MRMP: Multi-Rate Magnitude Pruning of Graph Convolutional Networks

Hichem Sahbi [1]

Abstract

In this paper, we devise a novel lightweight Graph Convolutional Network (GCN) design dubbed as Multi-Rate Magnitude Pruning (MRMP) that jointly trains network topology and weights. Our method is variational and proceeds by aligning the weight distribution of the learned networks with an a priori distribution. In the one hand, this allows implementing any fixed pruning rate, and also enhancing the generalization performances of the designed lightweight GCNs. In the other hand, MRMP achieves a joint training of multiple GCNs, on top of shared weights, in order to extrapolate accurate networks at any targeted pruning rate without retraining their weights. Extensive experiments conducted on the challenging task of skeleton-based recognition show a substantial gain of our lightweight GCNs particularly at very high pruning regimes.

1. Introduction

With the resurgence of deep neural networks (Krizhevsky et al., 2017), many computer vision tasks have been successfully revisited during the last decade (He et al., 2016; 2017; Jian et al., 2020; Ronneberger et al., 2015; Jiu & Sahbi, 2017; 2019). These tasks have been approached with increasingly accurate but *oversized* networks, and this makes their deployment on cheap devices highly challenging. Particularly, in hand-gesture recognition and human computer interaction, edge devices are endowed with limited computational resources. Therefore, fast and lightweight models with high recognition performances are vital for skeleton-based recognition. Recent learning models applying deep networks have shown saturated recognition accuracy without substantial improvement, while computational efficiency still remains a serious issue. Among these learning models, graph convolutional networks (GCNs) are known to be effective particularly on non-euclidean domains such as skeleton data (Zhu et al., 2016b; Zhang et al., 2020). At least two categories of GCNs are known in the literature; spatial and spectral. Spectral methods first project graph signals from the input to the Fourier domain prior to achieve convolution, and then back-project the convolved signals in the input domain (Kipf & Welling, 2016; Levie et al., 2018; Li et al., 2018b; Defferrard et al., 2016; Bruna et al., 2013; Henaff et al., 2015; Chung, 1997; Sahbi, 2021c; Mazari & Sahbi, 2019b). Spatial methods proceed differently by aggregating node signals using multi-head attention (MHA) prior to apply convolutions on the resulting node aggregates (Gori et al., 2005; Micheli, 2009; Wu et al., 2020; Hamilton et al., 2017; Knyazev et al., 2019; Sahbi et al., 2011; Sahbi, 2021b;a). Spatial GCNs are deemed more effective compared to spectral ones, however, their main downside resides in their high computational complexity. Hence, a major challenge is how to make these networks lightweight while maintaining their high accuracy (Huang et al., 2018a; Sandler et al., 2018; Howard et al., 2017; Tan & Le, 2019; Cai et al., 2019; He et al., 2018a;b; Sahbi, 2021d; 2023).

Many existing works tackle the issue of lightweight network design including tensor decomposition (Howard et al., 2019), quantization (Han et al., 2015a), distillation (Hinton et al., 2015; Mirzadeh et al., 2020; Zhang et al., 2018; Ahn et al., 2019; Sahbi et al., 2006) and pruning (LeCun et al., 1989; Hassibi & Stork, 1992; Han et al., 2015b; Sahbi, 2022). In particular, pruning methods are highly effective. Their principle consists in removing connections whose impact on the classification performances is the least noticeable. Two major classes of pruning techniques exist in the literature; structured (Li et al., 2016; Liu et al., 2017d) and unstructured (Han et al., 2015b;a). The former consists in zeroing-out weights of entire filters or channels whilst the latter seeks to remove weights independently. Whereas structured methods produce computationally more efficient networks, they are less accurate compared to unstructured techniques; indeed, the latter provide more flexible (and thereby more accurate) networks which are computationally still efficient. Magnitude pruning (MP) (Han et al., 2015a) is one of the mainstream unstructured methods that proceeds by removing the smallest weight connections prior to retrain the resulting pruned (lightweight) network. While being able to reach any targeted pruning rate exactly, MP is clearly suboptimal as its design *decouples* the training of network

[1] Sorbonne University, UPMC, CNRS, LIP6, France. Correspondence to: hichem.sahbi@sorbonne-universite.fr

Work presented at the ES-FoMo Workshop at the International Conference on Machine Learning – ICML 2023.

topology from weights. Therefore, any removed connection cannot be recovered when retraining the pruned networks, and this usually leads to suboptimal performances. Besides, the full retraining of the pruned networks (at multiple pruning rates) makes MP highly intractable.

In this paper, we investigate a novel alternative for magnitude pruning referred to as MRMP (Multi-Rate Magnitude Pruning) that allows (i) *coupling* the training of network topology and weights, (ii) *learning simultaneously multiple* network instances for different pruning rates, and (iii) *extrapolating* accurate networks at any unseen pruning rate without retraining. The proposed method constrains the distribution of the learned weights to match a priori targeted distributions and this allows, *via a band-stop mechanism*, to dropout all the connections up to a given targeted pruning rate. The advantage of the proposed contribution is twofold; in the one hand, it constrains the learned weights to fit a targeted distribution and this leads to better generalization. In the other hand, it allows obtaining fully trained networks at any unseen pruning rate instantaneously without weight retraining.

Considering all the aforementioned issues, the main contributions of our paper include

- A band-stop weight parametrization that achieves a *joint* training of GCN topology and weights (see section 4.1). This parametrization relies on shared latent weights that reduce the number of training parameters of the pruned GCNs.
- A KLD (Kullback Leibler Divergence) based regularizer that constrains the latent weights to fit an *a priori* distribution, and this allows implementing any targeted pruning rate *almost* exactly (see section 4.2).
- A multiple magnitude pruning that obtains optimal GCNs at any targeted pruning rate thanks to the band-stop parametrization and the KLD regularizer. The latter defines a continuum of *weight aggregates* associated to GCNs with increasing pruning rates. These *weight aggregates* allow generalizing across unseen pruning rates without retraining (see section 4.3).
- Extensive experiments conducted on the challenging task of skeleton-based recognition corroborate all these findings and show the outperformance of our method against the related work (see section 5).

2. Related work

We review and discuss subsequently the related work in variational pruning and skeleton-based recognition, and the limitations that motivate our contributions.

