Coarse-to-Fine Pruning of Graph Convolutional Networks for Skeleton-based Recognition

Hichem Sahbi

Coarse-to-Fine Pruning of Graph Convolutional Networks for Skeleton-based Recognition

Hichem Sahbi
Sorbonne University, CNRS, LIP6, F-75005, Paris, France
hichem.sahbi@sorbonne-universite.fr

***Abstract*—Magnitude Pruning is a staple lightweight network design method which seeks to remove connections with the smallest magnitude. This process is either achieved in a structured or unstructured manner. While structured pruning allows reaching high efficiency, unstructured one is more flexible and leads to better accuracy, but this is achieved at the expense of low computational performance.**

In this paper, we devise a novel coarse-to-fine (CTF) method that gathers the advantages of structured and unstructured pruning while discarding their inconveniences to some extent. Our method relies on a novel CTF parametrization that models the mask of each connection as the Hadamard product involving four parametrizations which capture channel-wise, column-wise, row-wise and entry-wise pruning respectively. Hence, fine-grained pruning is enabled only when the coarse-grained one is disabled, and this leads to highly efficient networks while being effective. Extensive experiments conducted on the challenging task of skeleton-based recognition, using the standard SBU and FPHA datasets, show the clear advantage of our CTF approach against different baselines as well as the related work.

***Index Terms*—Coarse and fine-grained pruning, graph convolutional networks, skeleton-based recognition.**

I. Introduction

Deep learning (DL) is a rapidly growing subfield of artificial intelligence (AI) which has made a significant advancement [1] in various pattern recognition tasks including action and hand-gesture classification [47]. Major actors in the realm of AI are nowadays deploying DL techniques (and particularly neural networks) to solve problems and gain a competitive edge. However, DL's progress has been achieved at the expense of a significant increase of time and memory demand, making it overpowering to deploy on cheap devices endowed with limited hardware resources. In the field of skeleton-based recognition, graph convolutional networks (GCNs) are peculiar neural networks that operate on non-euclidean domains (such as skeleton graphs) by learning relationships between nodes and edges. Two categories of GCNs exist in the literature: spectral [2], [3], [22], [37] and spatial [4], [6], [10], [52]. The former relies on the Fourier transform while the latter leverages message passing and multi-head attention layers. These layers extract node representations by aggregating features over their most salient neighboring nodes, prior to applying convolutions (as inner products) on the resulting node aggregates. With multi-head attention, spatial GCNs are deemed highly accurate on skeleton data, but oversized and computationally overwhelming, and their deployment on cheap devices requires designing their lightweight counterparts.

Existing work that addresses the issue of lightweight neural network design includes tensor decomposition [17], quantization [21], distillation [7], [40], [66], neural architecture search [69] and pruning [16], [23]–[25], [55]. Pruning techniques are particularly effective and consist in removing unnecessary connections leading to more compact and faster networks with a minimal decay on accuracy. One of the mainstream lightweight design methods is magnitude pruning (MP) [21]. The latter aims at ranking network connections according to the magnitude, *as a proxy to the importance*, of their weights, prior to remove the smallest magnitude connections, and this eventually leads to a minimal impact on performances. Two categories of MP techniques are widely known: structured [14], [41] and unstructured [21], [23], [27]. Structured MP consists in removing entire groups of weights, filters or neurons which significantly changes the model architecture and leads to higher compression rates and efficient computation on standard deep learning frameworks/hardware. However, structured MP suffers from a *coarse* pruning granularity as it cannot target individual (possibly important) weights within a group, and this may potentially result into a significant drop in accuracy particularly when aggressive pruning is achieved. On another hand, unstructured MP offers a *fine* control over granularity as it identifies and removes connections individually, and maintains the overall network architecture. Hence, it may potentially preserve important connections and ultimately achieve higher accuracy compared to structured MP. However, unstructured MP suffers from several downsides including lower compression rate compared to structured MP, and slower inference with *spread weight distributions* which may be inefficient for acceleration with most of the existing standard hardware.

In order to gather the upsides of both structured and unstructured pruning techniques, while mitigating their downsides, we devise in this paper a new pruning method for lightweight GCNs. The design principle of our approach is *coarse-to-fine* (CTF) and achieved using a novel multi-structured tensor parametrization; as we traverse this parametrization, pruning is getting relatively less structured and computationally less efficient, but more resolute (finer), allowing to reach the targeted pruning rate with a high accuracy. Given an unpruned network, we define our parametrization as the combination of three functions: (i) a band-stop parametrization which keeps only connections with the highest magnitudes, (ii) a weight-sharing parametrization that groups connections either channel-wise,

column-wise, row-wise or keeps them as singletons, and (iii) a gating mechanism which either keeps weights as singletons, or removes them in a structured manner. Besides being able to handle coarse as well as fine-grained pruning, our composed parametrization allow reaching a tradeoff between efficient computation and high accuracy as corroborated through extensive experiments conducted on the challenging task of skeleton-based action and hand-gesture recognition.

II. Related work

The following review discusses the related work in pruning and skeleton-based recognition, highlighting the limitations that motivate our contributions.

