Skeleton-based Hand-Gesture Recognition with Lightweight Graph Convolutional Networks

Hichem Sahbi
CNRS Sorbonne University

Abstract

Graph convolutional networks (GCNs) aim at extending deep learning to arbitrary irregular domains, namely graphs. Their success is highly dependent on how the topology of input graphs is defined and most of the existing GCN architectures rely on predefined or handcrafted graph structures.

In this paper, we introduce a novel method that learns the topology (or connectivity) of input graphs as a part of GCN design. The main contribution of our method resides in building an orthogonal connectivity basis that optimally aggregates nodes, through their neighborhood, prior to achieve convolution. Our method also considers a stochasticity criterion which acts as a regularizer that makes the learned basis and the underlying GCNs lightweight while still being highly effective. Experiments conducted on the challenging task of skeleton-based hand-gesture recognition show the high effectiveness of the learned GCNs w.r.t. the related work.

Keywords. Graph convolutional networks, lightweight connectivity design, skeleton-based hand-gesture recognition.

1 Introduction

Deep learning is currently witnessing a major interest in different fields including image processing and pattern recognition [1]. Its principle consists in learning multi-layered convolutional, pooling and fully connected operations that extract representations which capture low, mid and high-level characteristics of patterns while maximizing their classification performances. Most of the existing deep learning architectures [2]–[10], [12]–[16], [18], [19], [21], [54], [57] are targeted to vectorial data; i.e., data sitting on top of regular domains including images. However, other data require extending deep learning to irregular domains (namely graphs [12], [20], [22], [24], [25], [34], [72], [76], [88]) such as skeletons in action recognition. While convolutional operations on regular domains are well defined, their extension to irregular ones (i.e., graphs) is generally ill-posed and remains a major challenge.

Two different categories of GCNs exist in the literature, spatial and spectral [13], [35], [37]–[39], [41]. Spatial methods achieve node aggregations prior to apply convolutions using inner products while spectral techniques rely on the well defined graph Fourier transform [20], [22], [24], [25], [27]–[29], [31], [33]. Whereas spatial methods are known to be effective compared to spectral ones, their success is highly dependent on the topology of input graphs, and most of the existing solutions rely on handcrafted or predefined graph structures using similarities or the inherent properties of the targeted applications [30], [36], [42]–[46], [61], [65] (e.g., node relationships in social networks, edges in 3D modeling, etc). These structures are usually powerless to capture the most prominent relationships between nodes as their design is agnostic to the targeted application. For instance, when considering node relationships in skeletons, these links capture the anthropometric characteristics of individuals which are useful for their identification, while
other connections, yet to infer, are important for recognizing their actions. Hence, in spite of being relatively effective, the potential of these GCN methods is not fully explored as the setting of their graphs is either oblivious to the tasks at hand or achieved using the tedious cross validation.

Graph inference is generally ill-posed, NP-hard [47]–[49] and most of the existing approaches rely on constraints (similarity, smoothness, sparsity, band-limitedness, etc. [51]–[53], [55], [56], [58]–[60], [62], [63], [67]) for its conditioning [52], [56], [62]–[64], [66], [67]. Particularly in GCNs, recent advances aim at defining graph topology that best fits a given task [70], [71], [73]–[75], [77], [78], [80]. For instance, [75] proposes a graph network for semi-supervised classification that learns graph topology with sparse structure given a cloud of points; node-to-node connections are modeled with a joint probability distribution on Bernoulli random variables whose parameters are found using bi-level optimization. A computationally more efficient variant is introduced in [77] using a weighted cosine similarity and edge thresholding. Other solutions make improvement w.r.t. the original GCNs [25] by exploiting symmetric matrices [78] and discovering hidden structural relations (unspecified in the original graphs), using a so-called residual graph adjacency matrix and by learning a distance function over nodes. The work in [80] introduces a dual architecture with two parallel graph convolutional layers sharing the same parameters, and considers a normalized adjacency and a positive point-wise mutual information matrix to capture node co-occurrences through random walks sampled from graphs.

