IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. XX, NO. XX, XXXX 1

Coupled Noise Suppression and Feature Enhancement Network for Skeleton based Action Recognition

***Abstract*— In recent years, remarkable progress has been made in skeleton-based action recognition. However, during the process of acquisition, a significant amount of noise is introduced into skeleton data. This problem is simply overlooked by most existing methods. Some methods have designed specialized mechanisms to handle noise, but these mechanisms are either based on prior knowledge or require additional supervision information. To overcome these problems, we propose in this paper a fully implicit solution, which embeds a soft-thresholding based denoising mechanism as a module into existing graph neural networks. During the training the module can automatically learn to remove noise without any prior knowledge or additional supervision information. In addition, by relaxing the non-negative constraint of the threshold, the module also has the ability to adaptively enhance key features. Based on this, we further propose a two-staged method for coupled noise suppression and feature enhancement. The proposed method achieve state-of-the-art performance on public datasets such as NTU RGB+D (60 and 120) and NW-UCLA. Moreover, on noise polluted datasets, the proposed method demonstrates significant performance advantages over existing method.**

***Index Terms*— Skeleton-based action recognition, noise suppression, feature enhancement, graph convolution network**

I. Introduction

Human action recognition is an important problem in computer vision due to its relevance to a large number of realistic applications such as human-computer interaction, video surveillance *etc*. Unlike the huge volume of 3D point cloud or image data, the human skeleton data obtained from depth sensors not only has small data volume but also has good robustness to complex backgrounds. Therefore, in recent years, skeleton-based human action recognition has become a hot research topic. However, action recognition from skeleton data is a non-trivial task, since the skeleton data obtained from RGB-D sensor is not perfect which contains substantial noise or even errors.

Deep learning has been widely used in skeleton-based action recognition. Although the deep learning methods have relatively strong anti-noise ability, severe noise interference may still cause bias to the model prediction. However, there is no specific mechanism in most of the existing works to deal with noise in skeleton data. For example, ST-GCN [1] networks first applied graph convolution to the extraction of human skeleton features by modeling manually defined relationships of naturally connected joints of human body.

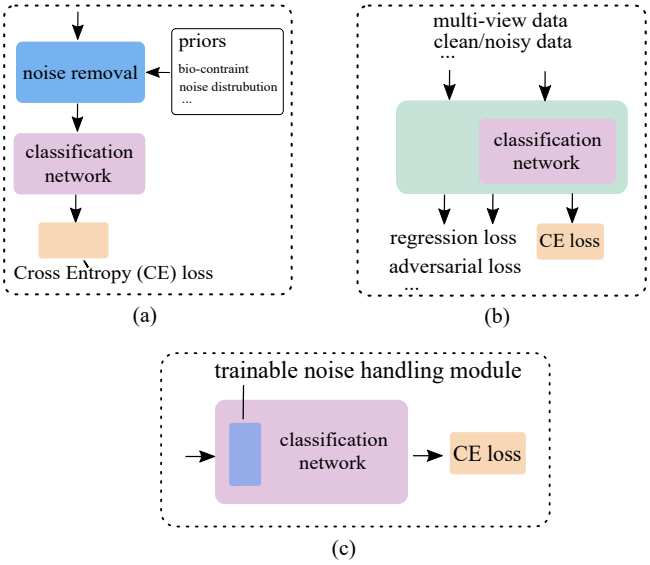

Fig. 1. Three types of strategy for noise handling in skeleton data. (a) Independent noise removal stage with priors. (b) Noise adaption with additional supervision information. (c) Fully implicit solution with trainable noise handling module.

And subsequently various graph convolutional network (GCN) [2]–[4] based action recognition networks were proposed. However, the effect of noise is not discussed in these studies.

Different from these works, there are a few works that designed special mechanisms to address the impact of noise. Prior knowledge is introduced to remove noise such as bio-constraint skeleton prior [5] or specific noise distribution [6] as shown in Fig. 1 (a). These priors may not be optimal and may not be able to handle complex noise. In [7], additional supervision information is introduced to train noise-adaptive models as shown in Fig. 1 (b), however this will increase the complexity of the model and the training process, and moreover the supervision information is sometimes difficult to obtain.

Thus fully implicit and trainable solution may be a better alternative, in which the network is trained in conventional way without introducing priors or additional supervision information. In this paper, we introduce a fully implicit adaptive soft-thresholding solution to address the noise problem. As shown in Fig. 2, in this solution there are two deep networks: the noise separation network is used to learn to project noisy features into an embedding space where the distributions of noise and signal are separable, and the threshold generation

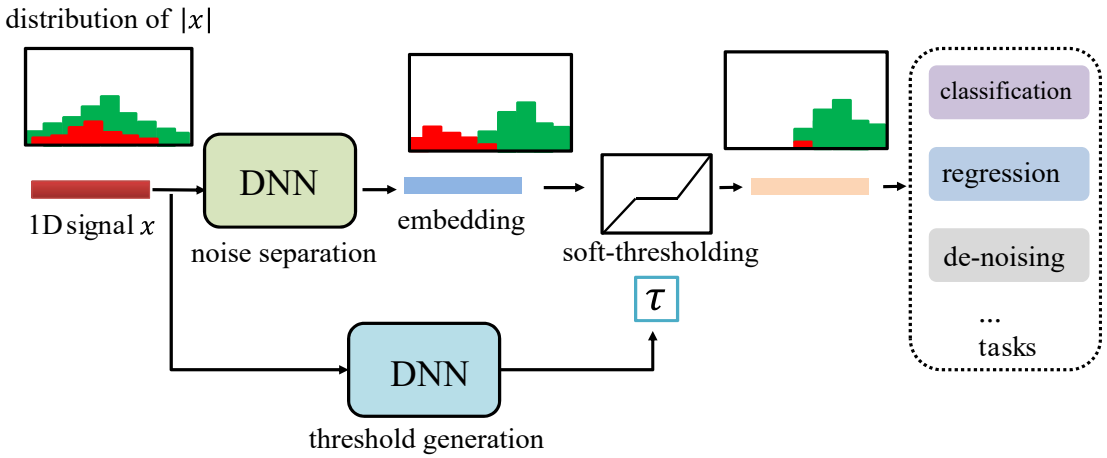

Fig. 2. Deep learning based soft-thresholding framework for totally implicit noise suppression in various learning tasks. There are two deep neural networks playing different roles, one for noise separation and the other for threshold generation.

network learns to predict a threshold with which the noise can be suppressed through the soft-thresholding function.

In addition to noise handling, we further relax the non-negative constraint in soft-thresholding, making the framework able to adaptively enhance key features. To make the network better exploit different dimensions of skeleton data, we decompose this framework into two stages: spatio-temporal suppression-enhancement (ST-S&E) and channel-wise suppression-enhancement (CW-S&E). These two stages as a whole are referred to as coupled noise suppression and feature enhancement module, which is incorporated into existing GCN based networks for skeleton based action recognition as shown in Fig. 3. Extensive experiments on NTU RGB+D (60 and 120) and NW-UCLA datasets show that the proposed method can achieve significant improvements with minor increase in the number of parameters and computational cost.