Variational Pruning. The latter seeks to learn weights and binary masks in order to capture the topology of pruned networks. This is obtained by minimizing a global loss which mixes a classification error and a regularizer that controls the cost of the pruned networks [13]–[15]. However, existing methods are powerless to implement targeted pruning rates without overtrying multiple weighting of the regularizers. Alternative methods explicitly use ℓ_0-based criteria to minimize the discrepancy between observed and targeted costs [15], [53]. Existing solutions rely on sampling heuristics or relaxation, which promote sparsity (via different regularizers including ℓ_1/ℓ_2-based, entropy, etc.) [8], [9], [11], [12] but are powerless to implement target costs exactly, and may lead to overpruned and thereby disconnected networks. Besides, most of the existing solutions including magnitude pruning are either structured [14], [41] or unstructured [21], [23], [46], and their benefit is not fully explored. This paper aims to gather the advantages of both structured and unstructured pruning while discarding their limitations.

Skeleton-based recognition. This task has gained increasing interest due to the emergence of sensors like Intel RealSense and Microsoft Kinect. Early methods for hand-gesture and action recognition used RGB [26], depth [51], shape/normals [5], [60]–[65], and skeleton-based techniques [49]. These methods were based on modeling human motions using handcrafted features [59], dynamic time warping [56], temporal information [19], [67], and temporal pyramids [47]. However, with the resurgence of deep learning, these methods have been quickly overtaken by 2D/3D Convolutional Neural Networks (CNNs) [18], [68], Recurrent Neural Networks (RNNs) [42]–[45], [47], [48], manifold learning [32]–[35], attention-based networks [54], [58], and GCNs [28]–[31]. The recent emergence of GCNs, in particular, has led to their increased use in skeleton-based recognition [3]. These models capture spatial and temporal attention among skeleton-joints with a better interpretability. However, when tasks involve relatively large input graphs, GCNs with multi-head attention become computationally inefficient and require lightweight design techniques. In this paper, we design efficient GCNs that make skeleton-based recognition highly efficient while also being effective.

III. Graph convnets at glance

Considering $\mathcal{S} = \{\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)\}_i$ as a collection of graphs with $\mathcal{V}_i$, $\mathcal{E}_i$ being respectively the nodes and the edges of $\mathcal{G}_i$, each graph $\mathcal{G}_i$ (denoted for short as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$) is empowered with a signal $\{\phi(u) \in \mathbb{R}^s : \ v \in \mathcal{V}\}$ and an adjacency matrix $\mathbf{A}$. Graph convolutional networks (GCNs) learn a set of C filters $\mathcal{F}$ that define convolution on n nodes of $\mathcal{G}$ as $(\mathcal{G} \star \mathcal{F})_\mathcal{V} = f\big(\mathbf{A}\ \mathbf{U}^\top\ \mathbf{W}\big)$, here $n = |\mathcal{V}|$, $^\top$ stands for transpose, $\mathbf{U} \in \mathbb{R}^{s\times n}$ is the graph signal, $\mathbf{W} \in \mathbb{R}^{s\times C}$ is the matrix of convolutional parameters corresponding to the C filters and $f(.)$ is a nonlinear activation applied entry-wise. With $(\mathcal{G}\star\mathcal{F})_\mathcal{V}$, the input signal $\mathbf{U}$ is projected using $\mathbf{A}$ providing for each node v, the aggregate set of its neighbors. Entries of $\mathbf{A}$ can either be handcrafted or learned in $(\mathcal{G}\star\mathcal{F})_\mathcal{V}$ forming a convolutional block with two layers: the first layer in $(\mathcal{G} \star \mathcal{F})_\mathcal{V}$ aggregates signals in $\mathcal{N}(\mathcal{V})$ (as the sets of neighbors of nodes in $\mathcal{V}$) by multiplying $\mathbf{U}$ with $\mathbf{A}$, while the second layer performs convolutions by multiplying the resulting aggregates with the C filters in $\mathbf{W}$. Learning multiple adjacency (also referred to as attention) matrices (denoted as $\{\mathbf{A}^k\}_{k=1}^K$) enable capturing various contexts and graph topologies when achieving aggregation and convolution. With multiple adjacency matrices $\{\mathbf{A}^k\}_k$ (and associated convolutional filter parameters $\{\mathbf{W}^k\}_k$), $(\mathcal{G} \star \mathcal{F})_\mathcal{V}$ is updated as $f\big(\sum_{k=1}^K \mathbf{A}^k\mathbf{U}^\top\mathbf{W}^k\big)$, so stacking multiple aggregation and convolutional layers makes GCNs more accurate but heavier. Our proposed method, in this paper, seeks to make GCNs lightweight yet effective.

IV. Proposed Method

Subsequently, we formalize a given GCN as a multi-layered neural network g_θ with weights defined by $\theta = \big\{\mathbf{W}^1, \ldots, \mathbf{W}^L\big\}$, and L its depth, $\mathbf{W}^\ell \in \mathbb{R}^{d_{\ell-1}\times d_\ell}$ its ℓ^{th} layer weight tensor, and d_ℓ its dimension. We define the output of a given layer ℓ as $\phi^\ell = f_\ell(\mathbf{W}^{\ell\top}\ \phi^{\ell-1})$, $\ell \in \{2, \ldots, L\}$, with f_ℓ an activation function; without a loss of generality, we omit the bias in the definition of ϕ^ℓ.

