Towards To-a-T Spatio-Temporal Focus for Skeleton-Based Action Recognition

Lipeng Ke[1], Kuan-Chuan Peng[2], Siwei Lyu[1]

[1] University at Buffalo, State University of New York
[2] Mitsubishi Electric Research Laboratories
lipengke@buffalo.edu, kpeng@merl.com, siweilyu@buffalo.edu

Abstract

Graph Convolutional Networks (GCNs) have been widely used to model the high-order dynamic dependencies for skeleton-based action recognition. Most existing approaches do not explicitly embed the high-order spatio-temporal importance to joints' spatial connection topology and intensity, and they do not have direct objectives on their attention module to jointly learn when and where to focus on in the action sequence. To address these problems, we propose the To-a-T Spatio-Temporal Focus (STF), a skeleton-based action recognition framework that utilizes the spatio-temporal gradient to focus on relevant spatio-temporal features. We first propose the STF modules with learnable gradient-enforced and instance-dependent adjacency matrices to model the high-order spatio-temporal dynamics. Second, we propose three loss terms defined on the gradient-based spatio-temporal focus to explicitly guide the classifier when and where to look at, distinguish confusing classes, and optimize the stacked STF modules. STF outperforms the state-of-the-art methods on the NTU RGB+D 60, NTU RGB+D 120, and Kinetics Skeleton 400 datasets in all 15 settings over different views, subjects, setups, and input modalities, and STF also shows better accuracy on scarce data and dataset shifting settings.

Introduction

As a fundamental task in computer vision, action recognition (Simonyan and Zisserman 2014; Varol, Laptev, and Schmid 2017; Carreira and Zisserman 2017; Si et al. 2018; Lei et al. 2019; Si et al. 2019; Shi et al. 2020a) has a wide range of applications in human-computer interaction (Liu, Liu, and Chen 2017), video surveillance (Ji et al. 2012), and sports analysis (Herath et al. 2017). Existing action recognition methods can be categorized into video-based (Simonyan and Zisserman 2014) and skeleton-based (Si et al. 2018) which take the video and skeleton sequences as inputs, respectively. In recent years, with the improvement in hardware (*e.g.*, MS Kinect) and skeleton extraction algorithms such as (Cao et al. 2017), the skeleton-based action recognition methods have received more attention for their low dimensional representation and robustness to the background changes (Johansson 1973).

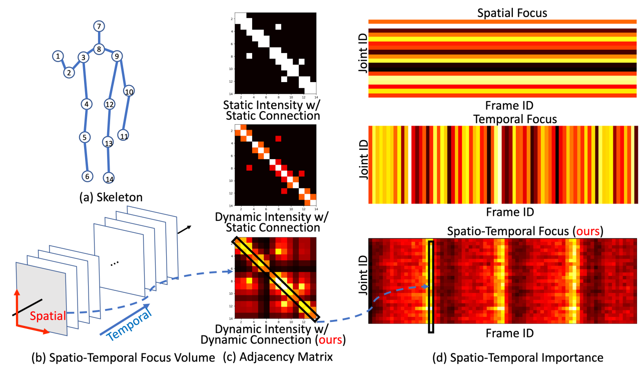

Figure 1: Spatio-temporal focus: (a) is the layout of joints with their ID numbers, (b) *Spatio-temporal focus volume* stacks spatial focus along temporal dimension (c) *Adjacency matrix* models the joints connection topology and connection intensity, the top two rows either model static connection topology or static connection intensity; the bottom row models the dynamic connection topology and dynamic connection intensity; ours further guides the learnable dynamic adjacency matrix with spatio-temporal focus. (d) *Spatio-temproal focus* indicates when (temporal) and where (spatial) to emphasize. The top two rows model the spatial and temporal focus *separately*, the bottom row (ours) models the spatio-temporal focus *jointly*.

Since the action sequence is a time series of human joint locations, we can represent it as a three-dimensional tensor, with the joints layout being the **spatial dimension** and joints movement in time series as the **temporal dimension**. To model the **temporal** information, earlier deep neural network based action recognition methods (*e.g.*, (Liu et al. 2017)) model the *temporal* movement of joints across frames directly using recurrent neural network with long short-term memory. However, these methods do not explicitly consider the *spatial* dependencies among different joints (Figure 1(d), middle-row). Subsequently, many methods (Si et al. 2018; Liu et al. 2020a; Peng et al. 2020; Lei et al. 2019; Li et al. 2019b) model the skeleton topology as graphs and use graph convolutional networks (GCNs) to model the **spatial** connections (topology and intensity) among joints (Figure 1(c), top row). The *static* topological connections

property / method (group)	G_1	G_2	G_3	STF
adjacency matrix w/ **learnable dynamic intensity**	Y	Y	N	Y
adjacency matrix w/ **learnable dynamic topology**	Y	N	N	Y
adjacency matrix w/ **spatio-temporal focus**	N	N	N	Y
attention (modules) w/ **direct supervision**	N	N	N	Y
conjunction of all the above	N	N	N	Y

Table 1: Comparison between STF and recent methods of skeleton-based action recognition. Among all the listed methods, STF models one of the most generalized types of adjacency matrices with explicit backward-knowledge spatio-temporal focus regularization. Method groups: G_1: (Ye et al. 2020; Shi et al. 2021; Chen et al. 2021b; Zeng et al. 2021), G_2: (Song et al. 2020b,a; Yang and Yin 2020; Li et al. 2020; Cheng et al. 2020; Heidari, Alex, and ros Iosifidis 2020a,b; Liu et al. 2020a; Yan, Xiong, and Lin 2018; Yang et al. 2020a; Ding, Yang, and Chen 2019; Fan et al. 2020; Yang et al. 2020b; Li et al. 2019b; Qin et al. 2021; Chen et al. 2021a; Plizzari, Cannici, and Matteucci 2020; Shi et al. 2020a,b; Lei et al. 2019), G_3: (Li, Zhang, and Li 2020; Pan, Chen, and Ortega 2021; Cho et al. 2020; Peng et al. 2020; Si et al. 2018; Liu et al. 2017; Song et al. 2017a; Xie et al. 2018; Huang et al. 2020; Si et al. 2019).

of the joints are captured by the adjacency matrix in Figure 1(c) top row, and the connection intensity learned from data is shown in Figure 1(c) middle row. Both types of methods (Figure 1(c) top & middle rows) model the fixed physically constrained topology connections that cannot be adapted for varying temporal dependencies in the input sequence. Some recent works (Ye et al. 2020; Shi et al. 2021; Chen et al. 2021b; Zeng et al. 2021) model dynamic connection topology from data, but the dynamic connection topology is generated from a forward pass through the model without any objective directly regularizing the adjacency matrices using spatio-temporal focus.

