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

Zhenyue Qin, Yang Liu, *Member, IEEE*, Pan Ji, Dongwoo Kim, Lei Wang, *Student Member, IEEE*, R. I. McKay, Saeed Anwar, and Tom Gedeon, *Senior Member, IEEE*

***Abstract*— Skeleton sequences are lightweight and compact and thus are ideal candidates for action recognition on edge devices. Recent skeleton-based action recognition methods extract features from 3-D 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, that is, 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 higher-order features in the form of angular encoding (AGE) 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*— Feature extraction, graph neural network, skeleton-based action recognition.**

I. Introduction

SKELETON-BASED action recognition is more robust to background information and easier to process, attracting increasing attention [25] in the community. Recently, deep graph neural networks fuel the recent surge of accuracy for skeleton-based action recognition [39]. By leveraging graph neural networks, action recognizers more thoroughly extract the topological information within the skeleton sequences.

To make graph neural networks applicable for skeleton-based action recognition, 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 [39]. Subsequently, [26] 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 the recognition accuracy of skeleton-based action recognizers.

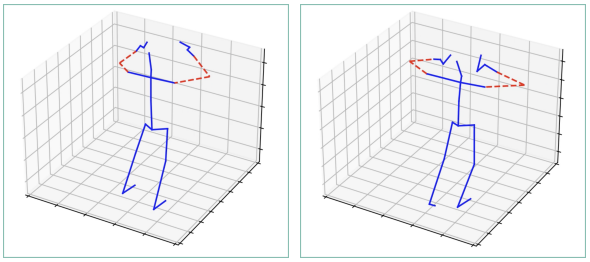


Fig. 1. Sample skeletons with similar motion trajectories: (left) taking off glasses versus (right) taking off headphones. The angles formed by red dashed lines (i.e., the fore- and upper arms) are distinctive, which are informative in distinguishing these two similar motions.

However, existing methods suffer from the poor performance of discriminating actions with similar motion trajectories (see Fig. 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 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 article, we propose the use of higher-order representations in the form of angles. We refer to the new proposed feature as angular encoding (AGE), 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 spatio-temporal graph convolutional network (STGCN) [39] and decoupling GCN [4], confusing action sequences can be classified 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) [29], [35], mainly due to two reasons. 1) *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.

Manuscript received 14 June 2021; revised 22 May 2022; accepted 20 August 2022. This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2020R1F1A1061667. *(Corresponding authors: Yang Liu; Zhenyue Qin.)*

Zhenyue Qin and R. I. McKay are with the School of Computing, Australian National University (ANU), Canberra, ACT 2601, Australia (e-mail: zhenyue.qin@anu.edu.au).

Yang Liu, Lei Wang, and Saeed Anwar are with the School of Computing, ANU, Canberra, ACT 2601, Australia, and also with Data61, CSIRO, Canberra, ACT 2601, Australia (e-mail: yang.liu3@anu.edu.au).

Pan Ji is with Tencent XR Laboratory, Shenzhen 518054, China.

Dongwoo Kim is with the Graduate School of Artifical Intelligence, POSTECH, Pohang 37673, South Korea.

Tom Gedeon is with the Optus-Curtin Centre of Excellence in AI, Curtin University, Perth, WA 6102, Australia.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3201518.

Digital Object Identifier 10.1109/TNNLS.2022.3201518

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 ENSEMBLING MODELS TRAINED WITH DIFFERENT INPUT FEATURES GIVEN IN THE PARENTHESIS. GFLOPS STANDS FOR THE FLOATING-POINT OPERATIONS PERFORMED BY A MODEL, WHICH IS THE NUMBER OF MULTIPLY–ADD OPERATIONS THAT A MODEL PERFORMS

Methods	Year	# Ens	NTU60				NTU120				# Params (M)	GFlops
			X-Sub	Acc ↑	X-View	Acc↑	X-Sub	Acc↑	X-Set	Acc↑		
HCN [11]	2018	1	86.5	-	91.1	-	-	-	-	-	-	-
MAN [36]	2018	1	82.7	-	93.2	-	-	-	-	-	-	-
ST-GCN [39]	2018	1	81.5	-	88.3	-	-	-	-	-	2.91	16.4
AS-GCN [14]	2019	1	86.8	-	94.2	-	-	-	-	-	7.17	35.5
AGC-LSTM [28]	2019	2	89.2	-	95.0	-	-	-	-	-	-	-
2s-AGCN [26]	2019	4	88.5	-	95.1	-	-	-	-	-	6.72	37.2
DGNN [25]	2019	4	89.9	-	96.1	-	-	-	-	-	8.06	71.1
Bayes-GCN [43]	2019	1	81.8	-	92.4	-	-	-	-	-	-	-
SGN [41]	2020	1	89.0	-	94.5	-	79.2	-	81.5	-	0.69	15.4
DeCoupleGCN [4]	2020	4	90.8	-	**96.6**	-	86.5	-	88.1	-	13.72	102.3
MS-G3D [18]	2020	2	91.5	-	96.2	-	86.9	-	88.4	-	6.44	98.0
MST [3]	2021	2	91.1	-	96.4	-	87.0	-	88.3	-	-	-
AdaSGN [27]	2021	4	90.5	-	95.3	-	85.9	-	86.8	-	-	-
Ta-CNN [37]	2022	2	90.7	-	95.1	-	85.7	-	87.3	-	-	-
Efficient-Self-Attention [22]	2022	4	90.5	-	96.1	-	85.7	-	86.8	-	-	-
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	**96.3**	0.6	**88.4**	1.1	**89.1**	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	**91.6**	0.7	**96.3**	0.4	**88.2**	0.7	**89.2**	2.0	5.80	78.0