Variational Pruning. The general recipe of variational pruning consists in learning both weights and binary masks that capture topology of pruned networks. This is achieved by minimizing a loss that combines (via a mixing hyperparameter) a classification error and a regularizer which controls the sparsity of the resulting masks (Liu et al., 2017d; Wen et al., 2016; Louizos et al., 2017). However, these methods are powerless to implement any given targeted pruning rate (cost) without overtrying multiple settings of the mixing hyperparameters. Alternative methods explicitly model the cost, using ℓ_0-based criteria (Louizos et al., 2017; Pan et al., 2016), in order to minimize the discrepancy between the observed cost and the targeted one. Nonetheless, the underlying optimization problems are highly combinatorial and existing solutions usually rely on sampling heuristics or relaxation, such as ℓ_1/ℓ_2-based, entropy, etc. (Gordon et al., 2018; Carreira-Perpinán & Idelbayev, 2018; Koneru & Vasudevan, 2019; Wiedemann et al., 2019); the latter promote sparsity, but are powerless to implement any given target cost exactly, and also result into overpruning effects leading to disconnected subnetworks, with weak generalization, especially at very high pruning regimes. Besides, most of the existing solutions, including magnitude pruning (Han et al., 2015a), decouple the training of network topology (masks) from weights, and this makes the learning of pruned networks clearly suboptimal.

Skeleton-based Recognition. With the emergence of sensors, including Intel RealSense (Keselman et al., 2017) and Microsoft Kinect (Zhang et al., 2017a), interest in skeleton-based recognition is increasingly growing (Cao et al., 2017). Hand-gesture and action recognition are two neighboring tasks which have initially been tackled using RGB (Liu & Yuan, 2018; Hu et al., 2015; Wang & Sahbi, 2013; Yuan et al., 2012; Wang & Sahbi, 2014), depth (Ohn-Bar & Trivedi, 2014; Wang et al., 2015), shape/normals (Oreifej & Liu, 2013; Rahmani & Mian, 2016; Yun et al., 2012; Ji et al., 2014; Li et al., 2015; Zanfir et al., 2013; Sahbi, 2007; Sahbi & Fleuret, 2004) and also skeleton-based techniques (Wang et al., 2018). In particular, early skeleton-based methods are based on modeling human motions using handcrafted features (Xia et al., 2012; Yang & Tian, 2014), dynamic time warping (Vemulapalli et al., 2014), temporal information (Zhang et al., 2016; Garcia-Hernando & Kim, 2017) as well as temporal pyramids (Zhu et al., 2016a; Q. De Smedt & Vandeborre, 2016). With the resurgence of deep learning, all these methods have been quickly overtaken by 2D/3D Convolutional Neural Networks (CNNs) (Feichtenhofer et al., 2016; Nunez et al., 2018a; Mazari & Sahbi, 2019a) that capture global skeleton posture together with local joint motion (Hou et al., 2018; Liu et al., 2020), by Recurrent Neural Networks (RNNs) which capture motion dynamics (Zhu et al., 2016b; Chen et al., 2017; Ke et al., 2017; Liu et al., 2017c; Liu & Yuan, 2018; Wang et al., 2016; Du et al., 2015; Wang & Wang, 2017; Liu et al., 2017b; Nunez et al., 2018b; Shahroudy et al., 2016; Zhang et al., 2017b; Lee et al., 2017; Liu et al., 2016; Maghoumi & LaViola, 2019;

Zhang et al., 2017a; Liu et al., 2017a), and manifold learning (Huang & Van Gool, 2017; Huang et al., 2018b; Nguyen et al., 2019; Liu et al., 2021; Kacem et al., 2018) as well as attention-based networks (Weng et al., 2018; Hou et al., 2018; Chen et al., 2019; Song et al., 2017). With the recent emergence of GCNs, the latter have been increasingly used in skeleton-based recognition (Huang & Van Gool, 2017; Huang et al., 2017; Li et al., 2018a; Yan et al., 2018; Wen et al., 2019; Shi et al., 2018; Nguyen et al., 2019; Li et al., 2019; 2020). These models explicitly capture, with a better interpretability, spatial and temporal attention among skeleton joints (Li et al., 2018b). However, on tasks involving large input graphs (such skeleton-based recognition), GCNs become computationally inefficient and require lightweight design techniques.

3. Graph convnets at a glance

Let $\mathcal{S} = \{\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)\}_i$ denote 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 endowed with a signal $\{\phi(u) \in \mathbb{R}^s : \ u \in \mathcal{V}\}$ and associated with an adjacency matrix $\mathbf{A}$. GCNs aim at learning a set of C filters $\mathcal{F}$ that define convolution on n nodes of $\mathcal{G}$ (with $n = |\mathcal{V}|$) as $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}} = f\big(\mathbf{A}\ \mathbf{U}^\top\ \mathbf{W}\big)$, here $^\top$ 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 entrywise. In $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}}$, the input signal $\mathbf{U}$ is projected using $\mathbf{A}$ and this provides for each node u, the aggregate set of its neighbors. Entries of $\mathbf{A}$ could be handcrafted or learned so $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}}$ corresponds to a convolutional block with two layers; the first one aggregates signals in $\mathcal{N}(\mathcal{V})$ (sets of node neighbors) by multiplying $\mathbf{U}$ with $\mathbf{A}$ while the second layer achieves convolution by multiplying the resulting aggregates with the C filters in $\mathbf{W}$. Learning multiple adjacency (also referred to as attention) matrices (denoted as $\{\mathbf{A}^k\}_{k=1}^K$) allows us to capture different contexts and graph topologies when achieving aggregation and convolution. With multiple matrices $\{\mathbf{A}^k\}_k$ (and associated convolutional filter parameters $\{\mathbf{W}^k\}_k$), $(\mathcal{G}\star\mathcal{F})_{\mathcal{V}}$ is updated as $f\big(\sum_{k=1}^K \mathbf{A}^k \mathbf{U}^\top \mathbf{W}^k\big)$. Stacking aggregation and convolutional layers, with multiple matrices $\{\mathbf{A}^k\}_k$, makes GCNs accurate but heavy. We propose, in what follows, a method that makes our networks lightweight and still effective.

4. Our Lightweight GCN Design

In the remainder of this paper, we formally subsume a given GCN as a multi-layered neural network g_θ whose weights are defined as $\theta = \big\{\mathbf{W}^1, \dots, \mathbf{W}^L\big\}$, with L being its depth, $\mathbf{W}^\ell \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ its ℓ^{th} layer weight tensor, and d_ℓ the dimension of ℓ. The output of a given layer ℓ is defined as $\phi^\ell = f_\ell({\mathbf{W}^\ell}^\top\ \phi^{\ell-1})$, $\ell \in \{2, \dots, L\}$, being f_ℓ an activation function; without a loss of generality, we omit the bias in the definition of ϕ^ℓ.