Pruning is the process of removing a subset of weights in θ by multiplying $\mathbf{W}^\ell$ with a binary mask $\mathbf{M}^\ell \in \{0,1\}^{d_{\ell-1}\times d_\ell}$. The binary entries of $\mathbf{M}^\ell$ are determined by pruning the underlying layer connections, so $\phi^\ell = f_\ell((\mathbf{M}^\ell \odot \mathbf{W}^\ell)^\top\ \phi^{\ell-1})$, with $\odot$ being the element-wise matrix product. In our definition of pruning, entries of the tensor $\{\mathbf{M}^\ell\}_\ell$ are set depending on the prominence and also on how the underlying connections in g_θ are grouped (or not); pruning that removes all the connections individually (resp. jointly) is referred to as *unstructured* (resp. *structured*) whilst pruning that removes some connections *first* group-wise and *then* individually is dubbed as *coarse-to-fine*. In what follows, we introduce our main contribution; a novel coarse-to-fine method that allows combining multiple pruning granularities resulting into efficient and also effective lightweight networks (as shown later in experiments).

A. Coarse-to-fine Pruning

We define our parametrization as the Hadamard product involving a weight tensor and a *cascaded* function applied

to the same tensor as

$$\mathbf{W}^\ell = \hat{\mathbf{W}}^\ell \odot \psi(\hat{\mathbf{W}}^\ell), \tag{1}$$

here $\hat{\mathbf{W}}^\ell$ is a latent tensor and $\psi(\hat{\mathbf{W}}^\ell)$ a continuous relaxation of $\mathbf{M}^\ell$ which enforces the prior that (i) weights $\hat{\mathbf{W}}^\ell$ with the smallest magnitude should be removed, (ii) entries in mask $\psi(\hat{\mathbf{W}}^\ell)$ are either removed group-wise (through rows, columns, channels) or removed individually. In the following, we expand the definition of fine and coarse parametrizations (respectively denoted as ψ_f and ψ_c) prior to their combination in Eq. 3. Unless stated otherwise, we omit ℓ in the definition of $\hat{\mathbf{W}}^\ell$ and we rewrite it (for short) as $\hat{\mathbf{W}}$.

Fine-grained parametrization. As subsequently described, the function $\psi_f(\hat{\mathbf{W}})$ is entry-wise applied to the tensor $\hat{\mathbf{W}}$ with the prior that small magnitude weights should be individually removed. The class of ψ_f functions must be: (1) differentiable, (2) symmetric, (3) bounded in $[0, 1]$, and (4) asymptotically reaching 1 when entries of $\psi_f(.)$ have large magnitude and 0 otherwise. Properties (1) and (2) respectively ensure that ψ_f has computable gradient and that only the magnitude of the latent weights matters whereas properties (3) and (4) guarantee that ψ_f is neither overflowing nor changing the sign of the latent weights, and also values in ψ_f behave as crisp (almost binary) masks reaching asymptotically 1 iff the latent weights in $|\hat{\mathbf{W}}|$ are sufficiently large, and 0 otherwise. In practice, a choice of ψ_f that satisfies these four conditions is the symmetrized shifted sigmoid $\psi_f(\hat{\mathbf{W}}) = 2\ \mathbf{sigmoid}(\sigma\hat{\mathbf{W}}^2) - 1$; here the power and the sigmoid are applied entry-wise and σ is a scaling factor that controls the crispness (binarization) of the mask entries in $\psi_f(\hat{\mathbf{W}})$. In practice, σ is annealed so as to cut-off the connections in the network in a smooth and differentiable manner — as the optimization of $\hat{\mathbf{W}}$ evolves — while obtaining at the end of the optimization process crisp (almost binary) masks.

Coarse-grained parametrization. The function $\psi_c(\hat{\mathbf{W}})$ implements a coarse-grained pruning by removing connections group-wise (row-wise, column-wise or block/channels-wise) in the tensor $\hat{\mathbf{W}}$. This function is formally defined as

$$\psi_c(\hat{\mathbf{W}}) \;=\; \underbrace{\phi^{-1}(\mathbf{P}_r\ \phi(\hat{\mathbf{W}}))}_{\text{row-wise pruning}} \odot \underbrace{\phi^{-1}(\phi(\hat{\mathbf{W}})^\top\ \mathbf{P}_c)}_{\text{column-wise pruning}} \odot \underbrace{\phi^{-1}(\mathbf{P}_r\mathbf{P}_c^\top\ \phi(\hat{\mathbf{W}}))}_{\text{block-wise pruning}}, \tag{2}$$

here ϕ (resp. ϕ^{-1}) reshapes a matrix into a vector (resp. vice-versa), and $\mathbf{P}_r \in \{0,1\}^{(d_{\ell-1}\times d_\ell)^2}$, $\mathbf{P}_c \in \{0,1\}^{(d_{\ell-1}\times d_\ell)^2}$ are two adjacency matrices that model the neighborhood across respectively the rows and the columns of $\hat{\mathbf{W}}$ whilst $\mathbf{P}_r\mathbf{P}_c^\top \in \{0,1\}^{(d_{\ell-1}\times d_\ell)^2}$ models this neighborhood through blocks/channels of the tensor $\hat{\mathbf{W}}$.