However, recent work implies that the performance of a GCN can be compromised when it goes deeper due to over-smoothing problems [21]. 2) *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 a rich collection of higher-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, that is, the joint and bone representations.
3) We are the first to incorporate multiple categories of angular features into modern spatial–temporal GCNs and achieve state-of-the-art results on several benchmarks, including NTU60 and NTU120. Meanwhile, if a simple model (employing fewer training parameters and requiring less inference time) has equipped with the proposed angular encoding, it becomes powerful. Thus, the proposed angular encoding supports real-time action recognition on edge devices.

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	**87.13**	0.91	86.87	0.52

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 [31], [32]. 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 [9], [10].

Later, as deep learning became a standard choice in video processing [1], [17] and understanding [12], [13], RGB-based videos started to tackle action recognition. However, they suffer from problems in domain adaptation [7], [42], [44] since they have varying backgrounds with different textures of subjects. On the other hand, skeleton data have relatively fewer issues with domain adaptation. Convolutional neural networks (CNNs) were introduced to tackle skeleton-based action recognition and achieved an improvement [33]. However, CNNs are designed for grid-based data and are not suitable for graph data since they cannot leverage the topology of a graph.

Recently, deep graph neural networks are accumulating attention [15], [20], [34], [40]. Graph neural networks also started to attract attention 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 [39], using graph convolution to aggregate joint features spatially and convolving consecutive frames along the temporal axis. Subsequently, actional-structural graph convolutional network (AS-GCN) [14] was proposed to further improve the spatial feature aggregation via the learnable adjacency matrix instead of using the skeleton as a fixed graph. Attention enhanced graph convolutional LSTM network (AGC-LSTM) [28] learned long-range temporal dependencies, using long short-term memory (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-adaptive graph convolutional network (AGCN) [26] made two major contributions: 1) applying a learnable residual mask to the adjacency matrix of the graph convolution, making the skeleton's topology more flexible; and 2) 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 semantics-guided neural network (SGN) [41] and Shift-GCN [5], employing self-attention and shift convolution, respectively. Recently, multi-scale-graph 3D (MS-G3D) [18] achieved high results by proposing graph 3-D convolutions (G3Ds) to aggregate features within a window of consecutive frames. However, 3-D convolutions demand a long running time.

In more recent times, Qin *et al*. [22] proposed some self-attention models that dynamically optimize the graph structure. Xu *et al*. [37] designed a pure CNN architecture that more effectively captures the topological information. Memmesheimer *et al*. [19] study the one-shot problem of skeleton-based action recognition. They apply the metric learning setting and map the problem to a nearest-neighbor search in a set of activity reference samples. Wang *et al*. [30] studied the adversarial attack problem in skeleton-based action recognition. They investigated a perceptual loss that ensures the imperceptibility of the attack. Diao *et al*. [6] investigated the black-box attack on skeleton-based action recognition. They proposed an attack mechanism called black-box attack on skeletal action recognition (BASKR) and showed that the adversarial attack is a threat and on-manifold adversarial samples are common for skeletal motions.

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. Some works show angle features similar to the local feature presented in this article [8], [38]. On the other hand, we propose a collection of angular encoding forms. Each category consists of further subcategories. Different categories of angular encoding are designed to capture motion features of distinct kinematic body parts.

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.

1) Static Angular Encoding: Suppose θ 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