Pruning consists in zeroing-out 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 set depending on whether the underlying layer connections are kept or removed, so $\phi^\ell = f_\ell((\mathbf{M}^\ell \odot \mathbf{W}^\ell)^\top\ \phi^{\ell-1})$, here $\odot$ stands for the element-wise matrix product. In this definition, entries of the tensor $\{\mathbf{M}^\ell\}_\ell$ are set depending on the prominence of the underlying connections in g_θ. However, such pruning suffers from several drawbacks. In the one hand, optimizing the discrete set of variables $\{\mathbf{M}^\ell\}_\ell$ is known to be highly combinatorial and intractable especially on large networks. In the other hand, the total number of parameters $\{\mathbf{M}^\ell\}_\ell$, $\{\mathbf{W}^\ell\}_\ell$ is twice the number of connections in g_θ and this increases training complexity and may also lead to overfitting.

4.1. Band-stop Weight Parametrization

In order to circumvent the above issues, we consider an alternative *parametrization*, related to magnitude pruning, that allows finding both the topology of the pruned networks together with their weights, without doubling the size of the training parameters, while making learning still effective. This parametrization corresponds to the Hadamard product involving a weight tensor and a function applied entry-wise 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 $\psi(\hat{\mathbf{W}}^\ell)$ is a continuous relaxation of $\mathbf{M}^\ell$ which enforces the prior that smallest weights should be removed from the network. In order to achieve this goal, ψ must be (i) bounded in $[0,1]$, (ii) differentiable, (iii) symmetric, and (iv) $\psi(\omega) \leadsto 1$ when $|\omega|$ is sufficiently large and $\psi(\omega) \leadsto 0$ otherwise. The first and the fourth properties ensure that the parametrization is neither acting as a scaling factor greater than one nor changing the sign of the latent weight, and also acts as the identity for sufficiently large weights, and as a contraction factor for small ones. The second property is necessary to ensure that ψ has computable gradient while the third condition guarantees that only the magnitudes of the latent weights matter. A choice, used in practice, that satisfies these four conditions is $\psi_{a,\sigma}(\hat{\mathbf{w}}) = (1 + \sigma \exp(a^2 - \hat{\mathbf{w}}^2))^{-1}$ with σ being a scaling factor and "a" threshold. As shown in Fig. 1, σ controls the smoothness of $\psi_{a,\sigma}$ around the support $\Omega \subseteq \mathbb{R}$ of the latent weights. This allows implementing an annealed (soft) thresholding function that cuts-off all the connections in smooth and differentiable manner as training of the latent parameters evolves. Put differently, the asymptotic behavior of $\psi_{a,\sigma}$ — that allows selecting the topology of the pruned subnetworks — is obtained as training reaches the latest epochs.

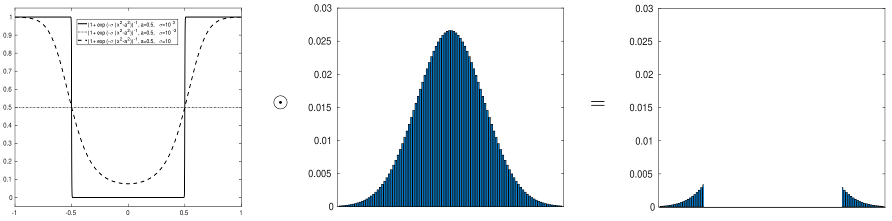

Figure 1. This illustration shows a Band-stop function $\psi_{a,\sigma}$ and its application to a given (gaussian) weight distribution. Depending on the setting of a, only large magnitude weights are kept and correspond to the fixed pruning rate. (Better to zoom the file).

4.2. SRMP: Single-Rate Magnitude Pruning

The aforementioned parameterization — while being effective (see later experiments) — it does not allow to implement any targeted pruning rate as the dynamic of learned latent weights $\{\hat{\mathbf{W}}^\ell\}_\ell$ is not known a priori. Hence, pruning rates could only be observed a posteriori or implemented, for instance, after training using a two stage process (such as magnitude pruning + retraining). In order to implement any a priori targeted pruning rate as a part of a single training process, we constrain the distribution of latent weights to fit an arbitrary probability distribution, so one may fix a in $\psi_{a,\sigma}$ and thereby achieve the targeted pruning rate. Let $\hat{W} \in \Omega$ denote a random variable standing for the latent weights in the pruned network g_θ; $\hat{W}$ is assumed drawn from any arbitrary distribution P (uniform, gaussian, laplace, etc). Fixing appropriately the distribution P *not only* allows implementing any targeted pruning rate, but has also a regularization effect which controls the dynamic of the learned weights and thereby the generalization properties of g_θ as shown subsequently and later in experiments.

Fitting a targeted distribution. Considering Q as the observed distribution of the latent weights $\{\hat{\mathbf{W}}^\ell\}_\ell$, and P the targeted one, our goal is to reduce the discrepancy between P and Q using a Kullback-Leibler Divergence (KLD) loss

$$D_{KL}(P||Q) = \int_\Omega P(\hat{W})(\log P(\hat{W}) - \log Q(\hat{W}))\, d\hat{W}. \quad (2)$$

Note that the analytic form of the above equation is known on the widely used probability density functions (PDFs), whilst for general (arbitrary) probability distributions, the exact form is not always known and requires sampling. Hence, we consider instead a discrete variant of this loss as well as P and Q; examples of targeted distributions P are given in Fig. 2 while the observed (and also differentiable) one Q is based on a relaxed variant of histogram estimation. Let $\{q_1, \dots, q_K\}$ denote a K-bin quantization of Ω (in practice $K = 100$), the k-th entry of Q is defined as

$$Q(\hat{W} = q_k) \propto \sum_{\ell=1}^{L-1} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_{\ell+1}} \exp\Big\{ -(\hat{\mathbf{W}}^\ell_{i,j} - q_k)^2/\beta_k^2 \Big\}, \quad (3)$$

here β_k is a scaling factor that controls the smoothness of the exponential function; larger values of β_k result into oversmoothed histogram estimation while a sufficiently (not very) small β_k leads to a surrogate histogram estimation close to the actual discrete distribution of Q. In practice, β_k is set to $(q_{k+1} - q_k)/2$; with this setting, one may replace $\propto$ (in Eq. 3) with an equality as the partition function of Q — i.e., $\sum_{k=1}^K Q(\hat{W} = q_k)$ — reaches almost one in practice.

