

# Fusing Higher-Order Features in Graph Neural Networks for Skeleton-Based Action Recognition

Zhenyue Qin, Yang Liu, Pan Ji, Dongwoo Kim, Lei Wang, R.I. (Bob) McKay, Saeed Anwar, Tom Gedeon

**Abstract**—Skeleton sequences are lightweight and compact, thus are 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. Nonetheless, 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 angular encoding 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 source code is publicly available at: <https://github.com/ZhenyueQin/Angular-Skeleton-Encoding>.

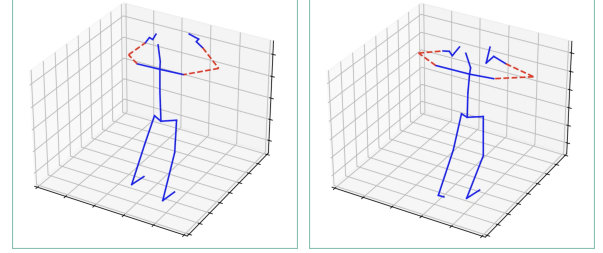
**Index Terms**—Skeleton-based action recognition, graph neural network, feature extraction

## I. INTRODUCTION

Action recognition is a long-standing problem in neural networks and learning systems. It has many useful real-world applications such as smart video surveillance, human-computer interaction, sports analysis, and health care [17]. Compared to the use of conventional RGB-based models [29], skeleton-based action recognition methods are more robust to background information and easier to process, attracting increasing attention [22] in the community. Moreover, mobile devices such as Kinect V2 and the new Azure Kinect for human pose estimation are readily available with decreasing cost, which increases the suitability of skeleton sequences for use in 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 [33]. Subsequently, [23] introduced a second-order feature: each bone is expressed 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.

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



Taking off glasses

Taking off headphones

**Fig. 1: Sample skeletons with similar motion trajectories: (left) taking off glasses vs (right) taking off headphones. The angles formed by red dashed lines (*i.e.*, the fore- and upper arms) are distinctive, which are informative in distinguishing these two similar motions.**

are similar in these actions, it is challenging to identify the cause of nuances between coordinates. It can be due to various body sizes, motion speeds, or actually performing different actions. To robustly capture the relative movements between body parts while maintaining invariance for different body sizes of human subjects, in this paper, we propose the use of third-order representations in the form of angles. We refer to the new proposed feature as angular encoding, which can be applied to both static and velocity domains of human body joints. Thus, the proposed encoding allows the model to recognize actions more precisely. Experimental results reveal that by fusing angular information into the existing modern action recognition architectures, such as STGCN [33], ShiftGCN [ ] and DecoupleGCN [ ], 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) [31], [26], mainly due to 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 [19]. (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.

Z. QIN, R. McKay and T. GEDEON were with Australian National University (ANU). Y. LIU L. Wang and S. ANWAR were with both ANU and Data61, CSIRO. P. JI was with OPPO US Research Center. D. KIM was with Postech. Corresponding authors: Yang Liu and Zhenyue Qin. Emails: yang.liu3@anu.edu.au, zhenyue.qin@anu.edu.au.

We summarize our contributions as follows:

- 1) We propose the third-order representations in the form of the angular encoding defined in both static and velocity domains. The encoding captures relative motion between body parts while maintaining 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 relative to 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.

## II. 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 [27], [28]. These models rarely explored the internal dependencies between body joints, resulting in missing rich information about actions. Kernel-based methods have also been proposed for action recognition [6], [7].

Later, as deep learning became a standard choice in video processing [15], [1] and understanding [10], [9] applications, RGB-based videos started to tackle action recognition tasks. However, they suffer from problems in domain adaptation [35], [5], [37] since they have varying background with different texture of subjects. On the other hand, skeleton data has relatively less number of issues with domain adaptation. Convolutional neural networks (CNNs) were introduced to tackle the problem and achieved an improvement in recognition capability [29]. However, CNNs are designed for grid-based data and are not suitable for graph data since they cannot leverage the topology of a graph.

Graph neural networks started to attract attention [18], [13], [30] in skeleton recognition. 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 [33], using graph convolution to aggregate joint features spatially and convolving consecutive frames along the temporal axis. Subsequently, AS-GCN [12] was proposed to further improve the spatial feature aggregation via the learnable adjacency matrix instead of using the skeleton as a fixed graph. AGC-LSTM [25] 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 topological information. 2s-AGCN [23] 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 graph convolution techniques have been proposed in skeleton-based action recognition, such as SGN [34], Shift-GCN [4] and DeCou-GCN [3], employing self-attention, shift convolution, and graph-based dropout, respectively. Recently, MS-G3D [16] 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. This motivates us to seek a new encoding to facilitate the model differentiating two confusing actions.

## III. ANGULAR FEATURE REPRESENTATION

### A. Angular Encoding

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.

*Static Angular Encoding*: suppose  $\theta$  is the angle between  $\vec{b}_{uw_1}$  and  $\vec{b}_{uw_2}$ ; we define the *static angular encoding*  $d_a(u)$  for joint  $u$  as

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

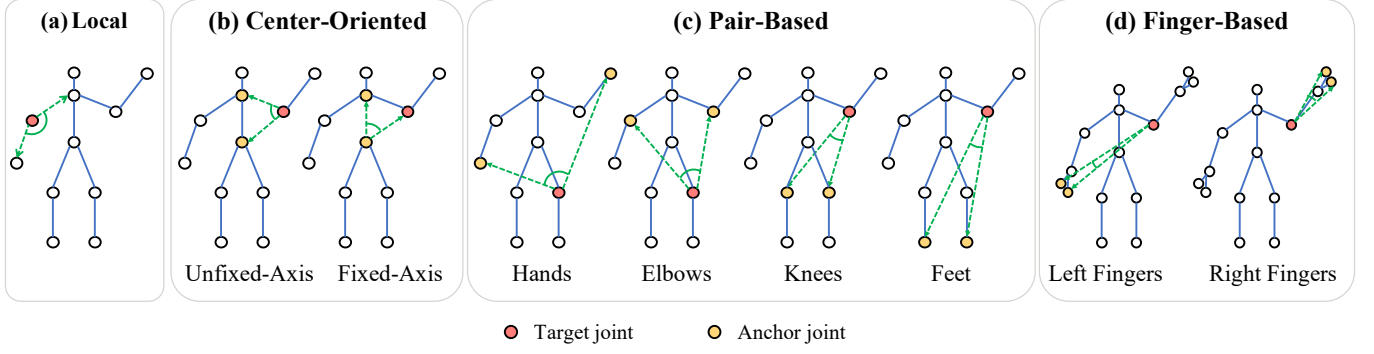
Note that  $w_1$  and  $w_2$  do not need to be adjacent nodes of  $u$ . The feature value increases monotonically as  $\theta$  goes from 0 to  $\pi$  radians. In contrast to the first-order features, representing the coordinate of a joint, and the second-order features, representing the lengths and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