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

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

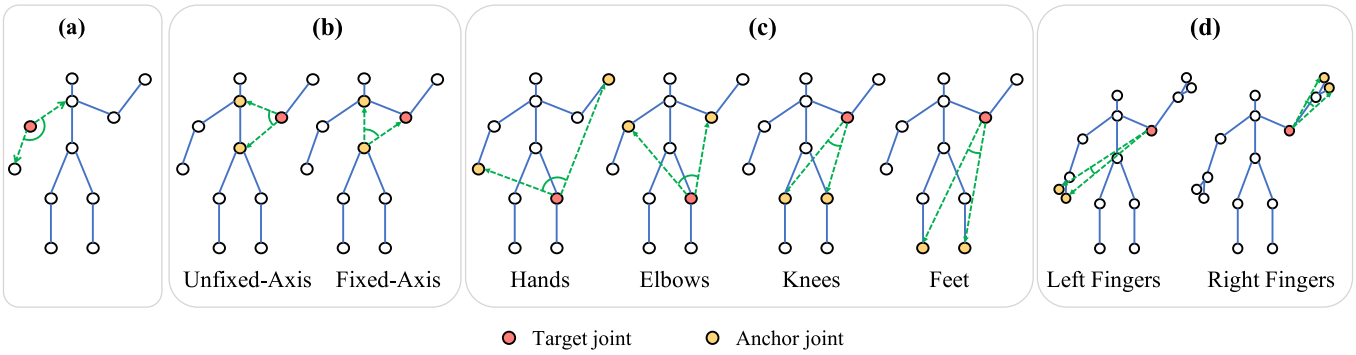


Fig. 2. Proposed four types of angular features. We extract angular features for the target joint (in red dots) which corresponds to the root of an angle. The anchor joints (in yellow dots) are fixed endpoints of angles. Green dashed lines represent the two sides of an angle. (a) Local. (b) Center-oriented. (c) Pair- and (d) Finger-based.

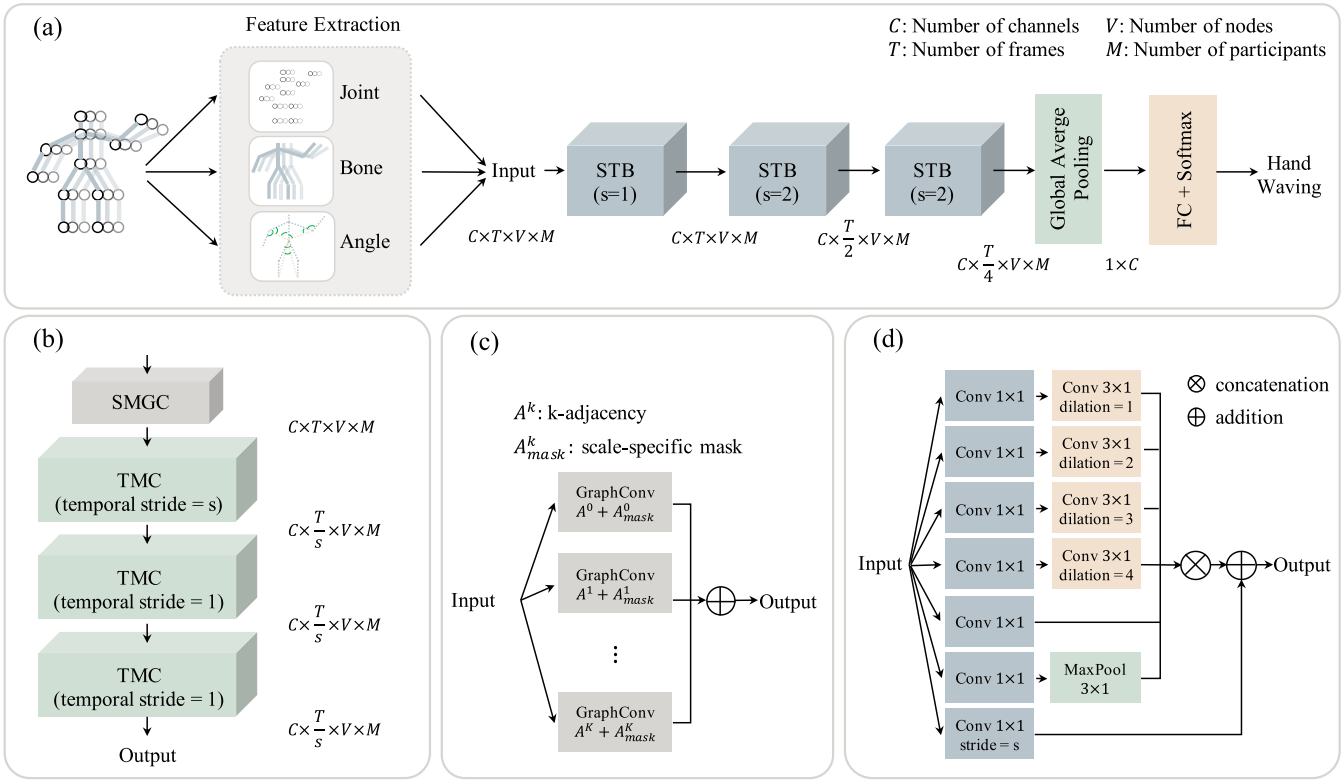


Fig. 3. Our backbone architecture is composed of three STBs, each consisting of a spatial multiscale graph convolution and a temporal multiscale convolution unit. The spatial multiscale unit extracts structural skeleton information with parallel graph convolutional layers. The temporal multiscale unit draws correlations with four functional groups. See Section III-B for more details. (a) Feature extraction. (b) STB. (c) SMGC. (d) TMC.

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.

2) Velocity Angular Encoding: The temporal differences of the angular features between consecutive frames, that is,

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

where $v_a^{(t+1)}(u)$ is the angular velocity of joint u at frame $(t+1)$, describing the dynamic changes of angles. The angular encoding is a third-order feature. Taking the velocity of these third-order features further increases the order. Hence, these velocity angular features enable an action recognizer to capture fourth-order information of motion sequences.