In this paper, we introduce a novel framework that designs graphs as a part of end-to-end GCN learning. Our design principle is based on the minimization of a constrained loss whose solution corresponds not only to the convolutional parameters of GCNs but also the underlying adjacency matrices that capture the topology of input graphs. Our contribution in this paper differs from the aforementioned related work in multiple aspects; on the one hand, in contrast to many existing methods – e.g., [81] which consider a single adjacency matrix shared through power series – the matrix operators designed in our contribution are non-parametrically learned and this provides more flexibility to our design. On the other hand, constraining these matrices, through orthogonality and stochasticity, allows achieving structured regularization that mitigates overfitting and allows learning lightweight GCN architectures\(^1\); in contrast to non structured lightweight network design (e.g., magnitude pruning), our proposed method (i) captures (through orthogonality and stochasticity) the structural relationships between parameters in the learned GCNs, and (ii) maintains completeness and minimality of the learned representations by finding the most discriminating and lightweight GCNs as also supported in our experiments.

2 LEARNING LIGHTWEIGHT CONNECTIVITY

Let \( S = \{G_i = (V_i, E_i)\}_i \) denote a collection of graphs with \( V_i, E_i \) being respectively the nodes and the edges of \( G_i \). Each graph \( G_i \) (denoted for short as \( G = (V, E) \)) is endowed with a signal \( \{\psi(u) \in \mathbb{R}^s : u \in V\} \) and associated with an adjacency matrix \( A \) with each entry \( A_{uu'} > 0 \) iff \( (u, u') \in E \) and 0 otherwise. GCNs aim at learning a set of filters \( \mathcal{F} = \{g_\theta = (V, E)\}_\theta \) that define convolution on \( n \) nodes of \( G \) (with \( n = |V| \)) as
\[
(G \ast \mathcal{F})_v = f(A \ U^\top W),
\]
here \( ^\top \) stands for transpose, \( U \in \mathbb{R}^{s \times n} \) is the graph signal, \( W \in \mathbb{R}^{s \times C} \) is the matrix of convolutional parameters corresponding to the \( C \) filters and \( f(.) \) is a nonlinear activation applied entrywise. In Eq. 1, the input signal \( U \) is projected using \( A \) and this provides for each node \( u \), the aggregate set

\(^1\) Adjacency matrices learned, in the related work, are usually fully dense and this introduces a lot of latency in the underlying GCNs, especially when handling large scale databases (even with reasonable size graphs).
Eq. (1) implements a convolutional block with two layers; the first one aggregates signals in \( U \) (sets of node neighbors) by multiplying \( U \) with \( A \) while the second layer achieves convolution by multiplying the resulting aggregates with the \( C \) filters in \( W \).

2.1 Orthogonality-driven connectivity

Learning multiple adjacency matrices (denoted as \( \{A_k\}_{k=1}^K \)) allows us to capture different contexts and graph topologies when achieving aggregation and convolution. With multiple matrices \( \{A_k\}_k \) (and associated convolutional filter parameters \( \{W_k\}_k \)), Eq. 1 is updated as

\[
(G \ast F)_{uv} = f \left( \sum_{k=1}^K A_k U^T W_k \right).
\]

If aggregation produces, for a given \( u \in \mathcal{V} \), linearly dependent vectors \( \mathcal{X}_u = \{\sum_{u'} A_{k,u'') \psi(u')\}_{k=1}^K \) then convolution will also generate linearly dependent representations with an overestimated number of training parameters in the null space of \( \mathcal{X}_u \). Besides, the tensor \( \{A_k\}_k \) used for aggregation, may also generate overlapping and redundant contexts.

Provided that \( \{\psi(u')\}_{u' \in \mathcal{N}_r(u)} \) are linearly independent, the sufficient condition that makes vectors in \( \mathcal{X}_u \) linearly independent reduces to constraining \( (A_{k,u''})_{k,u''} \) to lie on the Stiefel manifold (see for instance [82], [84], [85]) defined as \( V_K(\mathbb{R}^n) = \{M \in \mathbb{R}^{K \times n} : MM^T = I_K\} \) (with \( I_K \) being the \( K \times K \) identity matrix) which thereby guarantees orthonormality and minimality of \( \{A_1, \ldots, A_K\}^3 \). A less compelling condition is orthogonality, i.e., \( \langle A_k, A_{k'} \rangle_F = 0 \) and \( A_k \geq 0 \) for all matrices \( A_k \). Note that orthogonality (as designed subsequently) allows learning sparse adjacency matrices while column-stochasticity provides extra sparsity and acts as a structured regularizer that enhances further the generalization power of the learned GCNs\(^4\).