Spatio-temporal modeling Since not all joints and frames are equally important for the recognition (Shi et al. 2021), and only specific joints (**spatial**) with specific motion (**temporal**) are critical to distinguish different action classes (Ding, Yang, and Chen 2019), finding these critical joints (spatial) and the motion patterns (temporal) *jointly* in skeleton sequences is important for action recognition. However, most existing methods (Shi et al. 2020a; Si et al. 2019; Cho et al. 2020; Xie et al. 2018) simply create the attention modules using trainable parameters, which do not have the objectives to directly enforce the modules to capture the varying spatial and temporal patterns jointly.

To better guide the classifier about when and where to look at jointly, and model the learnable dynamic joints connection in spatio-temporal domain, we propose the *To-a-T Spatio-Temporal Focus* method (termed as STF), which uses the joint spatial and temporal information jointly (shown in Figure 1(c) bottom). Specifically, we extract the spatio-temporal focus that strongly influences the recognition in training by projecting the backward gradient to the spatio-temporal domain. Then we propose the STF modules to generate dynamic instance-dependent adjacency matrices, and we use the obtained spatio-temporal focus to regularize the dynamic adjacency matrices, such that the matrices reflect not only the high-order topology connections but also the spatio-temporal importance given the input skeleton sequence. Besides, we also use the gradient-based spatio-temporal focus to encourage the classifier to better emphasize on the critical spatio-temporal inputs and features. To achieve these goals, we introduce three loss terms: (1) the STF exploration loss to enforce the classifier to make prediction over *all* the critical joints; (2) the STF divergence loss to minimize the similarity of the focus for different classes, and (3) the STF coherence loss to focus on consistent spatio-temporal features across the stacked STF modules. Our proposed STF method outperforms the state-of-the-art skeleton-based action recognition methods in all 15 settings over different views, subjects, setups, and input modalities on the NTU RGB+D 60, NTU RGB+D 120, and Kinetics Skeleton 400 datasets.

Our contributions are summarized below:

- We propose the To-a-T Spatio-Temporal Focus framework (STF) as a flexible framework trained with spatio-temporal gradient for skeleton-based action recognition.
- We design the novel STF module that generates dynamic connection topology and intensity, and propose a loss to incorporate the spatio-temporal focus to regularize the spatio-temporal connection topology and intensity.
- We propose three loss terms defined on the gradient-based spatio-temporal focus to explicitly guide the classifier when and where to look at, distinguish confusing classes, and optimize the stacked STF modules.
- Our proposed STF framework outperforms the SOTA not only in three benchmarks but also in the scarce data and dataset shifting settings.

Related Works

Graph neural network for action recognition. Graph convolutional networks (GCNs) (Zhou et al. 2018) are a type of GNNs that extend convolutional operations to the adjacency matrices of structured data types that can be modeled as graphs. GCNs have been adapted widely to a number of applications including graph classification/embeding/prediction, link prediction, node classification etc. The first work applying GCN to skeleton-based action recognition is ST-GCN (Yan, Xiong, and Lin 2018), which constructs a spatio-temporal graph with the joints as the nodes and skeletal connectivity as the edges to model joint dependencies. However, ST-GCN only considers joints that are directly connected in the skeleton, which limits its representation capacity. Subsequently, multi-scale GCNs (Li et al. 2019b; Liu et al. 2020a) are proposed to capture dependencies among joints that are not neighbors in the skeleton graph. These methods use higher order polynomials of the adjacency matrix to aggregate features from non-adjacent joints. 2s-GCN (Lei et al. 2019) further adapts the adjacency matrix to model the **learnable dynamic intensity** of the joints connection using an embedding function. 2s-GCN also popularizes the use of multi-stream inputs, such as joint, bone, joint motion, bone motion, angular, etc., for skeleton-based action recognition (Lei et al. 2019; Shi et al. 2020b; Qin et al. 2021). A common drawback of these works is that they use the same adjacency matrices for different inputs, and the adjacency matrices they use only model the joints' physical topology connection, which limits their adaptivity to largely different and dynamic actions. Recently, Dynamic-GCN (Ye et al. 2020) introduces the Context-Encoding-GCN to learn skeleton topology automatically, but existing methods with **learnable dynamic topology** (Ye et al. 2020; Shi et al. 2021; Chen et al. 2021b; Zeng et al. 2021) (Table 1 G_1) do not have objectives directly enforced on their adjacency matrices to ensure that the spatio-temporal focus is correct. Thus we design the STF module that models dynamic connection topology and intensity with spatio-temporal focus that is embedded into the dynamic adjacency matrix for explicit enforcement.

Attention mechanism for action recognition. In parallel with the GCN approaches, the attention mechanism has been widely used in skeleton-based action recognition (Cho et al. 2020; Ding, Yang, and Chen 2019; Fan et al. 2020; Xie et al. 2018; Liu et al. 2017; Song et al. 2017b) since it was introduced to this task by Liu *et al.* (Liu et al. 2017) and Song *et al.* (Song et al. 2017b). The attention scheme can selectively focus on discriminative sets of joints within each frame of the inputs, and it can also exert different levels of influence on the outputs of different frames. Xie *et al.* (Xie et al. 2018) proposed the Temporal Attention Recalibration Module and Spatio-Temporal Convolution Module to re-calibrate the temporal and spatio-temporal attention across the frames, respectively. Fan *et al.* (Fan et al. 2020) further proposed a method consisting of a self-attention branch and a cross-attention branch to utilize the scenario context information. Ding *et al.* (Ding, Yang, and Chen 2019) improved the spatio-temporal attention by using an end-to-end attention-enhanced recurrent GCN to pay different levels of attention to different temporal and spatial joints. To capture long-term spatio-temporal relationships, Cho *et al.* (Cho et al. 2020) proposed the Self-Attention Network to extract high-level semantics capturing long-range correlations.