However, we face a computational challenge when we attempt to exploit these angular features: if we use all possible angles, that is, 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 Section III, we present the four categories of angles considered in this work.

a) Locally defined angles: As illustrated in Fig. 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

TABLE III

INDEPENDENT EVALUATION OF ANGULAR ENCODING FOR EACH CATEGORY. XSUB AND XVIEW REPRESENT CROSS-SUBJECT AND CROSS-VIEW. XSET MEANS CROSS-SETUP

Angular Types	NTU60 XSub	NTU60 XView	NTU120 XSub	NTU120 XSet
No angular encoding	87.2	93.7	81.9	83.5
With local	87.9	94.1	82.8	83.5
With center-based	88.4	94.3	83.0	83.7
With pair-based	87.8	94.2	82.4	83.5
With finger-based	88.0	94.1	82.7	83.6
Concatenating all	88.7	94.5	83.2	83.7

TABLE IV

COMPARISON OF RECOGNITION PERFORMANCE BETWEEN MSGCN AND MSG3D. MSG3D HAS HIGHER ACCURACY, MORE PARAMETERS, AND A LONGER RUNNING TIME. GFLOPS STANDS FOR THE FLOATING-POINT OPERATIONS PERFORMED BY A MODEL, WHICH IS THE NUMBER OF MULTIPLY–ADD OPERATIONS THAT A MODEL PERFORMS

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

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 Fig. 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: 1) hands; 2) elbows; 3) knees; and 4) feet, as illustrated in Fig. 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 Fig. 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.

B. Our Backbone Architecture

The overall network architecture is illustrated in Fig. 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 [18] as the backbone of our model. For simplification, we remove their heavy 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 (the persons conducting actions). We test various combinations of the joint, bone, and angular features in the experiments.

Each STB, as exhibited in Fig. 3(b), comprises a spatial multiscale graph convolution (SMGC) unit and three temporal multiscale convolution (TMC) units. The details of these components are illustrated as follows.

The SMGC unit, as shown in Fig. 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}. 1) *Powered adjacency matrices:* To prevent over-smoothing, we avoid sequentially stacking multiple graph convolutional layers to make the network deep. Following [18], 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 multihop 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. 2) *Learnable masks:* Using the skeleton as a fixed graph cannot capture the nonphysical 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 [26], we apply learnable masks to the adjacency matrices.

The TMC unit, shown in Fig. 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.

TABLE V

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

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

1) *Extracting multiscale temporal features:* The group contains four 3×1 temporal convolutions, applying four different dilations to obtain multiscale temporal receptive fields.
2) *Processing features within the current frame:* This group only has one 1×1 to concentrate features within a single frame.
3) *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.
4) *Preserving gradient:* The final group incorporates a residual path to preserve gradients during back-propagation [2].

IV. EXPERIMENTS

A. Datasets

1) NTU60 [24]: NTU60 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 skeletons is still challenging due to five aspects: 1) the skeletons are captured from different viewpoints; 2) the skeleton sizes of subjects vary; 3) so do their speeds of action; 4) different actions can have similar motion trajectories; and 5) there are limited joints to portray hand actions in detail.

2) NTU120 [16]: NTU120 is an extension of NTU60. It uses more camera positions and angles, as well as a larger number of performing subjects, leading to 1 13 945 videos.

B. Experimental Setups

We train deep learning models on four NVIDIA 2080-Ti graphics processing units (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 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 [25] in normalizing, translating each skeleton, and padding all clips to 300 frames via repeating the action sequences. The training loss function is cross-entropy [23].

C. Ablation Studies

There are two possible approaches for using angular features: 1) simply concatenate our proposed angular features with the existing joint, bone, or both features, and then train the model; and 2) 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 Nanyang Technological University (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.

1) 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, that is, the joint, bone, and a concatenation of both, respectively, as our BSLs. Then, we concatenate angular features to each of these three to compare the performance. We evaluate the accuracy with two data streams, that is, angular static and velocity. We observe that all the feature types

TABLE VI

COMPARISON OF THE EFFECT FOR IMPROVING ACTION RECOGNITION BY CONCATENATING CERTAIN ANGULAR FEATURES TO THE JOINT REPRESENTATION. EACH SUBTABLE IS SORTED BY THE INCREASE IN 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)

TABLE VII
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

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.

2) Training Solely With Angular Encoding: We are interested in the performance of the network when only feeding the angular encoding, that is, 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 skeleton motion trajectories.

3) 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 BSL results for each stream and compare them against the precision of ensembling the BSL 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 of the two streams is similar. We also observe that

TABLE VIII
STATIC: SECOND HALF

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	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

ensembling with *Bon* achieves considerable accuracy gain. An ensemble of *Jnt*, *Bon*, and *Ang* results in the highest accuracy in the static stream.