Budget-aware pruning. Let $F_{\hat{W}}(a) = P(\hat{W} \leq a)$ be the cumulative distribution function (CDF) of $P(\hat{W})$. For any given pruning rate r, one may find the threshold a of the parametrization $\psi_{a,\sigma}$ as

$$a(r) = F_{\hat{W}}^{-1}(r). \quad (4)$$

The above function, known as the quantile, defines the pruning threshold a on the targeted distribution P (and equivalently on the observed one Q thanks to the KLD loss) which guarantees that only a fraction $(1 - r)$ of the total weights are kept (i.e, nonzero) when applying the band-stop parametrization in Eq. 1. Note that the quantile at any given pruning rate r, can either be empirically evaluated on discrete random variables or can be analytically derived on the widely used PDFs (see table. 1).

Distributions	PDF $P(\hat{W})$	Quantile $a(r) = F_{\hat{W}}^{-1}(r)$
Uniform	$\frac{1}{T}$	$\frac{r}{T}$
Gaussian	$\frac{1}{\sigma\sqrt{2\pi}} \exp\big\{ -\frac{1}{2}\big(\frac{\hat{W}-\mu}{\sigma}\big)^2\big\}$	$\mu + \sigma\sqrt{2}\mathbf{erf}^{-1}(2r-1)$
Laplace	$\frac{1}{2b} \exp\big\{ -\frac{\lvert\hat{W}-b\rvert}{b}\big\}$	$\begin{cases} \mu + b\ \log(2r) & \text{if } r \leq \frac{1}{2} \\ \mu - b\ \log(2-2r) & \text{otherwise} \end{cases}$

Table 1. Different standard PDFs and the underlying quantile functions.

Considering the above budget implementation, pruning is achieved using a global loss as a combination of a cross-entropy term $\mathcal{L}_e$, and the KLD loss D_{KL} (which controls weight distribution and hence guarantees the targeted pruning rate/budget depending on the setting of a in $\psi_{a,\sigma}$ as shown in Eq. 4) resulting into

$$\min_{\{\hat{\mathbf{W}}^\ell\}_\ell} \mathcal{L}_e\big(\{\hat{\mathbf{W}}^\ell \odot \psi(\hat{\mathbf{W}}^\ell)\}_\ell\big) \ + \ \lambda\, D_{KL}(P||Q), \quad (5)$$

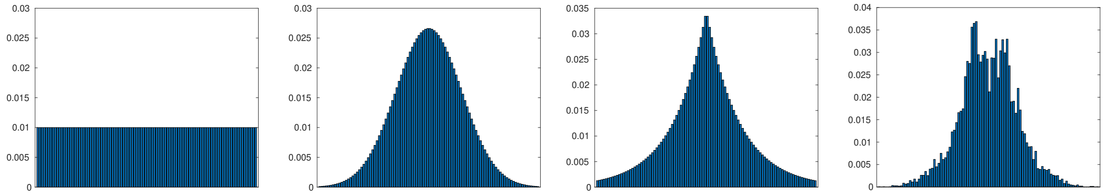

Figure 2. The first 3 figures correspond to targeted (uniform, gaussian and laplace) distributions. The 4th figure shows the actual weight distribution of the heavy/unpruned GCN which resembles to gaussian/laplacian. This may explain the good performances when the gaussian target is used particularly at low/mid pruning rates where the unpruned and pruned networks are more similar. At high pruning rates, laplace is better (see table 4).

here λ is sufficiently large (overestimated to $\lambda = 10$ in practice), so Eq. 5 focuses on implementing the budget and also constraining the pruning rate to reach r. As training evolves, D_{KL} reaches its minimum and stabilizes while the gradient of the global loss becomes dominated by the gradient of $\mathcal{L}_e$, and this maximizes further the classification performances. Note that the impact of $D_{KL}(P||Q)$ in Eq. 5 has some similarities and differences w.r.t. the usual regularizers particularly ℓ_0, ℓ_1 and ℓ_2. Whilst these three regularizers favor respectively uniform, laplace and gaussian distributions in Q, there is no guarantee that Q will *exactly match* an a priori distribution, so implementing any targeted pruning rate will require adding explicit (and difficult to solve) budget criteria or overtrying different mixing hyperparameteres on these regularizers. In contrast, as Q is constrained in $D_{KL}(P||Q)$, the Band-pass mechanism in Eq. 1 makes reaching any targeted pruning rate easily feasible. Note also that this Band-pass mechanism allows implementing a *partial* weight ranking — through the K-bins of the distribution Q — in a differentiable manner. In other words, as training evolves, this approach jointly trains network topology $\psi_{a,\sigma}(\hat{\mathbf{W}}^\ell)$ and weights $\hat{\mathbf{W}}^\ell$ by (i) changing the *bin assignment* of $\hat{\mathbf{W}}^\ell$ in Q, and by (ii) *activating and deactivating* these weights through $\psi_{a,\sigma}$ while maximizing generalization and satisfying exactly the targeted budget.

4.3. MRMP: Multi-Rate Magnitude Pruning

The aforementioned formulation is already effective (see later experiments), however, it requires rerunning a complete optimization, for any update of the pruning rate which is time and memory demanding. In what follows, we introduce a framework that allows training GCN instances with *shared latent weights* which achieve optimal performances at multiple pruning rates without retraining.

The guiding principle of our method relies on sharing the latent weights $\{\hat{\mathbf{W}}^\ell\}_\ell$ through multiple GCN instances defined by the hyperparameter $\{a(r)\}_r$ in $\psi_{a,\sigma}$. This makes it possible to reduce not only training time (as a unique training session is necessary for all the pruning rates) but also the memory footprint as the number of latent weights remains unchanged. Furthermore, training multiple GCN instances on top of shared latent parameters makes each instance as a proxy task to the other GCN learning tasks, and this turns out to improve generalization as shown in experiments. Last but not least, this also allows obtaining optimal pruned networks (at unseen pruning rates) instantaneously without retraining. In order to achieve these goals, we propose an updated loss as

$$\min_{\{\hat{\mathbf{W}}^\ell\}_\ell} \sum_r \mathcal{L}_e\big(\{\hat{\mathbf{W}}^\ell \odot \psi_{a(r),\sigma}(\hat{\mathbf{W}}^\ell)\}_\ell\big) \ + \ \lambda \, D_{KL}(P||Q), \tag{6}$$