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

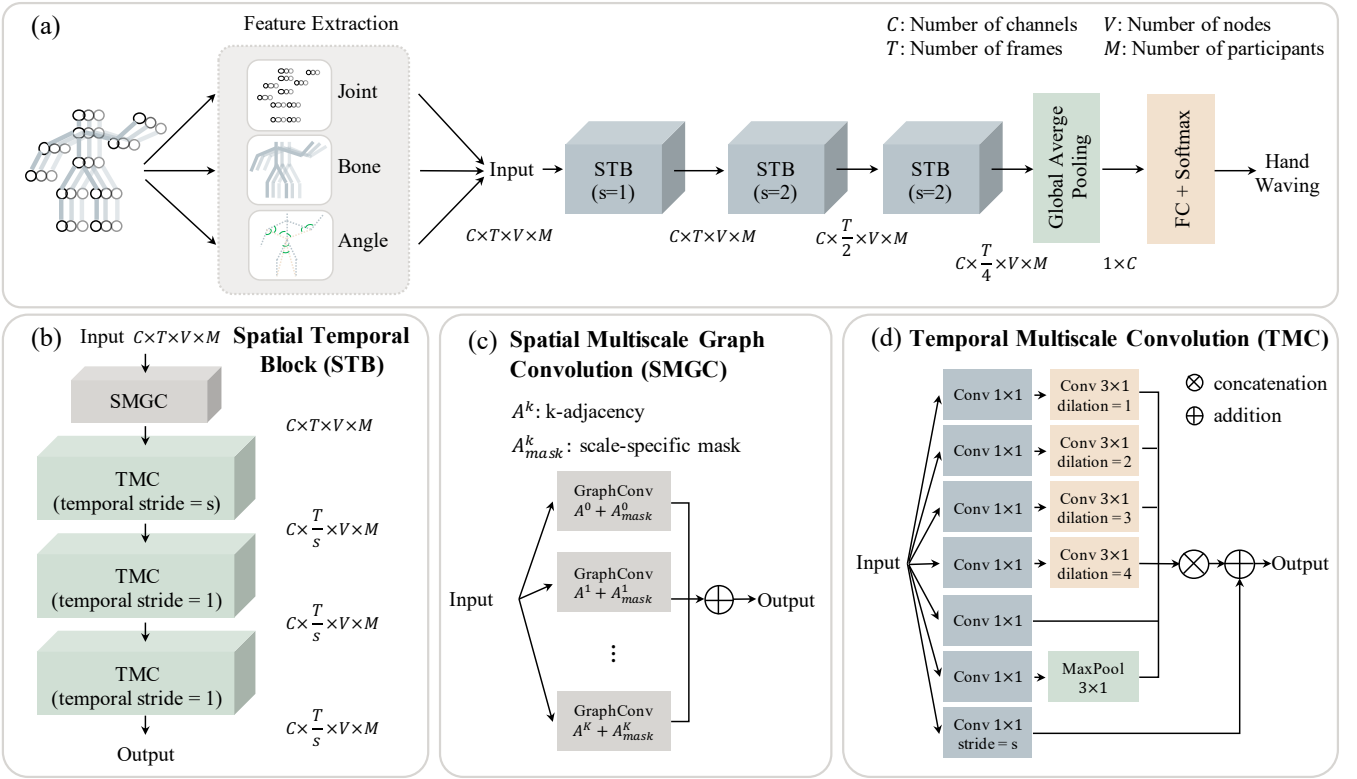
$$v_a^{(t+1)}(u) = d_a^{(t+1)}(u) - d_a^t(u), \quad (2)$$

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.



**Fig. 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.**



**Fig. 3: Our backbone architecture 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 III-B for more details.**

**(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, we use the two shoulders instead of the head and belly for the neck joint 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,

**TABLE I: Comparison of recognition performance on four settings of two benchmark datasets. 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. BSL means to use the original feature without employing angular encoding. AGE-S and AGE-V stand for concatenating the original representation with angular encoding in the static and velocity domains respectively. Joint/J and Bone/B denote the use of joint and bone features respectively. The top accuracy is highlighted in red bold, and the second best performance is highlighted in blue. Symbol & indicates to ensemble models trained with different input features given in the parenthesis.**

Methods	Year	# Ens	NTU60				NTU120				# Params (M)	GFlops
			X-Sub	Acc ↑	X-View	Acc ↑	X-Sub	Acc ↑	X-Set	Acc ↑		
HCN [8]	2018	1	86.5	-	91.1	-	-	-	-	-	-	-
MAN [32]	2018	1	82.7	-	93.2	-	-	-	-	-	-	-
ST-GCN [33]	2018	1	81.5	-	88.3	-	-	-	-	-	2.91	16.4
AS-GCN [11]	2019	1	86.8	-	94.2	-	-	-	-	-	7.17	35.5
AGC-LSTM [24]	2019	2	89.2	-	95.0	-	-	-	-	-	-	-
2s-AGCN [23]	2019	4	88.5	-	95.1	-	-	-	-	-	6.72	37.2
DGNN [22]	2019	4	89.9	-	96.1	-	-	-	-	-	8.06	71.1
Bayes-GCN [36]	2019	1	81.8	-	92.4	-	-	-	-	-	-	-
SGN [34]	2020	1	89.0	-	94.5	-	79.2	-	81.5	-	0.69	15.4
DeCoupleGCN [3]	2020	4	90.8	-	<b>96.6</b>	-	86.5	-	88.1	-	13.72	102.3
MS-G3D [16]	2020	2	91.5	-	96.2	-	86.9	-	88.4	-	6.44	98.0
Our Methods												
BSL-S (Joint)	-	1	87.2	-	93.7	-	81.9	-	83.5	-	1.42	19.0
AGE-S (Joint)	-	1	88.7	1.5	94.5	0.8	83.2	1.3	83.7	0.2	1.44	19.4
BSL-S (Bone)	-	1	88.2	-	93.6	-	84.0	-	85.3	-	1.42	19.0
AGE-S (Bone)	-	1	89.2	1.0	94.8	1.2	84.6	0.6	85.5	0.2	1.44	19.4
BSL-V (Joint)	-	1	86.0	-	93.3	-	79.3	-	80.8	-	1.42	19.0
AGE-V (Joint)	-	1	88.2	2.2	94.5	1.2	81.8	2.5	83.7	2.7	1.44	19.4
BSL-V (Bone)	-	1	86.4	-	92.7	-	80.3	-	82.0	-	1.42	19.0
AGE-V (Bone)	-	1	88.0	1.6	94.8	2.1	82.9	2.6	85.1	3.1	1.44	19.4
BSL-S (Joint+Bone)	-	1	89.2	-	95.1	-	84.1	-	86.0	-	1.44	19.4
AGE-S (Joint+Bone)	-	1	90.0	0.8	95.2	0.1	85.9	1.8	86.8	0.8	1.46	19.6
BSL-V (Joint+Bone)	-	1	86.1	-	92.6	-	80.5	-	81.5	-	1.44	19.4
AGE-V (Joint+Bone)	-	1	87.1	1.0	94.0	1.4	83.0	2.5	84.6	3.1	1.46	19.6
BSL-Ens: S(J)&V(J)	-	2	89.3	-	94.7	-	84.3	-	85.2	-	2.84	38.0
AGE-Ens: S(J)&V(J)	-	2	90.5	1.2	95.5	0.8	85.3	1.0	85.8	0.6	2.88	38.8
BSL-Ens: S(B)&V(B)	-	2	90.5	-	94.7	-	86.3	-	85.6	-	2.84	38.0
AGE-Ens: S(B)&V(B)	-	2	90.8	0.3	95.5	0.8	87.3	1.0	86.8	1.2	2.88	38.8
BSL-Ens: S(J+B)&V(J+B)	-	2	90.5	-	95.7	-	86.4	-	86.4	-	2.88	38.8
AGE-Ens: S(J+B)&V(J+B)	-	2	91.0	0.5	96.1	0.4	87.6	1.2	88.8	2.4	2.92	39.2
BSL-Ens: S(B)&S(J+B)&V(J+B)	-	3	90.7	-	95.7	-	87.3	-	86.9	-	4.30	57.8
AGE-Ens: S(B)&S(J+B)&V(J+B)	-	3	91.4	0.7	<b>96.3</b>	0.6	<b>88.4</b>	1.1	<b>89.1</b>	2.2	4.36	58.6
BSL-Ens: S(J)&S(B)&S(J+B)&V(J+B)	-	4	90.9	-	95.9	-	87.5	-	87.2	-	5.72	76.8
AGE-Ens: S(J)&S(B)&S(J+B)&V(J+B)	-	4	<b>91.6</b>	0.7	<b>96.3</b>	0.4	<b>88.2</b>	0.7	<b>89.2</b>	2.0	5.80	78.0

