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-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.

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.
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 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 |
|--------------------------|------|-------|-------|---------|-------------|--------|
| HCN [11]                 | 2018 | 1     | 86.5  | -       | -           | -      |
| MAN [56]                 | 2018 | 1     | 82.7  | -       | -           | -      |
| ST-GCN [39]              | 2018 | 1     | 81.5  | -       | -           | -      |
| AS-GCN [14]              | 2019 | 1     | 86.8  | -       | -           | -      |
| AGC-LSTM [28]            | 2019 | 2     | 89.2  | -       | -           | -      |
| 2s-AO-CNN [26]           | 2019 | 4     | 88.5  | -       | -           | -      |
| DGNN [25]                | 2019 | 4     | 89.9  | -       | -           | -      |
| Bayes-GCN [43]           | 2019 | 1     | 81.8  | -       | -           | -      |
| SGN [41]                 | 2020 | 1     | 89.0  | -       | -           | -      |
| DeCoupleGCN [4]          | 2020 | 4     | 90.8  | 96.6    | -           | -      |
| MS-GSD [18]              | 2020 | 2     | 91.5  | 96.2    | -           | -      |
| MST [3]                  | 2021 | 2     | 91.1  | 96.4    | -           | -      |
| AdaS-GCN [27]            | 2021 | 4     | 90.5  | 95.3    | -           | -      |
| Ta-CNN [37]              | 2022 | 2     | 90.7  | 95.1    | -           | -      |
| Efficient-Self-Attention [22] | 2022 | 4     | 90.5  | 96.1    | -           | -      |

| Our Methods              |      |       |       |         |     |     |
|--------------------------|------|-------|-------|---------|-----|-----|
| BSL-S (Joint)            | -    | 1     | 87.2  | 93.7    | -   | -   |
| AGE-S (Joint)            | -    | 1     | 88.7  | 94.5    | 0.8 | -   |
| BSL-S (Bone)             | -    | 1     | 88.2  | 93.6    | -   | -   |
| AGE-S (Bone)             | -    | 1     | 89.2  | 94.8    | 1.2 | -   |
| BSL-V (Joint)            | -    | 1     | 86.0  | 93.3    | -   | -   |
| AGE-V (Joint)            | -    | 1     | 88.2  | 94.5    | 1.2 | -   |
| BSL-V (Bone)             | -    | 1     | 88.0  | 94.8    | 2.1 | -   |
| BSL-S (Joint+Bone)       | -    | 1     | 90.0  | 95.1    | -   | -   |
| BSL-V (Joint+Bone)       | -    | 1     | 86.2  | 92.6    | -   | -   |
| ACE-V (Joint+Bone)       | -    | 1     | 87.1  | 94.0    | 1.4 | -   |
| BSL-Ens: S(J)&V(J)       | -    | 2     | 89.3  | 94.7    | -   | -   |
| AGE-Ens: S(J)&V(J)       | -    | 2     | 90.5  | 95.5    | 0.8 | -   |
| BSL-Ens: S(B)&V(B)       | -    | 2     | 90.5  | 94.7    | -   | -   |
| AGE-Ens: S(B)&V(B)       | -    | 2     | 90.8  | 95.5    | 0.8 | -   |
| BSL-Ens: S(J+B)&V(J+B)   | -    | 2     | 90.5  | 95.7    | -   | -   |
| AGE-Ens: S(J+B)&V(J+B)   | -    | 2     | 91.0  | 96.1    | 0.4 | -   |
| BSL-Ens: S(B)+S(J+B)&V(J+B) | 3     | 3     | 90.7  | 95.7    | -   | -   |
| AGE-Ens: S(B)+S(J+B)&V(J+B) | 3     | 3     | 91.4  | 96.3    | 0.6 | -   |
| BSL-Ens: S(J)&S(B)+S(J+B)&V(J+B) | -    | 4     | 90.9  | 95.9    | -   | -   |
| AGE-Ens: S(J)&S(B)+S(J+B)&V(J+B) | -    | 4     | 91.6  | 96.3    | 0.4 | -   |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE II
EVALUATION RESULTS ON ENSEMBLING WITH ANGULAR FEATURES. ENS IS THE ENSEMBLING, INT 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      | 81.90    | 81.90     | 83.53    | 83.53     |
| Jnt      | 79.31    | 79.31     | 81.84    | 81.84     |
| Ens: Jnt & Ang | 83.53 | 83.53 | 83.53 | 83.53 |
| Bon      | 83.53    | 83.53     | 83.53    | 83.53     |
| Ens: Bon & Ang | 83.53 | 83.53 | 83.53 | 83.53 |
| Ens: Jnt+Bon | 83.53 | 83.53 | 83.53 | 83.53 |
| Ens: Jnt+Bon & Ang | 83.53 | 83.53 | 83.53 | 83.53 |