However, the aforementioned attention mechanisms have two common problems. They do not consider spatial and temporal dependencies jointly, and they only receive implicit supervision from the final classification loss. These issues limit the accuracy of the attention-based methods when they are applied to the sequences corresponding to complex actions. To address these problems, we propose three loss terms defined on the gradient-based spatio-temporal focus to explicitly guide the classifier when and where to look at, distinguish confusing classes, and optimize the stacked STF modules.

We summarize the major weaknesses of existing works in Table 1 to highlight our motivation and novelty. Our STF is the only one with direct supervision to extract gradient-based spatio-temporal attention, and we use the spatio-temporal focus to regularize the adjacency matrix with learnable dynamic topology and intensity of joint connection.

Our Method — To-a-T Spatio-Temporal Focus

We propose a new framework, To-a-T Spatio-Temporal Focus (STF), for skeleton-based action recognition. STF utilizes the spatio-temporal focus acquired by backward gradients to improve the graph modeling in typical GCN and capture critical spatio-temporal features. We first propose the STF modules, which generate learnable and instance-dependent adjacency matrices. Then we use the gradient-based spatio-temporal focus to supervise the STF modules, such that the adjacency matrices capture both the high-order dynamic dependency and encode spatio-temporal importance. Second, we propose the STF exploration loss, STF divergence loss, and STF coherence loss to explicitly enforce the classifier to predict based on all critical joints and frames across the input, to focus on discriminative spatio-temporal features of confusing classes, and to have consistent spatio-temporal attention across the stacked STF modules, respectively. With these proposed losses, STF is better guided when and where to look at, distinguishing confusing classes, and distilling the high-level module's focus to low-level modules.

Spatio-Temporal Focus (STF) Module

A human skeleton graph is defined as (V, E), where $V = \{v_1, ..., v_N\}$ is the set of N nodes representing the joints, and E is the edge set representing the bones by an adjacency matrix $A \in R^{N\times N}$. The input skeleton sequence is formulated as a tensor $X \in R^{C\times T\times N}$, where C, T, and N are the joint feature dimension (for 2D/3D input, C=2/3), number of frames, and number of joints, respectively.

In the vanilla adjacency matrix (Figure 1(c) top row), $A_{i,j} = 1$ if the joints corresponding to nodes v_i and v_j are connected in the skeleton, and 0 otherwise. A is symmetric since the graph is undirected. We build up our STF module upon (Liu et al. 2020a), which encodes multi-scale adjacency matrices A_k. A_k considers connections via fewer than k intermediate joints as spatial connections at the scale of k. Since A_k is still constrained by the natural topology of human skeleton, which cannot model the high-order relationship in the spatio-temporal domain, we propose an instance-dependent and learnable dynamic adjacency matrix G_k (Figure 1(c) bottom row) to model the high-order dependency along with A_k. The features of the STF modules are computed as:

$$X^{l+1} = \sum_{k=1}^{K} W_k X^l (A_k + G_k) \tag{1}$$

where $K = 3$ denotes the scale size of the spatial dimension, W_k is the weight of the convolution operation, X^l is the feature at the l-th layer, and G_k is our proposed instance-dependent adjacency matrix generated from X^l.

As shown in Figure 2(d), STF firstly uses a three-layer convolutional module $\beta(\cdot)$ to extract the spatio-temporal embedding $\beta(X^l)$ of dimension $T\times N$ on input X^l. Then we apply a CNN with two convolutional layers $\alpha(\cdot)$ to convert the embedding to $G_k = \alpha(\beta(X^l))$ of dimension $N \times N$. Now the embedding $\beta(X^l)$ encodes a data-dependent graph that learns a unique graph for each sample via $\alpha(\cdot)$, but it does not guarantee that $\beta(X^l)$ can represent the high-order topology and importance in the spatio-temporal domain. To address this issue, we propose to use backward gradient to