**TABLE II: Evaluation results on ensembling with angular features. Ens is the ensembling. Jnt and Bon represent the joint and bone features respectively. The red bold number highlights the highest prediction accuracy. Acc↑ is the improvement 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	<b>87.13</b>	0.91	86.87	0.52

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' 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.



TABLE III: 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 ( $\text{Acc}\uparrow \geq 10\%$ ) due to using our angular features.

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc $\uparrow$ (%)	Similar Action
make victory sign	18.48	thumb up	53.04	<b>34.57</b>	make ok sign
staple book	26.67	staple book	37.13	<b>10.46</b>	cutting paper (using scissors)
writing	28.41	typing on a keyboard	48.90	<b>20.49</b>	typing on a keyboard
counting money	48.47	play magic cube	52.98	4.51	play magic cube
playing with phone/tablet	48.82	play magic cube	59.64	<b>10.82</b>	writing
wield knife towards other person	49.52	hit other person with something	62.50	<b>12.98</b>	hit other person with something
blow nose	55.35	yawn	59.65	4.30	yawn
fold paper	56.57	ball up paper	62.78	6.22	counting money
reading	58.34	cutting paper (using scissors)	64.10	5.76	writing
thumb up	58.65	make victory sign	72.35	<b>13.70</b>	make victory sign
yawn	59.00	hush (quite)	67.65	8.65	hush (quite)
snapping fingers	59.10	shake fist	65.51	6.40	make victory sign
open a box	59.98	fold paper	71.60	<b>11.63</b>	open bottle
pointing to something with finger	64.58	taking a selfie	79.71	<b>15.13</b>	taking a selfie
sneeze/cough	64.58	touch head (headache)	71.74	7.16	touch head (headache)
apply cream on hand back	67.82	open bottle	72.30	4.48	rub two hands together
cutting paper (using scissors)	68.28	staple book	70.16	1.87	staple book

## B. Our Backbone Architecture

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 a simplified version of MS-G3D [16] as the backbone of our model. For simplification, we remove their heavy graph 3D convolution (G3D) modules, weighing the performance gain against the computational cost. We call the resulting system MSGCN. Note that our proposed angular features are independent of the choice of the backbone.

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  $X \in \mathbb{R}^{C \times T \times V \times M}$ , 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  $A_{mask}^k$ . *Powered adjacency matrices*: To prevent over-smoothing, we avoid sequentially stacking multiple graph convolutional layers to make the network deep. Following [16], 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,  $A_{i,j}^k = 1$  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 [23], 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  $1 \times 1$  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  $3 \times 1$  temporal convolutions, applying four different dilations to obtain multiscale temporal receptive fields. (b) *Processing features within the current frame*: This group only has one  $1 \times 1$  to concentrate features within a single frame. (c) *Emphasizing the most salient information within the consecutive frames*: The group ends with a  $3 \times 1$  max-pooling layer to draw the most important features. (d) *Preserving Gradient*: The final group incorporates a residual path to preserve gradients during back-propagation [2].

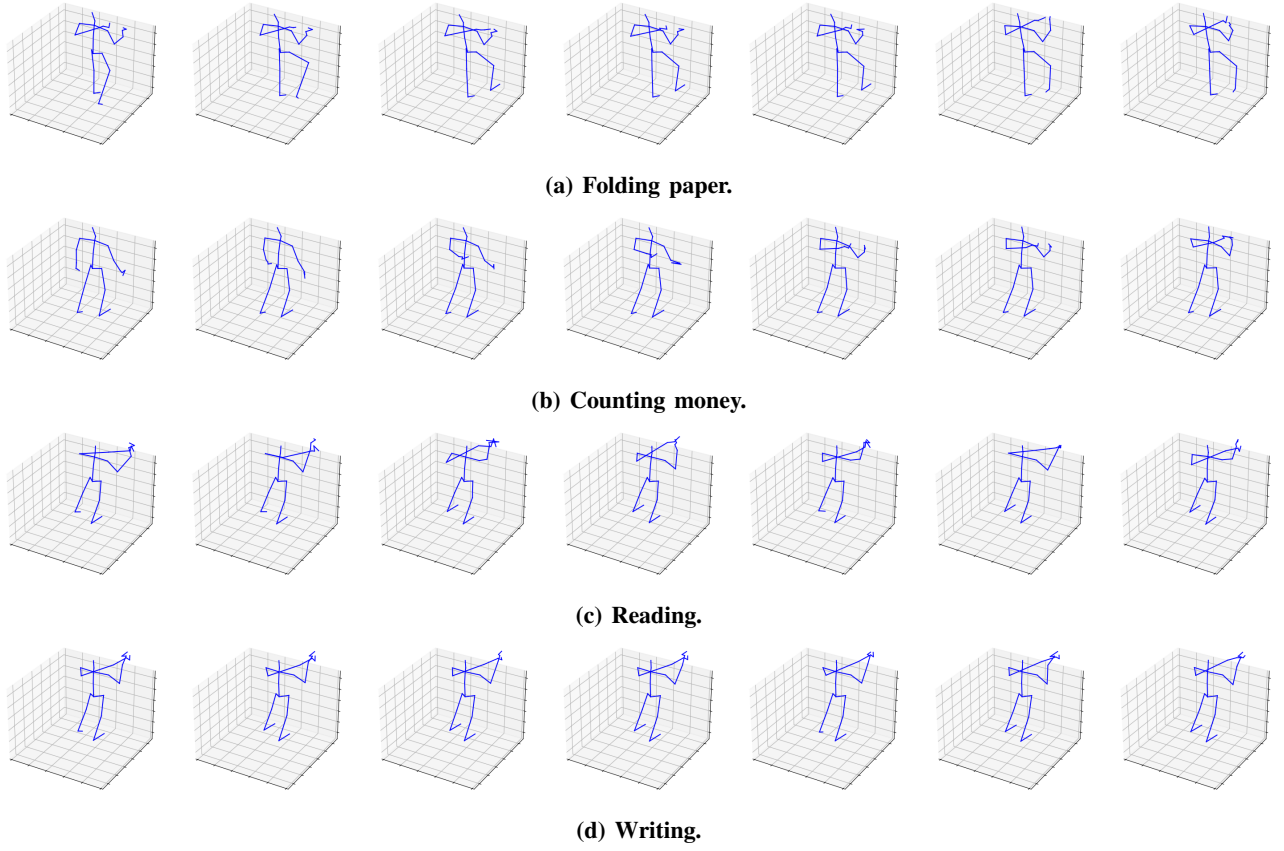
## IV. EXPERIMENTS

### A. Datasets

**NTU60** [21]. NTU60 is a widely-used benchmark dataset for skeleton-based action recognition, incorporating 56,000

**TABLE IV: A comparison of the effect for improving action recognition by concatenating certain angular feature to the joint representation. Each subtable is sorted by the increase of accuracy. The ‘Action’ column shows the ground truth labels, and the ‘Similar Action’ column shows the predictions from the model (with/without angular encoding).**

