Leveraging Third-Order Features in Skeleton-Based Action Recognition

Zhenyue Qin[1] , **Yang Liu**[1,2] ,
Pan Ji[3] , **Dongwoo Kim**[4] , **Lei Wang**[1,2] , **RI (Bob) McKay**[1] , **Saeed Anwar**[1,2] , **Tom Gedeon**[1]
[1]Australian National University
[2]Data61, CSIRO
[3]OPPO US Research
[4]GSAI POSTECH
{zhenyue.qin, yang.liu3}@anu.edu.au

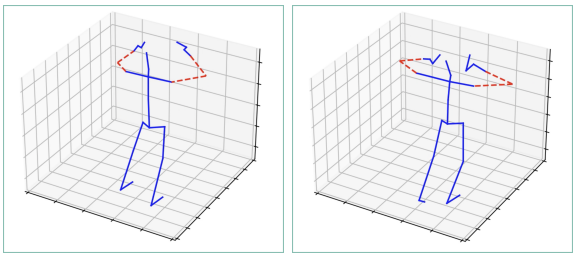
Abstract

Skeleton sequences are light-weight and compact, and thus ideal candidates for action recognition on edge devices. Recent skeleton-based action recognition methods extract features from 3D joint coordinates as spatial-temporal cues, using these representations in a graph neural network for feature fusion, to boost recognition performance. The use of first- and second-order features, *i.e.*, joint and bone representations has led to high accuracy, but many models are still confused by actions that have similar motion trajectories. To address these issues, we propose fusing third-order features in the form of angles into modern architectures, to robustly capture the relationships between joints and body parts. This simple fusion with popular spatial-temporal graph neural networks achieves new state-of-the-art accuracy in two large benchmarks, including NTU60 and NTU120, while employing fewer parameters and reduced run time. Our sourcecode is publicly available at: https://github.com/ZhenyueQin/Angular-Skeleton-Encoding.

1 Introduction

Action recognition is a long-standing problem in artificial intelligence and pattern recognition. It has many useful real-world applications such as smart video surveillance, human computer interaction, sports analysis and health care [Ma *et al.*, 2017]. Compared to the use of conventional RGB-based models [Wang *et al.*, 2019], skeleton-based action recognition methods are more robust to background information and easier to process, resulting in increasing attention [Shi *et al.*, 2019a]. Moreover, mobile devices such as Kinect V2 for human pose estimation are readily available with decreasing cost, which increases the suitability of skeleton sequences for use in edge devices for real-time action recognition.

In recent skeleton-based action recognition work, skeletons are treated as graphs, with each vertex representing a body joint, and each edge a bone. Initially, only first-order features were employed, representing the coordinates of the joints [Yan *et al.*, 2018]. Subsequently, [Shi *et al.*, 2019b] introduced a second-order feature: each bone is represented as the vector difference between one joint's coordinate and that of its nearest neighbor in the direction of the body center. Their experiments show that these second-order features improve recognition accuracy.



Taking off glasses Taking off headphones

Figure 1: Sample skeletons with similar motion trajectories: (left) taking off glasses vs (right) taking off headphones. Blue solid lines represent the skeleton, and red dashed lines form the angles shaped by the fore- and upper arms. The two angles are distinctive.

However, existing methods suffer from the poor performance of discriminating actions with similar motion trajectories (see Figure 1). Since the joint coordinates in each frame are similar in these actions, it is challenging to identify the cause of nuances between coordinates. It can be various body sizes, motion speeds or actually performing different actions. In this paper, we propose the use of third-order representations in the form of angles. Angular features, including angular distance and velocity, of human body joints capture the relative movements between body parts while maintain invariance for different body sizes of human subjects. Thus, the proposed angular features allow the model to more precisely recognize actions. Experimental results reveal that by fusing angular information into the existing modern action recognition architectures, such as ST-GCN [Yan *et al.*, 2018], SCK + DCK [Koniusz *et al.*, 2020], confusing actions can be classified much more accurately, especially when the actions have very similar motion trajectories.