4) Evaluating Angular Encoding of Each Category: We independently evaluate the boost of the angular encoding of the four categories, that is, local, center-oriented, pair-based, and finger-based. The utilized model is the BSL architecture. We discover that all these four categories can individually boost the recognition accuracy, as shown in Table III. Furthermore, the proposed angular encoding has been leveraged in an open challenge and revealed to be effective.[1]

[1]In ICCV 2021, the winning team of a skeleton-based action recognition challenge leveraged the angular encoding proposed in this article, achieving the first-place accuracy among 70+ teams. The utilized dataset was a newly collected skeleton dataset with drones. The winning team specifically evaluated the boost of accuracy from using our proposed angular encoding on the newly recorded dataset, showing the effectiveness of angular encoding. See their presentation (clickable) at 8:30.

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, that is, 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 a 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

TABLE IX
VELOCITY: FIRST HALF

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	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

and less than 70% of the run time while achieving *higher* skeleton-based recognition results.

Of note, the proposed angular features are compatible with the listed competing models. If one seeks even higher accuracy, the employed simple GCN can be replaced with a more sophisticated model, such as MS-G3D [18], although this change can lead to more parameters and longer inference time. For example, if we employ a more complicated MS-G3D [18] instead of our MSGCN, the accuracy can be further improved as Table IV 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.

A. Utilizing of All Types of Angular Encoding

First, we concatenate all kinds of angular encoding with joint features and train the BSL network. The results are illustrated in Table V. We observe two phenomena.

1) 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.
2) For some actions, after the angular features have been introduced, the most similar actions change. This

TABLE X
VELOCITY: SECOND HALF

Action	Joint		Concatenation: Joint + Angular		
	Acc (%)	Similar Action	Acc (%)	Acc↑ (%)	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

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 ground-truth 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 Fig. 4. Among them, folding paper and counting money are easily confused, and reading and writing are also likely to be mixed up. We see that 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 BSL 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 Fig. 5. We observe the following.

1) 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.
2) 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.

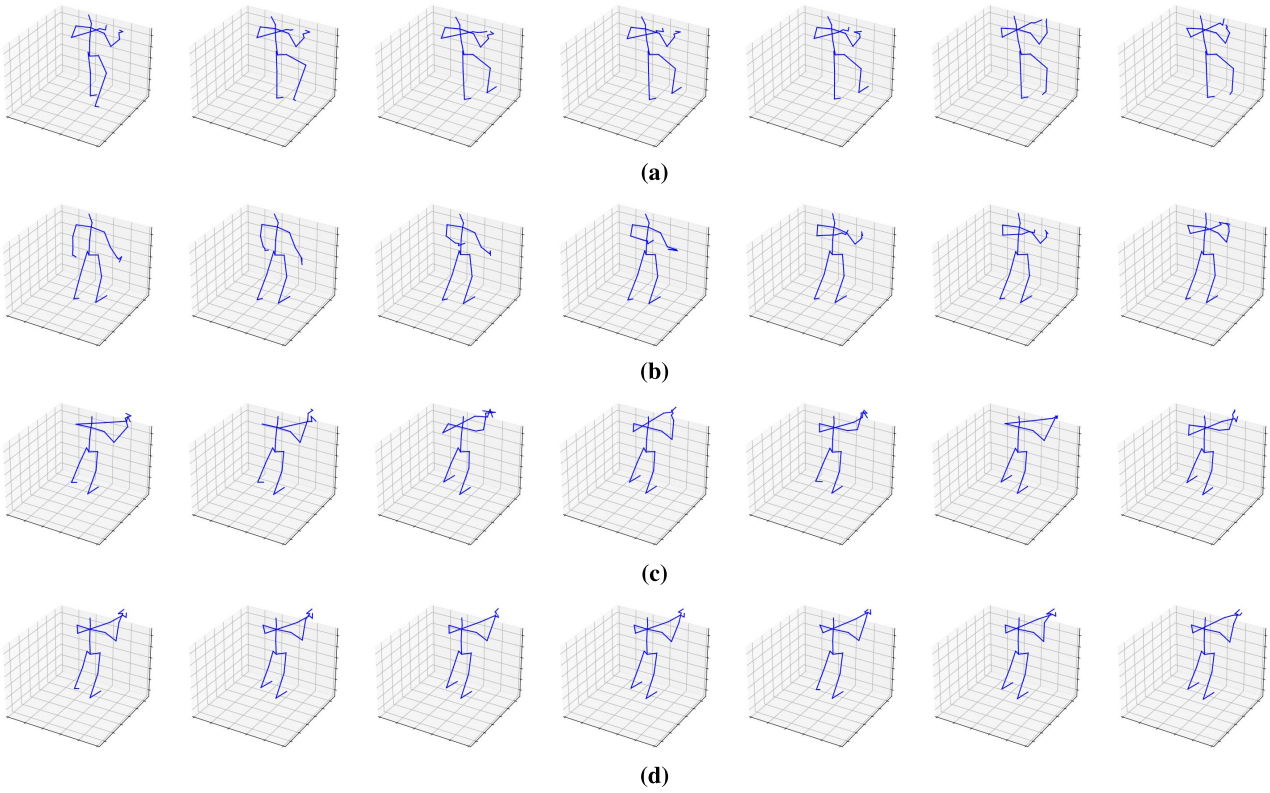


Fig. 4. Visualization examples of confusing actions. The action that the network gets most confused about has changed after employing angular encoding as a part of input features. (a) Folding paper. (b) Counting money. (c) Reading. (d) Writing.