The contributions of this paper are summarized as follows:

- We introduce for the first time the deep learning based soft-thresholding into the skeleton-based human action recognition network implicitly and adaptively suppress the effect of noise in the network.
- We propose to relax the non-negative constraint of soft-thresholding, and develop a general thresholding function which enables the network to perform noise suppresion and feature enhancement operations simultaneously.
- We propose a two staged network for better exploring the information of spatio-temporal and channel dimensions in coupled noise suppression and feature enhancement.

II. Related work

A. Non-GCN based skeleton action recognition

Human skeleton based action recognition has become increasingly important in many applications such as intelligent surveillance, motion capture, virtual reality, *etc*. As a result, related research topics have gained considerable attention. Prior to the widespread use of deep neural networks, research in skeleton based action recognition relied on traditional methods that utilized manually designed features and classifiers [8]. However, these traditional methods were often dependent on specific data and problems, requiring manual feature extraction and being prone to problems such as over-fitting.

Neural networks have been widely used in various fields due to their good performance and have also been introduced to skeleton based action recognition, achieving promising results. Methods based on neural networks can be classified into three categories: convolutional neural networks (CNN) based methods [9], [10], recurrent neural networks (RNN) based methods [11], [12], and graph convolution based methods. CNN based methods usually maps human skeletal data onto a two-dimensional plane by connecting them based on a fixed structure, forming a two-dimensional image. Subsequently, an image classification network is used to extract features from the two-dimensional image for classification. On the other hand, recurrent neural networks, such as LSTM and GRU, concatenate all the location coordinate data of each frame in a fixed order as the input for feature extraction. However, it is difficult for both CNN and RNN to model the topological structure human skeleton. Recently, large language model are introduced into skeleton-based action recognition in [13].

B. GCN based skeleton action recognition

In order to better explore topological structure human skeleton data, graph convolutional neural networks (GCNs) have been proposed, which have significant advantages in processing data with non-Euclidean structures. Human skeleton data is naturally graph-structured data consisting of nodes and edges, where nodes represent the skeletal positions of the data and edges represent the connections between them. Early work used GCN to extract spatial dependence among joints and used LSTM to capture temporal dynamics [14]. Yan et al proposed the ST-GCN network [1], which models the skeleton data as spatio-temporal graph which are repeated processed by spatial graph convolution and temporal convolution. Researchers found the original skeleton links are limited, and tried to enrich the connections in graph to capture latent and high-order dependencies [3].

Chen *et al* [15] proposed the channel-refined graph convolution network named CTR-GCN based on the graph convo-

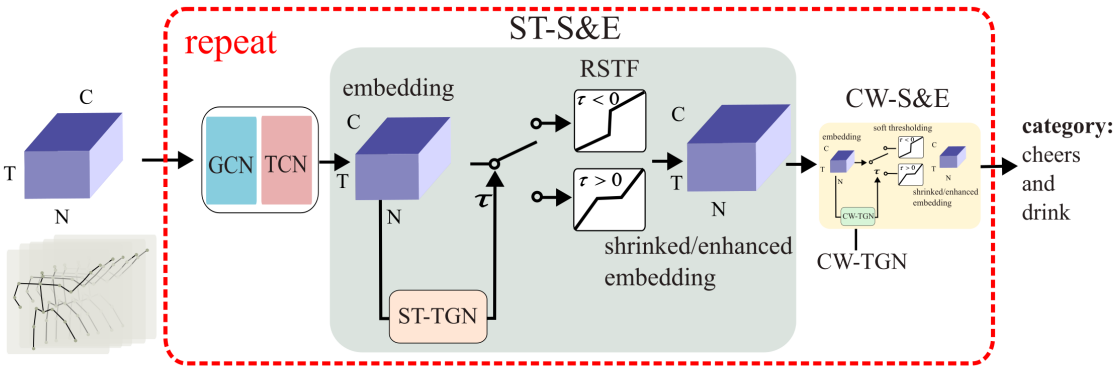

Fig. 3. Diagram of proposed coupled noise suppression and feature enhancement network. The added module consists of two stages: ST-S&E and CW-S&E.

lution refinement of each channel of the ST-GCN network. The CTR-GCN network tailors the graph convolution for the association information specific to each spatio-temporal node, which gets rid of the natural connection structure of the human skeleton, and improves the accuracy during the graph convolution feature extraction. High-order features are incorporated into graph neural networks in [16] to capture the relations between joints and body parts. Shi *et al* studied the problem of occluded skeleton data recognition and proposed their GCN based solutions [17]. Hierarchically decomposed graph was proposed in [18] to identify the relationships between distant joint nodes in the same semantic spaces. Zhou *et al* [19] propose a contrastive feature refinement head which discovers and calibrates ambiguous samples, which further enhances the performances. And recently, multi-modal data including skeleton, video and text are incorporated and co-learned in [20] to enhance robustness and generalization ability.

C. Noise handling for skeleton data

Despite noise being an important but under-investigated issue in skeleton-based action recognition, some efforts have been made to address this issue. Bio-constraint prior is introduced to remove noise in [5]. And Demisse *et al* [6] propose an auto-encoder based denoising method which makes prior assumptions about the distribution noise. As pointed out in [7], these priors are either too weak to remove noise effectively, or too strong to remove useful information as well. Song *et al* [7] formulate the problem as a domain adaption problem and proposed two kind of solutions. However, both of the two solutions require extra supervision information such as observations from multiple views, labeling of noisy and clean data *etc*, which complicates both the network structure and training strategy. And moreover, these supervision information can not always be satisfied. Thus, a trainable and fully implicit solution may be a better alternative.

III. Relaxed Soft-Thresholding

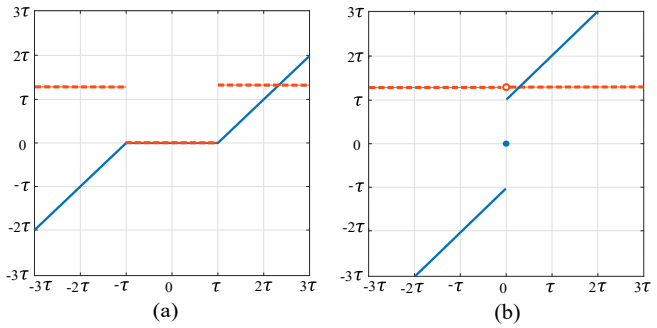

Fig. 4. (a) Original soft-thresholding function (blue) and its derivative (orange) with non-negative threshold τ which will suppress signal with low-amplitude (noise suppression). (b) Relaxed soft-thresholding function (blue) and its derivative (orange) with negative τ which will increase the amplitude of the signal (feature enhancement) by $|\tau|$.