It is worth considering whether it is possible to design a neural network to implicitly learn angular features. However, such a design would be challenging for current graph convolutional networks (GCNs). In brief, there are two reasons. (a) *Conflicts between more layers and higher performance*

of GCNs: GCNs are currently the best-performing models in classifying skeleton-based actions. To model the relationships among all the joints, a graph network requires many layers. However, recent work implies the performance of a GCN can be compromised when it goes deeper due to over-smoothing problems [Min *et al.*, 2020]. (b) *Limitation of adjacency matrices*: recent graph networks for action recognition learn the relationships among nodes via an adjacency matrix, which only captures pairwise relevance, whereas angles are third-order relationships involving three related joints.

We summarize our contributions as follows:

1. We propose third-order representations in the form of angular distance features as well as their velocity. They capture relative motion between body parts while maintain invariance against different human body sizes.
2. The angular features can be easily fused into existing action recognition architectures to further boost performance. Our experiments show that angular features are complementary information in terms of existing features, *i.e.*, the joint and bone representations.
3. We are the first to incorporate angular features into modern spatial-temporal GCNs and achieve state-of-the-art results on several benchmarks including NTU60 and NTU120. Meanwhile, our simple yet powerful model employs fewer training parameters and requires less inference time, thus capable of supporting real-time action recognition on edge devices.

2 Related Work

Many of the earliest attempts at skeleton-based action recognition encoded all human body joint coordinates in each frame into a feature vector for pattern learning [Wang, 2017; Wang *et al.*, 2020]. These models rarely explored the internal dependencies between body joints, resulting in missing rich information about actions. Later, convolutional neural networks (CNNs) were introduced to tackle the problem and achieved an improvement in recognition capability [Wang *et al.*, 2019]. However, CNNs are designed for grid-based data and are not suitable for graph data since they cannot leverage the topology of a graph.

In GCN-based models, a skeleton is treated as a graph, with joints as nodes and bones as edges. An early application was ST-GCN [Yan *et al.*, 2018], using graph convolution to aggregate joint features spatially and convolving consecutive frames along the temporal axis. Subsequently, AS-GCN [Li *et al.*, 2019a] was proposed to further improve the spatial feature aggregation via the learnable adjacency matrix instead of using the skeleton as a fixed graph. Afterwards, AGC-LSTM [Si *et al.*, 2019a] learned long-range temporal dependencies, using LSTM as a backbone, and changed every gate operation from the original fully connected layer to a graph convolution layer, making better use of the skeleton topology.

Concurrently, 2s-AGCN [Shi *et al.*, 2019b] made two major contributions: (a) applying a learnable residual mask to the adjacency matrix of the graph convolution, making the skeleton's topology more flexible; (b) proposing a second-order feature, the difference between the coordinates of two adjacent joints, to act as the bone information. An ensemble of two models, trained with the joint and bone features, substantially improved the classification accuracy. More novel graph convolution techniques have been proposed in skeleton-based action recognition, such as SGN [Zhang *et al.*, 2020], Shift-GCN [Cheng *et al.*, 2020b] and DeCoupGCN [Cheng *et al.*, 2020a], employing self-attention, shift convolution and graph-based dropout, respectively. Recently, MS-G3D [Liu *et al.*, 2020] achieved the current state-of-the-art results by proposing graph 3D convolutions to aggregate features within a window of consecutive frames. However, 3D convolutions demand a long running time.

All the existing methods suffer from low accuracy in discriminating actions sharing similar motion trajectories.

Features	Distance	Acc↑ (%)	Velocity	Acc↑ (%)
Jnt	81.90	–	79.31	–
Cct: Jnt + Ang	83.24	1.34	81.77	2.46
Bon	84.00	–	80.32	–
Cct: Bon + Ang	84.55	0.55	82.85	2.53
Cct: Jnt + Bon	84.10	–	80.54	–
Cct: Jnt + Bon + Ang	**85.87**	1.77	82.98	2.44

Table 1: Evaluation results on the concatenation of angular features. Cct indicates the concatenation. The red bold number highlights the highest prediction accuracy. Acc↑ is the improvment in accuracy.

Features	Distance	Acc↑ (%)	Velocity	Acc↑ (%)
Ang	81.97	–	79.83	–
Jnt	81.90	–	79.31	–
Ens: Jnt + Ang	83.53	1.63	83.81	4.5
Bon	84.00	–	80.32	–
Ens: Bon + Ang	86.47	2.47	86.13	5.81
Ens: Jnt + Bon	86.22	–	86.35	–
Ens: Jnt + Bon + Ang	**87.13**	0.91	86.87	0.52

Table 2: Evaluation results on ensembling with angular features. Ens is the ensembling. The red bold number highlights the highest prediction accuracy. Acc↑ is the improvment in accuracy.