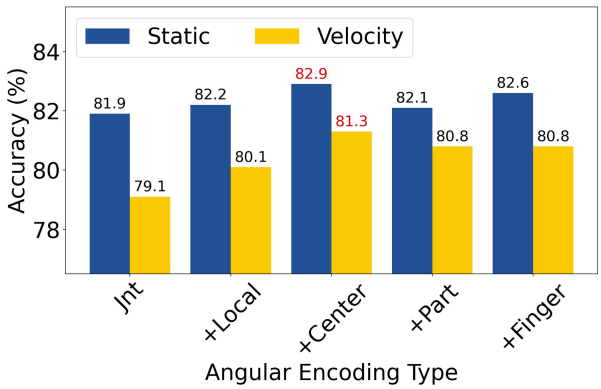


Fig. 5. Accuracy of recognizing skeleton-based actions using the multiscale GCN with different types of angular encoding. Both static and velocity domains are considered. The best accuracy of each domain is highlighted in red.

3) 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 investigate how each kind of angular encoding improves 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 VI. We see the following.

1) Equipping the velocity features with angular encoding boosts substantial accuracy for 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."
2) 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, finger-based encoding increases accuracy for hand-related actions, while part-based encoding benefits the actions heavily using arms and legs.

VI. Generalizability of Angular Encoding

A possible concern is the generalizability 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: 1) ShiftGCN [5]; 2) DecoupleGCN [4]; and 3) MSG3D [18]. The utilized dataset is the cross-subject setting of NTU120.

We display the results in Fig. 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

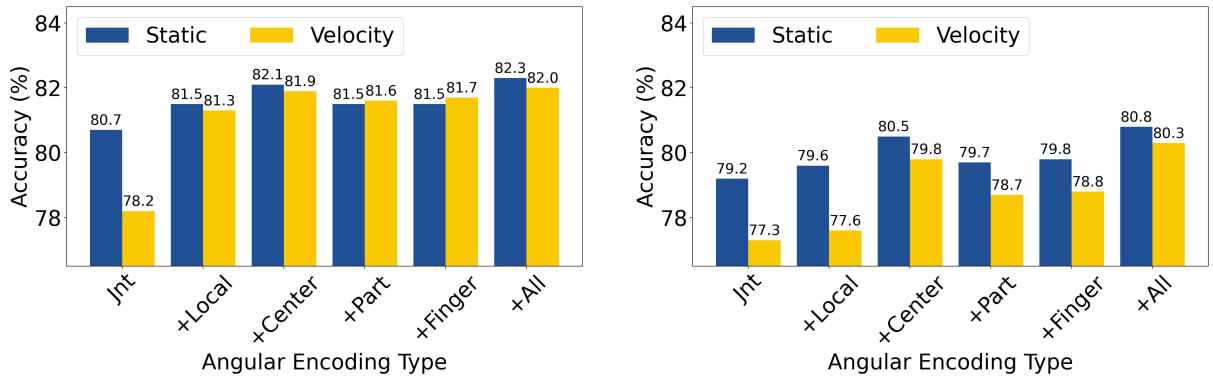


Fig. 6. Accuracy of recognizing skeleton-based actions using DecoupleGCN (left) and ShiftGCN (right) with different types of angular encoding. Both static and velocity domains are considered. The column All represents concatenating all types of angular encoding.

improves the accuracy of 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. These scenarios imply that using features in the velocity domain surpasses using the static joints.

VII. Discussion

As we have described in Section I, current GCNs are designed to extract features between two adjacent nodes. On the other hand, the angular features are higher-order ones beyond two adjacent vertices. We can theoretically view every angle as a hyperedge $e(v_1, v_2, v_3)$, where v_1, v_2, and v_3 are the constitutional joints of an angle. The angular encoding is their associated feature. The angular encoding extends the capability of existing GNNs to capture features of hyperedges.

From the perspective of treating a skeleton as a hypergraph, we have proposed four categories of hyperedges. In contrast, existing work that also makes use of angle features only contains one type of hyperedges.

VIII. Conclusion

To extend the capacity of GCNs in extracting body structural information, we propose higher-order representations 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, that is, 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.

Appendix

We provide the improvement of accuracy by angular encoding for each class. The results for the static domain are in Tables VII and VIII. The ones for the velocity domain are in Tables IX and X.

References

[1] S. Anwar and N. Barnes, "Densely residual Laplacian super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1192–1204, Mar. 2022.

[2] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4467–4475.

[3] Z. Chen, S. Li, B. Yang, Q. Li, and H. Liu, "Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 35, 2021, pp. 1113–1122.

[4] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling GCN with dropgraph module for skeleton-based action recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 536–553.

[5] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 183–192.

[6] Y. Diao, T. Shao, Y.-L. Yang, K. Zhou, and H. Wang, "BASAR: Black-box attack on skeletal action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7597–7607.

