Supplemental Materials for Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition

This supplemental materials include details about formula derivation, architecture setting, more visualizations and other ablation studies. Specifically, we give the derivation from Equation 12 to 13 and from Equation 8 to 14. Then we show the detailed architecture of CTR-GCN, including input size, output size and specific hyperparameters of each block. Moreover, we visualize shared topologies and channel-specific correlations. At last, we conduct ablation studies on the effect of CTR-GC's number per block, temporal convolution, and analyze the performance of different graph convolutions on hard classes.

1. Formula Derivation

We first give the derivation from Equation 12 to 13. The Equation 12 is

$$\mathbf{z_i^k} = \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{p_{ij}} \odot (\mathbf{x_j^k W}), \tag{1}$$

where $\mathbf{z_i^k} \in \mathbb{R}^{1\times C'}$ is the output feature of v_i and $\mathbf{p_{ij}} \in \mathbb{R}^{1\times C'}$ is the channel-wise relationship between v_i and v_j. $\mathbf{x_j^k} \in \mathbb{R}^{1\times C}$ is the input feature of v_j and $\mathbf{W} \in \mathbb{R}^{C\times C'}$ is weight matrix. The c-th element of $\mathbf{z_i^k}$ is formulated as

$$\begin{aligned} z_{ic}^k &= \sum_{v_j \in \mathcal{N}(v_i)} p_{ijc}(\mathbf{x_j^k W})_\mathbf{c} = \sum_{v_j \in \mathcal{N}(v_i)} p_{ijc}(\mathbf{x_j^k w_{:,c}}) \\ &= \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{x_j^k}(p_{ijc}\mathbf{w_{:,c}}), \end{aligned} \tag{2}$$

where p_{ijc} is the c-th element of $\mathbf{p_{ij}}$. $(\mathbf{x_j^k W})_\mathbf{c} \in \mathbb{R}^1$ is the c-th element of $\mathbf{x_j^k W}$ and $\mathbf{w_{:,c}} \in \mathbb{R}^{C\times 1}$ is the c-th column of $\mathbf{W}$. Therefore, $\mathbf{z_i^k}$ can be formulated as

$$\begin{aligned} \mathbf{z_i^k} &= \begin{bmatrix} \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{x_j^k}(p_{ij1}\mathbf{w_{:,1}}) \\ \vdots \\ \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{x_j^k}(p_{ijC'}\mathbf{w_{:,C'}}) \end{bmatrix}^T \\ &= \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{x_j^k}([p_{ij1}\mathbf{w_{:,1}}, \cdots, p_{ijC'}\mathbf{w_{:,C'}}]), \end{aligned} \tag{3}$$

which is the same as Equation 13.

Then we give the derivation from Equation 8 to 14. We add sample index $\mathbf{k}$ in Equation 8, which is formulated as

$$\mathbf{Z^k} = [\mathbf{R_1^k \tilde{x}_{:,1}^k} || \mathbf{R_2^k \tilde{x}_{:,2}^k} || \cdots || \mathbf{R_{C'}^k \tilde{x}_{:,C'}^k}]. \tag{4}$$

The c-th column of $\mathbf{Z^k} \in \mathbb{R}^{N\times C'}$ can be formulated as

$$\mathbf{z_{:,c}^k} = \mathbf{R_c^k \tilde{x}_{:,c}^k} = \mathbf{R_c^k}(\mathbf{X^k W})_{:,\mathbf{c}} = \mathbf{R_c^k}(\mathbf{X^k w_{:,c}}), \tag{5}$$

where $\mathbf{X^k} \in \mathbb{R}^{N\times C}$ is the input feature. The i-th element of $\mathbf{z_{:,c}^k}$, *i.e.*, the c-th element of v_i's output feature is

$$\begin{aligned} z_{ic}^k &= \mathbf{r_{i,:,c}^k}(\mathbf{X^k w_{:,c}}) = \sum_{v_j \in \mathcal{N}(v_i)} r_{ijc}^k(\mathbf{x_j^k w_{:,c}}) \\ &= \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{x_j^k}(r_{ijc}^k \mathbf{w_{:,c}}), \end{aligned} \tag{6}$$