3 Angular Feature Encoding

3.1 Angular Feature Representation

We propose using third-order features, which measure the angle between three body joints to depict the relative movements between body parts in skeleton-based action recognition. Given three joints u, w_1 and w_2, where u is the target joint to calculate the angular features and w_1 and w_2 are endpoints in the skeleton, $\vec{b}_{uw_i}$ denotes the vector from joint u to w_i $(i = 1, 2)$, we have $\vec{b}_{uw_i} = (x_{w_i} - x_u, y_{w_i} - y_u, z_{w_i} - z_u)$, where x_k, y_k, z_k represent the coordinates of joint k $(k = u, w_1, w_2)$. We define two kinds of angular features.

Angular distance: suppose θ is the angle between $\vec{b}_{uw_1}$ and

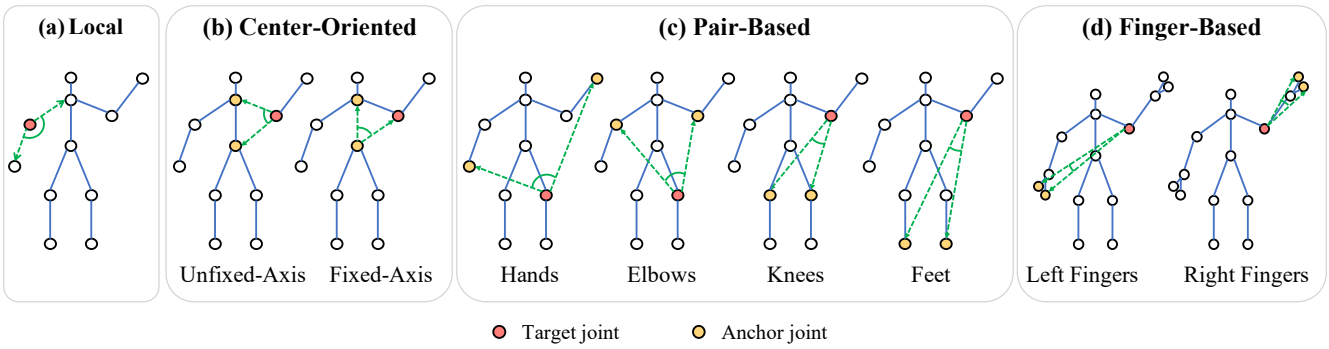


Figure 2: The proposed four types of angular features. We extract angular features for the target joint (in red dots) which corresponds to the root of an angle. The anchor joints (in yellow dots) are fixed endpoints of angles. Green dashed lines represent the two sides of an angle.

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	Similar Action
staple book	31.00	cutting paper (use scissors)	40.80	9.80	cutting paper (use scissors)
make victory sign	35.13	make ok sign	61.39	**26.26**	make ok sign
make ok sign	40.86	make victory sign	66.78	**25.92**	make victory sign
counting money	51.05	play magic cube	56.14	5.09	play magic cube
writing	53.30	typing on a keyboard	61.20	7.90	typing on a keyboard
reading	55.67	writing	63.73	8.06	writing
blow nose	59.13	**hush (quite)**	65.21	6.08	**yawn**
hit other person w sth	60.00	wield knife toward other person	68.17	8.17	wield knife toward other person
cutting paper	62.65	staple book	68.76	6.11	staple book
thumb up	64.00	**make victory sign**	78.78	**14.78**	**thumb down**
cutting nails	64.17	playing w phone/tablet	67.83	3.66	playing w phone/tablet
snapping fingers	64.28	**thumb up**	72.12	7.84	**make victory sign**
fold paper	64.32	**ball up paper**	76.00	**11.68**	**counting money**
playing w phone/tablet	65.91	**play magic cube**	68.09	2.18	**typing on a keyboard**
yawn	66.18	**hush (quiet)**	70.26	4.08	**blow nose**
wield knife toward other person	66.84	hit other person w sth	68.14	1.30	hit other person w sth
typing on a keyboard	68.11	writing	71.27	3.16	writing
open a box	68.36	reading	80.31	**11.95**	reading
sneeze/cough	68.84	touch head (headache)	70.65	1.81	touch head (headache)

Table 3: A comparison of with/without angular features on the most confusing actions that may share similar motion trajectories. The 'Action' column shows the ground truth labels, and the 'Similar Action' column shows the predictions from the model (with/without angular features). The similar actions highlighted in orange demonstrate the change of predictions after employing angular features. The accuracy improvements highlighted in red are the substantially increased ones (Acc↑ $\geq$ 10%) due to using our angular features.