Coarse-to-fine-grained parametrization. Considering the above definition of ψ_c and ψ_f, we obtain our complete coarse-to-fine mask parametrization as

$$\psi(\hat{\mathbf{W}}) = \underbrace{[\psi_c(\psi_f(\hat{\mathbf{W}}))]}_{\text{coarse-grained pruning}} \odot \underbrace{\psi_f(\hat{\mathbf{W}})}_{\text{fine-grained pruning}}. \tag{3}$$

From Eqs. 2 and 3, assuming crisp (almost binary) entries in $\hat{\mathbf{W}}$ (thanks to the sigmoid), block-wise pruning has the highest priority, followed by column-wise and row-wise pruning. This priority allows designing highly efficient lightweight networks with a coarse-granularity for block / column / row-wise (structured) pruning while the entry-wise (unstructured) parametrization is less computationally efficient but allows reaching the targeted pruning rate with a finer granularity (see Fig. 1). In sum, CTF allows efficient coarse-grained network design while also leveraging the accuracy of fine-grained one, thereby leading to *both* efficient and effective pruned networks as shown subsequently in experiments.

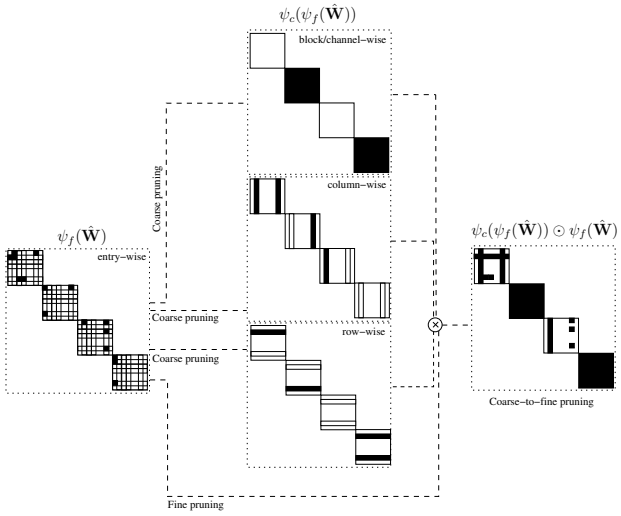

Fig. 1. This figure shows the CTF pruning process in Eq. 3; here each diagonal block corresponds to a channel.

B. Variational Pruning

By considering Eq. 1, we define our CTF pruning loss as

$$\mathcal{L}_e\big(\{\psi(\hat{\mathbf{W}}^\ell) \odot \hat{\mathbf{W}}^\ell\}_\ell\big) + \lambda\bigg(\sum_{\ell=1}^{L-1} \mathbf{1}_{d_\ell}^\top\ \psi(\hat{\mathbf{W}}^\ell)\ \mathbf{1}_{d_{\ell+1}}\ - c\bigg)^2, \tag{4}$$

here the left-hand side term is the cross entropy loss that measures the discrepancy between predicted and ground-truth labels. The right-hand side term is a budget loss that allows reaching any targeted pruning cost c. In the above objective function, λ is overestimated (to 1000 in practice) in order to make Eq. 4 focusing on the implementation of the budget. As training reaches its final epochs, the budget loss reaches its minimum and the gradient of the global objective function becomes dominated by the gradient of $\mathcal{L}_e$, and this allows improving further classification performances.

V. Experiments

In this section, we evaluate the performances of our pruned GCNs on skeleton-based recognition using two challenging datasets, namely SBU [64] and FPHA [20]. SBU is an interaction dataset acquired using the Microsoft Kinect sensor; it includes in total 282 moving skeleton sequences (performed by two interacting individuals) belonging to 8 categories. Each pair of interacting individuals corresponds to two 15 joint skeletons and each joint is characterized with a sequence of its 3D coordinates across video frames. In this dataset, we consider the same evaluation protocol as the one suggested in the original dataset release [64] (i.e., train-test split). The FPHA dataset includes 1175 skeletons belonging to 45 action categories with high inter and intra subject variability. Each skeleton includes 21 hand joints and each joint is again characterized with a sequence of its 3D coordinates across video frames. We evaluate the performance of our method on FPHA following the protocol in [20]. In all these experiments, we report the average accuracy over all the classes of actions.

TABLE I
Comparison of our baseline GCN against related work on the SBU database.