	Action	Joint		Concatenation: Joint + Angular		
		Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	Similar Action
Static	wear a shoe	70.43	take off a shoe	86.08	15.65	take off a shoe
	punching/slapping other person	72.36	hit other person with something	85.40	13.04	hit other person with something
	thumb up	58.65	make victory sign	71.13	12.48	make victory sign
	pointing to something with finger	64.58	taking a selfie	75.72	11.14	taking a selfie
	wield knife towards other person	49.52	hit other person with something	60.24	10.72	hit other person with something
	fold paper	56.57	ball up paper	66.61	10.04	counting money
	open a box	59.98	fold paper	68.47	8.49	fold paper
Velocity	cutting paper (using scissors)	27.27	staple book	45.90	18.63	staple book
	playing with phone/tablet	39.73	writing	57.45	17.73	typing on a keyboard
	drink water	72.72	brushing teeth	83.94	11.22	brushing teeth
	play magic cube	45.50	counting money	56.64	11.14	counting money
	reading	48.82	writing	59.71	10.89	writing
	typing on a keyboard	56.45	writing	67.27	10.82	writing
	wipe face	75.09	touch head (headache)	83.70	8.61	touch head (headache)
	Action	Joint		Concatenation: Joint + Angular		
		Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	Similar Action
Static	make victory sign	18.48	thumb up	40.35	21.87	make ok sign
	playing with phone/tablet	48.82	play magic cube	68.36	19.55	staple book
	wield knife towards other person	49.52	hit other person with something	65.80	16.28	hit other person with something
	wear a shoe	70.43	take off a shoe	85.35	14.92	take off a shoe
	take off a shoe	70.90	wear a shoe	85.40	14.50	wear a shoe
	punching/slapping other person	72.36	hit other person with something	83.21	10.85	hit other person with something
	yawn	59.00	hush (quite)	69.57	10.57	blow nose
	pointing to something with finger	64.58	taking a selfie	75.00	10.42	taking a selfie
	fold paper	56.57	ball up paper	66.09	9.52	ball up paper
Velocity	cutting paper (using scissors)	27.27	staple book	58.12	30.84	staple book
	playing with phone/tablet	39.73	writing	56.73	17.00	staple book
	make ok sign	27.17	make ok sign	43.65	16.48	make victory sign
	play magic cube	45.50	counting money	61.19	15.69	counting money
	drink water	72.72	brushing teeth	87.96	15.23	brushing teeth
	typing on a keyboard	56.45	writing	70.18	13.73	writing
	touch head (headache)	65.67	brushing teeth	77.90	12.23	drink water
	Action	Joint		Concatenation: Joint + Angular		
		Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	Similar Action
Static	make victory sign	18.48	thumb up	37.39	18.91	make ok sign
	open a box	59.98	fold paper	74.56	14.59	open bottle
	wear a shoe	70.43	take off a shoe	84.98	14.55	take off a shoe
	wield knife towards other person	49.52	hit other person with something	63.37	13.85	hit other person with something
	pointing to something with finger	64.58	taking a selfie	77.17	12.59	taking a selfie
	take off a shoe	70.90	wear a shoe	79.93	9.03	wear a shoe
	thumb down	75.52	thumb up	83.48	7.96	thumb up
Velocity	cutting paper (using scissors)	27.27	staple book	59.34	32.06	staple book
	playing with phone/tablet	39.73	writing	71.27	31.55	typing on a keyboard
	play magic cube	45.50	counting money	64.86	19.36	counting money
	typing on a keyboard	56.45	writing	72.00	15.55	writing
	pointing to something with finger	60.96	taking a selfie	73.55	12.59	taking a selfie
	drink water	72.72	brushing teeth	85.04	12.31	brushing teeth
	open a box	56.84	open bottle	68.82	11.98	open bottle
	Action	Joint		Concatenation: Joint + Angular		
		Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	Similar Action
Static	make victory sign	18.48	thumb up	39.48	21.00	make ok sign
	wield knife towards other person	49.52	hit other person with something	63.72	14.19	hit other person with something
	playing with phone/tablet	48.82	play magic cube	61.45	12.64	play magic cube
	punching/slapping other person	72.36	hit other person with something	82.85	10.49	wield knife towards other person
	fold paper	56.57	ball up paper	65.57	9.00	ball up paper
	play magic cube	62.81	counting money	71.15	8.34	playing with phone/tablet
	side kick	84.89	kicking something	93.21	8.32	kicking something
Velocity	playing with phone/tablet	39.73	writing	66.18	26.45	typing on a keyboard
	cutting paper (using scissors)	27.27	staple book	53.40	26.13	staple book
	play magic cube	45.50	counting money	64.86	19.36	counting money
	typing on a keyboard	56.45	writing	74.18	17.73	writing
	pointing to something with finger	60.96	taking a selfie	78.26	17.30	taking a selfie
	drink water	72.72	brushing teeth	85.04	12.31	brushing teeth
	nausea or vomiting condition	75.36	touch chest (stomachache/heart pain)	84.36	9.00	touch chest (stomachache/heart pain)



**Fig. 4: Visualization examples of confusing actions. The action that the network gets most confused have changed after employing angular encoding as a part of input features.**

**TABLE V: Comparison of recognition performance between MSGCN and MSG3D. MSG3D has higher accuracy, more parameters and longer running time.**

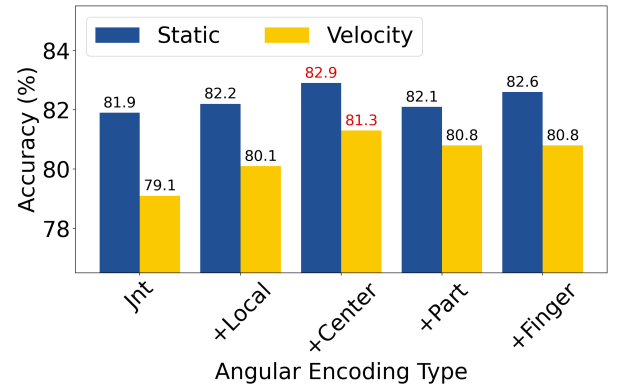
Architecture	Static:	Velocity:	# Params	GFlops
	Jnt+Bon+Ang	Jnt+Bon+Ang		
MSGCN	84.6	83.2	1.46	19.6
MSG3D	86.2	83.6	3.24	50.0

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** [14]. NTU120 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.