$\vec{b}_{uw_2}$; we define the *angular distance* $d_a(u)$ for joint u as

$$d_a(u) = \begin{cases} 1 - \cos\theta = 1 - \frac{\vec{b}_{uw_1} \cdot \vec{b}_{uw_2}}{|\vec{b}_{uw_1}||\vec{b}_{uw_2}|} & \text{if } u \neq w_1,\ u \neq w_2, \\ 0 & \text{if } u = w_1 \text{ or } u = w_2. \end{cases} \quad (1)$$

Note that w_1 and w_2 do not need to be adjacent nodes of u. The feature value increases monotonically as θ goes from 0 to π radians. In contrast to the first-order features, representing the coordinate of a joint and the second-order features, representing the lengths and directions of bones, these third-order features focus more on motions and are invariant to the scale of human subjects.

Angular velocity: the temporal differences of the angular features between consecutive frames, *i.e.*,

$$v_a^{(t+1)}(u) = d_a^{(t+1)}(u) - d_a^t(u), \quad (2)$$

where $v_a^{(t+1)}(u)$ is the angular velocity of joint u at frame $(t+1)$, describing the dynamic changes of angles.

However, we face a computational challenge when we attempt to exploit these angular features: if we use all possible angles, *i.e.*, all possible combinations of u, w_1 and w_2, the computational complexity is $O(N^3T)$, where N and T respectively represent the number of joints and frames. Instead, we manually define sets of angles that seem likely to facilitate distinguishing actions without drastically increasing computational cost. In the rest of this section, we present the four categories of angles considered in this work.

(a) Locally-Defined Angles. As illustrated in Figure 2(a), a locally-defined angle is measured between a joint and its two adjacent neighbors. If the target joint has only one adjacent joint, we set its angular feature to zero. When a joint has more than two adjacent joints, we choose the most active two. For example, for the neck joint, we use the two shoulders in-

