Vertex Feature Encoding and Hierarchical Temporal Modeling in a Spatio-Temporal Graph Convolutional Network for Action Recognition

Konstantinos Papadopoulos, Enjie Ghorbel, Djamila Aouada and Björn Ottersten
Interdisciplinary Centre for Security, Reliability and Trust (SnT) University of Luxembourg, Luxembourg
Email: {konstantinos.papadopoulos, enjie.ghorbel, djamila.aouada, bjorn.ottersten}@uni.lu

***Abstract*—Spatio-temporal Graph Convolutional Networks (ST-GCNs) have shown great performance in the context of skeleton-based action recognition. Nevertheless, ST-GCNs use raw skeleton data as vertex features. Such features have low dimensionality and might not be optimal for action discrimination. Moreover, a single layer of temporal convolution is used to model short-term temporal dependencies but can be insufficient for capturing both long-term. In this paper, we extend the Spatio-Temporal Graph Convolutional Network for skeleton-based action recognition by introducing two novel modules, namely, the Graph Vertex Feature Encoder (GVFE) and the Dilated Hierarchical Temporal Convolutional Network (DH-TCN). On the one hand, the GVFE module learns appropriate vertex features for action recognition by encoding raw skeleton data into a new feature space. On the other hand, the DH-TCN module is capable of capturing both short-term and long-term temporal dependencies using a hierarchical dilated convolutional network. Experiments have been conducted on the challenging NTU RGB-D 60, NTU RGB-D 120 and Kinetics datasets. The obtained results show that our method competes with state-of-the-art approaches while using a smaller number of layers and parameters; thus reducing the required training time and memory.**

I. Introduction

Skeleton-based human action recognition has received a huge amount of attention in various applications, such as video surveillance, coaching, and rehabilitation [1], [2], [3], [4], [5]. Recently, deep learning-based approaches have achieved impressive performance on large-scale datasets, by learning the appropriate features automatically from the data [6], [7], [8], [9], [10], [11]. These approaches rely either on Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN). However, they usually represent the skeleton sequences as vectors or 2D grids, ignoring inter-joint dependencies.

To express joint correlations both spatially and temporally, Yan et al. introduced the Spatial Temporal-Graph Convolutional Network (ST-GCN) [12]. Their work takes advantage of Graph Convolutional Networks (GCN) [13] extending the classical CNNs to graph convolutions. This architecture represents skeleton sequences as a graph composed of both temporal and spatial edges, by respectively considering the inter and intra-frame connections of joints. The effectiveness of this approach has motivated several extensions [14], [15], [16] which, consider the most informative connections between joints instead of the predefined natural skeleton structure or construct the spatio-temporal graphs using additional features such as bone lengths.

However, all these methods use only raw skeleton features (joint coordinates and/or bone lengths) for the construction of spatio-temporal graphs. While offering a high-level description of the human body structure, these features have low dimensionality and thus may be lacking discriminative power for action recognition. Indeed, hand-crafted approaches have shown the limitation of using only raw skeleton joints as features in action recognition [17], [18]. Furthermore, the temporal dependencies of the graph are modeled by a single temporal convolutional layer. As a result, critical long-term dependencies might be not consistently described. Moreover, these approaches make use of a considerable number of ST-GCN blocks (10, in most cases), which significantly increase the number of parameters and consequently the computational complexity and the required memory.

In this paper, we assume that by encoding the vertex features in an end-to-end manner and modeling temporal long-term and short-term dependencies, less number of layers (and consequently parameters) will be needed. For that reason, two modules are introduced. The first module, referred to as Graph Vertex Feature Encoder (GVFE), is a trainable layer that transforms the feature space from the Euclidean coordinate system of joints to an end-to-end learned vertex feature space, optimized jointly with the ST-GCN. The new feature space offers more robust discriminative capabilities as a result of its higher dimensionality [19]. The second module incorporates a hierarchical structure of dilated temporal convolutional layers for modeling short-term and long-term temporal dependencies by increasing the temporal receptive field in multiple levels. It is termed Dilated Hierarchical Temporal Graph Convolutional Network (DH-TCN) and replaces the standard temporal convolutional layers found in the ST-GCN block. With the use of these two modules, we show that fewer layers are needed to reach the same or even higher performance in action recognition while needing less memory and training time than previous ST-GCN based approaches such as [16].

In summary, our contributions are the following:

- Introduction of a Graph Vertex Feature Encoder (GVFE) module for encoding vertex features;
- Proposal of a Dilated Hierarchical Temporal Graph Convolutional Network (DH-TCN) module for modeling

short and long-term dependencies;
- Design of a more compact and efficient graph-based framework for action recognition trained in an end-to-end manner;
- Presentation of experimental validation and analysis of our approach on three challenging datasets.