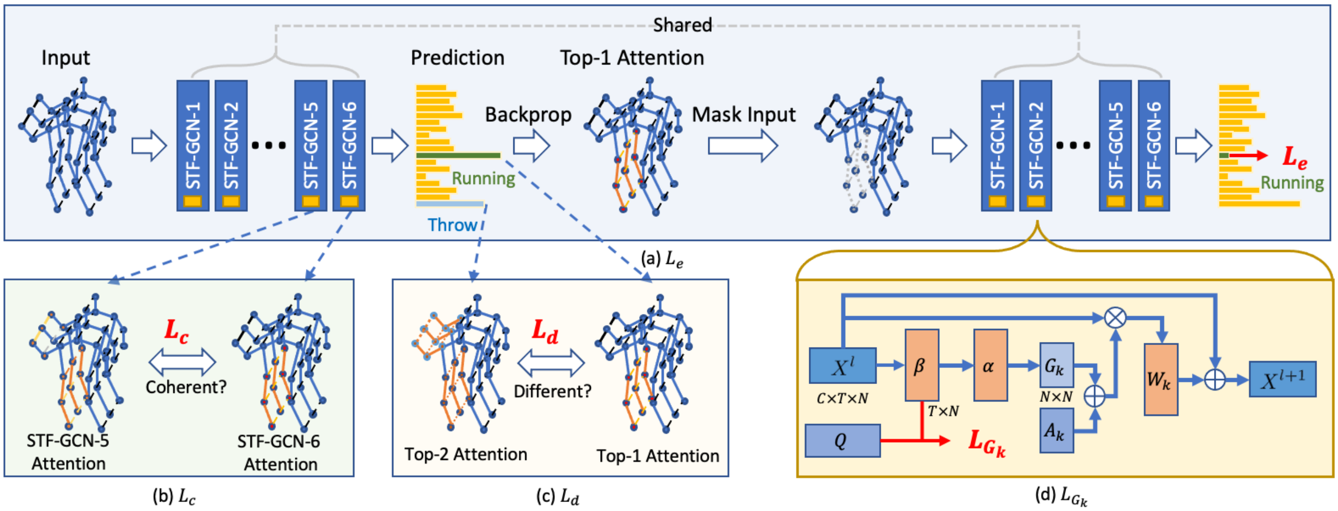

Figure 2: The illustration of the proposed objectives (all the spatio-temporal focus is projected to the input sequence for visualization): (a) STF exploration loss L_e, which masks the input sequence according to the focused parts and minimizes the probability of the originally predicted class of the masked input sequence; (b) STF coherence loss L_c enforces the focus to be coherent across the last two STF modules; (c) STF divergence loss L_d enforces the top two predicted classes to have different focused parts; (d) STF adjacency matrix and its loss function L_{G_k}, which encourages the consistency between the spatio-temporal focus and STF adjacency matrix such that the adjacency matrix is adaptive, high-order and instance-dependent. The yellow boxes in STF-GCN-1$\sim$STF-GCN-6 are not shared (*i.e.*, each STF module has its own STF adjacency matrix).

model spatio-temporal focus to supervise the learning of the STF modules.

To extract the spatio-temporal focus to guide the learning, we apply the Grad-CAM (Selvaraju et al. 2017) to the last layer (which contains more abstract features and has a larger receptive field than the previous layers) and extract the spatio-temporal focus Q as follows:

$$Q = ReLU\left(\sum_c \left(\frac{1}{Z}\sum_{t,v}\frac{\partial \hat{y}}{\partial X^l_{ctv}}\right) X^l_{ctv}\right), \quad (2)$$

where $\hat{y}$ is the probability of the predicted class y. c, t, and v denote the channel, temporal, and spatial dimensions of the intermediate feature map X^l_{ctv}, respectively, and Z is the normalization factor in the spatio-temporal dimension. Q is normalized to $[0, 1]$ by the min-max normalization. Then we learn the spatio-temporal embedding network β using Q in Eq. 2 as supervision via $L_{G_k} = \|Q - \beta(X^l)\|_2$, which ensures that the spatio-temporal embedding encodes the instance-dependent and high-order spatio-temporal dependencies. As we use the spatio-temporal focus Q as the guidance, the accuracy of Q becomes critical. We propose the following three objectives to further regularize Q.

STF Exploration

Firstly, the STF exploration loss is introduced to ensure the complete and global view of the spatio-temporal focus Q. Typical classification losses, such as the cross-entropy loss used in (Krizhevsky, Sutskever, and Hinton 2012), do not enforce the classifier to infer based on all critical skeleton joints in all important frames. Thus there is no constraint that the classifier will look over the entire critical parts in the spatio-temporal space. The classifier which only pays attention to partial critical parts can predict incorrectly when such information is occluded or noisy (Li et al. 2019a).

Therefore, we propose the STF exploration loss L_e to enforce the classifier's spatio-temporal focus to cover all the critical joints and all the critical frames. Specifically, suppose that the input sequence is classified to class y. An ideal classifier's focus should cover the entire critical parts across the input sequence. If we mask the corresponding parts of the spatio-temporal input sequence, the predicted probability of class y should be as low as possible. We formulate this process as:

$$L_e = g^y(\mathcal{X} - Q \odot \mathcal{X}), \quad (3)$$

where $g^y(\cdot)$ extracts the prediction score of the spatio-temporal focus masked input $(\mathcal{X} - Q \odot \mathcal{X})$ at the original prediction class y in Eq. 2, $\odot$ is the element-wise multiplication, and $\mathcal{X}$ is the input skeleton sequence. L_e uses an exclusion strategy by eliminating Q's corresponding spatio-temporal input to guide the network to focus on the critical parts across the input sequence. We observed that the focus Q expands to include more critical joints in the spatio-temporal space than Q without using L_e.

STF Divergence

Secondly, we propose the STF divergence loss to encourage the classifier to focus on different parts when predicting different classes. The skeleton sequences of different classes usually involve either different joints or different temporal movements, or both. For example, the reading and walking classes involve different joints (the upper body and lower body parts). In contrast, the walking and running classes involve similar parts (legs and arms) but have different temporal patterns (legs/arms' movement is different). These differences can be represented by the differences of their spatio-temporal focus. Inspired by the finding that overlapped attention of different classes causes visual confusion (Wang et al. 2019), we propose the STF divergence loss L_d to reduce the overlap of the focus of different classes. This loss discourages the network from covering all the parts across the sequence and focuses more on different classes' discriminative features. Although the concept of this loss can be applied to all the confusing classes, to simplify the optimization, we mainly focus on separating the most confusing classes (*e.g.*, touching neck vs. touching head). Specifically, given an input sequence, we select the top two predictions as the most confusing classes and enforce the top two classes' focus to overlap as little as possible.

Similar to L_e, we define L_d with the spatio-temporal focus Q via Eq. 2 as:

$$L_d = -||Q^{y_i} - Q^{y_j}||_2, \tag{4}$$

where Q^{y_i} and Q^{y_j} are the spatio-temporal focus of the top two prediction classes y_i and y_j, respectively. We observed that by introducing L_d, the top two predictions' focus overlaps less.

STF Coherence

Thirdly, we propose the STF coherence loss to utilize the high-level GCN module's focus to assist the low-level GCN module's learning. In skeleton-based action recognition, stacked GCNs are commonly used network structure (Si et al. 2018; Liu et al. 2020a; Peng et al. 2020; Lei et al. 2019). The modules closer to the final output (high-level modules) capture more abstract information with a larger receptive field. The modules closer to the input (low-level modules) have relatively smaller receptive fields. Thus, the high-level modules' focus has a more global view of the input skeleton sequence than that from the low-level modules. We can use the focus of a more global view from the high-level module to guide the learning of the low-level modules. Moreover, the stacked GCNs operate on the same input sequence; thus, if there is an optimal focus for these networks to pay attention to, it will be more likely to be the focus of the high-level module. Therefore, we propose the STF coherence loss L_c, which enforces the stacked GCNs to have coherent focus across the network:

$$L_c = ||Q_i - Q_j||_2, \tag{5}$$

where Q_i and Q_j are the spatio-temporal focus from the STF modules i and j, respectively. Ideally, the concept of L_c can be applied to all the GCN modules in addition to the GCN modules i and j. For the convenience and ease of optimization, we choose the last two GCN modules (STF-GCN-5 and STF-GCN-6 in Figure 2) as the GCN modules i and j given that they capture more abstract information of the input sequence.