here the right-hand side term remains unchanged and it again seeks to constrain the latent weights to fit a targeted distribution. In contrast, the left-hand side (cross entropy) term, is evaluated through multiple pruning rates using the shared latent weights; hence only $\psi_{a(r),\sigma}$ intervenes in order to prune connections according to the targeted rates. Once the optimization achieved, GCN instances may be obtained at any fixed pruning rates (including unseen ones) by multiplying each weight tensor with the binary mask tensor as $\{\hat{\mathbf{W}}^\ell \odot \psi_{a,\sigma}(\hat{\mathbf{W}}^\ell)\}_\ell$ here a is again obtained using Eq. 4 and thanks to the KLD criterion in Eq.6. It's worth noticing that the computational complexity of the above formulation is highly efficient (compared to running multiple independent instances of pruning); indeed, the gradient of the KLD term is exactly the same while all the gradients of the left-hand side terms (w.r.t. the GCN output) have similar analytic forms and their evaluation for different r (either during forward and backward steps of backpropagation) could be batched and efficiently vectorized. In practice, we observe a slight overhead of MRMP against SRMP (Single Rate Magnitude Pruning), in forward and backward steps, even when one hundred pruning rates (r) are jointly considered for MRMP.

5. Experiments

In this section, we evaluate the performance of our baseline and pruned GCNs on the task of skeleton-based recognition using two challenging skeleton datasets; SBU Interaction (Yun et al., 2012) and First-Person Hand Action

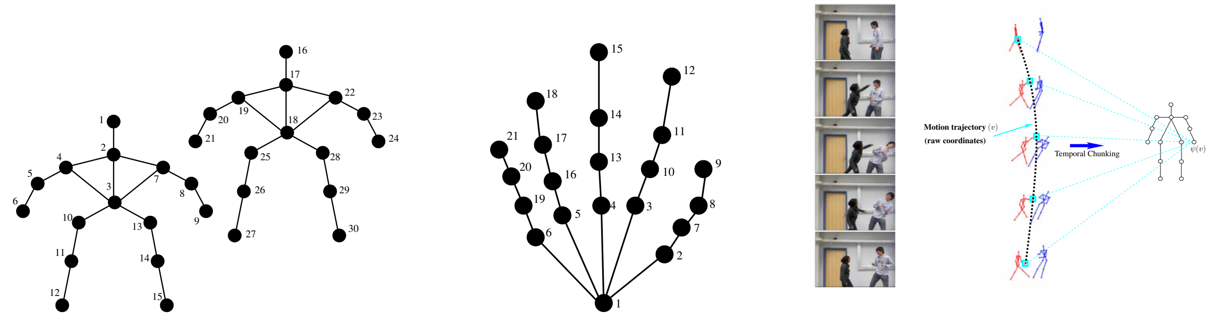

Figure 3. This figure shows original skeletons (left) on the SBU and (middle) the FPHA datasets. (Right) this figure shows the whole keypoint tracking and description process.

(FPHA) (Garcia-Hernando et al., 2018). The goal is to study the performance of our lightweight GCN design and its comparison against staple pruning techniques as well as the related work.

Dataset description. SBU is an interaction dataset acquired (under relatively well controlled conditions) using the Microsoft Kinect sensor; it includes in total 282 moving skeleton sequences (performed by two interacting individuals) belonging to 8 categories: “approaching”, “departing”, “pushing”, “kicking”, “punching”, “exchanging objects”, “hugging”, and “hand shaking”. Each pair of interacting individuals corresponds to two 15 joint skeletons and each joint is encoded 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 (Yun et al., 2012) (i.e., train-test split).
The FPHA dataset includes 1175 skeletons belonging to 45 action categories which are performed by 6 different individuals in 3 scenarios. In contrast to SBU, action categories are highly variable with inter and intra subject variability including style, speed, scale and viewpoint. Each skeleton includes 21 hand joints and each joint is again encoded with a sequence of its 3D coordinates across video frames. We evaluate the performance of our method using the 1:1 setting proposed in (Garcia-Hernando et al., 2018) with 600 action sequences for training and 575 for testing. In all these experiments, we report the average accuracy over all the classes of actions.

Skeleton normalization. Let $S^t = \{p_1^t, \dots, p_n^t\}$ denote the 3D skeleton coordinates at frame t. Without a loss of generality, we consider a particular order so that p_1^t, p_2^t and p_3^t correspond to three reference joints (e.g., neck, left shoulder and right shoulder); as shown in Fig. 3, this corresponds to joints 2, 4 and 7 for SBU and 1, 3 and 5 for FPHA. As the relative distance between these 3 joints is stable w.r.t. any motion, these 3 joints are used in order to estimate the rigid motion (similarity transformation) for skeleton normalization; see also (Meshry et al., 2016). Each graph sequence is processed in order to normalize its 3D coordinates using a similarity transformation; the translation parameters $\mathbf{t} = (t_x, t_y, t_z)$ of this transformation correspond to the shift that makes the reference point $(p_2^0 + p_3^0)/2$ coincide with the origin while the rotation parameters $(\theta_x, \theta_y, \theta_z)$ are chosen in order to make the plane formed by p_1^0, p_2^0 and p_3^0 coplanar with the x-y plane and the vector $p_2^0 - p_3^0$ colinear with the x-axis. Finally, the scaling γ of this similarity is chosen to make the $\|p_2^0 - p_3^0\|_2$ constant through all the action instances. Hence, each normalized joint is transformed as $\hat{p}_i^t = \gamma(p_i^t - \mathbf{t})R_x(\theta_x)R_y(\theta_y)R_z(\theta_z)$ with R_x, R_y, R_z being rotation matrices along the x, y and z axes respectively.

Input graphs. Considering a sequence of normalized skeletons $\{S^t\}_t$, each joint sequence $\{\hat{p}_j^t\}_t$ in these skeletons defines a labeled trajectory through successive frames (see Fig. 3-right). Given a finite collection of trajectories, we consider the input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node $v_j \in \mathcal{V}$ corresponds to the labeled trajectory $\{\hat{p}_j^t\}_t$ and an edge $(v_j, v_i) \in \mathcal{E}$ exists between two nodes iff the underlying trajectories are spatially neighbors. Each trajectory (i.e., node in $\mathcal{G}$) is processed using *temporal chunking*: first, the total duration of a sequence (video) is split into M equally-sized temporal chunks ($M = 4$ in practice), then the normalized joint coordinates $\{\hat{p}_j^t\}_t$ of the trajectory v_j are assigned to the M chunks (depending on their time stamps) prior to concatenate the averages of these chunks; this produces the description of v_j (again denoted as $\phi(v_j) \in \mathbb{R}^s$ with $s = 3 \times M$) and $\{\phi(v_j)\}_j$ constitutes the raw signal of nodes in a given sequence. Note that two trajectories v_j and v_i, with similar joint coordinates but arranged differently in time, will be considered as very different when using temporal chunking. Note that temporal chunking produces discriminant raw descriptions that preserve the temporal structure of trajectories while being *frame-rate* and *duration* agnostic.