The remainder of this paper is organized as follows: in Section II, an extensive literature review is presented. Section III recalls the background related to Spatio-Temporal Graph Convolutional Networks (ST-GCN) applied to action recognition. Section IV details the proposed framework. Section V presents the experiments and analyzes the results. Finally, Section VI concludes this paper and discusses possible extensions of this work.

II. Related Work

Over the last decade, the availability of 3D skeletons through RGB-D sensors has significantly boosted the development of numerous skeleton-based action recognition methods. Earlier methods have mainly introduced novel hand-crafted features aiming to describe the human motion. For example, human skeleton sequences can be modelled as trajectories lying in Euclidean or Riemannian spaces [3], [18], [4], [20], [21], as statistical-based representations [1], [2] or as pairwise relative positions of joints [22], [23], etc.

Recently, deep-learning based approaches have gained popularity and have shown notable performance, especially on large-scale datasets [24], [25], [26], [27], [10], [28], [29], [30], [31]. Instead of hand-crafting features, deep-learning based approaches are able to learn them automatically. Long Short-Term Memory (LSTM) networks, initially designed for modeling sequential data, have particularly shown great potential in action recognition. In fact, compared to conventional Recurrent Neural Networks (RNN), LSTM can handle long-term dependencies and, thus, mitigate the problem of vanishing gradients [26], [27], [27], [10]. However, LSTM-based models cannot be parallelized and thus are generally hard to train. CNN have also shown their efficiency for the action recognition task [24], [28], [29], [30], [7], [32], [33].

Nevertheless, both CNN and LSTM fail to exploit the spatio-temporal structure of 3D skeletons that can naturally be seen as graphs rather than Euclidean data. Recently, Graph Convolution Networks (GCN) [13], [34], [35], [36], [37] generalizing CNN from 2D grids to graphs have been introduced and adopted for skeleton-based action recognition [12], [38], [14], [16], [39]. Yan et al. [12] were among the first to utilize GCN in skeleton-based action recognition. They represented skeleton sequences as spatio-temporal graphs by preserving the inter-joint connections and linking temporally the same joints from different time steps and consequently designed a suitable network called Spatio-Temporal Graph Convolution Network (ST-GCN). Considering the fact that the graph edges defined by the natural skeleton structure might be not optimal for the task of action recognition, some approaches extended ST-GCN in order to capture more relevant dependencies among joints [14], [16]. To respectively capture action-specific and higher-order dependencies, an encoder-decoder module called A-link inference has been designed and a higher polynomial within the Spatial Graph Convolution has been used [16]. Shi et al. [14] proposed an Adaptive ST-GCN to adaptively learn joint connections in an end-to-end manner. Moreover, they made use of a two-stream network which combines first-order and second order joint information. On the other hand, Shi et al. [15] extended ST-GCN to Directed acyclic ST-GCN (D-ST-GCN) in order to capture the relationship between bones and joints. Si et al. [39] were the only ones attempting to extend the temporal modeling of ST-GCN that considers only short-term dependencies. To that aim, they introduced Attention Enhanced Graph Convolutional Long Short-Term Memory network (AGC-LSTM). Despite the relevance of such an approach, LSTM remain difficult to parallelize, as mentioned earlier in this section. Furthermore, all the presented graph-based approaches rely solely on joint and/or bone length features which might not be optimal for action recognition. In Section IV-A and Section IV-B, two novel graph-based modules aiming to overcome the two mentioned issues are presented.

III. Background

A. Skeleton Sequences as Graphs

Following a predefined structure indicating their interconnections, skeletons can be intuitively seen as graphs. Thus, Yan et al. [12] described skeleton sequences of J joints and T frames as spatio-temporal graphs in which, at each time instance t, each joint i is assumed to be a vertex. Then, two kinds of edges are constructed to connect vertices: spatial edges that are the natural spatial joint connections and temporal edges connecting the same joint across time. This spatio-temporal graph is denoted as $S = (V, E)$, with V the set of vertices and E the set of edges.

B. Spatio-Temporal Graph Convolutional Network

Considering the spatio-temporal graph S, a network called ST-GCN generalizing CNN to graphs has been proposed in [12]. In this work, for an input feature map $\mathbf{f_{in}}$, a spatial graph convolution is applied, such that:

$$\mathbf{f}_{out} = \mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{f}_{in}\mathbf{W}, \tag{1}$$

where $\mathbf{f_{out}}$ is the output feature map, $\mathbf{A}$ the adjacency matrix, $\mathbf{I}$ the identity matrix, $\mathbf{\Lambda} = [\Lambda^{ii}]_{i\in\{1,\dots,J\}}$ such that $\Lambda^{ii} = \sum_j (A^{ij} + I^{ij})$ and $\mathbf{W}$ is the weight matrix. For a graph of size (C_{in}, J, T), the dimension of the resulting tensor is (C_{out}, J, T), with C_{in} and C_{out} denoting respectively the number of input and output channels.