Method	Accuracy (%)
Raw Position [64]	49.7
Joint feature [60]	86.9
CHARM [61]	86.9
H-RNN [42]	80.4
ST-LSTM [43]	88.6
Co-occurrence-LSTM [47]	90.4
STA-LSTM [54]	91.5
ST-LSTM + Trust Gate [43]	93.3
VA-LSTM [45]	97.6
GCA-LSTM [44]	94.9
Riemannian manifold. traj [33]	93.7
DeepGRU [48]	95.7
RHCN + ACSC + STUFE [29]	98.7
Our baseline GCN	98.4

Implementation details and baseline GCNs. All the GCNs have been trained using the Adam optimizer for $2,700$ epochs with a batch size of 200 for SBU and 600 for FPHA, a momentum of 0.9, and a global learning rate (denoted as $\nu(t)$) inversely proportional to the speed of change of the loss used to train the networks; with $\nu(t)$ decreasing as $\nu(t) \leftarrow \nu(t-1) \times 0.99$ (resp. increasing as $\nu(t) \leftarrow \nu(t-1)/0.99$) when the speed of change of the loss in Eq. 4 increases (resp. decreases). Experiments were run on a GeForce GTX 1070 GPU device with 8 GB memory, without dropout or data augmentation. The baseline GCN architecture for SBU includes an attention layer of 1 head, a convolutional layer of 8 filters, a dense fully connected layer, and a softmax layer; notice that this architecture is not very heavy, nonetheless its pruning is very challenging (particularly at high pruning rates) as it may result into disconnected networks. The baseline GCN architecture for FPHA is heavier and includes 16 heads, a convolutional layer of 32 filters, a dense fully connected layer, and a softmax layer. Both the baseline (unpruned) GCN architectures, on the SBU and the FPHA benchmarks, are accurate (see tables. I and II), and our goal is to make them lightweight while maintaining their accuracy.

TABLE II
Comparison of our baseline GCN against related work on the FPHA database.

Method	Color	Depth	Pose	Accuracy (%)
2-stream-color [18]	✓	✗	✗	61.56
2-stream-flow [18]	✓	✗	✗	69.91
2-stream-all [18]	✓	✗	✗	75.30
HOG2-dep [51]	✗	✓	✗	59.83
HOG2-dep+pose [51]	✗	✓	✓	66.78
HON4D [62]	✗	✓	✗	70.61
Novel View [63]	✗	✓	✗	69.21
1-layer LSTM [47]	✗	✗	✓	78.73
2-layer LSTM [47]	✗	✗	✓	80.14
Moving Pose [65]	✗	✗	✓	56.34
Lie Group [56]	✗	✗	✓	82.69
HBRNN [42]	✗	✗	✓	77.40
Gram Matrix [67]	✗	✗	✓	85.39
TF [19]	✗	✗	✓	80.69
JOULE-color [26]	✓	✗	✗	66.78
JOULE-depth [26]	✗	✓	✗	60.17
JOULE-pose [26]	✗	✗	✓	74.60
JOULE-all [26]	✓	✓	✓	78.78
Huang et al. [32]	✗	✗	✓	84.35
Huang et al. [35]	✗	✗	✓	77.57
HAN [34]	✗	✗	✓	85.74
Our baseline GCN	✗	✗	✓	86.43

TABLE III
This table shows detailed performances and ablation study on SBU for different pruning rates. "none" stands for no-actual speedup is observed as the underlying tensors/architecture remain shaped identically to the unpruned network (despite having pruned connections); see also Fig. 2.

Pruning rates	Accuracy (%)	SpeedUp	Observation
0%	**98.40**	none	Baseline GCN
70%	93.84	none	Band-stop Weight Param.
90%	83.07	11×	Coarse MP (structured)
	96.92	none	Fine MP (unstructured)
	89.23	6×	Coarse-to-Fine MP (both)
95%	75.38	34×	Coarse MP (structured)
	93.84	none	Fine MP (unstructured)
	84.61	9×	Coarse-to-Fine MP (both)
98%	49.23	235×	Coarse MP (structured)
	90.76	none	Fine MP (unstructured)
	76.92	43×	Coarse-to-Fine MP (both)
Comparative (regularization-based) pruning			
98%	55.38	none	MP+ℓ_0-reg.
	73.84	none	MP+ℓ_1-reg.
	61.53	none	MP+Entropy-reg.
	75.38	none	MP+Cost-aware-reg.

Lightweight GCNs (Comparison and Ablation). Tables III-IV show a comparison and an ablation study of our method both on SBU and FPHA datasets. First, according to tables III-IV, when only the cross entropy loss is used without