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 (BASKKR) 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, $\vec{b}_{uw_1}$ denotes the vector from joint $u$ to $w_1$ ($i = 1, 2$), 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 $\theta$ 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

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

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.

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

\[ v(t+1)^a_u = d(t+1)^a_u - d^a_u \]  

(2)

where \( v(t+1)^a_u \) 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 \( O(N^3 T) \), 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 neighbor, if the target joint has only one adjacent neighbor, we set its angular feature to zero. When a joint has more adjacent neighbors, the target joint is considered to be in a pair-based angle.

b) Center-oriented: The center-oriented angles are those that are defined relative to a central joint, which is typically the root joint of an action.

c) Pair-based angles: These are defined between pairs of joints, with one joint being the target and the other being the anchor.

d) Finger-based: These angles are defined specifically for finger movements, where the target joint is usually a finger joint and the anchor joint is a neighboring joint.
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.

| 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        |

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 $1 \times 1$ 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.

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

TABLE III

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

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.
| Action                              | Joint         | Concatenation: Joint + Angular |
|------------------------------------|---------------|---------------------------------|
| make victory sign                  | 18.48         | 53.04                           |
| staple book                        | 26.67         | 34.57                           |
| writing                            | 28.41         | 10.46                           |
| counting money                     | 48.47         | 5.17                            |
| playing with phone/tablet          | 48.82         | 10.82                           |
| hit other person with something    | 49.52         | 10.92                           |
| blow nose                          | 55.35         | 4.30                            |
| ball up paper                      | 56.57         | 6.22                            |
| cutting paper (using scissors)     | 58.34         | 64.10                           |
| make victory sign                  | 58.65         | 72.35                           |
| hush (quite)                       | 59.00         | 67.65                           |
| shake fist                         | 59.10         | 65.51                           |
| open a box                         | 59.98         | 71.60                           |
| taking a selfie                    | 64.58         | 79.71                           |
| touch head (headache)              | 64.58         | 71.74                           |
| open bottle                        | 67.82         | 72.30                           |
| staple book                        | 68.28         | 70.16                           |

**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↑ ≥ 10%) due to using our angular features.

### 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 113,945 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 (%) | Similar Action |
| 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 |
| folding paper | 56.57 | ball playing paper | 66.61 | 10.04 | ball playing paper |
| open a box | 59.98 | fold paper | 68.47 | 8.49 | fold paper |

**Velocity**

| Action | Joint | Concatenation: Joint + Angular |
|--------|-------|--------------------------------|
|        | Acc (%) | Similar Action | Acc (%) | Similar Action |
| 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) |

**Static**

| Action | Joint | Concatenation: Joint + Angular |
|--------|-------|--------------------------------|
| make victory sign | 18.48 | thumb up | 40.35 | 21.87 | make ok sign |
| playing with phone/tablet | 48.82 | play magic cube | 64.36 | 19.55 | 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 |

**Velocity**

| Action | Joint | Concatenation: Joint + Angular |
|--------|-------|--------------------------------|
| cutting paper (using scissors) | 27.27 | staple book | 58.12 | 30.84 | staple book |
| make ok sign | 39.73 | writing | 56.73 | 17.00 | staple book |
| play magic cube | 45.50 | counting money | 61.19 | 15.69 | counting money |
| 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 |

**Static**

| Action | Joint | Concatenation: Joint + Angular |
|--------|-------|--------------------------------|
| 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 | 73.52 | thumb up | 83.48 | 7.96 | thumb up |

**Velocity**

| Action | Joint | Concatenation: Joint + Angular |
|--------|-------|--------------------------------|
| cutting paper (using scissors) | 27.27 | staple book | 59.34 | 32.06 | staple book |
| play magic cube | 39.73 | writing | 71.27 | 31.55 | typing on a keyboard |
| pointing to something with finger | 56.45 | writing | 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 |