A. Original soft-thresholding function

Soft-thresholding is a standard procedure in signal de-noising, whose optimality has been proven theoretically and it's effectiveness has also been verified in various applications. Donoho [21] proposed the soft-thresholding de-noising method based on the smoothness theory of functions and the statistical properties of noise. For wavelet transformed signal, noise component is mainly contained in the coefficients with small amplitude, the wavelet coefficients smaller than a threshold value is set to zero, and the amplitude of other coefficients is reduced. Since deep neural networks have strong approximation ability [22], they can approximate wavelet transform and achieve the effect of it after training. Thus introducing soft-thresholding into neural networks can suppress the noise contained in features, whose effectiveness has been demonstrated in signal denoising [23], fault diagnosis [24], *etc*. To the best of our knowledge, this paper is the first to introduce a deep learning based soft-thresholding framework to skeleton-based action recognition.

The soft-thresholding function can be written as:

$$soft(x,\tau) = \begin{cases} x+\tau & x < -\tau \\ 0 & -\tau \le x \le \tau \\ x-\tau & x > \tau \end{cases} \quad (1)$$

In this form, the threshold τ must be non-negative for the formula to make sense. And the function can be written in another form:

$$soft(x,\tau) = \max(|x| - \tau, 0) \cdot sign(x), \quad (2)$$

When τ is non-negative, this formula and Eq. 1 represent the same function as shown in Fig. 4(a).

B. Relaxed soft-thresholding function

Different from Eq. 1, with the formula in Eq. 2 if τ is negative, the formula also makes sense. However if τ is negative the curve of the function becomes a completely different form as shown in Fig. 4 (b). Consider a signal x, its absolute value at non-zero position increases by $|\tau|$ after passing through the function. The actual effect of setting τ to a negative number is to increase the amplitude of the signal as shown in Fig. 4 (b). If this function is applied to the features in the neural network, its effect is to enhance the amplitude of features, which will have greater impact on the output results of the entire neural network. The amplitude of the enhancement is $|\tau|$ which is dynamically generated by the threshold generating network.

Previous deep learning based soft-thresholding networks usually adopts Sigmoid activation function at the end of the threshold generation network to generate a positive threshold τ. We relax the non-negative condition by simply replacing the Sigmoid function (output range is $(0, 1)$) with the Tanh function whose output range is $(-1, 1)$. We can observe that relaxed soft-thresholding function (RSTF) exerts its influence on neurons through two modes: suppression model and enhancement modes, and the key to accessing these two modes lies in the threshold τ generated by the threshold-generating network. This threshold-generating network, through end-to-end training, possesses the capability to distinguish whether a neuron should be suppression or enhanced.

C. Characteristic Analysis

By further analyzing RSTF, we can discover that it possesses the following two characteristics:

1) RSTF can make potential noise-contaminated neurons stop propagating their gradient in suppression mode.
2) The gradient of weights associated with neurons can be made less active or more active by RSTF during training.

Firstly, let us look at the derivative of the RSTF function as shown in Fig. 4 (a) and (b). When a positive τ is given RSTF is in the suppression mode, the derivative of RSTF is shown in Fig. 4 (a). We notice that if the input value is in range $(-\tau, \tau)$ the derivative is zeros, this implies that during training, the neuron which is the output of RSTF that is in suppression model will have its gradient propagation stopped if its value is in range $(-\tau, \tau)$. Note that τ is generated by the threshold-generating network, thus this scenario can also be viewed as the threshold generation network halting the backward propagation of gradients for potentially noise-corrupted neurons.

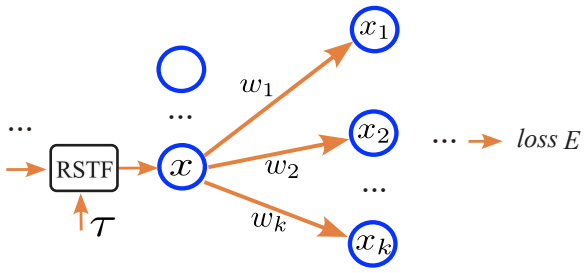

Fig. 5. An example of a neuron processed by relaxed soft-thresholding function (RSTF) in a neural network.

Next, we will investigate the weights associated with the neuron processed by RSTF. Suppose that neuron x is located in layer l of a neural network as shown in Fig. 5, and it is connected to k neurons $x_1, x_2, \ldots, x_k$ in layer $l+1$ with connection weights $w_1, w_2, \ldots, w_k$ respectively. The loss function of the network is denoted as E, the gradient which is back-propagated from E to $w_i, i = 1, 2, ..., k$ can be written as:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial w_i} \quad (3)$$

Since $x_i = w_i x + \Delta$ where Δ is independent of w_i, we have:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial x_i} x. \quad (4)$$

From the above equation, we can learn that the value of x directly affects $\frac{\partial E}{\partial w_i}$. Therefore, during the training process, when x is suppressed by the RSTF (becomes smaller), the gradient magnitudes of its corresponding connection weights $w_1, w_2, \ldots, w_k$ also decrease, leading to slower updates and reduced activity. Conversely, when x is enhanced by the RTS function (becomes larger), the gradient magnitudes of $w_1, w_2, \ldots, w_k$ also increase, resulting in faster updates and increased activity.

IV. Coupled Noise Suppression and Feature Enhancement

We propose a trainable and fully implicit coupled noise suppression and feature enhancement module which adaptively learns the threshold for relaxed soft-thresholding. In order to exploit spatial-temporal and channel-wise information effectively, we decompose this task into two stages. As shown in Fig. 6, the proposed module consists of two sequentially connected blocks: Spatio-temporal S&E module and Channel-wise S&E module, where S&E represents for suppression and enhancement.

A. Spatio-temporal S&E

In this block, temporal and spatial dimensions are at first exploited separately in two decoupled branches and finally combined to generate the threshold for relaxed soft-thresholding, we name the this part spatio-temporal threshold generation

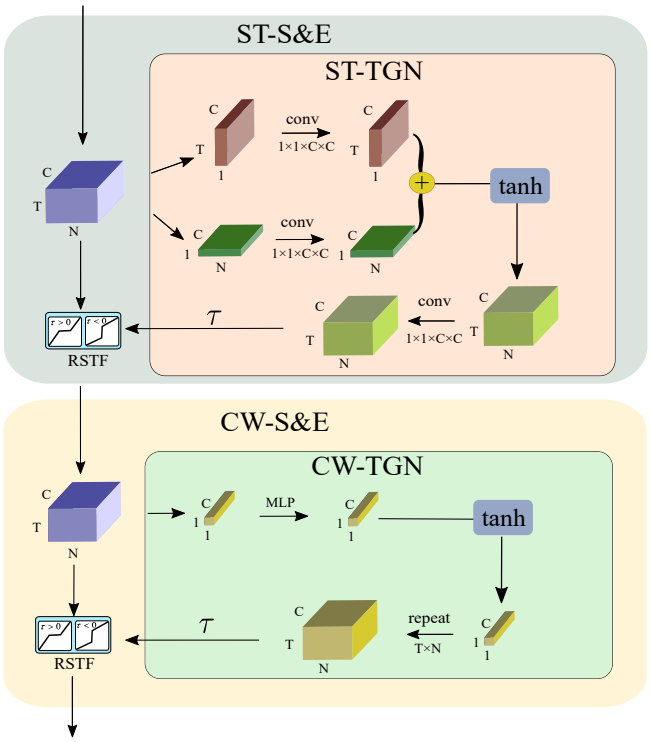

Fig. 6. The architecture detail of the two sequentially connected blocks: Spatio-temporal S&E module and Channel-wise S&E module of the proposed noise suppression and feature enhancement network. The key components of ST-S&E and CW-S&E are threshold generation network ST-TGN and CW-TGN respectively.