Considering orthogonality (as discussed above), the tensor \( \{A_k\}_k \) and \( W = \{W_k\}_k \) are learned as

\[
\min_{\{A_k \geq 0\}_k, W} \quad E(A_1, \ldots, A_K; W)
\]

s.t.

\[
\begin{align*}
A_k & \odot A_k > 0_{n \times n} \\
A_k & \odot A_{k'} = 0_{n \times n} \quad \forall k, k' \neq k \\
1^n \top A_k & = 1^n \end{align*}
\]

being \( E \) the cross entropy loss and \( 1^n \) a vector of \( n \) ones. In the above minimization problem, the first and the second constraints correspond to orthogonality while the third one to column-stochasticity. The latter is added in order to ensure that all of the entries in \( A_k \) are positive and each column sums to one; i.e., each matrix \( A_k \) models a Markov chain whose \( i \)-th row and \( j \)-th column provides the probability of transition from one node \( u_j \) to \( u_i \) in \( \mathcal{G} \). Note that orthogonality (as designed subsequently) allows learning sparse adjacency matrices while column-stochasticity provides extra sparsity and acts as a structured regularizer that enhances further the generalization power of the learned GCNs\(^4\).

\(^2\) e.g., when considering a common graph structure for all actions in videos.

\(^3\) Note that \( K \) should not exceed the rank of \( \{\psi(u')\}_{u' \in \mathcal{N}_r(u)} \), which is upper bounded by \( \min(|\mathcal{V}|, s) \); \( s \) is again the dimension of the graph signal.

\(^4\) Without stochasticity, one has to consider a normalization layer (with extra parameters), especially on graphs with heterogeneous degrees in order to reduce the covariate shift and distribute the transition probability evenly through nodes before achieving convolutions.
2.2 Optimization

A natural approach to solve Eq. (3) is to iteratively and alternately minimize over one matrix while keeping all the others fixed. However — and besides the non-convexity of the loss — the feasible set formed by these $O(K^2)$ bi-linear constraints is not convex w.r.t $\{A_k\}_k$. Moreover, this iterative procedure is computationally expensive as it requires solving multiple instances of constrained projected gradient descent and the number of necessary iterations to reach convergence is large in practice. All these issues make solving this problem challenging and computationally intractable even for reasonable values of $K$ and $n$. In what follows, we investigate a workaround that optimizes these matrices while guaranteeing their orthogonality and stochasticity as a part of optimization.

**Orthogonality.** Let \( \exp(\gamma \hat{A}_k) \odot (\sum_{r=1}^{K} \exp(\gamma \hat{A}_r)) \) be a softmax reparametrization of \( A_k \), with \( \odot \) being the entrywise hadamard division and \( \{A_k\}_k \) free parameters in \( \mathbb{R}^{n \times n} \), it becomes possible to implement orthogonality by choosing large values of \( \gamma \) to make this softmax crisp; i.e., only one entry \( A_{kij} \gg 0 \) while all others \( \{A_{k'j}\}_{k' \neq k} \) vanishing thereby leading to \( A_k \odot A_{k'} = 0_{n \times n}, \quad \forall k, k' \neq k \). By plugging this crispmax reparametrization into Eq. 3, the gradient of the loss \( E \) (now w.r.t \( \{A_k\}_k \)) is updated using the chain rule as

$$
\frac{\partial E}{\partial \text{vec}(\{A_k\}_k)} = J_{\text{orth}} \frac{\partial E}{\partial \text{vec}(\{A_k\}_k)},
$$

being \( \text{vec}(\{A_k\}_k) \) a vectorization of \( \{A_k\}_k \) and \((i,j) = (kij,k'j')\) an entry of the Jacobian \( J_{\text{orth}} \) as