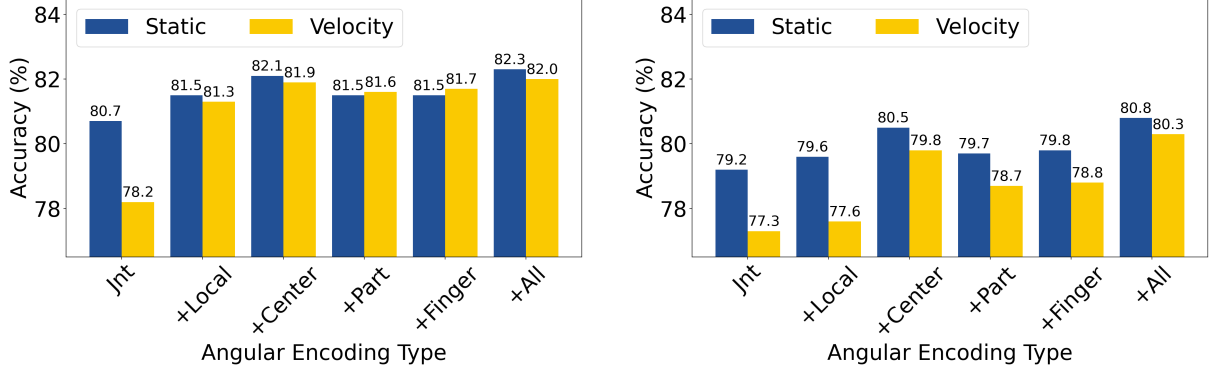
### B. Experimental Setups

We train deep learning models on four NVIDIA 2080-Ti GPUs and use PyTorch as our deep learning framework to compute the angular encoding. Furthermore, we apply stochastic gradient descent (SGD) with momentum 0.9 as the



**Fig. 5: Accuracy of recognizing skeleton-based actions using the multi-scale GCN with different types of angular encoding. Both static and velocity domains are considered. Best accuracy of each domain is highlighted as red.**

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 [22] in normalizing, translating each skeleton, and padding all clips to 300 frames via repeating the action sequences. The training loss function is cross-entropy [20].



**Fig. 6: Accuracy of recognizing skeleton-based actions using DecoupleGCN (left) and ShiftGCN (right) with different types of angular encoding. Both static and velocity domains are considered. The column All represents to concatenate all types of angular encoding.**

### C. 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. We report the results in Table I, including using different settings of both NTU and NTU120. To reduce clutter, we use the results of the cross-subject setting of NTU120 for ablation studies. We denote the accuracy without angular encoding with baseline (BSL). AGE means to concatenate the original feature with angular encoding. The suffix -S (in BSL-S and AGE-S) and -V (in BSL-V and AGE-V) represent feeding the static and velocity feature, respectively.

**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 static and velocity. We observe that all the feature types in both data streams receive accuracy boosting in response to incorporating angular features. For the static stream, concatenating angular features with the concatenation of joint and bone features leads to the most significant enhancement. As to the velocity stream, although the accuracy is lower than that of the static one, the improvement resulting from angular features is more substantial. In sum, concatenating all three features using the static data stream results in the highest accuracy.

**Training Solely with Angular Encoding.** We are interested in the performance of the network when only feeding the angular encoding, *i.e.*, no joint and bone features are used. The outcome is shown as the first row of Table II, denoted as *Ang*. We see training merely with angular encoding even outperforms that of utilizing the joint feature, indicating the completeness of angular encoding for depicting human skele-

ton motion trajectories.

**Ensembling with Angular Encoding.** We also study the change in accuracy when ensembling a network trained solely with angular features *Ang* with networks trained with joint and bone features, respectively, as well as their ensemble. The results are reported in Table II. 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. 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 static stream.

### D. 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 I. 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, MSGCN outperforms the existing state-of-the-art model by a wide margin.

Apart from the higher accuracy, MSGCN 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, MSGCN 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 accu-



racy, the simple GCN employed by us can be replaced with a more sophisticated model, such as MS-G3D [16], although such replacement can lead to more parameters and longer inference time. For example, if we employ more complicated MSG3D [16] instead of our MSGCN, the accuracy can be further improved as Table V shows. Nonetheless, both the number of parameters and the GFlops will also correspondingly increase.

## V. ANALYSIS OF ANGULAR ENCODING

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.

### A. Utilizing of All Types of Angular Encoding

First, we concatenate all kinds of angular encoding with joint features and train the baseline network. The results are illustrated in Table III. 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. This suggests that the angles are providing complementary information to the coordinate-based representations. For the new actions that still confuse the network after using the angular encoding, they are also challenging for humans to differentiate them from their corresponding correct actions by just observing skeletons. For better understanding, We provide some visual examples displaying the confusing actions whose mostly confused counterparts get altered after using angular encoding in Figure 4. Among them, folding paper and counting money are easily confused, and reading and writing are also likely to be mixed up. We see these confusing pairs of skeletons are visually similar to those of humans.

### B. Contributions from Different Angle Types

Next, we conduct ablation studies on different types of the proposed angular encoding for improving the accuracy of recognizing skeleton-based actions. The baseline accuracy is obtained merely using the joint feature. Then, we concatenate different types of angular encoding with the joint feature to evaluate the effectiveness of each encoding type. We study the effects of different types of angular features on improving the accuracy of recognizing actions.

The results are depicted in Figure 5. We observe: i) the center-oriented angular encoding boosts the accuracy with the largest margin for both static and velocity input features; the increases are 1.01% and 2.02% respectively. Since the center-oriented encoding reflects the distance from the joint to the body center, the results imply knowing such a distance is greatly beneficial to recognizing skeleton-based actions. This is consistent with our daily experience. To illustrate, people

normally pose the hand farther away from the body center for the victory sign than for the ok sign. ii) Angular encoding improves more accuracy for the velocity input features than the static joint coordinates. The average improvements are 0.58% and 1.42% respectively. This difference indicates angular encoding provides more additional information in capturing the dynamic motion trajectories of actions than depicting the spatial structural information. iii) The part-based angular encoding only marginally heightens the accuracy of using the static features, only 0.22%, whereas the increase improves substantially enlarges to 1.47% for the velocity input. We conjecture this is because the actions performed by arms and legs involve a lot of dynamics. Thus, when using the velocity input, angular encoding provides complementary dynamic information to these actions.

We also aim to understand how each kind of angular encoding improves the recognizing accuracy. To this end, we collect the top seven actions whose accuracy is improved by the angular encoding the most. The results are exhibited in Table IV. We see: i) Equipping the velocity features with angular encoding boosts substantial accuracy for the long-lasting actions, such as ‘staple book’. In contrast, for the static input, most actions whose accuracy is significantly improved are those that last for a short time, such as ‘thumb up’. ii) The majority of actions whose accuracy is improved by a type of angular encoding are those performed by the anchor joints corresponding to the angular encoding. To illustrate, the finger-based encoding increases accuracy for the hand-related actions, while the part-based encoding benefits the actions heavily using arms and legs.

## VI. GENERALISABILITY OF ANGULAR ENCODING

A possible concern is the generalisability of the proposed angular encoding. That is, will fusing angular encoding improve the accuracy of other backbone architectures? To answer this, we conduct experiments fusing angular encoding with the joint feature and feed the concatenated input to three recently-proposed backbone networks: ShiftGCN [4], DecoupleGCN [3] and MSG3D [16]. The utilized dataset is the cross-subject setting of NTU120.