The temporal graph convolution consists of classical convolutions, performed on the output feature tensor $\mathbf{f}_{out}$. K denotes the kernel size of the temporal convolutional layer.

The input features $\mathbf{f}_{in}^{(1)}$ incorporated in the first ST-GCN layer correspond to the joint coordinates such that $\forall i$, $\mathbf{f}_{in}^{(1)}(v_i, t) = \mathbf{P}_i(t)$ with $\mathbf{P}_i(t)$ the 3D coordinate of the joint i at an instant t and consequently $C_{in} = 3$. While these first

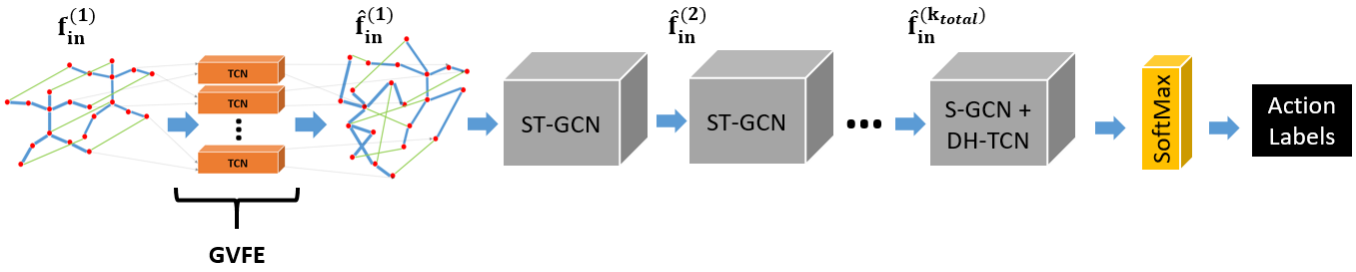

Fig. 1. Illustration of the proposed approach. In the first step, the GVFE module generates graph features. The new graph is given as an input to the Modified ST-GCN network in which the last block is composed of a Spatio-Graph Convolutional Network (S-GCN) and a Dilated Hierarchical Temporal Convolutional Network (DH-TCN). Finally, a SoftMax layer classifies the spatio-temporal graph features resulting from the last Modified ST-GCN block.

layer features $\mathbf{f}_{in}^{(1)}$ offer a representation easily understandable by the human, they might be not discriminative enough for the task of action recognition. Moreover, temporal graph convolutional layers capture only local dependencies, resulting in the presence of redundant information and neglecting long-term dependencies.

IV. Proposed Approach

In this section, the two novel modules, namely GVFE and DH-TCN, are presented. While GVFE aims at learning vertex features, DH-TCN temporally summarizes spatio-temporal graphs and consequently models long-term as well as short-term dependencies. These two modules are integrated with the original ST-GCN [12] framework. This full pipeline is depicted in Fig. 1 and is trained in an end-to-end manner. It is important to note that these modules are also complementary to other ST-GCN extensions such as AS-GCN [16].

A. Graph Vertex Feature Encoding (GVFE)

As mentioned in Section I, considering raw skeleton joint data as vertex features might not be informative enough for action recognition. The dimensionality of the raw skeleton joints is low and consequently not sufficient enough for effective feature discrimination. To enhance the discriminative power of vertex features, we introduce the GVFE module that is directly placed before the first ST-GCN block. GVFE maps 3D skeleton coordinates, traditionally used as input features to the first ST-GCN block $\mathbf{f}_{in}^{(1)}(v_i) = \mathbf{P}_i$ with $i \in \{1, ..., J\}$, from the Cartesian coordinate system $\mathbb{R}^3$ to a learned feature space $\mathcal{M} \subseteq \mathbb{R}^{C_{out}}$ of higher dimensionality $C_{out} > 3$. The higher dimensionality offers robust discriminative capabilities and better generalization, as discussed in [19]. GVFE module preserves the spatial structure of skeletons so that the joint dependencies are modeled. Since this module is trained in an end-to-end manner by optimizing the recognition error, we expect to obtain a more sufficient for action recognition feature space $\mathcal{M}$.