$$
J_{\text{orth}} = \begin{cases} 
\gamma A_{kij} (1 - A_{k'j}) & \text{if } k = k', i = i', j = j' \\
-\gamma A_{k'j} A_{k'ij} & \text{if } k \neq k', i = i', j = j' \\
0 & \text{otherwise},
\end{cases}
$$

here \( \frac{\partial E}{\partial \text{vec}(\{A_k\}_k)} \) is obtained from layerwise gradient backpropagation. However, with this reparametrization, large values of \( \gamma \) may lead to numerical instability when evaluating the exponential. We circumvent this by choosing \( \gamma \) that satisfies \( \epsilon \)-orthogonality: a surrogate property defined subsequently.

**Definition 1 (\( \epsilon \)-orthogonality).** A basis \( \{A_k\}_k \) is \( \epsilon \)-orthogonal if \( A_k \odot A_{k'} \leq \epsilon \mathbb{1}_{n \times n}, \quad \forall k, k' \neq k \), with \( \mathbb{1}_{n \times n} \) being the \( n \times n \) unitary matrix.

Considering the above definition, (nonzero) matrices belonging to an \( \epsilon \)-orthogonal basis are linearly independent w.r.t \((\cdot,\cdot)_F\) (provided that \( \gamma \) is sufficiently large) and hence this basis is also minimal. The following proposition provides a tight lower bound on \( \gamma \) that satisfies \( \epsilon \)-orthogonality.

**Proposition 1 (\( \epsilon \)-orthogonality bound).** Consider \( \{A_{kij}\}_{ij} \) as the entries of the crispmax reparametrized matrix \( A_k \) defined as \( \exp(\gamma \hat{A}_k) \odot (\sum_{r=1}^{K} \exp(\gamma \hat{A}_r)) \). Provided that \( \exists \delta > 0 : \forall i, j, \ell', \exists \ell, \hat{A}_{kij} \geq \hat{A}_{\ell'ij} + \delta \) (with \( \ell' \neq \ell \)) and if \( \gamma \) is at least

$$
\frac{1}{\delta} \ln \left( \frac{K \sqrt{(1 - 2\epsilon)}}{1 - \sqrt{(1 - 2\epsilon)} + 1} \right)
$$

then \( \{A_1, \ldots, A_K\} \) is \( \epsilon \)-orthogonal.
Proof 1. For any entry $i, j$, one may find $\ell, \ell'$ in $\{1, \ldots, K\}$ (with $\ell \neq \ell'$) s.t. $(A_k \odot A_{k'})_{ij}$

$$
\leq (A_{\ell} \odot A_{\ell'})_{ij}
$$

$$
= \frac{1}{2}(A_{\ell}^2_{ij} + A_{\ell'}^2_{ij}) - \frac{1}{2}(A_{\ell} - A_{\ell'})^2
$$

$$
\leq \frac{1}{2} - \frac{1}{2}(A_{\ell} - A_{\ell'})^2
$$

$$
= \frac{1}{2} - \frac{1}{2}\left(\frac{\exp(\gamma \hat{A}_{\ell}) - \exp(\gamma \hat{A}_{\ell'})}{\exp(\gamma \hat{A}_{\ell}) + \exp(\gamma \hat{A}_{\ell'}) + \sum_{r=3}^{K} \exp(\gamma \hat{A}_{r})}\right)^2
$$

$$
\leq \frac{1}{2} - \frac{1}{2}\left(\frac{\exp(\gamma \hat{A}_{\ell}) - \exp(\gamma \hat{A}_{\ell'})}{\exp(\gamma \hat{A}_{\ell}) + (K - 1) \exp(\gamma \hat{A}_{r})}\right)^2
$$

The sufficient condition is to choose $\gamma$ such as

$$
\frac{1}{2} - \frac{1}{2}\left[\frac{1}{1 + \frac{K}{\exp(\gamma \delta) - 1}}\right]^2 \leq \epsilon \implies \gamma \geq \frac{1}{\delta} \ln \left(\frac{K \sqrt{(1 - 2\epsilon)}}{1 - \sqrt{(1 - 2\epsilon)}} + 1\right).
$$