Method	Accuracy (%)
Raw Position (Yun et al., 2012)	49.7
Joint feature (Ji et al., 2014)	86.9
CHARM (Li et al., 2015)	86.9
H-RNN (Du et al., 2015)	80.4
ST-LSTM (Liu et al., 2016)	88.6
Co-occurrence-LSTM (Zhu et al., 2016b)	90.4
STA-LSTM (Song et al., 2017)	91.5
ST-LSTM + Trust Gate (Liu et al., 2016)	93.3
VA-LSTM (Zhang et al., 2017a)	97.6
GCA-LSTM (Liu et al., 2017a)	94.9
Riemannian manifold. traj (Kacem et al., 2018)	93.7
DeepGRU (Maghoumi & LaViola, 2019)	95.7
RHCN + ACSC + STUFE (Li et al., 2020)	98.7
Our baseline GCN	98.4

Table 2. Comparison of our baseline GCN against related work on the SBU database.

Implementation details & baseline GCNs. We trained the GCN networks end-to-end using the Adam optimizer (Kingma & Ba, 2014) for 2,700 epochs with a batch size equal to 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 our networks; when this speed increases (resp. decreases), $\nu(t)$ decreases as $\nu(t) \leftarrow \nu(t-1) \times 0.99$ (resp. increases as $\nu(t) \leftarrow \nu(t-1)/0.99$). All these experiments are run on a GeForce GTX 1070 GPU device (with 8 GB memory) and neither dropout nor data augmentation are used. The architecture of our baseline GCN includes an attention layer of 1 head on SBU (resp. 16 heads on FPHA) applied to skeleton graphs whose nodes are encoded with 8-channels (resp. 32 for FPHA), followed by a convolutional layer of 32 filters for SBU (resp. 128 filters for FPHA), and a dense fully connected layer and a softmax layer. The initial network for SBU is not very heavy, its number of parameters does not exceed 15,320, and this makes its pruning challenging as many connections will be isolated (not contributing in the evaluation of the network output). In contrast, the initial network for FPHA is relatively heavy (for a GCN) and its number of parameters reaches 2 millions. As shown in tables. 2 and 3, both GCNs are accurate compared to the related work on the SBU/FPHA benchmarks. Considering these GCN baselines, our goal is to make them highly lightweight while making their accuracy as high as possible.

Performances, Comparison & Ablation. Tables 4-5 and Fig. 4 show a comparison and an ablation study of our method both on SBU and FPHA datasets. First, from the results in Fig.4-top, we see the alignment between the targeted pruning rates and the observed ones when using the formulation in Eq. 5 for different PDFs; the quantile functions of the gaussian and laplace PDFs allow implementing *fine-steps* of the targeted pruning rates r particularly when r is large. In contrast, the quantile functions of the gaussian and laplace PDFs are coarse around mid r values (55%).

Method	Color	Depth	Pose	Accuracy (%)
2-stream-color (Feichtenhofer et al., 2016)	✓	✗	✗	61.56
2-stream-flow (Feichtenhofer et al., 2016)	✓	✗	✗	69.91
2-stream-all (Feichtenhofer et al., 2016)	✓	✗	✗	75.30
HOG2-dep (Ohn-Bar & Trivedi, 2014)	✗	✓	✗	59.83
HOG2-dep+pose (Ohn-Bar & Trivedi, 2014)	✗	✓	✓	66.78
HON4D (Oreifej & Liu, 2013)	✗	✓	✗	70.61
Novel View (Rahmani & Mian, 2016)	✗	✓	✗	69.21
1-layer LSTM (Zhu et al., 2016b)	✗	✗	✓	78.73
2-layer LSTM (Zhu et al., 2016b)	✗	✗	✓	80.14
Moving Pose (Zanfir et al., 2013)	✗	✗	✓	56.34
Lie Group (Vemulapalli et al., 2014)	✗	✗	✓	82.69
HBRNN (Du et al., 2015)	✗	✗	✓	77.40
Gram Matrix (Zhang et al., 2016)	✗	✗	✓	85.39
TF (Garcia-Hernando & Kim, 2017)	✗	✗	✓	80.69
JOULE-color (Hu et al., 2015)	✓	✗	✗	66.78
JOULE-depth (Hu et al., 2015)	✗	✓	✗	60.17
JOULE-pose (Hu et al., 2015)	✗	✗	✓	74.60
JOULE-all (Hu et al., 2015)	✓	✓	✓	78.78
Huang et al. (Huang & Van Gool, 2017)	✗	✗	✓	84.35
Huang et al. (Huang et al., 2018b)	✗	✗	✓	77.57
HAN (Liu et al., 2021)	✗	✗	✓	85.74
Our baseline GCN	✗	✗	✓	86.43

Table 3. Comparison of our baseline GCN against related work on the FPHA database.

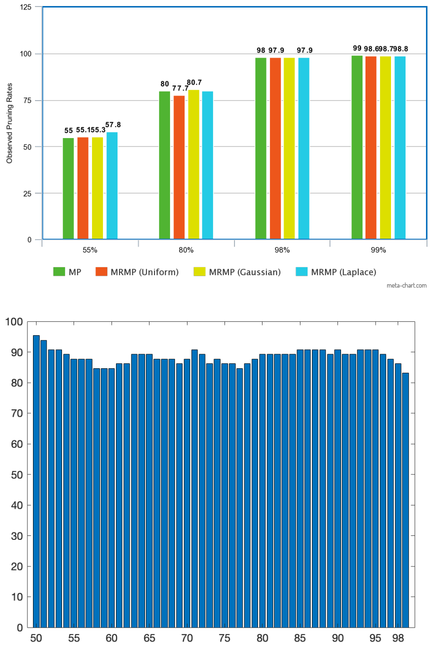

Figure 4. Performances on the SBU dataset: (Top) Fixed and observed pruning rates when different PDFs are used in the KLD regularizer. (Bottom) Performances for different (seen and unseen) pruning rates of MRMP; again seen pruning rates (during training) correspond to 50, 55, 60, 65, 70, 75, 80, 85, 90, 95 and 98% while unseen ones correspond to all the remaining pruning rates in $[50, 100[$. Better to zoom the file.