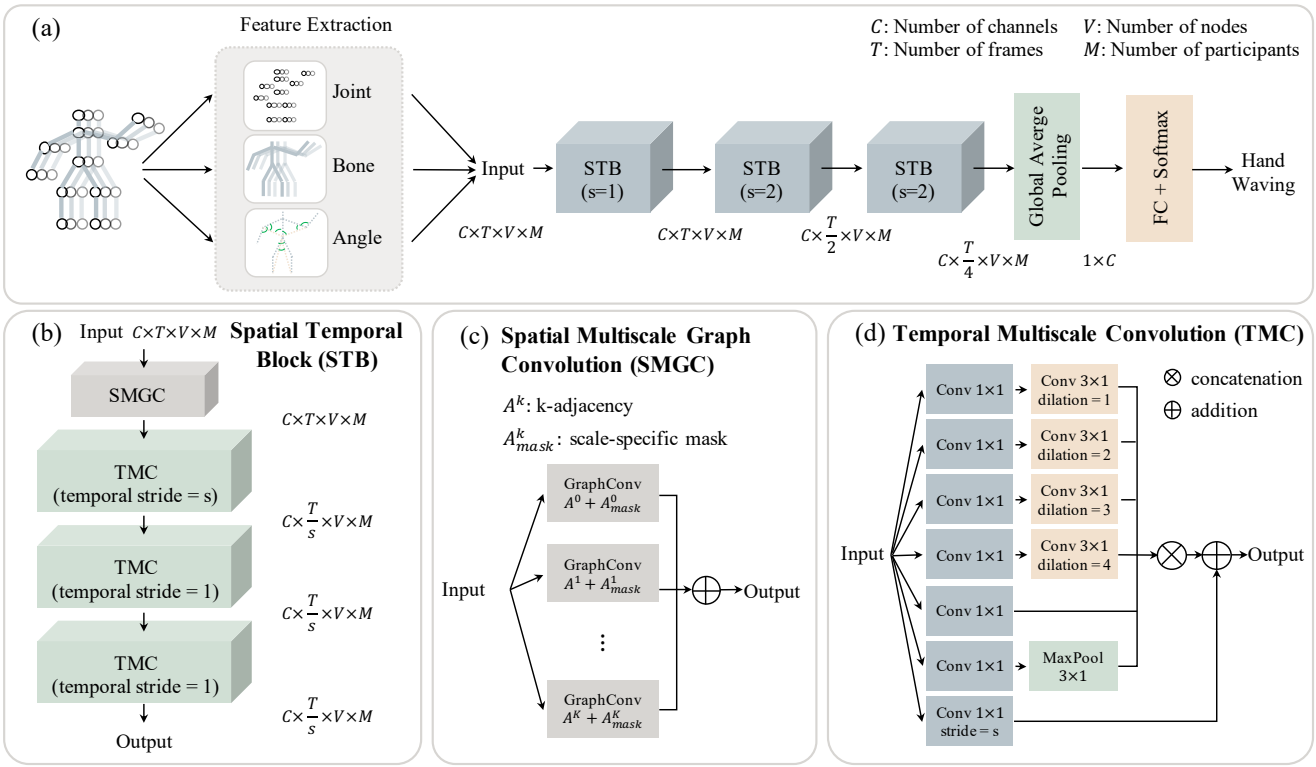


Figure 3: Our Angular Network (AngNet) is composed of three spatial temporal blocks, each consisting of a spatial multiscale graph convolution and a temporal multiscale convolution unit. The spatial multiscale unit extracts structural skeleton information with parallel graph convolutional layers. The temporal multiscale unit draws correlations with four functional groups. See Section 3.2 for more details.

stead of the head and belly since the latter rarely move. These angles can capture relative motions between two bones.

(b) Center-Oriented Angles. A center-oriented angle measures the angular distance between a target joint and two body center joints representing the neck and pelvis. As in Figure 2(b), given a target joint, we use two center-oriented angles: 1) neck-target-pelvis, dubbed as unfixed-axis and 2) neck-pelvis-target, dubbed as fixed-axis. For the joints representing the neck and pelvis, we set their angular features to zero. Center-oriented angles measure the relative position between a target joint and the body center joints. For example, given an elbow as a target joint moving away horizontally from the body center, the unfixed-axis angle decreases, while the fixed-axis angle increases.

(c) Pair-Based Angles. Pair-based angles measure the angle between a target joint and four pairs of endpoints: hands, elbows, knees and feet, as illustrated in Figure 2(c). If the target joint is one of the endpoints, we set the feature value to zero. We select these four pairs due to their importance in performing actions. The pair-based angles are beneficial for recognizing object-related actions. For example, when a person is holding a box, the angle between a target joint and hands can indicate the box's size.

(d) Finger-Based Angles. Fingers are actively involved in human actions. When the skeleton of each hand has finger joints, we include more detailed finger-based angles to incorporate them. As demonstrated in Figure 2(d), the two joints corresponding to fingers are selected as the anchor endpoints of an angle. The finger-based angles can indirectly depict gestures. For instance, an angle with a wrist as the root and a hand tip as well as a thumb as two endpoints can reflect the degree of hand opening.

3.2 Our Backbone

The overall network architecture is illustrated in Figure 3. Three different features are extracted from the skeleton and input into the stack of three spatial temporal blocks (STBs). Then, the output passes sequentially to a global average pooling, a fully-connected layer, and then a softmax layer for action classification. We use MS-G3D [Liu *et al.*, 2020] as the backbone of our model, but we remove their heavy graph 3D convolution (G3D) modules, weighing the performance gain against the computational cost. We call the resulting system AngNet. Note that our proposed angular features are independent of the choice of the backbone network.

We extract the joint, bone and angular features from every action video. For the bone feature, if a joint has more than one adjacent node, we choose the joint closer to the body's center. So, given an elbow joint, we use the vector from the elbow to the shoulder rather than the vector from the elbow to the wrist. For the angle, we extract seven or nine angular features (without/with finger-based angles) for every joint, constituting seven or nine channels of features. Eventually, for each action, we construct a feature tensor $X \in \mathbb{R}^{C \times T \times V \times M}$,