network (ST-TGN). As shown in Fig. 6, given input tensor $X \in \mathbb{R}^{T\times N\times C}$, where T is the sequence length, N is the number of skeleton joints, C is the number of feature channels. The first branch of the block keeps the temporal dimension while reduces the spatial dimension by computing spatial mean. The second branch of the block keeps the spatial dimension while reduces the temporal dimension by computing temporal mean. Two 1×1 convolution operators are connected separately in two branches to integrate information from different feature channels. The output tensors of two branches are added together and then activated with $tanh$ which generate a tensor of $\mathbb{R}^{T\times N\times C}$ whose element range is $(-1, 1)$. If the activation output is positive, noise suppression will be performed, and if it's negative, feature enhancement will be performed. The generation of threshold τ_{st} can be formulated as:

$$\tau_{st} = \tanh(W_1(\varphi(X)) + W_2(\psi(X))) \tag{5}$$

where, $\varphi(.)$ and $\psi(.)$ compute spatial mean and temporal mean respectively. $W_1(.)$ and $W_2(.)$ are two 1×1 convolution operators. Then the input tensor X and threshold tensor τ_{st} are fed to Eq. 2 for spatio-temporal feature S&E.

B. Channel-wise S&E

In CW-S&E block, the input tensor is firstly reduced to a $1\times1\times C$ tensor by computing spatio-temporal mean (global mean pooling), which makes the block focus solely on feature channels. As shown in Fig. 6, then 1×1 convolution is used

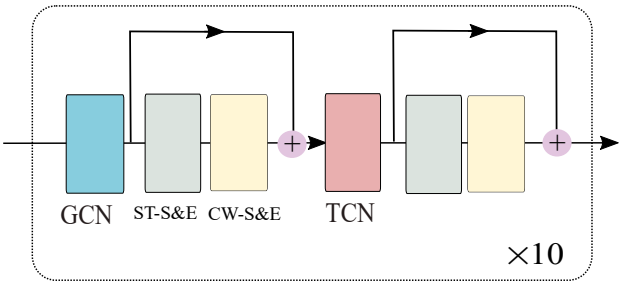

Fig. 7. General architecture of networks with ST-S&E and CW-S&E incorporated. Two residual links are added to make gradient propagate more effectively.

to exploit the relation among channels. The output tensor is activated with $tanh$ which generates a $1\times1\times C$ threshold tensor. we name the this part channel-wise threshold generation network (CW-TGN). The process can be formulated as:

$$\tau_c = \tanh(W_4(\phi(X'))) \tag{6}$$

where X' is the input tensor, $\phi(.)$ computes the spatio-temporal mean, $W_4(.)$ is the 1×1 convolution operator. Then the input tensor X and threshold tensor τ_c are fed to Eq. 2 for channel-wise feature S&E.

C. GCN with ST-S&E and CW-S&E

We incorporate proposed ST-S&E and CW-S&E into three baseline networks. The first one is the CTR-GCN [15] which is a powerful GCN based method and is currently among the top-ranked methods. The second one is a modified version of ST-GCN [1] with adaptive topology and temporal modeling as in [15], we name this baseline network modified ST-GCN (m-ST-GCN). Although there are slightly more parameters in m-ST-GCN, it can be trained much faster than CTR-GCN. The third baseline is HR-Head [19], we obtain the same number of parameters and FLOPs as CTR-GCN, suggesting that their network structures may be identical, the difference lies in the loss function. GCN and temporal convolution are stacked and repeated in all of the three baseline networks. ST-S&E and CW-S&E are added to each stacked block as shown in Fig. 7. In order to make gradient propagate more effectively, we introduce residual structure as shown in Fig. 7. The incorporation of ST-S&E and CW-S&E brings about only minor increment in the number of parameters and FLOPs as shown in Tab. I.

Real-time performance is crucial for industrial applications. We have listed the inference speed (instance/second) for three baseline methods and the method incorporating the module proposed in this paper, as shown in Tab. I. These tests were conducted on an RTX2080Ti GPU. Each instance is a 64 frame skeleton sequence. It can be observed from the table that, despite a slight decrease in inference speed, the method proposed in this paper can still guarantee real-time performance.

Cross-entropy loss is adopted as the loss function of the network. The model was trained with Stochastic Gradient Descent (SGD) that incorporated a momentum of 0.9 and a weight decay of 0.0004. The training was conducted over 65

TABLE I
PARAMETERS, FLOPS, INFERENCE SPEED (INSTANCE/SECOND) AND TRAINING TIME (MINUTE/EPOCH) OF THE BASELINE NETWORKS WITH AND WITHOUT ST-S&E AND CW-S&E.

Methods	Param. (M)	FLOPs (G)	inf. speed (ins./s)	Training time (min/epoch)
m-ST-GCN	2.09	2.41	228	3.5
CTR-GCN	1.44	1.79	47.4	15.5
FR-Head	1.44	1.79	47.4	19.5
m-ST-GCN+2S&E	3.19	2.5	80.6	8.5
CTR-GCN+2S&E	2.54	1.88	34	20.5
FR-Head+2S&E	2.54	1.88	34	24.5

epochs, with the first 5 epochs employing a warmup strategy [25] to enhance the stability of the training process. The initial learning rate was set to 0.1, and it was reduced by a factor of 0.1 at epochs 35 and 55. For both the NTU RGB+D and NTU RGB+D120 datasets, the batch size was set to 64. We also list in Tab. I the training times of three baselines with and without proposed module for one epoch on the X-Sub benchmark of NTU RGB+D 120 dataset.

V. EXPERIMENTS

A. Datasets

The proposed method is evaluated on three widely used skeleton based action recognition datasets: NTU RGB+D 60 [26], NTU RGB+D 120 [27] and Northwestern-UCLA [28]. NTU RGB+D 60 is a large-scale 3D human activity dataset having 56,880 videos of 60 classes. Two benchmarks are recommended: (1) cross-subject (Xsub): training data comes from 20 subjects, and testing data comes from the other 20 subjects. (2) cross-view (X-view): training data comes from camera views 2 and 3, and testing data comes from camera view 1. NTU RGB+D 120 is currently the largest dataset in skeleton based action recognition which contains 114,480 samples of 120 classes captured from with 32 different camera setups. Two benchmarks are recommended: (1) cross-subject (X-Sub): training samples are from 53 subjects, and testing samples are from the other 53 subjects. (2) cross-setup (X-Set): training samples are from setups with even setup IDs, and testing samples are from setups with odd IDs. Northwestern-UCLA dataset is collected with three Kinect sensors from multiple viewpoints. It contains 1494 video clips of 10 action categories. As in [28], training data is from the first two sensors, and testing data is from the other sensor.