We qualitatively verify the efficacy of L_c where we noticed that with L_c, the last two GCN modules' focus is more coherent than that without L_c, and that enforcing the coherence of the focus from different GCN modules can correct the misclassification from the classifier which does not use L_c.

Overall STF Loss

We use our proposed STF modules and objectives together with the cross-entropy loss L_{ce} for skeleton-based action recognition. Specifically, the overall loss is:

$$L = L_{ce} + \lambda_e L_e + \lambda_d L_d + \lambda_c L_c + \lambda_{G_k} L_{G_k}, \tag{6}$$

where $\lambda_e, \lambda_d, \lambda_c, \lambda_{G_k}$ are the weights of the losses, so that each loss term has comparable absolute range. We separate L_e from L_d, L_c, and L_{G_k} during training, and ensemble them during testing because we find that optimizing L_e and L_d together makes the training process unstable. We hypothesize that it is because the goals of L_e and L_d can be conflicting implicitly – L_e tends to expand the focus to include all critical joints, but L_d typically shrinks the focus to reduce the overlap of the focus corresponding to confusing classes.

Experiment

Datasets

We conduct experiments on three benchmark datasets, namely, the NTU RGB+D 60 (Shahroudy et al. 2016), NTU RGB+D 120 (Liu et al. 2019), and Kinetics Skeleton 400 (Kay et al. 2017) datasets (denoted as NTU-60, NTU-120, and Kinetics-400, respectively).

NTU RGB+D 60 (Shahroudy et al. 2016) is a large-scale skeleton-based action recognition dataset with over 60 action classes of 40 subjects for indoor scenarios. Each sequence contains one or two persons' 3D skeletons captured by three Kinect v.2 cameras in three views (termed as views 1, 2, and 3). The recommended two settings are (1) cross-Subject (x-sub), where the dataset is equally split as training and testing sets of 20 subjects each; and (2) cross-View (x-view), where all samples from view 1 are used for testing and the samples from views 2/3 are used for training.

NTU RGB+D 120 (Liu et al. 2019) extends the NTU-60 dataset with 60 extra action classes, resulting in a total of 120 action classes, for 106 subjects. The recommended two settings are (1) cross-Subject (x-sub), the 106 subjects are split into 53/53 subjects for training/testing; and (2) cross-Setup (x-set), where 16/16 setups are used for training/testing.

Kinetics Skeleton 400 (Kay et al. 2017) is a skeleton-based action recognition dataset converted from the Kinetics 400 video dataset (Kay et al. 2017) using the OpenPose (Cao et al. 2017) toolbox in 2D keypoints modality.

Implementation Details and Protocols

Data preparation. We preprocess the input skeleton sequences by subtracting each joint position by the center joint position and normalizing the results by the body height, in the same fashion as (Si et al. 2018). No other data processing or augmentation is used for fair comparisons. All skeleton sequences are padded to T=300 frames by repeating the sequences. To keep the semantic information of the skeleton sequence in Eq. 3, as done in (Si et al. 2018),

dataset	NTU RGB+D 60						NTU RGB+D 120						Kinetics-400		
setting	x-sub			x-view			x-sub			x-set					
method \ input modality	J	B	J+B	J	B	J+B	J	B	J+B	J	B	J+B	J	B	J+B
SR-TSL (Si et al. 2018)	84.80	-	-	92.40	-	-	-	-	-	-	-	-	-	-	-
2s-AGCN (Lei et al. 2019)	-	-	88.50	93.70	93.20	95.10	-	-	-	-	-	-	35.10	33.30	36.10
TS-SAN (Cho et al. 2020)	87.20	-	-	92.70	-	-	-	-	-	-	-	-	35.10	-	-
GCN-NAS (Peng et al. 2020)	-	-	89.40	94.60	94.70	95.70	-	-	-	-	-	-	35.50	34.90	37.10
MS-TGN (Li, Zhang, and Li 2020)	86.60	87.50	89.50	94.10	93.90	95.90	-	-	-	-	-	-	35.20	33.30	37.30
MS-AAGCN (Shi et al. 2020b)	88.00	88.40	89.40	95.10	94.70	96.00	-	-	-	-	-	-	36.00	34.70	37.40
3s RA-GCN (Song et al. 2020a)	-	-	87.30⋆	-	-	93.60⋆	-	-	81.10⋆	-	-	82.70⋆	-	-	-
DC-GCN+ADG (Cheng et al. 2020)	-	-	90.80‡	-	-	96.60‡	-	-	86.50‡	-	-	88.10‡	-	-	-
DSTA-Net (Shi et al. 2020a)	-	-	91.50⋄	-	-	96.40⋄	-	-	86.60⋄	-	-	89.00⋄	-	-	-
STIGCN (Huang et al. 2020)	90.10	-	-	96.10	-	-	-	-	-	-	-	-	37.90	-	-
PA-ResGCN-B19 (Song et al. 2020b)	-	-	90.90†	-	-	96.00†	-	-	87.30†	-	-	88.30†	-	-	-
Dynamic-GCN (Ye et al. 2020)	-	-	91.50	-	-	96.00	-	-	87.30	-	-	88.60	-	-	37.90
MS TE-GCN (Li et al. 2020)	87.40	88.50	90.80‡	93.40	93.30	96.20‡	-	-	84.40‡	-	-	85.90‡	-	-	-
UNIK (Yang et al. 2021)	-	-	86.80	-	-	94.40	-	-	80.80	-	-	86.50	-	-	-
AdaSGN (Shi et al. 2021)	-	-	89.10	-	-	94.70	-	-	85.90	-	-	86.80	-	-	-
Yang's (Siyuan et al. 2021)	88.00	-	-	94.90	-	-	-	-	-	-	-	-	-	-	-
MS-G3D (paper) (Liu et al. 2020a)	89.40	90.10	91.50	95.00	95.30	96.20	-	-	86.90	-	-	88.40	35.80	35.44	38.00
MS-G3D (code)* (Liu et al. 2020b)	88.77	89.59	90.67	94.88	94.86	95.82	82.35	84.86	86.42	84.14	86.79	87.98	35.74	34.77	37.23
STF (ours)	**91.34**	**91.09**	**92.47**	**96.46**	**96.51**	**96.86**	**85.06**	**86.80**	**88.85**	**86.40**	**88.86**	**89.92**	**38.20**	**37.56**	**39.87**