**Static**

| Action | Joint | Concatenation: Joint + Angular |
|--------|-------|--------------------------------|
| 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 |
| play magic cube | 48.82 | play magic cube | 61.45 | 12.64 | play magic cube |
| pointing to something with finger | 72.36 | hit other person with something | 82.85 | 10.49 | wield knife towards other person |
| fold paper | 56.57 | ball playing paper | 63.57 | 9.00 | ball playing paper |
| play magic cube | 62.81 | counting money | 71.15 | 8.34 | playing with phone/tablet |
| ride bike | 84.89 | kicking something | 93.21 | 8.32 | kicking something |

**Velocity**

| Action | Joint | Concatenation: Joint + Angular |
|--------|-------|--------------------------------|
| 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 |
| nauseate or vomiting condition | 75.36 | touch chest (stomachache/heart pain) | 84.36 | 9.00 | touch chest (stomachache/heart pain) |
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 \( \text{Ang} \). 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 \( \text{Ang} \) 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 \( \text{Ang} \). We note that ensembling \( \text{Ang} \) 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

| Action | Joint | Similar Action | Concatenation: Joint + Angular | Similar Action |
|--------|-------|----------------|-------------------------------|---------------|
| make victory sign | 18.88 | thumb up | 34.87 | make ok sign |
| staple book | 26.67 | staple book | 10.46 | cutting paper (using scissors) |
| writing | 28.41 | typing on a keyboard | 20.49 | typing on a keyboard |
| counting money | 48.47 | play magic cube | 4.51 | play magic cube |
| playing with phone/tablet | 48.82 | play magic cube | 10.82 | writing |
| wield knife towards other person | 49.52 | hit other person with something | 12.98 | hit other person with something |
| blow nose | 55.35 | yawn | 4.30 | yawn |
| fold paper | 56.57 | ball up paper | 6.22 | counting money |
| reading | 58.34 | cutting paper (using scissors) | 5.76 | writing |
| thumb up | 58.65 | make victory sign | 13.70 | make victory sign |
| yawn | 59.00 | bush (quite) | 8.65 | brush (quite) |
| snapping fingers | 59.10 | shake fist | 6.40 | make victory sign |
| open a box | 59.98 | fold paper | 11.63 | open bottle |
| pointing to something with finger | 64.58 | taking a selfie | 15.13 | taking a selfie |
| sneeze/cough | 64.58 | touch head (headache) | 7.16 | touch head (headache) |
| apply cream on hand back | 67.82 | open bottle | 4.48 | rub two hands together |
| cutting paper (using scissors) | 68.28 | staple book | 1.87 | staple book |
| typing on a keyboard | 68.45 | cutting paper (using scissors) | 0.64 | writing |
| brush (quite) | 69.16 | yawn | 2.92 | yawn |
| ball up paper | 69.26 | fold paper | 2.04 | fold paper |
| eat meal/snack | 69.55 | brushing teeth | 1.73 | brushing teeth |
| wear a shoe | 70.43 | take off a shoe | 14.92 | take off a shoe |
| take off a shoe | 70.90 | wear a shoe | 10.85 | wear a shoe |
| punch/scratching other person | 72.36 | hit other person with something | 10.49 | hit other person with something |
| open bottle | 73.17 | play magic cube | 5.87 | take something out of a bag |
| put something into a bag | 73.26 | take something out of a bag | 1.69 | snapping fingers |
| shake fist | 74.69 | hand waving | 6.80 | touch neck (neckache) |
| touch head (headache) | 75.45 | drink water | 5.35 | pointing to something with finger |
| thumb down | 75.52 | thumb up | 8.07 | point down |
| sniff (smell) | 76.04 | blow nose | 5.00 | blow nose |
| make a phone call/answer phone | 80.09 | reading | 7.55 | playing with phone/tablet |
| apply cream on face | 80.71 | wipe face | 2.39 | wipe face |
| rub two hands together | 80.88 | clapping | 1.72 | apply on hand back |
| touch neck (neckache) | 80.88 | drink water | 6.43 | flick hair |
| nausea or vomiting condition | 81.18 | sneeze/cough | 3.55 | touch chest (stomachache/heart pain) |
| drink water | 81.48 | brushing teeth | 2.46 | brushing teeth |
| move heavy objects | 81.57 | carry something with other person | 4.52 | carry something with other person |
| take something out of a bag | 81.64 | put something into a bag | 3.26 | put something into a bag |
| brushing teeth | 81.78 | brush water | 5.76 | touch head (headache) |
| drop | 81.91 | staple book | 3.91 | tear up paper |
| put the palms together | 81.97 | cross hands in front (say stop) | 10.78 | sniff (smell) |
| point finger at the other person | 81.97 | put back of other person | 6.80 | put back of other person |
| use a fan (with hand or paper/feeling warm) | 82.27 | hand waving | 7.55 | shake fist |
| check time (from watch) | 82.33 | open bottle | 8.25 | put the palms together |
| support somebody with hand | 82.65 | follow other person | 6.04 | knock over other person (hit with body) |
| take off headphones | 82.92 | take off glasses | 3.47 | take off glasses |
| tennis bat swinging | 83.13 | throw up cap/hat | 0.30 | throw up cap/hat |
| take off glasses | 83.67 | take off headphone | 9.39 | take off headphone |
| knock over other person (hit with body) | 83.72 | whisper in other person’s ear | 5.17 | whisper in other person’s ear |
| wipe face | 83.78 | brushing hair | 3.90 | brushing hair |
| reach into pocket | 84.04 | touch back (backache) | 1.36 | typing on a keyboard |
ensembleing 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

