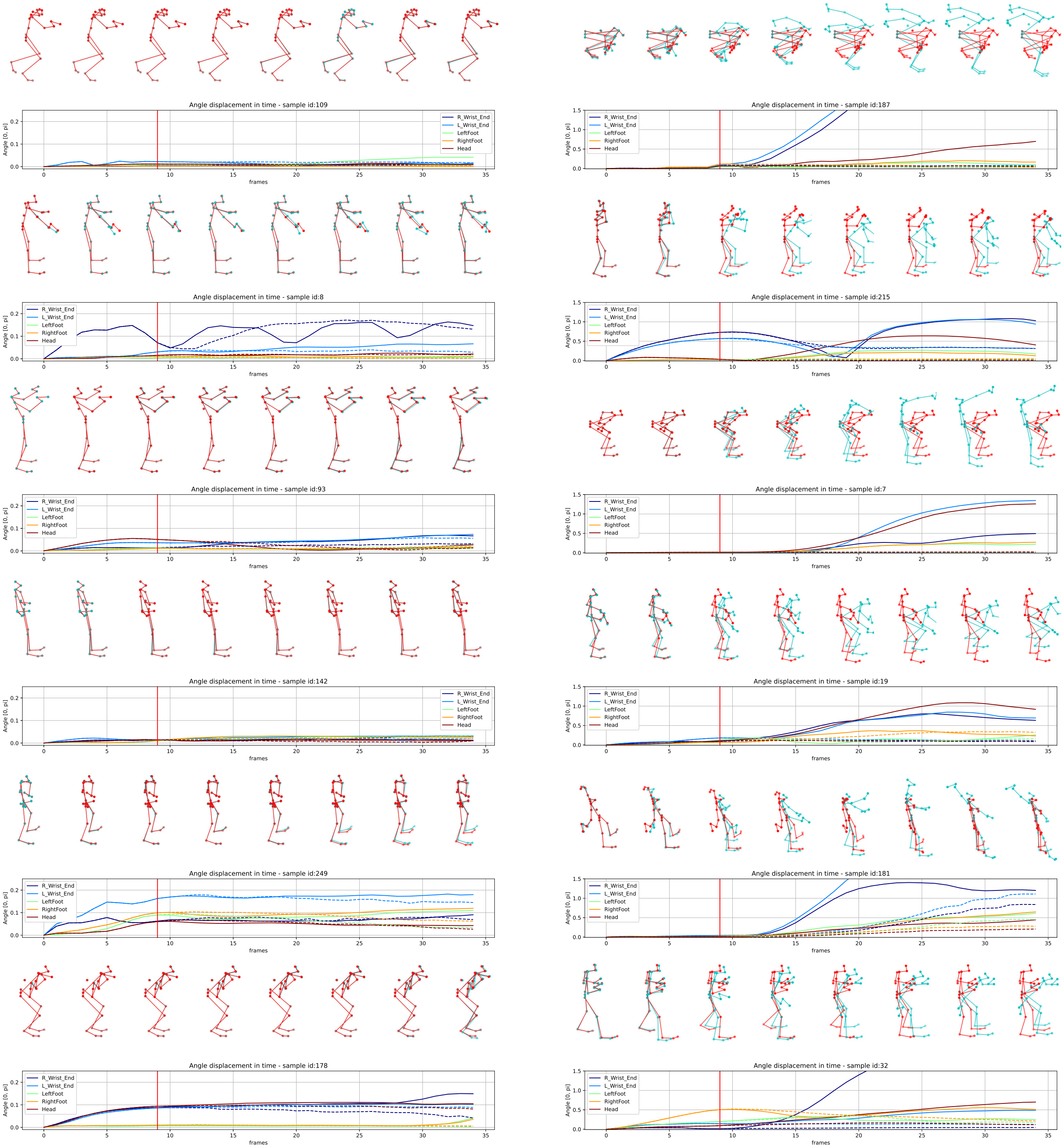


Figure 6. Normalized (0-1) feature maps for different motion categories. (right) input and output feature maps for spatial and temporal domains. (left) Relative angle variations. Yellow means 1 and blue means 0



(a)

(b)

Figure 7. Motion prediction results on several motion classes from the H3.6M dataset. Sorted by (a) the lowest and (b) the largest errors. Solid lines are ground truth. Dashed lines are predictions from the $M32$ model. The input (from 0 to 9 frames) and output poses are split by a vertical red line. Blue color of the poses represents ground truth while the red color of the poses represents the predicted ones.