B. Comparison With State-of-The-Arts

The performance of proposed method on three datasets are compared with state-of-the-art methods [1], [4], [15], [19], [29]–[32], and the comparison results are listed in Tab. II, Tab. III and Tab. IV. We give the performance of proposed ST-S&E and CW-S&E incorporated into two baseline networks: m-ST-GCN and CTR-GCN. Incorporating ST-S&E and CW-S&E into m-ST-GCN enhances the performance significantly in all the three datasets. All these performances reach state-of-the-art levels. We also gain improvements by incorporating ST-S&E and CW-S&E into CTR-GCN which is already among the top performers.

TABLE II
PERFORMANCES OF PROPOSED MODULES INCORPORATED INTO THREE BASELINES AGAINST STATE-OF-THE-ART METHODS ON NTU RGB+D 60 DATASET.

Methods (year)	X-Sub(%)	X-View(%)
ST-GCN [1] (2018)	81.5	88.3
2s-AGCN [29] (2019)	88.5	95.1
MSTGCN [33] (2021)	91.5	96.6
4s-MTT+Shift-GCN [30] (2022)	90.8	96.7
EfficientGCN-B4 [34] (2022)	90.8	96.7
Ta-CNN [35] (2022)	90.4	94.8
GSTLN [36] (2023)	91.9	96.6
TD-GCN [37](2023)	92.8	96.8
LKA-GCN [38](2023)	90.7	96.1
TranSkeleton [39](2023)	92.8	97.0
SiT-MLP [40](2024)	92.3	96.8
m-ST-GCN [15] (2021)	91.5	96
CTR-GCN [15] (2021)	92.4	96.8
HR-Head [19](2023)	92.8	96.8
m-ST-GCN+2S&E	92.1	96.4
CTR-GCN+2S&E	92.6	96.9
HR-Head+2S&E	**93.0**	**97.1**

Apart from these two baselines, we also embedded the proposed module in this paper into another baseline: HR-Head [19]. This method is also based on GCN and outperforms CTR-GCN. After incorporating our module, it achieved significant improvements on three datasets as shown in Tab. II, Tab. III and Tab. IV. This once again demonstrates the effectiveness of our method.

TABLE III
PERFORMANCES OF PROPOSED MODULES INCORPORATED INTO THREE BASELINES AGAINST STATE-OF-THE-ART METHODS ON NTU RGB+D 120 DATASET.

Methods (year)	X-Sub(%)	X-Set(%)
2s-AGCN [29] (2019)	82.9	84.9
MSTGCN [33] (2021)	87.5	88.8
2s-MTT+AGCN [30] (2022)	86.1	87.6
EfficientGCN-B4 [34] (2022)	88.7	88.9
Ta-CNN [35] (2022)	85.4	86.8
GSTLN [36] (2023)	88.1	89.3
LKA-GCN [38](2023)	86.3	87.8
TranSkeleton [39](2023)	89.4	90.5
SiT-MLP [40](2024)	89.0	90.5
m-ST-GCN [15] (2021)	88.4	88.3
CTR-GCN [15] (2021)	88.9	90.6
HR-Head [19](2023)	89.5	90.9
m-ST-GCN+2S&E	89.1	90.1
CTR-GCN+2S&E	89.2	90.7
HR-Head+2S&E	**89.7**	**91.0**

C. Ablation Studies

Roles of ST-S&E and CW-S&E. We validate the effectiveness of the two components in ablation experiments on NTU RGB+D 120 and NTU RGB+D 60. We adopt the m-ST-GCN as the baseline network, and two modules: ST-S&E and CW-S&E are added to the baseline network as baseline+2S&E. And removing ST-S&E and CW-S&E respectively result in the dropping of the performance compared with baseline+2S&E. The network with either block out of two yields better results than the baseline. The results are listed in Tab. V and Tab. VI. This demonstrates the effectiveness of both ST-S&E and CW-S&E.

TABLE IV
PERFORMANCES OF PROPOSED MODULES INCORPORATED INTO THREE BASELINES AGAINST STATE-OF-THE-ART METHODS ON NORTHWESTERN-UCLA DATASET.

Methods (year)	Northwestern-UCLA Top-1 (%)
AGC-LSTM [31] (2021)	93.3
shift-GCN [4] (2020)	94.6
DC-GCN+ADG [32] (2020)	95.3
Ta-CNN [35] (2022)	96.1
GSTLN [36] (2023)	94.8
SiT-MLP [40](2024)	96.5
m-ST-GCN [15] (2021)	95.3
CTR-GCN [15] (2021)	96.5
HR-Head [19](2023)	96.8
m-ST-GCN+2S&E	96.3
CTR-GCN+2S&E	96.8
HR-Head+2S&E	**97.0**

Roles of noise suppression and feature enhancement. By relaxing the non-negative constraint of soft-thresholding, we are able to achieve simultaneous noise suppression and feature enhancement. The Tanh function is placed at the end of the network to generate threshold values within the range of -1 to 1. If the Tanh functions is replaced with the Sigmoid function, the generated threshold values are within the range of 0 to 1, only noise suppression operation is performed. We evaluate the performance of networks with ST-S&E and CW-S&E using Sigmoid function (baseline+2S&E (Sigmoid)), which means the non-negative constraint is kept, while feature enhancement is disabled. The results are listed in Tab. V and Tab. VI. We can see from the table that after adopting the Sigmoid function, the performance of the network has decreased. At this time, the two modules only retain the function of noise suppression. Nevertheless, the performance is still improved compared with the baseline. This can indicate that both noise suppression and feature enhancement are helpful for improving performance, and single noise suppression is also effective for improving performance.

TABLE V
ABLATION EXPERIMENTS ON NTU RGB+D 60 DATASET.

Methods	X-Sub(%)	X-Set(%)
baseline	91.5	96
baseline+ST-S&E	91.7	96.2
baseline+CW-S&E	91.9	96.3
baseline+2S(Sigmoid)	91.6	96.2
baseline+2S&E	**92.1**	**96.4**

TABLE VI
ABLATION EXPERIMENTS ON NTU RGB+D 120 DATASET.

Methods	X-Sub(%)	X-Set(%)
baseline	88.4	88.3
baseline+ST-S&E	88.5	89.1
baseline+CW-S&E	88.7	89.4
baseline+2S(Sigmoid)	88.7	89.7
baseline+2S&E	**89.1**	**90.1**

Evaluation with varying widths and depths. We evaluated the performance of our proposed method in this paper when applied to baseline networks with varying widths and depths. First, we adjusted the network's depth by altering the number of GCN-TCN blocks from 10 to 5, 8, 12, and 15, respectively, and tested its performance in each case. Then, we kept the number of GCN-TCN blocks constant at 10 and adjusted the intermediate representation feature dimensions of each block. The original dimensions were set as follows: 64 for blocks 1-4, 128 for blocks 5-7, and 256 for blocks 8-10. We tested the performance under two scenarios: reducing these dimensions by 25% (i.e., 48 for blocks 1-4, 96 for blocks 5-7, and 192 for blocks 8-10) and increasing them by 25% (i.e., 80 for blocks 1-4, 160 for blocks 5-7, and 320 for blocks 8-10). The results are listed in Tab. VII. As can be seen from the table, the default depth and width provide the optimal performance. Reducing either the depth or the width results in a significant decline in performance, while increasing either the depth or the width maintains the performance relatively stable with a slight decline.