1In 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 ensembleing 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 VIII**

**Static: Second Half**

| Action                              | Joint (%) | Similar Action | Concentration: Joint + Angular (%) | Similar Action |
|-------------------------------------|-----------|----------------|------------------------------------|----------------|
| put on headband                     | 84.20     | wear on glasses | 87.52                             | 3.32 take off headband |
| throw                               | 84.42     | 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                     | 85.04     | drink water    | 88.28                             | 3.20 eat milk/milk |
| nod headbow                         | 85.23     | nausea or vomiting condition | 94.93                       | 4.97 nausea or vomiting condition |
| kicking other person                | 85.59     | step on foot   | 91.30                             | 5.71 punching/lashing other person |
| touch chest (stomach/heart pain)    | 85.96     | touch back (backpack) | 91.30                      | 5.35 touch back (backpack) |
| 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 two 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 (headaches) |
| 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                        | 87.73     | 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                          | 87.84     | cross hands in front (say stop) | 94.09                  | 5.76 put the palms together |
| stretch oneself                     | 88.95     | 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 hair/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 |
| hand up (both hands)               | 90.81     | stretch oneself | 94.23                             | 3.44 stretch oneself |
| take off a bat/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 |
| fall down                           | 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 put back on other person |
| touch back (backpack)               | 91.39     | touch chest (stomach/heart pain) | 94.20                  | 2.81 touch chest (stomach/heart pain) |
| pinion down                         | 91.67     | take off a shirt | 95.97                             | 4.30 nausea or vomiting condition |
| standing up (from sitting position) | 91.67     | touch back (backpack) | 92.00                  | 0.27 make victory sign |
| shake head                          | 91.73     | walking apart from each other | 98.91                     | 7.16 follow other person |
| staggering                          | 91.75     | side kick      | 94.43                             | 2.57 side kick    |
| butt kicks (kick backward)          | 91.86     | sitting down   | 96.86                             | 4.14 falling      |
| squat down                          | 92.73     | cross hands in front (say stop) | 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) | 93.20 | shake fist | 94.97                             | 2.04 shake fist    |
| kicking something                   | 93.20     | shake ball     | 94.20                             | 1.00 staggering   |
| hopping (one foot jumping)          | 93.35     | staggering    | 96.00                             | 2.45 kicking something |
| take off jacket                     | 93.93     | tear up paper  | 96.38                             | 2.45 wear jacket   |
| wear on bag                         | 94.64     | tear up paper  | 98.91                             | 4.57 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 | 96.24                     | 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 | 97.37                     | 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   |
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