We display the results in Figure 6. We not only demonstrate the accuracy of fusing all kinds of proposed angular encoding, but we also separately concatenate every type of encoding with the joint feature and report the corresponding accuracy. We see fusing angular encoding with the original features consistently improves the accuracy on all three backbones. On the other hand, the effectiveness of different angular encoding varies in boosting accuracy. We observe the center-oriented angular encoding increases accuracy with the largest magnitude. Furthermore, angular encoding improves accuracy more when deployed in the velocity domain than in the static domain. These two observations are consistent with those on our simple backbone network. For DecoupleGCN, the part- and finger-based angular encoding more substantially improve accuracy than they do for our simple backbone. Specifically, although feeding the velocity input to DecoupleGCN initially leads to lower accuracy than using the static feature, the situation is

reversed after fusing with these two types of angular encoding. That is, using features in the velocity domain surpasses using the static joints.

## VII. 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 maintaining robustness against variations of subjects. Hence, they are able to discriminate between challenging actions having similar motion trajectories, which causes 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

- [1] Saeed Anwar and Nick Barnes. Densely residual laplacian super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2020. 2
- [2] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *Advances in Neural Information Processing Systems*, pages 4467–4475, 2017. 5
- [3] 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*, 2020. 2, 4, 9
- [4] Ke Cheng, Yifan Zhang, Xiangyu He, Weihai Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 9
- [5] Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. Open set domain adaptation: Theoretical bound and algorithm. *IEEE Transactions on Neural Networks and Learning Systems (T-NNLS)*, 2020. 2
- [6] Piotr Koniusz, Anoop Cherian, and Fatih Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European conference on computer vision*, pages 37–53. Springer, 2016. 2
- [7] Piotr Koniusz, Lei Wang, and Anoop Cherian. Tensor representations for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition (T-PAMI)*, 2020. 2
- [8] 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*, 2018. 4
- [9] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Winter Conference on Applications of Computer Vision*, pages 1459–1469, 2020. 2
- [10] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Transferring cross-domain knowledge for video sign language recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6205–6214, 2020. 2
- [11] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4
- [12] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [13] Ming Li, Zheng Ma, Yu Guang Wang, and Xiaosheng Zhuang. Fast haar transforms for graph neural networks. *Neural Networks*, 128, 2020. 2
- [14] 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 Conference on Computer Vision and Pattern Recognition (T-PAMI)*, 2019. 7
- [15] Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon. Invertible denoising network: A light solution for real noise removal. *Conference Computer Vision and Pattern Recog.*, 2021. 2
- [16] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4, 5, 9
- [17] 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*, 2017. 1
- [18] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks and Learning Systems (T-NNLS)*, 20(3):498–511, 2009. 2
- [19] Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering gcn: Overcoming oversmoothness in graph convolutional networks. *Advances in Neural Information Processing Systems*, 2020. 1
- [20] Zhenyue Qin, Dongwoo Kim, and Tom Gedeon. Rethinking softmax with cross-entropy: Neural network classifier as mutual information estimator. *arXiv:1911.10688*, 2019. 7
- [21] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (T-PAMI)*, 2016. 5
- [22] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 4, 7
- [23] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 4, 5
- [24] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4
- [25] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [26] Petar Velićković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv:1710.10903*, 2017. 1
- [27] 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, 2017. 2
- [28] Lei Wang, Du Q. Huynh, and Piotr Koniusz. A comparative review of recent kinect-based action recognition algorithms. *TIP*, 29:15–28, 2020. 2
- [29] Lei Wang, Piotr Koniusz, and Du Huynh. Hallucinating idt descriptors and i3d optical flow features for action recognition with cnns. In *International Conference Computer Vision*, 2019. 1, 2
- [30] Yu Guang Wang, Ming Li, Zheng Ma, Guido Montufar, Xiaosheng Zhuang, and Yanan Fan. Haar graph pooling. In *International Conference Machine Learning*, pages 9952–9962. PMLR, 2020. 2
- [31] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems (T-NNLS)*, 2020. 1
- [32] 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*, 2018. 4
- [33] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference Artificial Intelligence*, 2018. 1, 2, 4
- [34] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4
- [35] Yiyang Zhang, Feng Liu, Zhen Fang, Bo Yuan, Guangquan Zhang, and Jie Lu. Clarinet: A one-step approach towards budget-friendly unsupervised domain adaptation. *arXiv:2007.14612*, 2020. 2
- [36] Rui Zhao, Kang Wang, Hui Su, and Qiang Ji. Bayesian graph convolution lstm for skeleton based action recognition. In *International Conference Computer Vision*, 2019. 4
- [37] Li Zhong, Zhen Fang, Feng Liu, Jie Lu, Bo Yuan, and Guangquan Zhang. How does the combined risk affect the performance of unsupervised domain adaptation approaches? *arXiv:2101.01104*, 2020. 2

## APPENDIX

We provide the improvement of accuracy by angular encoding for each class. The results for the static domain are in [Table VI](#) and [Table VII](#). The ones for the velocity domain are in [Table VIII](#) and [Table IX](#).

TABLE VI: Static: First Half.

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	Similar Action
make victory sign	18.48	thumb up	53.04	34.57	make ok sign
staple book	26.67	staple book	37.13	10.46	cutting paper (using scissors)
writing	28.41	typing on a keyboard	48.90	20.49	typing on a keyboard
counting money	48.47	play magic cube	52.98	4.51	play magic cube
playing with phone/tablet	48.82	play magic cube	59.64	10.82	writing
wield knife towards other person	49.52	hit other person with something	62.50	12.98	hit other person with something
blow nose	55.35	yawn	59.65	4.30	yawn
fold paper	56.57	ball up paper	62.78	6.22	counting money
reading	58.34	cutting paper (using scissors)	64.10	5.76	writing
thumb up	58.65	make victory sign	72.35	13.70	make victory sign
yawn	59.00	hush (quite)	67.65	8.65	hush (quite)
snapping fingers	59.10	shake fist	65.51	6.40	make victory sign
open a box	59.98	fold paper	71.60	11.63	open bottle
pointing to something with finger	64.58	taking a selfie	79.71	15.13	taking a selfie
sneeze/cough	64.58	touch head (headache)	71.74	7.16	touch head (headache)
apply cream on hand back	67.82	open bottle	72.30	4.48	rub two hands together
cutting paper (using scissors)	68.28	staple book	70.16	1.87	staple book
typing on a keyboard	68.45	cutting paper (using scissors)	69.09	0.64	writing
hush (quite)	69.16	yawn	72.08	2.92	yawn
ball up paper	69.26	fold paper	71.30	2.04	fold paper
eat meal/snack"	69.55	brushing teeth	71.27	1.73	brushing teeth
wear a shoe	70.43	take off a shoe	85.35	14.92	take off a shoe
take off a shoe	70.90	wear a shoe	81.75	10.85	wear a shoe
punching/slapping other person	72.36	hit other person with something	82.85	10.49	hit other person with something
open bottle	73.17	play magic cube	73.82	0.65	open a box
put something into a bag	73.26	take something out of a bag	79.13	5.87	take something out of a bag
shake fist	74.69	hand waving	76.39	1.69	snapping fingers
touch head (headache)	75.45	drink water	82.25	6.80	touch neck (neckache)
thumb down	75.52	thumb up	80.87	5.35	pointing to something with finger
sniff (smell)	76.04	blow nose	81.04	5.00	blow nose
make a phone call/answer phone	80.09	reading	87.64	7.55	playing with phone/tablet
apply cream on face	80.71	wipe face	83.10	2.39	wipe face
rub two hands together	80.88	clapping	82.61	1.72	apply cream on hand back
touch neck (neckache)	80.88	drink water	87.32	6.43	flick hair
nausea or vomiting condition	81.18	sneeze/cough	84.73	3.55	touch chest (stomachache/heart pain)
drink water	81.48	brushing teeth	83.94	2.46	brushing teeth
move heavy objects	81.57	carry something with other person	86.09	4.52	carry something with other person
take something out of a bag	81.64	put something into a bag	84.90	3.26	put something into a bag
brushing teeth	81.78	drink water	87.55	5.76	touch head (headache)
drop	81.91	staple book	85.82	3.91	tear up paper
put the palms together	81.97	cross hands in front (say stop)	92.75	10.78	sniff (smell)
point finger at the other person	81.97	pat on back of other person	88.77	6.80	pat on back of other person
use a fan (with hand or paper)/feeling warm	82.27	hand waving	89.82	7.55	shake fist
check time (from watch)	82.33	open bottle	90.58	8.25	put the palms together
support somebody with hand	82.65	follow other person	88.70	6.04	knock over other person (hit with body)
take off headphone	82.92	take off glasses	86.40	3.47	take off glasses
tennis bat swing	83.15	throw up cap/hat	83.45	0.30	throw up cap/hat
take off glasses	83.67	take off headphone	93.07	9.39	take off headphone
knock over other person (hit with body)	83.72	whisper in other person's ear	88.89	5.17	whisper in other person's ear
wipe face	83.78	brushing hair	87.68	3.90	brushing hair
reach into pocket	84.04	touch back (backache)	85.40	1.36	typing on a keyboard