[7] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang, "Open set domain adaptation: Theoretical bound and algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4309–4322, Oct. 2021.

[8] K. Hu, Y. Ding, J. Jin, L. Weng, and M. Xia, "Skeleton motion recognition based on multi-scale deep spatio-temporal features," *Appl. Sci.*, vol. 12, no. 3, p. 1028, Jan. 2022.

[9] P. Koniusz, A. Cherian, and F. Porikli, "Tensor representations via kernel linearization for action recognition from 3D skeletons," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer, 2016, pp. 37–53.

[10] P. Koniusz, L. Wang, and A. Cherian, "Tensor representations for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 648–665, 2021.

[11] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–8.

[12] D. Li, C. R. Opazo, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1459–1469.

[13] D. Li, X. Yu, C. Xu, L. Petersson, and H. Li, "Transferring cross-domain knowledge for video sign language recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6205–6214.

[14] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3595–3603.

[15] M. Li, Z. Ma, Y. G. Wang, and X. Zhuang, "Fast Haar transforms for graph neural networks," *Neural Netw.*, vol. 128, pp. 188–198, Aug. 2020.

[16] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.

[17] Y. Liu *et al.*, "Invertible denoising network: A light solution for real noise removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021.

[18] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020.

[19] R. Memmesheimer, S. Haring, N. Theisen, and D. Paulus, "Skeleton-DML: Deep metric learning for skeleton-based one-shot action recognition," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3702–3710.

[20] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 498–511, Mar. 2009.

[21] Y. Min, F. Wenkel, and G. Wolf, "Scattering GCN: Overcoming oversmoothness in graph convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 1–11.

[22] X. Qin, R. Cai, J. Yu, C. He, and X. Zhang, "An efficient self-attention network for skeleton-based action recognition," *Sci. Rep.*, vol. 12, no. 1, pp. 1–10, Dec. 2022.

[23] Z. Qin, D. Kim, and T. Gedeon, "Neural network classifier as mutual information estimator," in *Proc. Int. Conf. Mach. Learn. (ICML-XAI)*, 2021.

[24] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1010–1019.

[25] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7904–7913.

[26] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12026–12035.

[27] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "AdaSGN: Adapting joint number and model size for efficient skeleton-based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13413–13422.

[28] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1227–1236.

[29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[30] H. Wang *et al.*, "Understanding the robustness of skeleton-based action recognition under adversarial attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14656–14665.

[31] L. Wang, "Analysis and evaluation of Kinect-based action recognition algorithms," M.S. thesis, School Comput. Sci. Softw. Eng., Univ. Western Australia, Perth, WA, USA, 2017.

[32] L. Wang, D. Q. Huynh, and P. Koniusz, "A comparative review of recent Kinect-based action recognition algorithms," *IEEE Trans. Image Process.*, vol. 29, pp. 15–28, 2020.

[33] L. Wang, P. Koniusz, and D. Huynh, "Hallucinating IDT descriptors and I3D optical flow features for action recognition with CNNs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8698–8708.

[34] Y. G. Wang, M. Li, Z. Ma, G. Montufar, X. Zhuang, and Y. Fan, "Haar graph pooling," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 9952–9962.

[35] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[36] C. Xie, C. Li, B. Zhang, C. Chen, J. Han, and J. Liu, "Memory attention networks for skeleton-based action recognition," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–7.

[37] K. Xu, F. Ye, Q. Zhong, and D. Xie, "Topology-aware convolutional neural network for efficient skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 36, 2022, pp. 2866–2874.

[38] S. K. Yadav, K. Tiwari, H. M. Pandey, and S. A. Akbar, "Skeleton-based human activity recognition using ConvLSTM and guided feature learning," *Soft Comput.*, vol. 26, no. 2, pp. 877–890, Jan. 2022.

[39] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018.

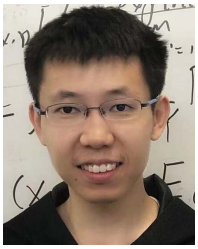
[40] K. Yao, J. Liang, J. Liang, M. Li, and F. Cao, "Multi-view graph convolutional networks with attention mechanism," *Artif. Intell.*, vol. 307, Jun. 2022, Art. no. 103708.

[41] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1112–1121.

[42] Y. Zhang, F. Liu, Z. Fang, B. Yuan, G. Zhang, and J. Lu, "Clarinet: A one-step approach towards budget-friendly unsupervised domain adaptation," 2020, *arXiv:2007.14612*.

[43] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution LSTM for skeleton based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6882–6892.

[44] L. Zhong, Z. Fang, F. Liu, J. Lu, B. Yuan, and G. Zhang, "How does the combined risk affect the performance of unsupervised domain adaptation approaches?" 2020, *arXiv:2101.01104*.

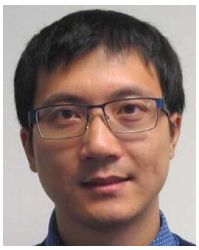


Zhenyue Qin received the bachelor's degree (Hons.) from Australian National University, Canberra, ACT, Australia, in 2017, where he is currently pursuing the Ph.D. degree.