Methods	# Ens	NTU60		NTU120		# Params (M)	GFlops
		X-Sub	X-View	X-Sub	X-Set		
HCN [Li *et al.*, 2018]	1	86.5	91.1	-	-	-	-
MAN [Xie *et al.*, 2018]	1	82.7	93.2	-	-	-	-
ST-GCN [Yan *et al.*, 2018]	1	81.5	88.3	-	-	2.91	16.4
AS-GCN [Li *et al.*, 2019b]	1	86.8	94.2	-	-	7.17	35.5
AGC-LSTM [Si *et al.*, 2019b]	2	89.2	95.0	-	-	-	-
2s-AGCN [Shi *et al.*, 2019b]	4	88.5	95.1	-	-	6.72	37.2
DGNN [Shi *et al.*, 2019a]	4	89.9	96.1	-	-	8.06	71.1
Bayes-GCN [Zhao *et al.*, 2019]	1	81.8	92.4	-	-	-	-
SGN [Zhang *et al.*, 2020]	1	89.0	94.5	79.2	81.5	0.69	15.4
MS-G3D [Liu *et al.*, 2020]	2	91.5	96.2	86.9	88.4	6.44	98.0
Our Results							
Baseline (Jnt)	1	87.2	93.7	81.9	83.7	1.44	19.4
Baseline (Bon)	1	88.2	93.6	84.0	85.7	1.44	19.4
Ens: Baseline (Jnt+Bon)	2	89.3	94.7	85.9	87.4	2.88	38.8
Cct: AngNet-JA (Jnt + Ang)	1	88.7	94.5	83.2	83.7	1.46	19.6
Cct: AngNet-BA (Bon + Ang)	1	89.2	94.8	84.6	85.5	1.46	19.6
Cct: AngNet-JBA: (Jnt + Bon + Ang)	1	90.0	94.8	85.9	86.8	1.50	20.6
Cct: AngNet-VJBA: (Jnt + Bon + Ang)	1	87.1	94.0	83.0	84.6	1.50	20.6
Ens: AngNet-JBA + VJBA	2	91.0	96.1	87.6	88.8	3.00	41.2
Ens: AngNet-BA + JBA + VJBA	3	91.6	96.3	**88.4**	89.1	4.38	60.8
Ens: AngNet-JA + BA + JBA + VJBA	4	**91.7**	**96.4**	88.2	**89.2**	5.92	80.4

Table 4: Comparison of recognition performance on three benchmarks. We compare not only the recognition accuracy but also the total number of parameters (#params) in the networks. #Ens is the number of models used in an ensemble. Cct indicates the concatenation, and J/Jnt, B/Bon as well as A/Ang denote the use of joint, bone and angular features, respectively. V indicates that the input data stream is velocity. The top accuracy is highlighted in red bold, and the second best performance is highlighted in blue.

where C, T, V and M respectively correspond to the numbers of channels, frames, joints and participants. We test various combinations of the joint, bone and angular features in the experiments.

Each STB, as exhibited in Figure 3(b), comprises a spatial multiscale graph convolution (SMGC) unit and three temporal multiscale convolution (TMC) units.

The SMGC unit, as shown in Figure 3(c), consists of a parallel combination of graph convolutional layers. The adjacency matrix of graph convolutions results from the summation of a powered adjacency matrix A^k and a learnable mask A^k_{mask}. *Powered adjacency matrices:* To prevent oversmoothing, we avoid sequentially stacking multiple graph convolutional layers to make the network deep. Following [Liu *et al.*, 2020], to create graph convolutional layers with different sizes of receptive fields, we directly use the powers of the adjacency matrix A^k instead of A itself to aggregate the multi-hop neighbor information. Thus, $A^k_{i,j} = 1$ indicates the existence of a path between joint i and j within k-hops. We feed the input into K graph convolution branches with different receptive fields. K is no more than the longest path within the skeleton graph. *Learnable masks:* Using the skeleton as a fixed graph cannot capture the non-physical dependencies among joints. For example, two hands may always perform actions in conjunction, whereas they are not physically connected in a skeleton. To infer the latent dependencies among joints, following [Shi *et al.*, 2019b], we apply learnable masks to the adjacency matrices.