| Action | Joint | Similar Action | Concactenation: Joint + Angular |
|--------|-------|----------------|-------------------------------|
| make ok sign | 27.17 | make ok sign | 46.78 | make victory sign |
| cutting paper (using scissors) | 27.27 | staple book | 60.38 | staple book |
| staple book | 30.17 | cutting paper (using scissors) | 31.52 | cutting paper (using scissors) |
| playing with phone/tablet | 39.73 | writing | 64.00 | typing on a keyboard |
| play magic cube | 45.50 | counting money | 65.21 | counting money |
| reading | 48.82 | writing | 58.61 | writing |
| counting money | 49.00 | play magic cube | 50.70 | play magic cube |
| blow nose | 52.04 | yawn | 62.96 | yawn |
| thumb up | 53.43 | make victory sign | 63.30 | make victory sign |
| cutting nails | 54.89 | writing | 58.10 | playing with phone/tablet |
| hit other person with something | 55.35 | wield knife towards other person | 58.43 | wield knife towards other person |
| typing on a keyboard | 56.45 | writing | 66.18 | writing |
| open a box | 56.84 | open bottle | 65.16 | open bottle |
| shoot at other person with a gun | 57.78 | point finger at the other person | 63.83 | point finger at the other person |
| fold paper | 57.96 | ball up paper | 65.91 | ball up paper |
| yawn | 58.30 | hush (quite) | 64.00 | hush (quite) |
| wield knife towards other person | 59.76 | hit other person with something | 61.28 | hit other person with something |
| pointing to something with finger | 60.96 | taking a selfie | 72.46 | taking a selfie |
| snapping fingers | 63.46 | shake fist | 65.68 | shake fist |
| open bottle | 64.10 | open a box | 73.65 | open a box |
| sneezegrunt | 65.30 | touch head (headache) | 68.12 | touch head (headache) |
| touch head (headache) | 65.67 | brushing teeth | 73.91 | brushing teeth |
| hush (quite) | 68.46 | yawn | 74.17 | yawn |
| touch neck (neckache) | 71.83 | touch head (headache) | 82.25 | touch head (headache) |
| flick hair | 71.87 | blow nose | 77.74 | blow nose |
| drink tea | 72.72 | brushing teeth | 85.77 | brushing teeth |
| shoot at the basket | 73.65 | throw | 82.34 | throw |
| put something into a bag | 73.78 | take something out of a bag | 78.09 | take something out of a bag |
| shake face | 74.17 | hand waving | 76.22 | hand waving |
| throw up cap/hat | 74.57 | toss a coin | 79.23 | toss a coin |
| wipe face | 75.09 | touch head (headache) | 87.68 | touch head (headache) |
| nausea or vomiting condition | 75.36 | touch chest (stomachache/heart pain) | 80.36 | touch chest (stomachache/heart pain) |
| take off a shoe | 76.74 | wear a shoe | 83.21 | wear a shoe |
| taking a selfie | 77.26 | drink water | 83.33 | drink water |
| knock over other person (hit with body) | 77.65 | wield knife towards other person | 82.12 | wield knife towards other person |
| take something out of a bag | 78.17 | put something into a bag | 82.47 | put something into a bag |
| wear a shoe | 78.49 | take off a shoe | 81.69 | take off a shoe |
| take off headphones | 78.68 | take off glasses | 84.98 | take off glasses |
| make a phone call/answer phone | 79.00 | drink water | 81.82 | drink water |
| thumb down | 79.87 | thumb up | 82.78 | thumb up |
| support somebody with hand | 79.87 | follow other person | 82.26 | follow other person |
| point finger at the other person | 80.16 | pat on back of other person | 88.41 | pat on back of other person |
| reach into pocket | 80.39 | eat meal/snack' | 83.21 | eat meal/snack' |
| drop | 80.45 | check time (from watch) | 83.64 | check time (from watch) |
| pat on back of other person | 81.25 | point finger at the other person | 86.96 | point finger at the other person |
| tear up paper | 82.03 | fold paper | 86.35 | fold paper |
| whisper in other person’s ear | 82.13 | knock over other person (hit with body) | 86.96 | knock over other person (hit with body) |
| throw | 82.27 | tennis bat swing | 88.00 | tennis bat swing |
| brushing teeth | 82.52 | writing | 89.01 | writing |
| put on headphones | 82.56 | wear on glasses | 87.52 | wear on glasses |
| check time (from watch) | 83.42 | eat meal/snack' | 89.49 | eat meal/snack' |
| step on foot | 83.52 | kicking other person | 86.26 | kicking other person |
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
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 constitutive 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, 2020, pp. 536–553.
[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.
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. 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).