For each joint i, a separate Temporal Convolutional Network (TCN) is employed to encode raw data, as illustrated in Fig. 2. In this context, TCNs show strong potential since (a) they do not allow information to flow from the future states to the past states, (b) the input and output sequences have the same length and (c) they model temporal dependencies. For each joint i, the new graph vertex features $\hat{\mathbf{f}}_{in}^{(1)}(v_i)$ obtained after applying the TCN are computed as follows,

$$\hat{\mathbf{f}}_{in}^{(1)}(v_i) = \mathbf{W}_i^{TCN} * \mathbf{f}_{in}^{(1)}(v_i) = \mathbf{W}_i^{TCN} * \mathbf{P}_i, \qquad (2)$$

where $\{\mathbf{W}_i^{TCN}\}_{1 \leq i \leq J}$ is the collection of tensors containing the kernel filters $\{\mathbf{W}_{i,j}^{TCN}\}$ of dimension $\mathbb{R}^{C_{out} \times T_w \times C_{in}}$, with $j \in \{1, ..., C_{out}\}$ the index of the filter and T_w the temporal size of the filters. Note that we use the identity activation function. This module preserves the skeleton structure and has the advantage of being applicable to any graph-based network, regardless of the application.

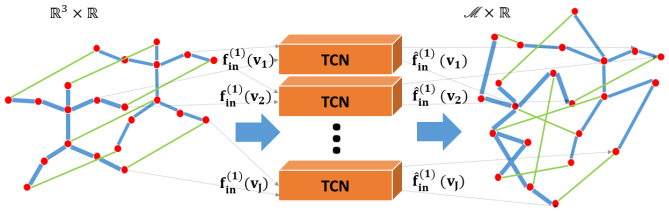

Fig. 2. Illustration of the GVFE module structure: it is composed of J TCN blocks. For each joint, one TCN block is separately used in order to conserve the natural skeleton structure. Note that this module preserves the skeleton joint connectivity.

B. Dilated Hierarchical Temporal Graph Convolutional Network

The modeling of temporal dependencies is crucial in action recognition. However, in several ST-GCN-based approaches [12], [14], [15], [16], temporal dependencies are modeled using only one convolutional layer. As a result, long-term dependencies that can be important for modeling actions are not well encoded.

To that end, we propose to replace the temporal convolutions of the ST-GCN block with a module that encodes both short-term and long-term dependencies. Given the output feature map $\mathbf{f}_{out}^{(k)}$ resulting from the k^{th} Spatial GCN (S-GCN) block (with $k \in [1, k_{total}]$ and k_{total} the total number of ST-GCN blocks), this module, termed Dilated Hierarchical Temporal Convolutional Network (DH-TCN), is composed of N successive dilated temporal convolutions. The association of these

two blocks is illustrated in Fig. 4. Each layer output $f_{temp}^{(k,n)}$ of order n of DH-TCN is obtained as follows,

$$\mathbf{f}_{temp}^{(k,n)} = F\Big(\mathbf{W}_i^{DH} *_l \mathbf{f}_{temp}^{(k,n-1)}\Big), \text{ with } \mathbf{f}_{temp}^{(k,0)} = \mathbf{f}_{out}^{(k)}, \quad (3)$$

where $\{\mathbf{W}^{DH}\}_{1\leq i\leq J}$ is the tensor containing the trainable temporal filters of dimension $\mathbb{R}^{C_{out}\times T_{w_1}\times C_{out}}$ with T_{w_1} their temporal dimension and $*_l$ refers to the convolution operator with a dilation of $l = 2^n, n \in [0, N-1]$.

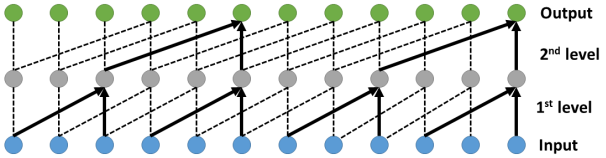

Fig. 3. Example of a 2-level dilated convolution on an input sequence. The first level encodes short-term dependencies, while the second level increases the receptive field and encodes longer-term dependencies.

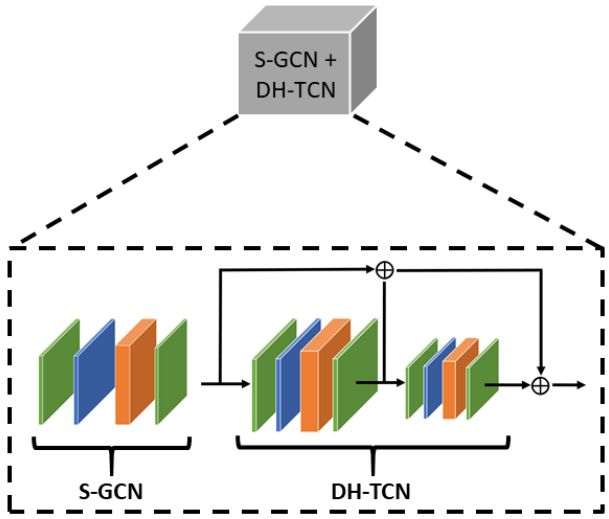

Fig. 4. Illustration of S-GCN + DH-TCN block. Spatial features are extracted from the S-GCN module and are, then, fed into DH-TCN module. Green color is used for Batch Normalization units, blue for ReLU and orange for 2D Convolutional Layers.