Table 2: Performance comparison of skeleton-based action recognition in top-1 accuracy (%). *: For MS-G3D (Liu et al. 2020a), the publicized code (Liu et al. 2020b) we use as our baseline has lower accuracy than what was reported in their paper. Annotations: J: joint; B: bone; "-": results not provided in the reference. The methods requiring more inputs than J+B: "⋆": (Song et al. 2020a) uses 3 streams; "‡": (Cheng et al. 2020; Li et al. 2020) use J+B+J motion+B motion; "⋄": (Shi et al. 2020a) uses spatial-temporal, spatial, slow-temporal, and fast-temporal streams; "†": (Song et al. 2020b) uses J+B+velocity.

we add a visibility channel alongside the X-Y-Z location channel to indicate the joint's visibility, where 1/0 means visible/invisible. The initial visibility channel is set to 1. The visibility channel is applied to the NTU-60 and NTU-120 datasets. For the Kinetics-400 dataset, the visibility is embedded with the confidence score channel of joints. Unless otherwise specified, we use the default settings for other parameters.

Experimental protocol. Following the experimental protocol of Shi *et al.* (Lei et al. 2019), we report the classification accuracies under three different input modalities: (i) J: the joints only, where the joint stream is the absolute skeleton joints location sequence, (ii) B: the bones only, where the bone stream is the second-order information (the lengths and directions of bones) of the skeleton data, and (iii) J+B: the 2-stream input with both joints and bones.

Training scheme. We implement STF using MS-G3D (Liu et al. 2020a) as the backbone. The MS-G3D baseline model is trained using SGD with momentum 0.9, batch size 32, initial learning rate 0.05, and weight decay 0.0005, and the base learning rate is adjusted accordingly for different settings. For NTU-60, NTU-120, and Kinetics-400, the learning rate is decayed at [20, 35, 45], [20, 35, 50], [25, 40, 55] epochs, respectively. After that, we pre-train the STF model from the MS-G3D baseline model, with lower initial learning rates $\{10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$. We empirically set $\lambda_e/\lambda_d/\lambda_c/\lambda_{G_k}$ as 0.01/0.1/0.1/0.01, respectively, such that all loss terms have comparable absolute ranges.

Joint-bone stream fusion. Moreover, we follow Shi *et al.* (Lei et al. 2019) to combine the results of joint and bone streams by averaging the prediction probability for each sequence, and report them in Table 2. Specifically, we merge the J and B streams by averaging the prediction probability from both streams to get the J+B results of STF.

Experimental Results

Baselines. We summarize the experimental results in Table 2, where we compare STF with the SOTA skeleton-based action recognition methods. Of all the methods in Table 2, MS-G3D (Liu et al. 2020a) is one of the most competitive methods, so we build STF on top of it. However, when we run the publicized code (Liu et al. 2020b) of MS-G3D, we found that its accuracy is lower than that reported in the paper (Liu et al. 2020a). Since we implement STF based on the publicized code of MS-G3D, we use the accuracy obtained from running the publicized code of MS-G3D for fair comparisons. We also include the performance reported in the original paper of MS-G3D (Liu et al. 2020a) in Table 2, where we mark the best performance in **bold**.

Comparison with SOTAs. Table 2 shows that STF outperforms the listed baselines in all settings. We find that although 3s RA-GCN (Song et al. 2020a), DC-GCN+ADG (Cheng et al. 2020), DSTA-Net (Shi et al. 2020a), PA-ResGCN-B19 (Song et al. 2020b), and MS TE-GCN (Li et al. 2020) have the unfair advantage of using strictly more information than J+B as input, STF still outperforms them in most settings. In addition, we compare the accuracy gains of STF and MS-G3D over their baselines in Table 3. We find that the accuracy gain of STF is 3.8/1.9/2.3 times of that of MS-G3D on the challenging Kinetics-400 dataset, given that STF only uses 10% more FLOPs than MS-G3D.

settings	Kinetics-J	Kinetics-B	Kinetics-J+B
Δ(MS-G3D, 2s-AGCN)	0.64% (1.0x)	1.47% (1.0x)	1.13% (1.0x)
Δ(STF, MS-G3D)	**2.46**% (3.8x)	**2.79**% (1.9x)	**2.64**% (2.3x)

Table 3: Comparison of accuracy gain over the baseline.

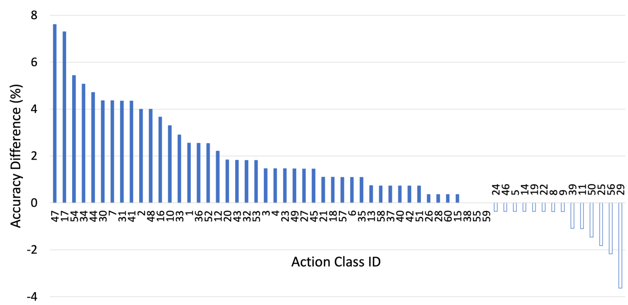

Figure 3: The class-wise accuracy difference (%) between STF in Table 2) and the baseline MS-G3D (Liu et al. 2020a) (MS-G3D (code) in Table 2) for the NTU-60 dataset under the x-sub setting with the joint input modality.