The TMC unit, shown in Figure 3(d), consists of seven parallel temporal convolutional branches. Each branch starts with a 1×1 convolution to aggregate features between different channels. The functions of different branches diverge as the input passes forward, which can be divided into four groups. In detail: (a) *Extracting multiscale temporal features*: the group contains four 3×1 temporal convolutions, applying four different dilations to obtain multiscale temporal receptive fields. (b) *Processing features within the current frame*: This group only has one 1×1 to concentrate features within a single frame. (c) *Emphasizing the most salient information within the consecutive frames*: The group ends with a 3×1 max-pooling layer to draw the most important features. (d) *Preserving Gradient*: The final group incorporates a residual path to preserve the magnitude of gradients during back-propagation [Chen *et al.*, 2017].

4 Experiments

4.1 Datasets

NTU60. [Shahroudy *et al.*, 2016] is a widely-used benchmark dataset for skeleton-based action recognition, incorporating 56,000 videos. The action videos were collected in a laboratory environment, resulting in accurately extracted skeletons. Nonetheless, recognizing actions from these high-quality skeletons is challenging due to five aspects: the skeletons are captured from different viewpoints; the skeleton sizes

of subjects vary; so do their speeds of action; different actions can have similar motion trajectories; there are limited joints to portray hand actions in detail.

NTU120. [Liu *et al.*, 2019] is an extension of NTU60. It uses more camera positions and angles, as well as a larger number of performing subjects, leading to 113,945 videos. Thus, it is more challenging than NTU60.

4.2 Experimental Setups

We train deep learning models on four NVIDIA 2080-Ti GPUs and use PyTorch as our deep learning framework to compute the angular features. Furthermore, we apply stochastic gradient descent (SGD) with momentum 0.9 as the optimizer. The training epochs for NTU60 and NTU120 are set to 55 and 60, respectively, with learning rates decaying to 0.1 of the original value at epochs 35, 45 and 55. We follow [Shi *et al.*, 2019a] in normalizing, translating each skeleton, and padding all clips to 300 frames via repeating the action sequences.

4.3 Ablation Studies

There are two possible approaches to use angular features: (a) simply concatenate our proposed angular features with the existing joint, bone or both features, and then train the model; (b) feed the angular features into our model, and ensemble it with other models that are trained using joint, bone or both features to predict the action label. We study the differences between these approaches. All the experiments in this section are conducted on the cross-subject setting of NTU120 dataset.

Concatenating with Angular Features. Here, we study the effects of concatenating angular features with others. We first obtain the accuracy of three models trained with three feature types, *i.e.*, the joint, bone and a concatenation of both, respectively, as our baselines. Then, we concatenate angular features to each of these three to compare the performance. We evaluate the accuracy with two data streams, *i.e.*, angular distance and velocity. The results are reported in Table 1. We observe that all the feature types in both data streams receive accuracy boosting in response to incorporating angular features. For the distance stream, concatenating angular features with the concatenation of joint and bone features leads to the greatest enhancement. As to the velocity stream, although the accuracy is lower than that of the distance one, the improvement resulting from angular features is more substantial. In sum, concatenating all three features using the distance data stream results in the highest accuracy.

Ensembling with Angular Features. We also study the change in accuracy when ensembling a network trained solely with angular features, denoted as *Ang*, with networks trained with joint and bone features, respectively, as well as their ensemble. The three models are referred to as *Jnt*, *Bon* and *Jnt* + *Bon* separately. We still use both the distance and velocity streams of data. The results are given in Table 2. We obtain the accuracy of the above three models as the baseline results for each stream, and compare them against the precision of ensembling the baseline models with *Ang*. We note that ensembling *Ang* consistently leads to an increase in accuracy. As with the concatenation studies, angular features are more beneficial for the velocity stream. However, unlike the case with concatenation, the accuracy from the two streams are similar for the ensembling models. We also observe that ensembling with *Bon* achieves considerable accuracy gain. An ensemble of *Jnt*, *Bon* and *Ang* results in the highest accuracy in the distance stream.

Discussion. We want to provide an intuitive understanding of how angular features help in differentiating actions. To this end, we compare the results from two models trained with the joint features and the concatenation of joint and angular features respectively, as illustrated in Table 3.

We observe two phenomena: (a) the majority of the action categories receiving a substantial accuracy boost from angular features are hand-related, such as making a victory sign vs thumbs up. We hypothesize that the enhancement may result from our explicit design of angles for hands and fingers, so that the gestures can be portrayed more comprehensively. (b) for some actions, after the angular features have been introduced, the most similar actions change. Taking the action 'blow nose' as an example, the most confusing action is changed to 'yawn', which a human has difficulty in identifying between them by observing skeletons. This suggests that the angles are providing complementary information to the coordinate-based representations.