The hierarchical architecture with different dilation ensures the modeling of long-term dependencies. Dilated convolutions are proven to be efficient in modeling long-term dependencies [40] while at the same time they maintain efficiency. Architectures with dilated convolutions have been successful for audio generation [41], semantic segmentation [42] and machine translation [43]. An example of how dilated convolutions are applied in a two-level manner is illustrated in Fig. 3. At the same time, the residual connection depicted in Fig. 4 enables the preservation of the information of short-term dependencies.

The entire DH-TCN module is illustrated in Figure 4. Each hierarchical layer is composed of a dilated temporal convolution, a ReLU activation function, and a batch normalization. We use the DH-TCN module only on the last (k_{total}) ST-GCN block to avoid feature over-summarization and to reduce the number of trainable parameters.

V. Experiments

Our framework has been tested on three well-known benchmarks, namely NTU RGB+D 60 (NTU-60) [44], NTU RGB+D 120 (NTU-120) [45] and Kinetics [46] datasets.

A. Datasets and Experimental Settings

NTU RGB+D Dataset (NTU-60): NTU RGB+D is a Kinect-acquired dataset which consists of $56,880$ videos. This dataset includes 60 actions performed by 40 subjects. The data are collected from 3 different angles, at $-45°, 0°$ and $45°$ with respect to the human body. In our experiments, we follow the same protocols (cross-view and cross-subject settings) proposed in [44].

NTU RGB+D 120 Dataset (NTU-120): NTU RGB+D 120 Dataset extends the original NTU dataset by adding 60 additional action classes to the existing ones and 66 more subjects. The recording angles remain the same at $-45°, 0°$ and $45°$ with respect to the human body, but more setups (height and distance) are considered (32 instead of 18). We consider the same evaluation protocol (cross-setup and cross-subject settings) suggested in [45].

Kinetics Dataset (Kinetics): Kinetics is a large-scale dataset of $300,000$ RGB videos collected from YouTube. It includes 400 action classes. In order to add skeleton data to this dataset, Yan et al. [12] used the OpenPose toolbox [49] on every frame of each video clip. We follow the evaluation method reported in [12], where $240,000$ videos are used for training and $20,000$ videos are used for validation and we report the Top-1 and Top-5 accuracies.

B. Implementation Details

The implementation of our approach is based on the PyTorch ST-GCN [12] and AS-GCN [16] codes. In both approaches, we include the GVFE module before the first ST-GCN block and we replace the temporal convolutions of the last block with the DH-TCN module. For the spatial GCN, we use the same parameters suggested in [12]. The number of output channels in GVFE is set to $C_{out} = 8$ for AS-GCN and to 9 for ST-GCN and the temporal window of the DH-TCN module is set to $T_w = 9$. These parameters were chosen according to the analysis we conducted and presented in Tables II and III. As shown in these tables, the best performance is obtained for $T_w = 9$ and $C_{out} = 9$. The number of hierarchical temporal convolutional layers N in DH-TCN is set to 2 for two reasons. First, two layers offer adequate summarization of features without omitting important temporal dependencies and second we wanted to maintain a small number of trainable parameters. The Stochastic Gradient Descent optimizer is used with a decaying learning rate of 0.01. In contrast to [12], [16] that makes use of 10 ST-GCN or AS-GCN blocks, we use only 4 blocks with $k \in \{1, ..., 4\}$.

TABLE I

ACCURACY OF RECOGNITION (%) ON NTU-60 AND NTU-120 DATASETS. THE EVALUATION IS PERFORMED USING CROSS-VIEW AND CROSS-SUBJECT SETTINGS ON NTU-60 AND CROSS-SUBJECT AND CROSS-SETUP SETTINGS ON NTU-120. *THESE VALUES HAVE NOT BEEN REPORTED IN THE STATE-OF-THE-ART AND THE AVAILABLE CODES HAVE BEEN USED TO OBTAIN THE RECOGNITION ACCURACY OF THESE ALGORITHMS ON NTU-120.