His research interests include human action recognition, graph neural networks, and human-centered computing.

Yang Liu (Member, IEEE) received the bachelor's and master's degrees from Sun Yat-sen University, Guangzhou, China, in 2012, and Peking University, Beijing, China, in 2016, respectively. She is currently pursuing the Ph.D. degree with Australian National University, Canberra, ACT, Australia.

Her research interests include image restoration, human action recognition, and human-centered computing.



Pan Ji received the Ph.D. degree from Australian National University, Canberra, ACT, Australia, in 2016.

From September 2020 to June 2022, he was a Senior Staff Research Engineer and the Manager at OPPO U.S. Research Center, InnoPeak Technology Inc., Palo Alto, CA, USA. Previously, he worked as a Researcher at NEC Laboratories America, Princeton, NJ, USA, from February 2018 to September 2020. Before moving to the U.S. in February 2018, he has been working as an ARC Senior Research Associate (Post-Doctoral Researcher) at The University of Adelaide, Adelaide, Australia, since July 2016. He is currently the Director of the visual perception at the XR Vision Laboratories, Tencent, Shenzhen, China. His research interests lie in computer vision (especially 3-D vision), unsupervised learning (e.g., clustering), and various other aspects of machine learning.

Dr. Ji received the Best Student Paper Award at the International Conference on Image Processing (ICIP) in 2014.

Dongwoo Kim received the Ph.D. degree from KAIST, Daejeon, South Korea, in 2015.

He is currently an Assistant Professor at the Department of Computer Science and Engineering and the Graduate School of Artificial Intelligence, POSTECH, Pohang, South Korea. Before POSTECH, he worked as a Lecturer and a Research Fellow at Australian National University, Canberra, ACT, Australia.

Lei Wang (Student Member, IEEE) received the M.E. degree in software engineering from The University of Western Australia (UWA), Perth, WA, Australia, in 2018. He is currently pursuing the Ph.D. degree with Australian National University (ANU), Canberra, ACT, Australia, and Data61/CSIRO, Canberra.

He was a Visiting Researcher at the Machine Learning Research Group, Data61/CSIRO (former NICTA). He was also a Visiting Researcher with the Department of Computer Science and Software Engineering, UWA. Since 2018, he has been a full-time Computer Vision Researcher with iCetana Pty Ltd., Perth. His research interests include action recognition, anomaly detection, computer vision, and machine learning.

Mr. Wang is an ACM Student Member.

R. I. (Bob) McKay is an Honorary Professor at Australian National University, Canberra, ACT, Australia. Before that, he was a Professor at Seoul National University, Seoul, South Korea.

Prof. McKay was the Program Co-Chair of the 2002 Australian Conference on Artificial Intelligence and the General Chair of the 2003 IEEE Congress on Evolutionary Computation. He has been an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and *Genetic Programming and Evolvable Machines* (Springer).

Saeed Anwar received the master's degree in Erasmus mundus vision and robotics (Vibot), Heriot-Watt University, Universitat de Girona, and Université de Bourgogne, and the Ph.D. degree from Australian National University (ANU), Canberra, ACT, Australia, in 2012 and 2018, respectively.

He is currently a Research Scientist with the Commonwealth Scientific and Industrial Research Organization (CSIRO), Canberra, and a Lecturer at ANU. He also holds honorary positions, such as a Visiting Fellow at the University of Technology Sydney (UTS), Ultimo, NSW, Australia, and an Assistant Professor with the University of Canberra, Canberra. He has a strong teaching experience in reputed universities and a substantial industry presence. Moreover, he has published in top-tier conferences and journals, including one best paper nomination in CVPR. He leads commercial projects and supervises B.S., M.S., and Ph.D. students.

Tom Gedeon (Senior Member, IEEE) received the B.Sc. (Hons.) and Ph.D. degrees from the University of Western Australia, Perth, WA, Australia.

He holds the Optus Chair in AI and the Director of the Optus Centre for AI, Curtin University, Perth. Before this, he was a Professor of computer science and the former Deputy Dean of the College of Engineering and Computer Science, Australian National University, Canberra, ACT, Australia. He remains an Honorary Professor at ANU. He has over 400 publications. His main research interests include responsive and responsible AI and underpinned by edge computing efficient AI. His focus is on the development of automated systems for information extraction, from eye gaze and physiological data, as well as textual and other data, and for the synthesis of the extracted information into humanly useful information resources, primarily using neural/deep networks and fuzzy logic methods.

Prof. Gedeon has run a number of international conferences. He is the former President of the Asia Pacific Neural Network Assembly and the Computing Research and Education Association of Australasia. He is currently a member of the Australian Research Council's Medical Research Advisory Group. He has been nominated for VC's awards for postgraduate supervision at three universities. He has been the General Chair of the International Conference on Neural Information Processing (ICONIP) three times. He is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the *Neural Networks* (INNS/Elsevier).