4.4 Comparison with State of the Art Models

The ablation studies indicate fusing angular features in both concatenating and ensembling forms can boost accuracy. Hence, we include the results of both approaches as well as their combination in Table 4. In practice, the storage and the run time may become bottlenecks. Thus, we consider not only the recognition accuracy but also the number of parameters (in millions) and the inference time (in gigaFLOPs). The unavailable results are marked with a dash.

We achieve new state-of-the-art accuracies for recognizing skeleton actions on both datasets, *i.e.*, NTU60 and NTU120. For NTU120, AngNet outperforms the existing state-of-the-art model by a wide margin.

Apart from higher accuracy, AngNet requires fewer parameters and shorter inference time. We evaluate the inference time of processing a single NTU120 action video for all the methods. Compared with the existing most-accurate model, AngNet requires fewer than 70% of the parameters and less than 70% of the run time while achieving *higher* recognition results.

Of note, the proposed angular features are compatible with the listed competing models. If one seeks even higher accuracy, the simple GCN employed by us can be replaced with a more sophisticated model, such as MS-G3D [Liu *et al.*, 2020], although such replacement can lead to more parameters and longer inference time.

5 Conclusion

To extend the capacity of GCNs in extracting body structural information, we propose a third-order representation in the form of angular features, The proposed angular features comprehensively capture the relative motion between different body parts while maintain robustness against variations of subjects. Hence, they are able to discriminate between

challenging actions having similar motion trajectories, which cause problems for existing models. Our experimental results show that the angular features are complementary to existing features, *i.e.*, the joint and bone representations. By incorporating our angular features into a simple action recognition GCN, we achieve new state-of-the-art accuracy on several benchmarks while maintaining lower computational cost, thus supporting real-time action recognition on edge devices.

References

[Chen *et al.*, 2017] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4467–4475, 2017. 5

[Cheng *et al.*, 2020a] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. *European Conference of Computer Vision (ECCV)*, 2020. 2

[Cheng *et al.*, 2020b] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[Koniusz *et al.*, 2020] Piotr Koniusz, Lei Wang, and Anoop Cherian. Tensor representations for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2020. 1

[Li *et al.*, 2018] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018. 5

[Li *et al.*, 2019a] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[Li *et al.*, 2019b] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[Liu *et al.*, 2019] Jun Liu, Amir Shahroudy, Mauricio Lisboa Perez, Gang Wang, Ling-Yu Duan, and Alex Kot Chichung. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2019. 6

[Liu *et al.*, 2020] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4, 5, 6

[Ma *et al.*, 2017] Qianli Ma, Lifeng Shen, Enhuan Chen, Shuai Tian, Jiabing Wang, and Garrison W Cottrell. Walking walking walking: Action recognition from action echoes. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. 1

[Min *et al.*, 2020] Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering gcn: Overcoming oversmoothness in graph convolutional networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[Shahroudy *et al.*, 2016] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[Shi *et al.*, 2019a] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 5, 6

[Shi *et al.*, 2019b] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5

[Si *et al.*, 2019a] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[Si *et al.*, 2019b] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[Wang *et al.*, 2019] Lei Wang, Piotr Koniusz, and Du Huynh. Hallucinating idt descriptors and i3d optical flow features for action recognition with cnns. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[Wang *et al.*, 2020] Lei Wang, Du Q. Huynh, and Piotr Koniusz. A comparative review of recent kinect-based action recognition algorithms. *TIP*, 29:15–28, 2020. 2

[Wang, 2017] Lei Wang. Analysis and Evaluation of Kinect-based Action Recognition Algorithms. Master's thesis, School of the Computer Science and Software Engineering, The University of Western Australia, Nov 2017. 2

[Xie *et al.*, 2018] Chunyu Xie, Ce Li, Baochang Zhang, Chen Chen, Jungong Han, and Jianzhuang Liu. Memory attention networks for skeleton-based action recognition. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018. 5

[Yan *et al.*, 2018] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 1, 2, 5

[Zhang *et al.*, 2020] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5

[Zhao *et al.*, 2019] Rui Zhao, Kang Wang, Hui Su, and Qiang Ji. Bayesian graph convolution lstm for skeleton

based action recognition. In *International Conference on Computer Vision (ICCV)*, 2019. 5