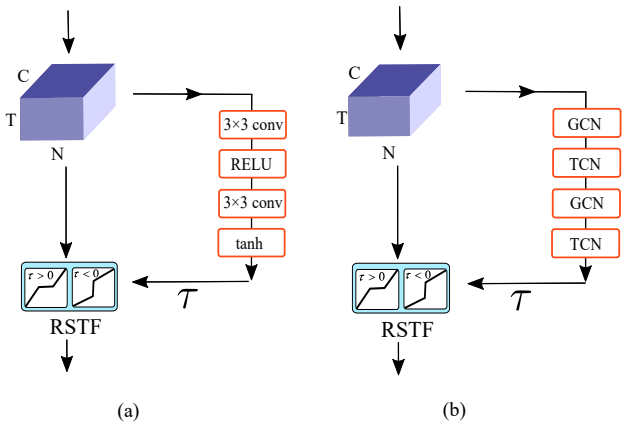

Fig. 8. Two alternative threshold generation network architectures: (a) 2D convolution based method, (b) GCN and TCN based method.

TABLE VII
ABLATION EXPERIMENTS WITH VARYING WIDTHS AND DEPTHS.

Methods	X-Sub(%)	X-Set(%)
baseline+2S&E	**89.1**	**90.1**
depth-5	84.2	83.7
depth-8	87.6	87.9
depth-12	89.0	89.7
depth-15	88.7	89.4
width+%25	88.7	89.7
width-%25	87.7	88.1

Evaluation with other alternatives of TGN. We previously presented a scheme for the threshold generation network (TGN), which primarily considers the efficiency of feature encoding. Therefore, we adopted average pooling for this operation. We also tested other design schemes for TGN, as shown in Fig. 8. The first scheme uses two consecutive 2D convolutions, followed by activation with the tanh function, which we denote as +2S&E(conv). The second scheme employs two consecutive combinations of GCN-TCN, activated by the tanh function, denoted as +2S&E(gcn-tcn). Evaluations were conducted on the xsub benchmark of the NTU RGB+D 120 dataset. The experimental results are presented in Tab. VIII.

We can observe that compared to the baseline, both TGN variants achieved some degree of improvement. Specifically,

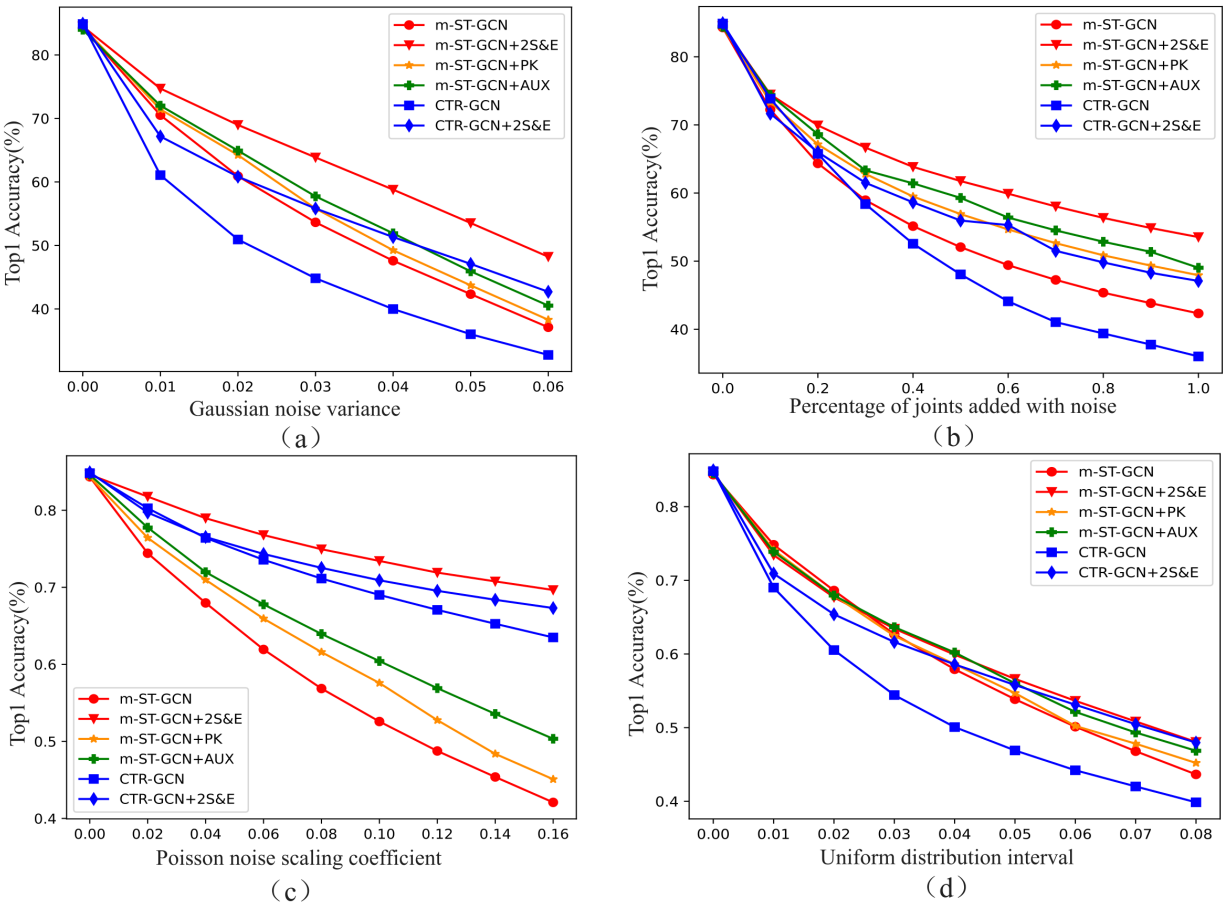

Fig. 9. The evaluation results of anti-noise experiments by adding noise in four different ways.

+2S&E(conv) achieved a minor improvement, which may be due to the mismatch between 2D convolutions and the topological structure of skeleton data. In contrast, +2S&E(gcn-tcn) obtained better results which is near that of the proposed approach. However, the FLOPS of these both the two schemes are significantly higher as shown in Tab. VIII.

TABLE VIII
ABLATION EXPERIMENTS ON DIFFERENT SCHEMES OF THRESHOLD-GENERATION NETWORK.

Methods	X-Sub(%)	Param.	FLOPS
baseline	88.4	2.09M	2.41G
+2S&E	**89.1**	3.19M	2.5G
+2S&E(conv)	88.6	2.46M	3.03G
+2S&E(gcn-tcn)	89.0	2.48M	3.07G

D. Anti-noise performance

In order to evaluate the anti-noise ability of proposed method, we have carried out the anti-noise experiments. Four of the networks compared are the baseline network m-ST-GCN and m-ST-GCN+2S&E, CTR-GCN [15] and CTR-GCN+2S&E. We have also incorporated two additional de-noising methods into the m-ST-GCN baseline for comparison. The first method is based on prior knowledge [6]. We implemented the auto-encoder described in [6] and used it to preprocess both the training and testing sets. This method is denoted as m-ST-GCN+PK. The second method is based on auxiliary supervision data [7]. We implemented the multi-view regression loss and supervised the training process with multi-view data. This method is denoted as m-ST-GCN+AUX.