Following the above proposition, setting $\gamma$ to the above lower bound guarantees $\epsilon$-orthogonality; for instance, when $K = 2$, $\delta = 0.01$ and provided that $\gamma \geq 530$, one may obtain $0.01$-orthogonality which is almost a strict orthogonality. This property is satisfied as long as one slightly disrupts the entries of $\{\hat{A}_k\}_k$ with random noise during training. However, this may still lead to another limitation; precisely, bad local minima are observed due to an early convergence to crisp adjacency matrices. We prevent this by steadily annealing the temperature $1/\gamma$ of the softmax through training epochs (using $\gamma_{\text{epoch max epochs}}$ instead of $\gamma$) in order to make optimization focusing first on the loss, and then as optimization evolves, temperature cools down and allows reaching the aforementioned lower bound (thereby crispmax) and $\epsilon$-orthogonality at convergence.

Lightweight connectivity with stochasticity. Unless explicitly mentioned, $A_k$ is simply rewritten as $A$. We consider a reparametrization $A = h(\hat{A})D(h(\hat{A}^T))^{-1}$, with $D(.)$ being the degree matrix operator, $h$ a strictly monotonic positive function and this allows a free setting of the matrix $\hat{A}$ during optimization while guaranteeing stochasticity. In practice, $h$ is set to $\exp$ and the original gradient is obtained, similarly to Eq. 4, from layerwise gradient back propagation by multiplying the original gradient by the Jacobian $[J_{\text{stc}}]_{ij, i'j'} = [A_{ij} \odot A_{i'j'}] + (\delta_{ii'} - A_{ij})$ with $\delta_{ii'} = 1_{i = i'}$. Note that stochasticity, when combined with orthogonality, lightens connectivity by a factor $n$ compared to orthogonality whose factor does not exceed $K$; this combination is obtained by multiplying the underlying Jacobians, so the final gradient becomes

$$
\frac{\partial E}{\partial \text{vec}(\{\hat{A}_k\}_k)} = J_{\text{stc}} \cdot J_{\text{orth}} \cdot \frac{\partial E}{\partial \text{vec}(\{A_k\}_k)},
$$

and this order of application is strict, as orthogonality sustains after stochasticity while the converse is not necessarily guaranteed at the end of the optimization process.

5. whatever the range of entries in these matrices $\{\hat{A}_k\}_k$. 

\[\text{Proof 1.}\] For any entry $i, j$, one may find $\ell, \ell'$ in $\{1, \ldots, K\}$ (with $\ell \neq \ell'$) s.t. $(A_k \odot A_{k'})_{ij}$
3 Experiments

Database and settings. We evaluate the performance of our GCN on the task of action recognition using the First-Person Hand Action (FPHA) dataset [89]. The latter includes 1175 skeletons belonging to 45 action categories which are performed by 6 different individuals in 3 scenarios. Action categories are highly variable with inter and intra subject variability including style, speed, scale and viewpoint. Each video (sequence of skeletons) is initially described with a handcrafted graph $G = (V, E)$ where each node $v_j \in V$ corresponds to the $j$-th hand-joint trajectory (denoted as $\{\hat{p}_t^j\}$) and an edge $(v_j, v_i) \in E$ exists if the $j$-th and the $i$-th trajectories are spatially connected. Each trajectory in $G$ is processed using temporal chunking: first, the total duration of a sequence is split into $M$ equally-sized temporal chunks ($M = 4$ in practice), then the trajectory coordinates $\{\hat{p}_t^j\}$ are assigned to the $M$ chunks (depending on their time stamps) prior to concatenate the averages of these chunks; this produces the raw description of $v_j$, again denoted as $\psi(v_j)$.