TABLE VII: Static: Second Half.

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc <sup>†</sup> (%)	Similar Action
put on headphone	84.20	wear on glasses	87.52	3.32	take off headphone
throw	84.82	wear jacket	91.27	6.45	stretch oneself
side kick	84.89	kicking something	90.77	5.88	kicking something
tear up paper	84.98	fold paper	88.19	3.21	wear jacket
wear on glasses	85.08	drink water	88.28	3.20	eat meal/snack"
nod head/bow	85.23	nausea or vomiting condition	94.93	9.70	nausea or vomiting condition
kicking other person	85.59	step on foot	91.30	5.71	punching/slapping other person
touch chest (stomachache/heart pain)	85.96	touch back (backache)	91.30	5.35	touch back (backache)
toss a coin	86.09	throw up cap/hat	89.01	2.92	make victory sign
exchange things with other person	86.48	giving something to other person	89.04	2.57	giving something to other person
step on foot	86.65	kicking other person	89.39	2.74	kicking other person
cross toe touch	86.80	move heavy objects	89.90	3.09	move heavy objects
brushing hair	86.91	wipe face	88.64	1.73	touch head (headache)
taking a selfie	87.04	reading	90.22	3.17	pointing to something with finger
put on bag	87.35	take something out of a bag	93.91	6.57	wear jacket
take off bag	87.72	tennis bat swing	92.36	4.65	take off jacket
whisper in other person's ear	87.87	knock over other person (hit with body)	88.87	1.00	knock over other person (hit with body)
cross arms	88.74	cross hands in front (say stop)	94.09	5.35	put the palms together
stretch oneself	88.93	hands up (both hands)	93.06	4.13	hands up (both hands)
cheer up	89.15	hand waving	90.51	1.36	use a fan (with hand or paper)/feeling warm
put on a hat/cap	89.44	wear on glasses	94.85	5.41	wear on glasses
salute	89.58	shake head	92.03	2.45	shake head
hand waving	89.88	use a fan (with hand or paper)/feeling warm	90.15	0.27	shake fist
take a photo of other person	90.32	shoot at other person with a gun	94.27	3.95	shoot at other person with a gun
hands up (both hands)	90.81	stretch oneself	94.25	3.44	stretch oneself
take off a hat/cap	90.94	throw up cap/hat	96.70	5.76	apply cream on face
juggling table tennis balls	91.15	toss a coin	95.81	4.66	snapping fingers
falling	91.36	move heavy objects	93.82	2.45	staggering
touch other person's pocket	91.36	giving something to other person	94.91	3.55	pat on back of other person
touch back (backache)	91.39	touch chest (stomachache/heart pain)	94.20	2.81	touch chest (stomachache/heart pain)
sitting down	91.67	falling	94.51	2.83	kicking something
standing up (from sitting position)	91.67	take off a shoe	95.97	4.30	nausea or vomiting condition
shake head	91.73	touch back (backache)	92.00	0.27	make victory sign
staggering	91.75	walking apart from each other	98.91	7.16	follow other person
butt kicks (kick backward)	91.86	side kick	94.43	2.57	side kick
squat down	92.73	sitting down	96.86	4.14	falling
cross hands in front (say stop)	92.84	taking a selfie	93.12	0.28	put the palms together
bounce ball	92.86	running on the spot	94.21	1.35	finger-guessing game (playing rock-paper-scissors)
finger-guessing game (playing rock-paper-scissors)	92.92	shake fist	94.97	2.04	shake fist
kicking something	93.20	side kick	94.20	1.00	staggering
hopping (one foot jumping)	93.55	staggering	96.00	2.45	kicking something
take off jacket	93.93	tear up paper	96.38	2.45	wear jacket
wear jacket	94.64	tear up paper	98.91	4.27	put on bag
running on the spot	95.17	hopping (one foot jumping)	97.39	2.22	butt kicks (kick backward)
high-five	95.18	hit other person with something	97.05	1.87	giving something to other person
walking apart from each other	95.38	walking towards each other	96.74	1.36	walking towards each other
arm swings	95.52	arm circles	98.61	3.09	arm circles
pushing other person	95.74	hugging other person	96.01	0.28	walking apart from each other
follow other person	95.88	walking apart from each other	96.01	0.13	walking apart from each other
arm circles	96.22	stretch oneself	98.96	2.74	stretch oneself
cheers and drink	96.22	take a photo of other person	97.57	1.35	drink water
hugging other person	96.45	falling	97.81	1.36	falling
jump up	96.83	running on the spot	98.19	1.36	kicking something
walking towards each other	97.17	follow other person	99.27	2.10	staggering



TABLE VIII: Velocity: First Half.