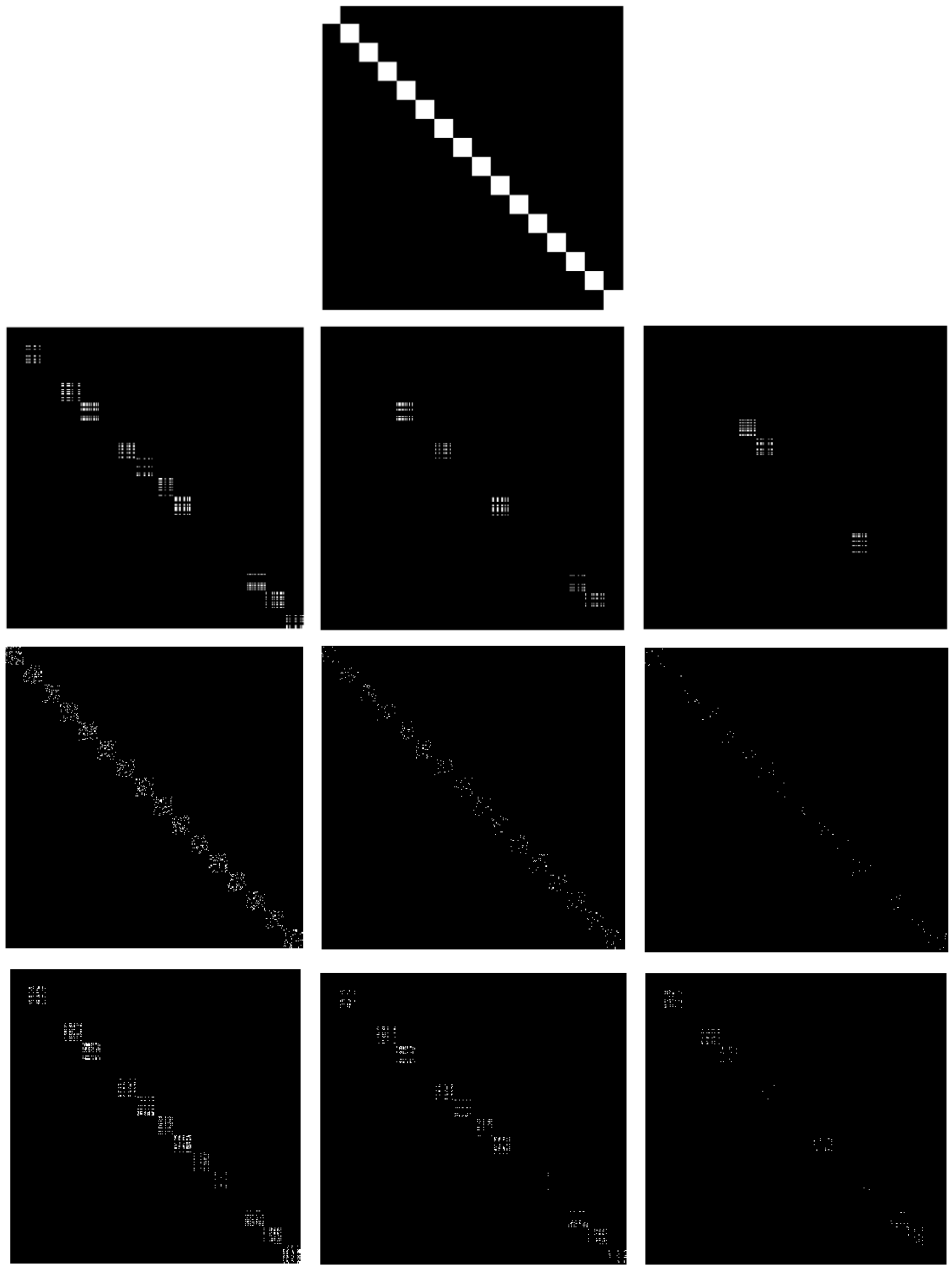

Fig. 2. This figure shows a crop of the mask tensor of the second (multi-head-attention) layer of our GCNs when trained on the FPHA dataset. Top row corresponds to the original mask (without pruning) while the second and the third rows correspond to masks obtained with structured and unstructured pruning respectively (with increasing pruning rates; from left-to-right equal to 90%, 95% and 98% respectively). The final row corresponds to masks obtained with semi-structured pruning (with again increasing pruning rates; from left-to-right, equal to 90%, 95% and 98% respectively). In all these masks, each diagonal block corresponds to a channel. **Better to zoom the PDF.**

budget (i.e., $\lambda = 0$ in Eq. 4), performances are close to the initial heavy GCNs (particularly on FPHA), with less parameters[1] as this produces a regularization effect similar to [57]. Then, when pruning is achieved with the coarse-grained parametrization, the accuracy is relatively low but the speedup is high particularly for high pruning regimes. When pruning is performed with the fine-grained parametrization, the accuracy reaches its highest value, but no speedup is observed as pruning is unstructured and the architecture of the pruned networks remains unchanged. When the coarse-to-fine parametrization is used, we observe the best tradeoff between accuracy and speedup; in other words, coarsely pruned parts of the network lead to high speedup and efficient computation, while finely pruned parts allow reaching better accuracy with a limited impact on computation, so a significant speedup is still observed.

Extra comparison against other regularizers shows

[1]Pruning rate does not exceed 70% and no control on this rate is achievable.

TABLE IV

THIS TABLE SHOWS DETAILED PERFORMANCES AND ABLATION STUDY ON FPHA FOR DIFFERENT PRUNING RATES. "NONE" STANDS FOR NO-ACTUAL SPEEDUP IS OBSERVED AS THE UNDERLYING TENSORS/ARCHITECTURE REMAIN SHAPED IDENTICALLY TO THE UNPRUNED NETWORK (DESPITE HAVING PRUNED CONNECTIONS); SEE ALSO FIG. 2.