where $\mathbf{r_{i,:,c}^k} \in \mathbb{R}^{1\times N}$ is the i-th row of $\mathbf{R_c^k} \in \mathbb{R}^{N\times N}$. It can be seen that Equation 6 has the similar form with Equation 2. Thus Equation 6 can be reformulated to the similar form with Equation 3, which is

$$\mathbf{z_i^k} = \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{x_j^k}([r_{ij1}^k \mathbf{w_{:,1}}, \cdots, r_{ijC'}^k \mathbf{w_{:,C'}}]). \tag{7}$$

It can be seen that Equation 7 is the same as Equation 14, *i.e.*, Equation 8 can be reformulated to Equation 14.

2. Detailed Architecture

The detailed architecture of the proposed CTR-GCN is shown in Table 1, CTR-GCN contains ten basic blocks and a classification layer which consists of a global average pooling, a fully connected layer and a softmax operation. M refers to the number of people in the sequences, which is set to 2, 2, and 1 for NTU RGB+D, NTU RGB+D 120, and NW-UCLA respectively. In a sequence, M skeleton sequences are processed independently by ten basic blocks and are average pooled by the classification layer to obtain the final score. T and N refer to the length and the number of joints of input skeleton sequences, which are {64, 25}, {64, 25} and {52, 20} for NTU-RGB+D, NTU-RGB+D 120, and NW-UCLA respectively. C is the basic channel number which is set to 64 for CTR-GCN. "SM" and "TM" indicate the spatial modeling module and temporal modeling module respectively. The two numbers after SM are the input channel and output channel of SM. The three numbers after TM are the input channel, output channel and temporal stride. At the Basic Blocks 5 and 8, the strides of convolutions in temporal modeling module (TM) are set to 2 to reduce the temporal dimension by half. n_c is the number of action classes, which is 60, 120, 10 for NTU-RGB+D, NTU-RGB+D120, and NW-UCLA respectively.

3. Visualization

As shown in Figure 1, we visualize the shared topologies and channel-specific correlations of our CTR-GCN. The input sample belongs to "typing on a keyboard". It can be seen that (1) the shared topologies in three layers

Layers	Output Sizes	Hyperparameters
Basic Block 1	M×T×N	SM: 3, C TM: C, C, 1
Basic Block 2	M×T×N	SM: C, C TM: C, C, 1
Basic Block 3	M×T×N	SM: C, C TM: C, C, 1
Basic Block 4	M×T×N	SM: C, C TM: C, C, 1
Basic Block 5	M×$\frac{T}{2}$×N	SM: C, 2C TM: 2C, 2C, 2
Basic Block 6	M×$\frac{T}{2}$×N	SM: 2C, 2C TM: 2C, 2C, 1
Basic Block 7	M×$\frac{T}{2}$×N	SM: 2C, 2C TM: 2C, 2C, 1
Basic Block 8	M×$\frac{T}{4}$×N	SM: 2C, 4C TM: 4C, 4C, 2
Basic Block 9	M×$\frac{T}{4}$×N	SM: 4C, 4C TM: 4C, 4C, 1
Basic Block 10	M×$\frac{T}{4}$×N	SM: 4C, 4C TM: 4C, 4C, 1
Classification	1×1×1	global averge pool n_c-d fc softmax

Table 1. Detailed architecture of CTR-GCN. M, T, and N refer to the number of people, the length, and the number of joints of input sequences. "SM" and "TM" indicate the spatial modeling module and temporal modeling module respectively. The two numbers after SM are the input channel and output channel of SM. The three numbers after TM are the input channel, output channel and temporal stride. n_c is the number of action classes.