And the dataset for evaluation is the joint modality of cross-subject benchmark of NTU RGB+D 120. All networks are trained with original training data and are tested on testing data with different levels and different types of noise added. We adopt four ways to add noise:

- Zero-mean Gaussian noise with variable standard deviations is added to all joint points.
- zero-mean Gaussian noise of fixed standard deviation (0.05) is added to randomly selected variable proportions of joint points.
- Poisson-distributed noise with $\lambda = 10$, multiplied by a variable scaling coefficient, is added to all joint nodes.
- Noise uniformly distributed within a variable interval $[-d, d]$ is added to all joint nodes.

The evaluation results are shown in Fig. 9. From the figure, we can see that under these various noise conditions, m-ST-GCN+2S&E exhibits the strongest noise resistance. When the CTR-GCN is equipped with our shrink&enhancement module, its noise resistance also improves, which proves the effective-

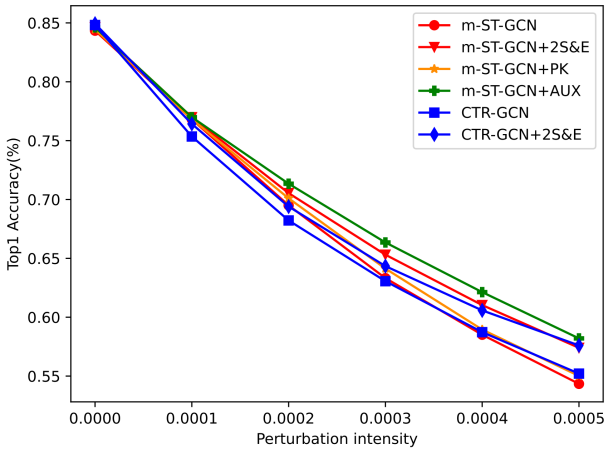

Fig. 10. The evaluation results of adversarial attack experiment.

ness of the method proposed in this paper. Additionally, the other two noise reduction methods, m-ST-GCN+PK and m-ST-GCN+AUX, both demonstrate better noise resistance than the baseline, with m-ST-GCN+AUX performing slightly better. This may be due to the introduction of additional supervision data.

E. Adversarial attack experiment

Finally, we attempt to test the robustness of these methods against adversarial attacks. We employ a classic adversarial attack method, the Fast Gradient Sign Method (FGSM) [41], to introduce adversarial perturbations to all input nodes during the inference phase. The intensity of these perturbations is controlled by the parameter ε, allowing us to obtain different levels of adversarial attacks by adjusting ε.

The experimental results are shown in the Fig. 10, which reveals that m-ST-GCN+AUX demonstrates superior robustness against adversarial attacks, followed closely by the methods proposed in this paper, namely m-ST-GCN+2S&E and CTR-GCN+2S&E. Both of these methods outperform their respective baselines. Although the methods presented in this paper were not specifically designed for adversarial attacks, these adversarial perturbations can still be regarded as noise, and the proposed methods remain effective.

F. Result visualization

Finally, we present some visualization results as shown in Fig. 11. Five joints are randomly selected in the test dataset of the joint modality of NTU RGB+D 60 and Gaussian noise of zero mean and variance 0.05 was added to these joints. Two networks: m-ST-GCN and m-ST-GCN+2S&E are evaluated which are trained beforehand with original training dataset. And we take the activation results from the end of the first of the 10 units in Fig. 7. we do not choose the activation results from deeper layers because as the network goes deeper, the spatial correspondence between the tensor and the initial data will become weaker, thus it is hard to distinguish those noise-polluted elements from the feature map.

As shown in Fig. 11, different colors represent the different activation degree of the nodes, and the nodes marked with a red circle indicate the joints that have been added with noise, which can have a negative impact on the classification results. After adding the ST-S&E and CW-S&E module, most of the noisy joints were suppressed (note the color change of the joints in (a) to (f)). The blue arrows indicate the joints that need to be focused on, as their activation degree leads to different classification results. For example, in the instance (a), the original method incorrectly classified the sample as belonging to the ”A30. typing on a keyboard” category, whereas after adding the ST-S&E and CW-S&E module, it was correctly identified as belonging to the ”A36. shake head” category. The former focused on the performer’s hand movements, while the latter focused on the performer’s head movements. In example (b), the proposed method successfully focuses on the importance of both hand and head, leading to correct prediction.

VI. Conclusion and Future Work

In this paper, we firstly introduce the deep learning based soft-thresholding framework into the problem of noise handling for skeleton based action recognition, which is fully implicit and does not require priors or additional supervision information. Then we relax the non-negative condition of soft-thresholding enabling the framework to perform feature enhancement and noise suppression simultaneously. In order to fully exploit the inherent relations among spatio-temporal-channel dimensions, we propose two decoupled suppression and enhancement stages.

Experimental results have demonstrated the effectiveness of proposed method on three public skeleton based action recognition datasets. And experimental results on datasets polluted by different types of noise have demonstrated the effectiveness and robustness of proposed method in noise handling.

We believe that the solution of simultaneous noise suppression and feature enhancement we propose can also be applied to other tasks, as long as the data of these tasks contain noise and need to enhance key features. We will further investigate this in our future work. In addition, the function in the figure may not be the optimal one, whether there exist other functions with better performance is also a problem that needs further exploration.

References

[1] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[2] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, “Semantics-guided neural networks for efficient skeleton-based human action recognition,” in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1112–1121.

[3] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Actional-structural graph convolutional networks for skeleton-based action recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3595–3603.

[4] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, “Skeleton-based action recognition with shift graph convolutional network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 183–192.