Method	**NTU-60**		**NTU-120**		**Kinetics**	
	X-subject	X-view	X-subject	X-setup	Top-1	Top-5
SkeleMotion [7]	76.5	84.7	67.7	66.9	-	-
Body Pose Evolution Map [47]	**91.7**	**95.3**	64.6	66.9	-	-
Multi-Task CNN with RotClips [8]	81.1	87.4	62.2	61.8	-	-
Two-Stream Attention LSTM [48]	76.1	84.0	61.2	63.3	-	-
Skeleton Visualization (Single Stream) [9]	80.0	87.2	60.3	63.2	-	-
Multi-Task Learning Network [24]	79.6	84.8	58.4	57.9	-	-
ST-GCN (10 blocks) [12]	81.5	88.3	72.4*	71.3*	30.7	52.8
GVFE + ST-GCN w/ DH-TCN (4 blocks - ours)	79.6	88.0	72.3	71.7	29.0	50.9
AS-GCN (10 blocks) [16]	86.8	94.2	77.7*	78.9*	**34.8**	**56.5**
GVFE + AS-GCN w/ DH-TCN (4 blocks - ours)	86.4	92.9	**79.2**	**81.2**	-	-

TABLE II

ACCURACY OF RECOGNITION (%) OF OUR PROPOSED METHOD USING DIFFERENT TEMPORAL LENGTHS OF CONVOLUTIONAL FILTERS (T_w). THE EXPERIMENTS WERE CONDUCTED ON THE CROSS-VIEW SETTING OF GVFE + ST-GCN W/ DH-TCN USING THE NTU-60 DATASET.

$\mathbf{T_w}$	5	7	9	11
Accuracy	83.4	74.6	88.0	79.9

TABLE III

ACCURACY OF RECOGNITION (%) OF OUR PROPOSED METHOD USING DIFFERENT GVFE OUTPUT CHANNELS (C_{out}). THE EXPERIMENTS WERE CONDUCTED ON THE CROSS-VIEW SETTING OF GVFE + ST-GCN W/ DH-TCN USING THE NTU-60 DATASET.

$\mathbf{C_{out}}$	3	6	8	9	10
Accuracy	51.1	74.5	81.0	88.0	71.0

C. Results

1) Comparison with state-of-the-art: In this section, we compare our approach with recent skeleton-based methods, such as SkeleMotion [7], Body Pose Evolution Map [47], Multi-Task CNN with RotClips [8], Two-Stream Attention LSTM [48], Skeleton Visualization (Single Stream) [9], Multi-Task Learning Network [24] and more particularly with the two graph-based baselines namely ST-GCN [12] and AS-GCN [16]. GVFE and DH-TCN modules are incorporated in both ST-GCN [12] and AS-GCN [16] methods. The obtained accuracy of recognition on NTU-60 and NTU-120 datasets are reported in Table I.

On NTU-120, we obtain the best accuracy of recognition of the state-of-the-art for both settings. Indeed, our approach used with AS-GCN (GVFE+AS-GCN w/ DH-TCN) reaches 79.18% and 81.22% for cross-subject and cross-setup settings, respectively. These positive results are also confirmed when testing our approach with ST-GCN (GVFE + ST-GCN w/ DH-TCN). For instance, we improve the accuracy of the original ST-GCN in cross-setup setting by 0.4%.

On NTU-60, the achieved scores are among the best of the state-of-the-art but remain slightly inferior to the original ST-GCN and AS-GCN (with respectively $79.6\% - 88.0\%$against $81.5\% - 88.3\%$ and $86.4\% - 92.9\%$ against $86.8\% - 94.2\%$). Although being slightly inferior, it is important to highlight that only 4 blocks are used in our case (against 10 for ST-GCN and AS-GCN). The method based on Body Pose Evolution Map [47] remains the best performing approach on NTU-60. However, on NTU-120, this method registers an accuracy inferior to our approach by $14.6\% - 14.3\%$, while the difference is less important on NTU-60 with only a gap of $5.3\% - 2.4\%$ making our method more stable.

Moreover, on Kinetics dataset [46], ST-GCN + GVFE w/ DH-TCN achieved a Top-1 accuracy of 29.0% and a Top-5 accuracy of 50.9% by using 4 ST-GCN blocks. Although our approach shows a slight drop of 1.7% in accuracy compared the original ST-GCN approach, it is important to note that our approach is more compact since it requires 6 blocks less than the original ST-GCN.

The initial feature space of skeleton joints seems to be sufficiently discriminated when applied to NTU-60 dataset. Nevertheless, NTU-120 dataset contains a significantly larger amount of videos and action classes, making our approach a more suitable solution. This is justified by the need of a more discriminative feature space for such a large dataset that is offered by the GVFE module. A more detailed analysis concerning this follows in the ablation study.