Implementation details. We trained the GCNs end-to-end using the Adam optimizer for 2,800 epochs with a batch size equal to 600, a momentum of 0.9 and a global learning rate (denoted as $\nu(t)$) inversely proportional to the speed of change of the cross entropy loss used to train our networks; when this speed increases (resp. decreases), $\nu(t)$ decreases as $\nu(t) \leftarrow \nu(t-1) \times 0.99$ (resp. increases as $\nu(t) \leftarrow \nu(t-1)/0.99$). In all these experiments, we use a GeForce GTX 1070 GPU device (with 8 GB memory), we evaluate the performances using the 1:1 setting proposed in [89] with 600 action sequences for training and 575 for testing, and we report the average accuracy over all the classes of actions.

Performances and comparison. We compare the performances of our GCN design against two baselines: handcrafted and learned. In the first baseline (known as power map), all the matrices $\{A_k\}_k$ are evaluated upon the adjacency matrix $A$ (taken from the input skeletons) as $A_k = A^{(k)}$ with $A^{(k)} = A^{(k-1)}A$, $A^{(0)} = I$ and this defines nested supports for convolutions while in the second baseline, all the adjacency matrices $\{A_k\}_k$ are learned using the objective function (3) but w/o orthogonality and stochasticity constraints. Table 1 shows a comparison with these baselines and an ablation study of our complete model and the impact of orthogonality (separately and combined) on the performances. These results show that orthogonality has a clear and a consistent positive impact on the performances while stochasticity (when combined with orthogonality) provides lightweight GCNs with an extra gain in accuracy. Clearly, these two constraints act as regularizers that also reduce the number of training parameters thereby leading to highly effective and also efficient GCNs. In order to further investigate the impact of these two constraints, we compare the underlying GCNs against lightweight ones obtained differently, with magnitude pruning; the latter consists first in zeroing the smallest parameters in the learned GCNs, and then fine-tuning the remaining parameters. As shown in table 1, lightweight GCNs, trained with orthogonality and stochasticity, clearly outperform those obtained with magnitude pruning+fine-tuning. Finally, we compare the classification performances of our GCN against other related methods in action recognition ranging from sequence based such as LSTM to deep graph (non-vectorial) methods, etc. (see table 2 and references within). From the results in these tables, our GCN brings a noticeable gain w.r.t. related state of the art methods.

4 Conclusion

We introduce in this paper a novel framework that designs graph topology as a part of an “end-to-end” GCN learning. This topology is captured using multiple adjacency matrices whose optimization is constrained with orthogonality and stochasticity. The former makes it possible to remove the redundancy while the latter allows learning lightweight and highly effective GCNs. These two constraints also act as regularizers that model structural relationships between
TABLE 1: Detailed performances, for different $K$, using handcrafted and learned connectivity w/o and with our constraints. We also compare these results with those of GCNs obtained using magnitude pruning (for the same pruning rates: $\lfloor (1 - \frac{1}{K}) \times 100 \rfloor$ for $L$-orth vs. $L$+MP and $\lfloor (1 - \frac{1}{n}) \times 100 \rfloor$ for $L$-orth+stc vs. $L$+MP), here $H$, $L$, orth, stc and MP stands respectively for handcrafted, learned, orthogonality, stochasticity and magnitude pruning.

| $K$ | Accuracy (%) | Pruning rate (%) | $L$ | $L$-orth | $L$-MP | $L$-orth+stc | $L$+MP |
|-----|--------------|------------------|-----|-----------|---------|-------------|--------|
| 3   | 84.17        | none             | 83.30| 84.52     | 84.52   | 83.65       | 81.56  |
| 4   | 82.95        | none             | 83.82| 85.21     | 83.13   | 85.73       | 82.95  |
| 8   | 72.69        | none             | 85.04| 83.65     | 86.78   | 84.00       |        |

TABLE 2: Comparison against state of the art methods.