Improvement on scarce data. To simulate the real-world scenarios where only small amount of training data are available, we train on 10/20/25% of randomly sampled training data from NTU-60 under the x-sub setting with the joint input modality, and evaluate the accuracy on the entire testing set of NTU-60. Table 4 shows that STF consistently outperforms the baseline in all settings. We also verify this on the randomly sampled 25% Kinetics-400, the STF (joint 25.56, bone 26.30) outperforms MS-G3D (joint 24.69, bone 25.22), regardless of the input modality.

Efficacy under data shift. To test the efficacy of the features learned from our proposed STF across different datasets, we test STF model on the dataset different from the training set. Specifically, we freeze the model trained on the NTU-60 dataset under the x-sub setting with joint input modality, and replace the output fully connected layer with a new fully connected layer of 120 output nodes. The new fully connected layer is fine-tuned using the training set of the NTU-120 under the x-sub setting with joint input modality. After fine-tuning, we evaluate its performance on the NTU-120 testing set. Our experimental result shows that STF outperforms the baseline MS-G3D by 1.04% (83.38% vs. 82.34%). This result supports that the features learned from STF has better generalization ability than MS-G3D.

Improvement on confusing classes. In addition, we also show a break-down evaluation of the class-wise accuracy difference (%) for the joint input modality between STF in Table 2) and the baseline MS-G3D (Liu et al. 2020a) (MS-G3D (code) in Table 2) for the x-sub setting on NTU-60. Figure 3 shows that STF outperforms MS-G3D for most classes. The highest performance gain occurs among the more challenging classes with more multiple joint correlations (*e.g.*, touch neck +7.61% vs. touch head +4.71%, and take off a shoe +7.30% vs. wear a shoe +3.66%). This is likely due to the explicit consideration of the high-order dependencies in STF.

method \ p	10	20	25	100
MS-G3D	72.01	79.11	81.91	88.77
STF	**72.73** (↑0.72)	**80.77** (↑1.66)	**84.27** (↑2.36)	**91.34** (↑2.57)

Table 4: The accuracy (%) using joint input modality on the NTU-60 dataset under the x-sub setting. We use p% of the randomly sampled training data from the NTU-60 dataset.

method	MS-G3D	L_c	L_d	L_{G_k}	L_e	accuracy	Δ
M_1	✓					88.77	—
M_2	✓	✓				89.33	↑ 0.56
M_3	✓	✓	✓			89.92	↑ 1.15
M_4	✓	✓	✓	✓		90.65	↑ 1.88
M_5: STF	✓	✓	✓	✓	✓	**91.34**	↑ 2.57

Table 5: Ablation study of top-1 accuracy (%) using joint only modality on the NTU-60 dataset under the x-sub setting. Δ shows the accuracy improvement over the baseline, MS-G3D (Liu et al. 2020a).

Ablation Study

Our learning objectives are composed of several different loss terms, so it is important to know the contribution of these losses to accuracy. To this end, we perform an ablation study of the contribution of different loss terms using the joint input modality on the NTU-60 dataset under the x-sub setting in Table 5. We report the accuracy and Δ, the accuracy improvement over MS-G3D (Liu et al. 2020a) by using different combinations of loss terms as the learning objectives. STF results in $\Delta = 2.57\%$ improvement with joint only modality on the NTU-60 x-sub setting.

Conclusion

We propose the To-a-T Spatio-Temporal Focus (STF) method for skeleton-based action recognition. First, we propose the STF modules to generate flexible adjacency matrices, and use the spatio-temporal focus to guide the learning of STF modules, such that the adjacency matrices capture the high-order dependency and spatio-temporal importance. To capture critical spatio-temporal features, we propose the STF exploration, STF divergence, and STF coherence losses to encourage the spatio-temporal focus which supports the classifier's prediction to include all critical spatio-temporal features, to distinguish different classes on spatio-temporal focus, and to make the spatio-temporal focus across the stacked GCNs consistent, respectively. STF outperforms the SOTA methods on the NTU RGB+D 60, NTU RGB+D 120, and Kinetics Skeleton 400 datasets. Our proposed objectives in STF are not tied to any specific network architecture and have the potential to be applied to other network architectures, which is our future work. We also plan to explore multi-stream input to train all our proposed objectives jointly in the future.

References

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Real-time multi-person 2d pose estimation using part affinity fields. In *CVPR*.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.

Chen, T.; Zhou, D.; Wang, J.; Wang, S.; Guan, Y.; He, X.; and Ding, E. 2021a. Learning Multi-Granular Spatio-Temporal Graph Network for Skeleton-based Action Recognition. arXiv:2108.04536.

Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; and Hu, W. 2021b. Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition. *arXiv preprint arXiv:2107.12213*.

Cheng, K.; Zhang, Y.; Cao, C.; Shi, L.; Cheng, J.; and Lu, H. 2020. Decoupling GCN with DropGraph Module for Skeleton-Based Action Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Cho, S.; Maqbool, M.; Liu, F.; and Foroosh, H. 2020. Self-Attention Network for Skeleton-based Human Action Recognition. In *The IEEE Winter Conference on Applications of Computer Vision*, 635–644.

Ding, X.; Yang, K.; and Chen, W. 2019. An Attention-Enhanced Recurrent Graph Convolutional Network for Skeleton-Based Action Recognition. In *Proceedings of the 2019 2nd International Conference on Signal Processing and Machine Learning*, 79–84.

Fan, Y.; Weng, S.; Zhang, Y.; Shi, B.; and Zhang, Y. 2020. Context-Aware Cross-Attention for Skeleton-Based Human Action Recognition. *IEEE Access*, 8.

Heidari, N.; Alex; and ros Iosifidis. 2020a. On the spatial attention in Spatio-Temporal Graph Convolutional Networks for skeleton-based human action recognition. *arXiv preprint arXiv:2011.03833*.

Heidari, N.; Alex; and ros Iosifidis. 2020b. Progressive Spatio-Temporal Graph Convolutional Network for Skeleton-Based Human Action Recognition. *arXiv preprint arXiv:2011.05668*.