2) Impact of the number of blocks: As mentioned earlier, our approach utilizes only 4 ST-GCN or AS-GCN blocks instead of 10. For a fair comparison with the baselines, we also test ST-GCN [12] and AS-GCN [16] when using only 4 blocks. The recognition accuracy of these experiments is reported in Table IV. Our method (GVFE + ST-GCN w/ DH-TCN) shows a significant performance boost in both settings of over 19% compared to ST-GCN with 4 blocks. Similarly, the recognition accuracy remains higher than the original AS-GCN compared to our method (GVFE + AS-GCN w/ DH-TCN). However, in this case, the accuracy boost is less impressive with an increase of 2.3% for cross-subject settings and 1.8% for cross-setup settings. This could be explained by the 7 extra spatio-temporal convolutional blocks after the *maxPooling* layer in the AS-GCN network, which add more discriminative power to the full pipeline.

TABLE IV
ACCURACY OF RECOGNITION (%) USING ONLY 4 ST-GCN OR AS-GCN BLOCKS ON NTU-120 DATASET FOR CROSS-SUBJECT AND CROSS-SETUP SETTINGS. *THESE VALUES ARE NOT REPORTED IN THE STATE-OF-THE-ART. THUS, THE AVAILABLE CODES HAVE BEEN USED TO OBTAIN THESE RESULTS.

Method	X-subject	X-setup
ST-GCN (4 blocks) [12]	45.3*	51.8*
GVFE + ST-GCN w/ DH-TCN (4 blocks - ours)	**72.3**	**71.7**
AS-GCN (4 blocks) [16]	76.9*	79.4*
GVFE + AS-GCN w/ DH-TCN (4 blocks - ours)	**79.2**	**81.2**

3) Ablation Study: To analyze the contribution of each component of our framework, an ablation study was conducted. For this purpose, we removed each time a component and report the obtained performance on both NTU-120 dataset for the cross-setup setting. The results are reported in Table V.

Our approach, which combines both the GVFE and the DH-TCN modules, achieves 71.7% mean accuracy, which is higher by 19.9% than the original ST-GCN approach with 4 ST-GCN blocks. When using only the GVFE, the mean accuracy reaches 70.9%. We tested different configurations in this case, such as attaching a Rectified Linear Unit (ReLU) or a Batch Normalization Unit (BN). In both cases, the performance was degraded (68.9% and 66.7%, respectively), since these units distort the joint motion trajectories.

Moreover, we conducted experiments by incorporating only the DH-TCN module. The mean accuracy, in this case, reached 68.3%, showing that GVFE and DH-TCN modules trained in an end-to-end manner can offer a significant performance boost.

TABLE V
ABLATION STUDY: ACCURACY OF RECOGNITION (%) ON NTU-120 DATASET FOR CROSS-SETUP SETTINGS USING ST-GCN AS A BASELINE. *THESE VALUES ARE NOT REPORTED IN THE STATE-OF-THE-ART. THUS, THE AVAILABLE CODES HAVE BEEN USED TO OBTAIN THESE RESULTS

Method	Accuracy (%)
ST-GCN (4 blocks) [12]	51.8*
GVFE + ST-GCN (4 blocks)	70.9
ST-GCN w/ DH-TCN (4 blocks)	68.3
GVFE + ST-GCN w/ DH-TCN (4 blocks - ours)	**71.7**

4) Number of parameters and training time: Although our method makes use of two additional modules compared to the baselines, the use of only 4 blocks reduces the number of parameters. For instance, When using our method (GVFE + AS-GCN w/ DH-TCN) with 4 blocks, the number of parameters drops from 7420696 to 6596002 (-12.5%) compared to the original AS-GCN with 10 blocks, while keeping almost the same accuracy on NTU-60 or even increasing it on NTU-120. Consequently, the training time is also reduced. As an example, on NTU-120 for cross-subject settings, our approach requires 8308 seconds less than the original AS-GCN for training.

VI. CONCLUSION

In this paper, two novel modules for ST-GCN based methods have been proposed called GVFE and DH-TCN. These modules enable the reduction of the number of needed blocks and parameters while conserving almost the same or improving the recognition accuracy. Instead of relying on raw skeleton features such as skeleton joints, GVFE learns and generates graph vertex features in an end-to-end manner. To model simultaneously long-term and short-term dependencies, DH-TCN makes use of hierarchical dilated temporal convolutional layers. The relevance of these modules has been confirmed thanks to the performance achieved on two well-known datasets. Some future extensions are under consideration, such as applying a similar hierarchical model to replace the spatial graph convolutional layer.

VII. ACKNOWLEDGEMENTS

This work was funded by the National Research Fund (FNR), Luxembourg, under the project C15/IS/10415355/3D-ACT/Björn Ottersten. We would, also, like to thank Christian Hundt from NVIDIA AI Technology Center Luxembourg for his valuable input and fruitful discussions.

REFERENCES

[1] L. Xia, C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *2012 CVPR Workshops*, June 2012, pp. 20–27.