| Method                  | Color | Depth | Pose | Accuracy (%) |
|------------------------|-------|-------|------|--------------|
| Two stream-color [90]  | ✓     | ✗     | ✗    | 61.56        |
| Two stream-flow [90]   | ✓     | ✗     | ✗    | 69.91        |
| Two stream-all [90]    | ✓     | ✗     | ✗    | 75.30        |
| HOG2-depth [92]        | ✗     | ✓     | ✗    | 59.83        |
| HOG2-depth+pose [92]   | ✗     | ✓     | ✓    | 66.78        |
| HON4D [94]             | ✗     | ✓     | ✗    | 70.61        |
| Novel View [95]        | ✗     | ✓     | ✗    | 69.21        |
| 1-layer LSTM [97]      | ✗     | ✗     | ✓    | 78.73        |
| 2-layer LSTM [97]      | ✗     | ✗     | ✓    | 80.14        |
| Moving Pose [98]       | ✗     | ✗     | ✓    | 56.34        |
| Lie Group [99]         | ✗     | ✗     | ✓    | 82.69        |
| HBRNN [101]            | ✗     | ✗     | ✓    | 77.40        |
| Gram Matrix [103]      | ✗     | ✗     | ✓    | 85.39        |
| TF [105]               | ✗     | ✗     | ✓    | 80.69        |
| JOULE-color [107]      | ✓     | ✗     | ✗    | 66.78        |
| JOULE-depth [107]      | ✓     | ✓     | ✗    | 60.17        |
| JOULE-pose [107]       | ✓     | ✗     | ✓    | 74.60        |
| JOULE-all [107]        | ✓     | ✓     | ✓    | 78.78        |
| Huang et al. [108]     | ✗     | ✗     | ✓    | 84.35        |
| Huang et al. [110]     | ✗     | ✗     | ✓    | 77.57        |
| Our best (table 1)     | ✗     | ✗     | ✓    | 86.78        |

network parameters in order to enhance both their generalization and lightweightness. Experiments conducted on the challenging task of skeleton-based hand-gesture recognition, shows the outperformance of the proposed lightweight GCNs against different baselines as well as the related work.
[84] L. Huang, X. Liu, B. Lang, A. W. Yu, Y. Wang, and B. Li. Orthogonal weight normalization: Solution to optimization over multiple dependent Stiefel manifolds in deep neural networks. In AAAI, 2017.

[85] A. Shukla, S. Bhagat, S. Uppal, S. Anand, P. Turaga. PrOSe: Product of Orthogonal Spheres Parameterization for Disentangled Representation Learning. In BMVC, 2019.

[86] S. Thiernert, H. Sahbi, and M. Steinebach, “Applying interest operators in semi-fragile video watermarking,” in Security, Steganography, and Watermarking of Multimedia Contents VII, vol. 5681. International Society for Optics and Photonics, 2005, pp. 353–363.

[87] D.P. Kingma, and J. Ba. “Adam: A method for stochastic optimization.” arXiv preprint arXiv:1412.6980 (2014)

[88] H. Sahbi, J.-Y. Audibert, and R. Keriven, “Context-dependent kernels for object classification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, pp. 699–708, 2011.

[89] S. Thiemert, H. Sahbi, and M. Steinebach, “Applying interest operators in semi-fragile video watermarking,” in Security, Steganography, and Watermarking of Multimedia Contents VII, vol. 5681. International Society for Optics and Photonics, 2005, pp. 353–363.

[90] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In CVPR, 2018.

[91] C. Feichtenhofer, A. P., and A. Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. In CVPR, 2016.

[92] Q. Oliveau, H. Sahbi. Learning attribute representations for remote sensing ship category classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2017.

[93] E.Ohn-Barand, M.M. Trivedi. Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations. IEEE TITS, 15(6), pages 2368–2377, 2014.

[94] N. Bourdis, D. Marraud, H. Sahbi. “Constrained optical flow for aerial image change detection.” 2011 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2011.

[95] G. Garcia-Hernando and T.-K. Kim. Transition Forests: Learning Discriminative Temporal Transitions for Action Recognition. In CVPR, 2017.

[96] H. Sahbi. “A particular Gaussian mixture model for clustering and its application to image retrieval.” Soft Computing 12.7 (2008): 667-676.

[97] N. Bourdis, D. Marraud, H. Sahbi. “A particular Gaussian mixture model for clustering and its application to image retrieval.” Soft Computing 12.7 (2008): 667-676.

[98] J. Hu et al., Jointly Learning Heterogeneous Features for RGB-D Activity Recognition. In CVPR, 2015.