Pruning rates	Accuracy (%)	SpeedUp	Observation
0%	**86.43**	none	Baseline GCN
50%	85.56	none	Band-stop Weight Param.
90%	76.69	13×	Coarse MP (structured)
	83.13	none	Fine MP (unstructured)
	80.17	6×	Coarse-to-Fine MP (both)
95%	70.08	37×	Coarse MP (structured)
	81.56	none	Fine MP (unstructured)
	77.56	13×	Coarse-to-Fine MP (both)
98%	63.30	96×	Coarse MP (structured)
	76.86	none	Fine MP (unstructured)
	70.95	41×	Coarse-to-Fine MP (both)
Comparative (regularization-based) pruning			
98%	64.69	none	MP+ℓ_0-reg.
	70.78	none	MP+ℓ_1-reg.
	67.47	none	MP+Entropy-reg.
	69.91	none	MP+Cost-aware-reg.

the substantial gain of our method. Indeed, our method is compared against variational pruning with different regularizers plugged in Eq. 4 instead of our budget loss, namely ℓ_0 [15], ℓ_1 [11], entropy [12] and ℓ_2-based cost [39]; all without parametrization. From the observed results, the impact of our method is substantial for different settings and for equivalent pruning rate (namely 98%). Note that when alternative regularizers are used, multiple settings (trials) of the underlying hyperparameter λ (in Eq. 4) are considered prior to reach the targeted rate, and this makes the whole training and pruning process overwhelming. While cost-aware regularization makes training more tractable, its downside resides in the observed collapse of trained masks; this is a well known effect that affects performances at high pruning rates. Finally, Fig. 2 shows examples of obtained mask tensors taken from the second (attention) layer of the pruned GCNs; we observe compact tensor weight distributions with some individually pruned connections when using CTF, while coarse-grained and fine-grained pruning, when taken separately, either produce *compact* or *spread* tensors with a negative impact on either *accuracy* or *speed* respectively. CTF gathers both fine and coarse-grained advantages while discarding their downsides.

VI. CONCLUSION

We introduce in this paper a CTF approach for pruning. The strength of the proposed method resides in its ability to combine the advantages of coarse-grained (structured) and fine-grained (unstructured) pruning while discarding their downsides. The proposed method relies on a novel weight parametrization that first prunes tensors channel-wise, then column-wise and row-wise, and finally entry-wise enabling both high efficiency and high accuracy. Experiments conducted on the challenging tasks of action and hand-gesture recognition, using two standard datasets, corroborate all these findings.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[2] T-N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[3] R. Li, S. Wang, F. Zhu, and J. Huang. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[4] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In IJCNN, 2005.

[5] H. Sahbi. Kernel pca for similarity invariant shape recognition. *Neurocomputing*, 70(16-18):3034–3045, 2007.

[6] B. Knyazev, G-W. Taylor, and M. Amer. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[7] S-I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of AAAI*, vol 34, p5191–5198, 2020.

[8] M-A. Carreira-Perpinán and Y. Idelbayev. "learning-compression" algorithms for neural net pruning. In *Proceedings of IEEE CVPR*, pages 8532–8541, 2018.

[9] A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T-J. Yang, and E. Choi. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In CVPR 2018.

[10] H. Sahbi. Learning connectivity with graph convolutional networks. In *25th International Conference on Pattern Recognition (ICPR)*, pages 9996–10003. IEEE, 2021.

[11] B. Naga, G. Koneru and V. Vasudevan. Sparse artificial neural networks using a novel smoothed lasso penalization. *IEEE TCS II: Express Briefs*, 66(5):848–852, 2019.

[12] S. Wiedemann, A. Marban, K-R. Müller, and W. Samek. Entropy-constrained training of deep neural networks. In *2019 IJCNN*, pages 1–8. IEEE, 2019.

[13] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

[14] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of IEEE ICCV*, p2736–2744, 2017.

[15] C. Louizos, M. Welling, and D-P. Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

[16] H. Sahbi. Topologically-consistent magnitude pruning for very lightweight graph convolutional networks. In *IEEE International Conference on Image Processing (ICIP)*, pages 3495–3499. IEEE, 2022.

[17] A. Howard, M. Sandler, G. Chu, L-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. Searching for mobilenetv3. In CVPR 2019.

[18] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of IEEE CVPR*, pages 1933–1941, 2016.

[19] G. Garcia-Hernando and T-K. Kim. Transition forests: Learning discriminative temporal transitions for action recognition and detection. In *Proceedings of IEEE CVPR*, pages 432–440, 2017.

[20] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 409–419, 2018.

[21] S. Han, H. Mao, and W-J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[22] H. Sahbi. Learning laplacians in chebyshev graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2064–2075, 2021.

[23] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[24] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

[25] B. Hassibi and D. Stork. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5, 1992.

[26] J-F. Hu, W-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5344–5352, 2015.

[27] H. Sahbi. TCMP: End-to-End Topologically Consistent Magnitude Pruning for Miniaturized Graph Convolutional Networks. In IEEE ICASSP 2024.