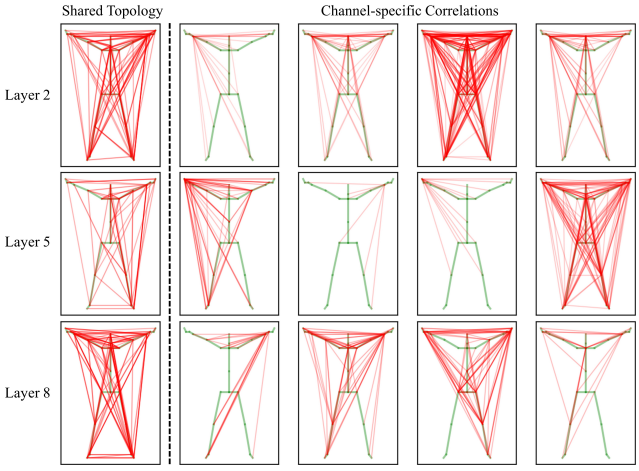

Figure 1. Visualization of the shared topologies and channel-specific correlations. The green lines show the natural connections of human skeleton. The intensity of red lines indicates the connection strength of correlations.

Number	Param.	Acc (%)
3(CTR-GCN)	1.46M	84.9
1	0.85M	84.3 $^{\downarrow 0.5}$
2	1.15M	84.7 $^{\downarrow 0.2}$
4	1.76M	85.2 $^{\uparrow 0.3}$
5	2.07M	**85.4** $^{\uparrow \mathbf{0.5}}$
6	2.37M	85.0 $^{\uparrow 0.1}$

Table 2. Comparisons of model performances with different number of CTR-GCs.

Temporal Modeling	Acc (%)
Temporal Conv(CTR-GCN)	84.9
Temporal Pooling	72.8 $^{\downarrow 12.1}$

Table 3. Comparisons of model performances with different number of CTR-GCs.

tend to be coarse and dense, which captures global features for recognizing actions; (2) the channel-specific correlations varies with different channels, indicating that our CTR-GCN models individual joints relationships under different types of motion features; (3) most channel-specific correlations focus on two hands, which capture subtle interactions on hands and are helpful for recognizing "typing on a keyboard".

4. Ablation Study

Effect of CTR-GC's number. In CTR-GCN, we use three CTR-GCs for fair comparison with other methods (e.g., AGCN, MSG3D), which mostly use three or more GCs to increase model capacity. To verify the effectiveness of CTR-GC's number to our method, We test the model with 1 6 CTR-GCs. As shown in Table 2, accuracies first increase due to increased model capacity, but drops at 6 CTR-GCs, which may be caused by overfitting.

Effect of temporal convolutions. It's a common practice to use (multi-scale) temporal convolutions for temporal modeling in skeleton-based action recognition. To validate the effect of temporal convolutions, we try to use global average pooling for temporal modeling. As shown in Table 3, the performance drops from 84.9% to 72.8%, probably because the pooling loses too much temporal information to extract joints' trajectory features effectively.

Performance on hard classes. We further analyze the performance of different graph convolutions on hard classes on NTU-RGB+D 120, *i.e.*, "staple book", "count money", "play with phone" and "cut nails", "playing magic cube" and "open bottle". These actions mainly involve subtle interactions between fingers, making them difficult to be recognized correctly. As shown in Figure 2, CTR-GC outperforms other graph convolutions on all classes. Especially, CTR-GC exceeds other methods at least by 7.03%

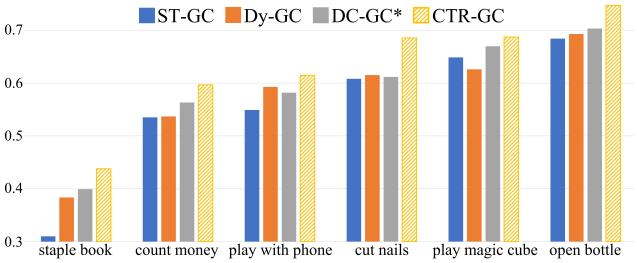

Figure 2. Comparison of classification accuracy of different graph convolutions on hard action classes.

and 4.36% on "cut nails" and "open bottle" respectively, showing that, compared with other GCs, our CTR-GC can effectively extract features of subtle interactions and classify them more accurately.