Pruning rates	Accuracy (%)	Observation
0%	**98.40**	Baseline GCN
70%	93.84	Band-stop Weight Param.
55%	89.23	MP
	83.07	SRMP: MP+KLD (Uniform)
	93.84	MRMP: MP+KLD (Uniform)+MR
	81.53	SRMP: MP+KLD (Gaussian)
	90.76	MRMP: MP+KLD (Gaussian)+MR
	80.00	SRMP: MP+KLD (Laplace)
	89.23	MRMP: MP+KLD (Laplace)+MR
80%	87.69	MP
	84.61	SRMP: MP+KLD (Uniform)
	89.23	MRMP: MP+KLD (Uniform)+MR
	78.46	SRMP: MP+KLD (Gaussian)
	89.23	MRMP: MP+KLD (Gaussian)+MR
	80.00	SRMP: MP+KLD (Laplace)
	90.76	MRMP: MP+KLD (Laplace)+MR
98%	69.23	MP
	78.46	SRMP: MP+KLD (Uniform)
	86.15	MRMP: MP+KLD (Uniform)+MR
	47.69	SRMP: MP+KLD (Gaussian)
	86.15	MRMP: MP+KLD (Gaussian)+MR
	60.00	SRMP: MP+KLD (Laplace)
	80.00	MRMP: MP+KLD (Laplace)+MR
Comparative (regularization-based) pruning		
98%	55.38	MP+ℓ_0-reg.
	73.84	MP+ℓ_1-reg.
	61.53	MP+Entropy-reg.
	75.38	MP+Cost-aware-reg.

Table 4. Detailed performances and ablation study, for different pruning rates, PDFs used for KLD and also MR; MR stands for Multi-Rate pruning.

Pruning rates	Accuracy (%)	Observation
0%	**86.43**	Baseline GCN
50%	85.56	Band-stop Weight Param.
55%	87.82	MP
	87.82	SRMP: MP+KLD (Uniform)
	88.92	MRMP: MP+KLD (Uniform)+MR
	88.52	SRMP: MP+KLD (Gaussian)
	89.58	MRMP: MP+KLD (Gaussian)+MR
	87.65	SRMP: MP+KLD (Laplace)
	88.48	MRMP: MP+KLD (Laplace)+MR
80%	86.78	MP
	85.91	SRMP: MP+KLD (Uniform)
	86.43	MRMP: MP+KLD (Uniform)+MR
	87.47	SRMP: MP+KLD (Gaussian)
	88.93	MRMP: MP+KLD (Gaussian)+MR
	86.95	SRMP: MP+KLD (Laplace)
	87.89	MRMP: MP+KLD (Laplace)+MR
98%	60.34	MP
	70.26	SRMP: MP+KLD (Uniform)
	71.18	MRMP: MP+KLD (Uniform)+MR
	70.60	SRMP: MP+KLD (Gaussian)
	74.73	MRMP: MP+KLD (Gaussian)+MR
	70.80	SRMP: MP+KLD (Laplace)
	72.97	MRMP: MP+KLD (Laplace)+MR
Comparative (regularization-based) pruning		
98%	64.69	MP+ℓ_0-reg.
	70.78	MP+ℓ_1-reg.
	67.47	MP+Entropy-reg.
	69.91	MP+Cost-aware-reg.

Table 5. Same caption as tab 4 but for FPHA dataset.

Second, according to tables 4-5, when training is achieved with only the band-stop weight parametrization (i.e., $\lambda = 0$ in Eq. 5), performances are close to the initial heavy GCNs (particularly on FPHA), with less parameters[1] as this produces a regularization effect similar to (Wan et al., 2013). Third, we observe a positive impact when the KLD term (in Eq. 5) is used both with single and multi-rate magnitude pruning (resp. SRMP and MRMP) with an extra-advantage of MRMP against SRMP; again MP+KLD in tables 4-5 corresponds to SRMP and MP+KLD+MR refers to MRMP. Extra comparison of KLD against other regularizers shows the substantial gain of our method. Indeed, KLD is compared against different alternatives plugged in Eq. 5 instead of KLD, namely ℓ_0 (Louizos et al., 2017), ℓ_1 (Koneru & Vasudevan, 2019), entropy (Wiedemann et al., 2019) and ℓ_2-based cost (Lemaire et al., 2019). From the observed results, the impact of KLD is substantial for different PDFs and for equivalent pruning rate (namely 98%). Note that when alternative regularizers are used, multiple settings (trials) of the underlying hyperparameter λ (in Eq. 5) are considered prior to reach the targeted pruning 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.4-bottom shows the generalization performance of MRMP from seen to unseen pruning rates. In these results, only a few pruning rates are used for MRMP (namely 50, 55, 60, 65, 70, 75, 80, 85, 90, 95 and 98%) and all the remaining rates in $[50, 100[$ are used for instantaneous pruning without retraining. We observe stable and high performances *from seen to unseen* pruning rates and this shows that MRMP is able to extrapolate highly accurate GCNs (even) on unseen pruning rates.

6. Conclusion

We introduce in this paper a novel lightweight GCN design based on multi-rate magnitude pruning. The strength of the proposed method resides in its ability to constrain the probability distribution of the learned GCNs to match an a priori distribution, and this allows implementing, via a band-stop mechanism, any given targeted pruning rate while also enhancing the generalization performances of the resulting lightweight GCNs. Besides, our method allows training multiple network instances simultaneously, on top of shared latent weights, at different pruning rates and extrapolating GCNs at unseen rates without retraining their weights. Experiments conducted on the challenging task of skeleton-based recognition shows a significant gain of our method. As a future work, we are currently investigating the extension of the current approach to other networks and databases.

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

References

Ahn, S., Hu, S. X., Damianou, A., Lawrence, N. D., and Dai, Z. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171, 2019.

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

Cai, H., Gan, C., Wang, T., Zhang, Z., and Han, S. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299, 2017.

Carreira-Perpinán, M. A. and Idelbayev, Y. “learning-compression” algorithms for neural net pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8532–8541, 2018.

Chen, X., Guo, H., Wang, G., and Zhang, L. Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 2881–2885. IEEE, 2017.

Chen, Y., Zhao, L., Peng, X., Yuan, J., and Metaxas, D. N. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. *arXiv preprint arXiv:1907.08871*, 2019.

Chung, F. R. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

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

Feichtenhofer, C., Pinz, A., and Zisserman, A. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1933–1941, 2016.