[28] C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang. Spatio-temporal graph convolution for skeleton based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[29] S. Li, T. Jiang, T. Huang, and Y. Tian. Global co-occurrence feature learning and active coordinate system conversion for skeleton-based action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 586–594, 2020.

[30] Y-H. Wen, L. Gao, H. Fu, F-L. Zhang, and S. Xia. Graph cnns with motif and variable temporal block for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8989–8996, 2019.

[31] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[32] Z. Huang and L. Van Gool. A riemannian network for spd matrix learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[33] A. Kacem, M. Daoudi, B. Ben Amor, S. Berretti, and J-C. Alvarez-Paiva. A novel geometric framework on gram matrix trajectories for human behavior understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):1–14, 2018.

[34] J. Liu, Y. Wang, S. Xiang, and C. Pan. Han: An efficient hierarchical self-attention network for skeleton-based gesture recognition. *arXiv preprint arXiv:2106.13391*, 2021.

[35] Z. Huang, J. Wu, and L. Van Gool. Building deep networks on grassmann manifolds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[36] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan. Decoupled representation learning for skeleton-based gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5751–5760, 2020.

[37] A. Mazari and H. Sahbi. "MLGCN: Multi-Laplacian graph convolutional networks for human action recognition." The British Machine Vision Conference (BMVC). 2019.

[38] D-P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] C. Lemaire, A. Achkar, and P-M. Jodoin. Structured pruning of neural networks with budget-aware regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9108–9116, 2019.

[40] H. Sahbi. "Coarse-to-fine deep kernel networks." IEEE ICCV-W, 2017.

[41] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H-P. Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[42] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[43] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 816–833. Springer, 2016.

[44] 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*, 27(4):1586–1599, 2017.

[45] 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 *Proceedings of IEEE ICCV*, pages 2117–2126, 2017.

[46] H. Sahbi. DAMP: Distribution-Aware Magnitude Pruning for Budget-Sensitive Graph Convolutional Networks. In IEEE ICASSP 2024.

[47] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In AAAI, 2016.

[48] M. Maghoumi and J-J. LaViola. Deepgru: Deep gesture recognition utility. In *Advances in Visual Computing: 14th International Symposium on Visual Computing, ISVC 2019, Lake Tahoe, NV, USA, October 7–9, 2019, Proceedings, Part I 14*, pages 16–31. Springer, 2019.

[49] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera. Rgb-d-based human motion recognition with deep learning: A survey. *Computer Vision and Image Understanding*, 171:118–139, 2018.

[50] M. Meshry, M-E. Hussein, and M. Torki. Linear-time online action detection from 3d skeletal data using bags of gesturelets. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–9. IEEE, 2016.

[51] E. Ohn-Bar and M-M. Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

[52] H. Sahbi. Kernel-based graph convolutional networks. In *25th International Conference on Pattern Recognition (ICPR)*, pages 4887–4894. IEEE, 2021.

[53] W. Pan, H. Dong, and Y. Guo. Dropneuron: Simplifying the structure of deep neural networks. *arXiv preprint arXiv:1606.07326*, 2016.

[54] 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*, volume 31, 2017.

[55] H. Sahbi. Lightweight connectivity in graph convolutional networks for skeleton-based recognition. In *IEEE International Conference on Image Processing (ICIP)*, pages 2329–2333. IEEE, 2021.

[56] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[57] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066. PMLR, 2013.

[58] H. Sahbi, J-Y. Audibert, and R. Keriven. Context-dependent kernels for object classification. *IEEE transactions on pattern analysis and machine intelligence*, 33(4):699–708, 2011.

[59] X. Yang and Y. Tian. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25(1):2–11, 2014.

[60] Y. Ji, G. Ye, and H. Cheng. Interactive body part contrast mining for human interaction recognition. In *2014 IEEE international conference on multimedia and expo workshops (ICMEW)*, pages 1–6. IEEE, 2014.

[61] W. Li, L. Wen, M-C. Chuah, and S. Lyu. Category-blind human action recognition: A practical recognition system. In *Proceedings of IEEE ICCV*, pages 4444–4452, 2015.

[62] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of IEEE CVPR*, pages 716–723, 2013.

[63] H. Rahmani and A. Mian. 3d action recognition from novel viewpoints. In *Proceedings of IEEE CVPR*, pages 1506–1515, 2016.

[64] K. Yun, J. Honorio, D. Chattopadhyay, Tamara L Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE CVPRW*, pages 28–35. IEEE, 2012.

[65] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2752–2759, 2013.

[66] H. Sahbi and D. Geman. A hierarchy of support vector machines for pattern detection. *Journal of Machine Learning Research*, 7(10), 2006.

[67] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In CVPR 2016.

[68] J-C. Nunez, R. Cabido, J-J. Pantrigo, A-S. Montemayor, and J-F. Velez. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, 76:80–94, 2018.

[69] LI, Yanyu, ZHAO, Pu, YUAN, Geng, et al. Pruning-as-search: Efficient neural architecture search via channel pruning and structural reparameterization. arXiv preprint arXiv:2206.01198, 2022.