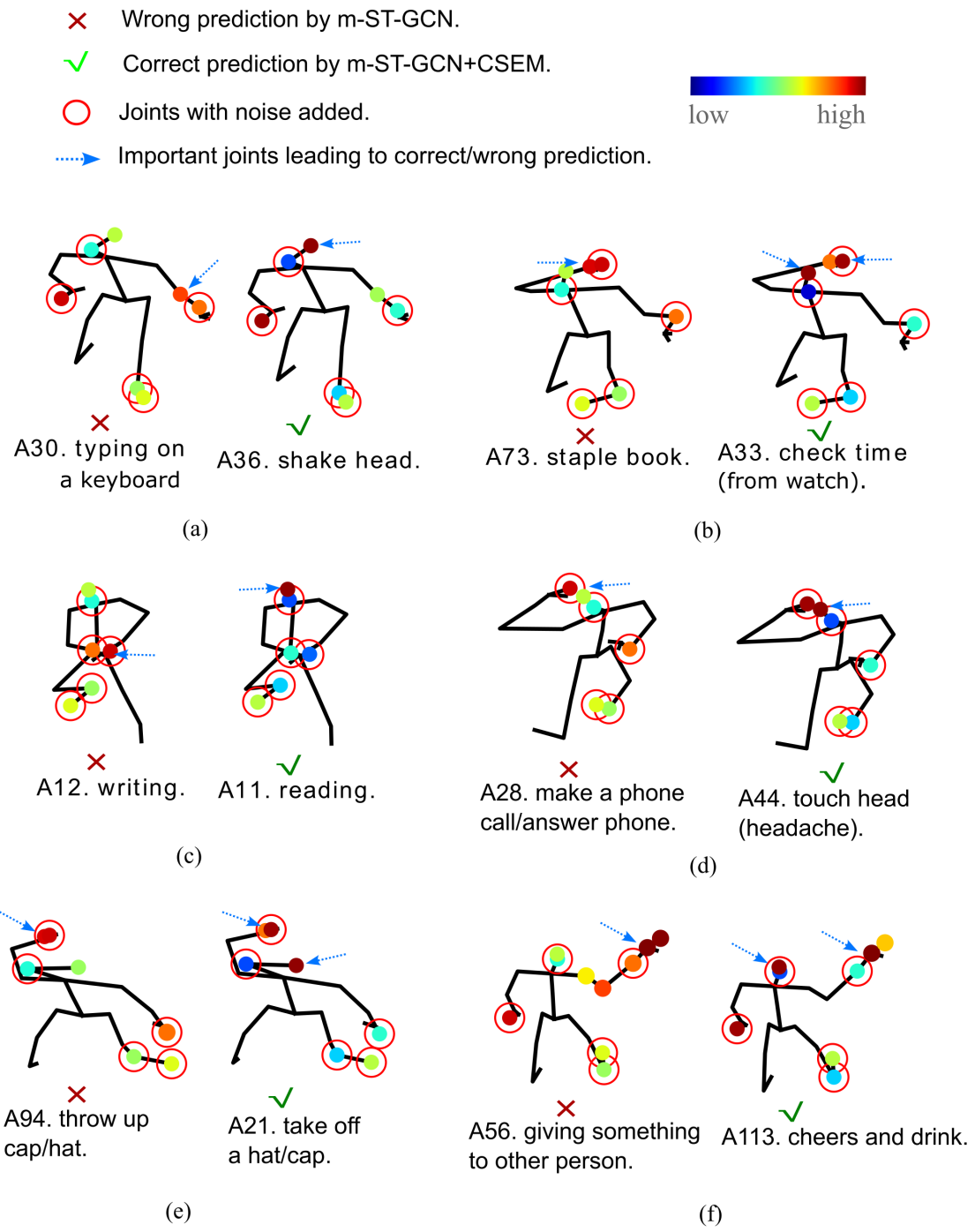

Fig. 11. Visualization of several examples in evaluation results.

[5] Q. Nie, J. Wang, X. Wang, and Y. Liu, "View-invariant human action recognition based on a 3d bio-constrained skeleton model," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3959–3972, 2019.
[6] G. G. Demisse, K. Papadopoulos, D. Aouada, and B. Ottersten, "Pose encoding for robust skeleton-based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 188–194.
[7] S. Song, J. Liu, L. Lin, and Z. Guo, "Learning to recognize human actions from noisy skeleton data via noise adaptation," *IEEE Transactions on Multimedia*, vol. 24, pp. 1152–1163, 2021.
[8] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1290–1297.
[9] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3288–3297.
[10] T. Huynh-The, C.-H. Hua, and D.-S. Kim, "Encoding pose features to images with data augmentation for 3-d action recognition," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3100–3111, 2019.
[11] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.
[12] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer lstm networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 148–157.
[13] H. Qu, Y. Cai, and J. Liu, "Llms are good action recognizers," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 18 395–18 406.

[14] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution lstm for skeleton based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6882–6892.

[15] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 359–13 368.

[16] Z. Qin, Y. Liu, P. Ji, D. Kim, L. Wang, R. I. McKay, S. Anwar, and T. Gedeon, "Fusing higher-order features in graph neural networks for skeleton-based action recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 4783–4797, 2022.

[17] W. Shi, D. Li, Y. Wen, and W. Yang, "Occlusion-aware graph neural networks for skeleton action recognition," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 10, pp. 10 288–10 298, 2023.

[18] J. Lee, M. Lee, D. Lee, and S. Lee, "Hierarchically decomposed graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 444–10 453.

[19] H. Zhou, Q. Liu, and Y. Wang, "Learning discriminative representations for skeleton based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 608–10 617.

[20] J. Liu, C. Chen, and M. Liu, "Multi-modality co-learning for efficient skeleton-based action recognition," *arXiv preprint arXiv:2407.15706*, 2024.

[21] D. L. Donoho, "De-noising by soft-thresholding," *IEEE transactions on information theory*, vol. 41, no. 3, pp. 613–627, 1995.

[22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[23] K. Isogawa, T. Ida, T. Shiodera, and T. Takeguchi, "Deep shrinkage convolutional neural network for adaptive noise reduction," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 224–228, 2017.

[24] M. Zhao, S. Zhong, X. Fu, B. Tang, and M. Pecht, "Deep residual shrinkage networks for fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4681–4690, 2019.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010–1019.

[27] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2684–2701, 2019.

[28] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2649–2656.

[29] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 026–12 035.

[30] J. Kong, Y. Bian, and M. Jiang, "Mtt: Multi-scale temporal transformer for skeleton-based action recognition," *IEEE Signal Processing Letters*, vol. 29, pp. 528–532, 2022.

[31] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "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*, 2019, pp. 1227–1236.

[32] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling gcn with dropgraph module for skeleton-based action recognition," in *European Conference on Computer Vision*. Springer, 2020, pp. 536–553.

[33] Z. Chen, S. Li, B. Yang, Q. Li, and H. Liu, "Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 2, 2021, pp. 1113–1122.

[34] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Constructing stronger and faster baselines for skeleton-based action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 2, pp. 1474–1488, 2022.

[35] K. Xu, F. Ye, Q. Zhong, and D. Xie, "Topology-aware convolutional neural network for efficient skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2866–2874.

[36] M. Dai, Z. Sun, T. Wang, J. Feng, and K. Jia, "Global spatio-temporal synergistic topology learning for skeleton-based action recognition," *Pattern Recognition*, p. 109540, 2023.

[37] J. Liu, X. Wang, C. Wang, Y. Gao, and M. Liu, "Temporal decoupling graph convolutional network for skeleton-based gesture recognition," *IEEE Transactions on Multimedia*, vol. 26, pp. 811–823, 2023.

[38] Y. Liu, H. Zhang, Y. Li, K. He, and D. Xu, "Skeleton-based human action recognition via large-kernel attention graph convolutional network," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2575–2585, 2023.

[39] H. Liu, Y. Liu, Y. Chen, C. Yuan, B. Li, and W. Hu, "Transkeleton: Hierarchical spatial–temporal transformer for skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 4137–4148, 2023.

[40] S. Zhang, J. Yin, Y. Dang, and J. Fu, "Sit-mlp: A simple mlp with point-wise topology feature learning for skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[41] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.