[2] E. Ghorbel, J. Boonaert, R. Boutteau, S. Lecoeuche, and X. Savatier, "An extension of kernel learning methods using a modified log-euclidean distance for fast and accurate skeleton-based human action recognition," *Computer Vision and Image Understanding*, vol. 175, pp. 32–43, 2018.

[3] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *2014 CVPR*, June 2014, pp. 724–731.

[4] E. Ghorbel, K. Papadopoulos, R. Baptista, H. Pathak, G. Demisse, D. Aouada, and B. Ottersten, "A view-invariant framework for fast skeleton-based action recognition using a single rgb camera," in *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Prague, 25-27 February 2018*, 2019.

[5] K. Papadopoulos, E. Ghorbel, R. Baptista, D. Aouada, and B. Ottersten, "Two-stage rgb-based action detection using augmented 3d poses," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2019, pp. 26–35.

[6] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang, "Graph based skeleton motion representation and similarity measurement for action recognition," in *ECCV 2016*. Springer International Publishing, 2016, pp. 370–385.

[7] C. Caetano, J. Sena, F. Brémond, J. A. d. Santos, and W. R. Schwartz, "Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition," *arXiv preprint arXiv:1907.13025*, 2019.

[8] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3d action recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2842–2855, 2018.

[9] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[10] G. G. Demisse, K. Papadopoulos, D. Aouada, and B. Ottersten, "Pose encoding for robust skeleton-based action recognition," in *CVPR Workshops*, 2018, pp. 188–194.

[11] R. Baptista, E. Ghorbel, K. Papadopoulos, G. G. Demisse, D. Aouada, and B. Ottersten, "View-invariant action recognition from rgb data via 3d pose estimation," in *ICASSP 2019*. IEEE, 2019, pp. 2542–2546.

[12] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, 2018.

[13] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[14] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019, pp. 12 026–12 035.

[15] ——, "Skeleton-based action recognition with directed graph neural networks," in *CVPR*, 2019, pp. 7912–7921.

[16] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019, pp. 3595–3603.

[17] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," in *ICCV*, 2013, pp. 2752–2759.

[18] E. Ghorbel, R. Boutteau, J. Boonaert, X. Savatier, and S. Lecoeuche, "Kinematic spline curves: A temporal invariant descriptor for fast action recognition," *Image and Vision Computing*, vol. 77, pp. 60–71, 2018.

[19] J. C. Meza and M. Woods, "A numerical comparison of rule ensemble methods and support vector machines," Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), Tech. Rep., 2009.

[20] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, "3-d human action recognition by shape analysis of motion trajectories on riemannian manifold," *IEEE transactions on cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2014.

[21] A. Ben Tanfous, H. Drira, and B. Ben Amor, "Coding Kendall's Shape Trajectories for 3D Action Recognition," in *CVPR 2018*, Salt Lake City, United States, Jun. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01713295

[22] 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*, June 2012, pp. 1290–1297.

[23] X. Yang and Y. L. Tian, "Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2012, pp. 14–19.

[24] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *CVPR*, 2017, pp. 3288–3297.

[25] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on lie groups for skeleton-based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6099–6108.

[26] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 1012–1020.

[27] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *2017 ICCV*, Oct 2017, pp. 2136–2145.

[28] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 786–792.

[29] ——, "Skeleton-based action recognition with convolutional neural networks," in *2017 ICME Workshops (ICMEW)*. IEEE, 2017, pp. 597–600.

[30] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2015, pp. 579–583.

[31] D. Xie, C. Deng, H. Wang, C. Li, and D. Tao, "Semantic adversarial network with multi-scale pyramid attention for video classification," in *AAAI*, vol. 33, 2019, pp. 9030–9037.

[32] D. Liang, G. Fan, G. Lin, W. Chen, X. Pan, and H. Zhu, "Three-stream convolutional neural network with multi-task and ensemble learning for 3d action recognition," in *CVPR Workshops*, 2019, pp. 0–0.

[33] G. Hu, B. Cui, and S. Yu, "Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention," in *2019 ICME*. IEEE, 2019, pp. 1216–1221.

[34] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[35] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[36] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[37] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[38] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition," in *2018 CVPR*, June 2018, pp. 5323–5332.

[39] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *CVPR*, 2019, pp. 1227–1236.

[40] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *CVPR*, 2019.

[41] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[42] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[43] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. v. d. Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *arXiv preprint arXiv:1610.10099*, 2016.

[44] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *CVPR*, June 2016.

[45] 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*, 2019.

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

[47] M. Liu and J. Yuan, "Recognizing human actions as the evolution of pose estimation maps," in *CVPR*, 2018, pp. 1159–1168.

[48] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention lstm networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2017.

[49] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017, pp. 7291–7299.