Garcia-Hernando, G. and Kim, T.-K. Transition forests: Learning discriminative temporal transitions for action recognition and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 432–440, 2017.

Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. 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*, pp. 409–419, 2018.

Gordon, A., Eban, E., Nachum, O., Chen, B., Wu, H., Yang, T.-J., and Choi, E. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1586–1595, 2018.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 729–734. IEEE, 2005.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

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

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

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

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

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

He, Y., Kang, G., Dong, X., Fu, Y., and Yang, Y. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018a.

He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., and Han, S. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 784–800, 2018b.

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

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hou, J., Wang, G., Chen, X., Xue, J.-H., Zhu, R., and Yang, H. Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

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

Huang, G., Liu, S., Van der Maaten, L., and Weinberger, K. Q. Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2752–2761, 2018a.

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

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

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

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

Jian, W., Zhou, Y., and Liu, H. Densely connected convolutional network optimized by genetic algorithm for fingerprint liveness detection. *IEEE Access*, 9:2229–2243, 2020.

Jiu, M. and Sahbi, H. Nonlinear deep kernel learning for image annotation. *IEEE Transactions on Image Processing*, 26(4):1820–1832, 2017.

Jiu, M. and Sahbi, H. Deep representation design from deep kernel networks. *Pattern Recognition*, 88:447–457, 2019.

Kacem, A., Daoudi, M., Amor, B. B., Berretti, S., and Alvarez-Paiva, J. C. 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.

Ke, Q., Bennamoun, M., An, S., Sohel, F., and Boussaid, F. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3288–3297, 2017.

Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., and Bhowmik, A. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1–10, 2017.

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

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

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

Koneru, B. N. G. and Vasudevan, V. Sparse artificial neural networks using a novel smoothed lasso penalization. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(5):848–852, 2019.

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

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

Lee, I., Kim, D., Kang, S., and Lee, S. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1012–1020, 2017.

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

Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

Li, B., Li, X., Zhang, Z., and Wu, F. Spatio-temporal graph routing for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8561–8568, 2019.

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

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

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

Li, S., Jiang, T., Huang, T., and Tian, Y. 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*, pp. 586–594, 2020.

Li, W., Wen, L., Chuah, M. C., and Lyu, S. Category-blind human action recognition: A practical recognition system. In *Proceedings of the IEEE international conference on computer vision*, pp. 4444–4452, 2015.

Liu, J., Shahroudy, A., Xu, D., and Wang, G. 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*, pp. 816–833. Springer, 2016.

Liu, J., Wang, G., Duan, L.-Y., Abdiyeva, K., and Kot, A. C. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2017a.

Liu, J., Wang, G., Hu, P., Duan, L.-Y., and Kot, A. C. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1647–1656, 2017b.

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

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

Liu, M. and Yuan, J. Recognizing human actions as the evolution of pose estimation maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1159–1168, 2018.

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

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017d.

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

Maghoumi, M. and LaViola, J. J. 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*, pp. 16–31. Springer, 2019.

Mazari, A. and Sahbi, H. Deep temporal pyramid design for action recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2077–2081. IEEE, 2019a.

Mazari, A. and Sahbi, H. Mlgcn: Multi-laplacian graph convolutional networks for human action recognition. In *The British Machine Vision Conference (BMVC)*, 2019b.

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

Micheli, A. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., and Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Nguyen, X. S., Brun, L., Lézoray, O., and Bougleux, S. A neural network based on spd manifold learning for skeleton-based hand gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12036–12045, 2019.

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

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

Ohn-Bar, E. and Trivedi, M. M. 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.

Oreifej, O. and Liu, Z. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 716–723, 2013.

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

Q. De Smedt, H. W. and Vandeborre, J.-P. Skeleton-based dynamic hand gesture recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Las Vegas, NV, United states, june, pp 1206-1214*, 2016.

Rahmani, H. and Mian, A. 3d action recognition from novel viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1506–1515, 2016.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

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

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

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

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

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

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

Sahbi, H. Phase-field models for lightweight graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4643–4649, 2023.

Sahbi, H. and Fleuret, F. Kernel methods and scale invariance using the triangular kernel. Technical report, INRIA, 2004.

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

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

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

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

Shi, L., Zhang, Y., Cheng, J., and Lu, H. Non-local graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1805.07694*, 1(2):3, 2018.

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. 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.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Vemulapalli, R., Arrate, F., and Chellappa, R. 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*, pp. 588–595, 2014.

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

Wang, H. and Wang, L. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 499–508, 2017.

Wang, L. and Sahbi, H. Directed acyclic graph kernels for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3168–3175, 2013.

Wang, L. and Sahbi, H. Bags-of-daglets for action recognition. In *IEEE International Conference on Image Processing (ICIP)*, pp. 1550–1554. IEEE, 2014.

Wang, P., Li, W., Gao, Z., Zhang, J., Tang, C., and Ogunbona, P. O. Action recognition from depth maps using deep convolutional neural networks. *IEEE Transactions on Human-Machine Systems*, 46(4):498–509, 2015.

Wang, P., Li, Z., Hou, Y., and Li, W. Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the 24th ACM international conference on Multimedia*, pp. 102–106, 2016.

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

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

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

Weng, J., Liu, M., Jiang, X., and Yuan, J. Deformable pose traversal convolution for 3d action and gesture recognition. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 136–152, 2018.

Wiedemann, S., Marban, A., Müller, K.-R., and Samek, W. Entropy-constrained training of deep neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2019.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Xia, L., Chen, C.-C., and Aggarwal, J. K. View invariant human action recognition using histograms of 3d joints. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pp. 20–27. IEEE, 2012.

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

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

Yuan, F., Xia, G.-S., Sahbi, H., and Prinet, V. Mid-level features and spatio-temporal context for activity recognition. *Pattern Recognition*, 45(12):4182–4191, 2012.

Yun, K., Honorio, J., Chattopadhyay, D., Berg, T. L., and Samaras, D. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pp. 28–35. IEEE, 2012.

Zanfir, M., Leordeanu, M., and Sminchisescu, C. 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*, pp. 2752–2759, 2013.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE international conference on computer vision*, pp. 2117–2126, 2017a.

Zhang, S., Liu, X., and Xiao, J. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 148–157. IEEE, 2017b.

Zhang, X., Wang, Y., Gou, M., Sznaier, M., and Camps, O. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4498–4507, 2016.

Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4320–4328, 2018.

Zhang, Z., Cui, P., and Zhu, W. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):249–270, 2020.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016a.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016b.