Herath, S.; Har; i, M.; and Porikli, F. 2017. Going deeper into action recognition: A survey. *Image and vision computing*, 60: 4–21.

Huang, Z.; Shen, X.; Tian, X.; Li, H.; Huang, J.; and Hua, X.-S. 2020. Spatio-Temporal Inception Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2122–2130.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 221–231.

Johansson, G. 1973. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2): 201–211.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. arXiv:1705.06950.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Lei, S.; Yifan, Z.; Jian, C.; and Hanqing, L. 2019. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12018–12027.

Li, J.; Xie, X.; Zhao, Z.; Cao, Y.; Pan, Q.; and Shi, G. 2020. Temporal Graph Modeling for Skeleton-based Action Recognition. *arXiv preprint arXiv:2012.08804*.

Li, K.; Wu, Z.; Peng, K.-C.; Ernst, J.; and Fu, Y. 2019a. Guided Attention Inference Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42: 2996–3010.

Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; and Tian, Q. 2019b. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*.

Li, T.; Zhang, R.; and Li, Q. 2020. Multi Scale Temporal Graph Networks For Skeleton-based Action Recognition. In *International Conference on Computer Science and Information Technology*.

Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.-Y.; and Kot, A. C. 2019. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, J.; Wang, G.; Hu, P.; Duan, L.-Y.; and Kot, A. C. 2017. Global Context-Aware Attention LSTM Networks for 3D Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, M.; Liu, H.; and Chen, C. 2017. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68: 346–362.

Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; and Ouyang, W. 2020a. Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 143–152.

Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; and Ouyang, W. 2020b. MS-G3D official code. https://github.com/kenziyuliu/MS-G3D.git. Accessed: 2020-08-08.

Pan, C.; Chen, S.; and Ortega, A. 2021. Spatio-Temporal Graph Scattering Transform. In *International Conference on Learning Representations*.

Peng, W.; Hong, X.; Chen, H.; and Zhao, G. 2020. Learning Graph Convolutional Network for Skeleton-based Human Action Recognition by Neural Searching. In *AAAI*.

Plizzari, C.; Cannici, M.; and Matteucci, M. 2020. Spatial Temporal Transformer Network for Skeleton-based Action Recognition. In *International Conference on Pattern Recognition Workshop*.

Qin, Z.; Liu, Y.; Ji, P.; Kim, D.; Wang, L.; McKay, B.; Anwar, S.; and Gedeon, T. 2021. Fusing Higher-Order Features in Graph Neural Networks for Skeleton-Based Action Recognition. arXiv:2105.01563.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Shahroudy, A.; Liu, J.; Ng, T.-T.; and Wang, G. 2016. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2020a. Decoupled Spatial-Temporal Attention Network for Skeleton-Based Action Recognition. In *Asian Conference on Computer Vision*.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2020b. Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks. *IEEE Transactions on Image Processing*, 29: 9532–9545.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2021. AdaSGN: Adapting Joint Number and Model Size for Efficient Skeleton-Based Action Recognition. *arXiv preprint arXiv:2103.11770*.

Si, C.; Chen, W.; Wang, W.; Wang, L.; and Tan, T. 2019. An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Si, C.; Jing, Y.; Wang, W.; Wang, L.; and Tan, T. 2018. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 103–118.

Simonyan, K.; and Zisserman, A. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems*, 568–576.

Siyuan, Y.; Jun, L.; Shijian, L.; Meng Hwa, E.; and Kot, A. C. 2021. Skeleton Cloud Colorization for Unsupervised 3D Action Representation Learning. *ICCV*.

Song, S.; Lan, C.; Xing, J.; Zeng, W.; and Liu, J. 2017a. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In *AAAI*.

Song, S.; Lan, C.; Xing, J.; Zeng, W.; and Liu, J. 2017b. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, 4263–4270.

Song, Y.-F.; Zhang, Z.; Shan, C.; and Wang, L. 2020a. Richly Activated Graph Convolutional Network for Robust Skeleton-based Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*.

Song, Y.-F.; Zhang, Z.; Shan, C.; and Wang, L. 2020b. Stronger, Faster and More Explainable: A Graph Convolutional Baseline for Skeleton-based Action Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, 1625–1633.

Varol, G.; Laptev, I.; and Schmid, C. 2017. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6): 1510–1517.

Wang, L.; Wu, Z.; Karanam, S.; Peng, K.-C.; Singh, R. V.; Liu, B.; and Metaxas, D. N. 2019. Sharpen focus: Learning with attention separability and consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, 512–521.

Xie, C.; Li, C.; Zhang, B.; Chen, C.; Han, J.; Zou, C.; and Liu, J. 2018. Memory Attention Networks for Skeleton-based Action Recognition. In *International Joint Conference on Artificial Intelligence*, 1639–1645.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *AAAI*.

Yang, D.; Li, M. M.; Fu, H.; Fan, J.; and Leung, H. 2020a. Centrality Graph Convolutional Networks for Skeleton-based Action Recognition. *arXiv preprint arXiv:2003.03007*.

Yang, D.; Wang, Y.; Dantcheva, A.; Garattoni, L.; Francesca, G.; and Bremond, F. 2021. Unik: A unified framework for real-world skeleton-based action recognition. *arXiv preprint arXiv:2107.08580*.

Yang, H.; Gu, Y.; Zhu, J.; Hu, K.; and Zhang, X. 2020b. PGCN-TCA: Pseudo Graph Convolutional Network With Temporal and Channel-Wise Attention for Skeleton-Based Action Recognition. *IEEE Access*, 8: 10040–10047.

Yang, Z.; and Yin, K. 2020. Improving Skeleton-based Action Recognitionwith Robust Spatial and Temporal Features. *arXiv preprint arXiv:2008.00324*.

Ye, F.; Pu, S.; Zhong, Q.; Li, C.; Xie, D.; and Tang, H. 2020. Dynamic GCN: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, 55–63.

Zeng, A.; Sun, X.; Yang, L.; Zhao, N.; Liu, M.; and Xu, Q. 2021. Learning Skeletal Graph Neural Networks for Hard 3D Pose Estimation. *ICCV*.

Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*.