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc $\uparrow$ (%)	Similar Action
make ok sign	27.17	make ok sign	46.78	19.61	make victory sign
cutting paper (using scissors)	27.27	staple book	60.38	33.11	staple book
staple book	30.17	cutting paper (using scissors)	31.52	1.35	staple book
playing with phone/tablet	39.73	writing	64.00	24.27	typing on a keyboard
play magic cube	45.50	counting money	65.21	19.71	counting money
reading	48.82	writing	58.61	9.79	writing
counting money	49.00	play magic cube	50.70	1.70	play magic cube
blow nose	52.04	yawn	62.96	10.91	yawn
thumb up	53.43	make victory sign	63.30	9.87	make victory sign
cutting nails	54.89	writing	58.17	3.28	playing with phone/tablet
hit other person with something	55.35	wield knife towards other person	58.43	3.09	wield knife towards other person
typing on a keyboard	56.45	writing	66.18	9.73	writing
open a box	56.84	open bottle	65.16	8.32	open bottle
shoot at other person with a gun	57.78	point finger at the other person	63.83	6.04	point finger at the other person
fold paper	57.96	ball up paper	65.91	7.96	ball up paper
yawn	58.30	hush (quite)	64.00	5.70	hush (quite)
wield knife towards other person	59.76	hit other person with something	61.28	1.52	hit other person with something
pointing to something with finger	60.96	taking a selfie	72.46	11.51	taking a selfie
snapping fingers	63.46	shake fist	65.68	2.22	shake fist
open bottle	64.10	open a box	73.65	9.55	play magic cube
sneeze/cough	65.30	touch head (headache)	68.12	2.81	touch head (headache)
touch head (headache)	65.67	brushing teeth	73.91	8.25	brushing teeth
hush (quite)	68.46	yawn	74.17	5.71	blow nose
touch neck (neckache)	71.83	touch head (headache)	82.25	10.42	touch head (headache)
flick hair	71.87	blow nose	77.74	5.87	brushing hair
drink water	72.72	brushing teeth	85.77	13.04	brushing teeth
shoot at the basket	73.65	throw	82.34	8.69	hands up (both hands)
put something into a bag	73.78	take something out of a bag	78.09	4.30	take something out of a bag
shake fist	74.17	hand waving	76.22	2.04	hand waving
throw up cap/hat	74.57	toss a coin	79.23	4.66	toss a coin
wipe face	75.09	touch head (headache)	87.68	12.59	touch head (headache)
nausea or vomiting condition	75.36	touch chest (stomachache/heart pain)	80.36	5.00	touch chest (stomachache/heart pain)
take off a shoe	76.74	wear a shoe	83.21	6.47	wear a shoe
taking a selfie	77.26	drink water	83.33	6.07	pointing to something with finger
knock over other person (hit with body)	77.65	wield knife towards other person	82.12	4.47	wield knife towards other person
take something out of a bag	78.17	put something into a bag	82.47	4.30	put something into a bag
wear a shoe	78.49	take off a shoe	81.69	3.20	take off a shoe
take off headphone	78.68	take off glasses	84.98	6.30	take off glasses
make a phone call/answer phone	79.00	drink water	81.82	2.82	playing with phone/tablet
thumb down	79.87	thumb up	82.78	2.91	thumb up
support somebody with hand	79.87	follow other person	82.26	2.39	follow other person
point finger at the other person	80.16	pat on back of other person	88.41	8.25	pat on back of other person
reach into pocket	80.39	eat meal/snack"	83.21	2.82	wear on glasses
drop	80.45	check time (from watch)	83.64	3.18	sniff (smell)
pat on back of other person	81.25	point finger at the other person	86.96	5.71	point finger at the other person
tear up paper	82.03	fold paper	86.35	4.32	open a box
whisper in other person's ear	82.13	knock over other person (hit with body)	86.96	4.83	knock over other person (hit with body)
throw	82.27	tennis bat swing	88.00	5.73	wear jacket
brushing teeth	82.52	writing	89.01	6.49	touch head (headache)
put on headphone	82.96	wear on glasses	87.52	4.57	wear on glasses
check time (from watch)	83.42	eat meal/snack"	89.49	6.07	rub two hands together
step on foot	83.52	kicking other person	86.26	2.74	kicking other person

TABLE IX: Velocity: Second Half.

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc <sup>†</sup> (%)	Similar Action
grab other person's stuff	83.52	wield knife towards other person	87.30	3.78	touch other person's pocket
toss a coin	83.99	thumb up	86.56	2.57	snapping fingers
brushing hair	84.35	brushing teeth	87.91	3.56	use a fan (with hand or paper)/feeling warm
shake head	84.45	typing on a keyboard	92.73	8.27	touch neck (neckache)
cross hands in front (say stop)	84.51	put the palms together	90.58	6.07	put the palms together
take a photo of other person	84.59	shoot at other person with a gun	90.10	5.51	shoot at other person with a gun
move heavy objects	86.15	carry something with other person	89.96	3.82	carry something with other person
take off bag	86.15	take off jacket	90.97	4.82	take off jacket
cheer up	86.23	hand waving	89.42	3.19	use a fan (with hand or paper)/feeling warm
use a fan (with hand or paper)/feeling warm	86.27	hand waving	88.36	2.09	shake fist
put on a hat/cap	86.50	wear on glasses	96.32	9.82	wear on glasses
punching/slapping other person	86.59	hit other person with something	86.86	0.27	hit other person with something
nod head/bow	86.68	touch chest (stomachache/heart pain)	94.93	8.25	take off a shoe
hand waving	86.96	use a fan (with hand or paper)/feeling warm	90.15	3.19	use a fan (with hand or paper)/feeling warm
touch back (backache)	87.41	touch chest (stomachache/heart pain)	91.30	3.90	touch chest (stomachache/heart pain)
put on bag	87.52	take off jacket	93.22	5.70	wear jacket
exchange things with other person	87.52	giving something to other person	88.52	1.00	giving something to other person
salute	87.77	kicking something	90.22	2.45	brushing teeth
cross toe touch	87.85	move heavy objects	89.37	1.52	move heavy objects
wear on glasses	88.01	eat meal/snack"	92.67	4.66	brushing hair
hands up (both hands)	88.72	stretch oneself	91.29	2.57	stretch oneself
touch chest (stomachache/heart pain)	89.22	touch back (backache)	90.22	1.00	touch back (backache)
touch other person's pocket	89.55	pat on back of other person	93.82	4.27	giving something to other person
put the palms together	89.58	check time (from watch)	90.94	1.36	cross hands in front (say stop)
cross arms	90.13	cross hands in front (say stop)	94.43	4.30	put the palms together
kicking other person	90.30	kicking something	91.30	1.00	punching/slapping other person
take off glasses	90.97	wear on glasses	92.70	1.73	take off headphone
pickup	91.00	take off a shoe	93.82	2.82	take off a shoe
hopping (one foot jumping)	91.36	running on the spot	96.73	5.36	staggering
juggling table tennis balls	91.50	open a box	94.42	2.92	open a box
pushing other person	91.75	punching/slapping other person	93.84	2.09	touch other person's pocket
bounce ball	91.98	juggling table tennis balls	93.51	1.53	finger-guessing game (playing rock-paper-scissors)
side kick	92.55	kicking something	94.95	2.39	kicking something
high-five	92.58	finger-guessing game (playing rock-paper-scissors)	95.66	3.08	make victory sign
carry something with other person	92.75	support somebody with hand	94.44	1.69	support somebody with hand
sitting down	92.77	falling	93.04	0.27	nod head/bow
butt kicks (kick backward)	93.08	side kick	95.30	2.22	side kick
handshaking	93.20	hugging other person	95.65	2.45	pat on back of other person
wear jacket	93.55	take off jacket	97.09	3.55	take off jacket
take off a hat/cap	93.87	take off glasses	95.24	1.37	shake head
falling	94.64	squat down	97.09	2.45	staggering
squat down	94.82	sitting down	96.86	2.05	sitting down
jump up	95.38	running on the spot	98.19	2.81	hopping (one foot jumping)
cheers and drink	96.22	grab other person's stuff	97.39	1.17	high-five
staggering	96.46	kicking something	97.83	1.36	walking towards each other
running on the spot	96.74	bounce ball	97.04	0.30	hopping (one foot jumping)
hugging other person	96.81	check time (from watch)	98.18	1.36	check time (from watch)
arm swings	96.91	arm circles	97.91	1.00	arm circles
arm circles	97.43	stretch oneself	98.78	1.35	stretch oneself