

Are Spatial-Temporal Graph Convolution Networks for Human Action Recognition Over-Parameterized?

Jianyang Xie¹, Yitian Zhao², Yanda Meng³, He Zhao¹, Anh Nguyen¹, Yalin Zheng^{1*}

¹ University of Liverpool, UK. ² Ningbo Institute of Materials Technology and Engineering, CAS, China. ³ University of Exeter, UK.

{Jianyang.Xie, yzheng}@liverpool.ac.uk

Abstract

Spatial-temporal graph convolutional networks (ST-GCNs) showcase impressive performance in skeleton-based human action recognition (HAR). However, despite the development of numerous models, their recognition performance does not differ significantly after aligning the input settings. With this observation, we hypothesize that ST-GCNs are over-parameterized for HAR, a conjecture subsequently confirmed through experiments employing the lottery ticket hypothesis. Additionally, a novel sparse ST-GCNs generator is proposed, which trains a sparse architecture from a randomly initialized dense network while maintaining comparable performance levels to the dense components. Moreover, we generate multi-level sparsity ST-GCNs by integrating sparse structures at various sparsity levels and demonstrate that the assembled model yields a significant enhancement in HAR performance. Thorough experiments on four datasets, including NTU-RGB+D 60(120), Kinetics-400, and FineGYM, demonstrate that the proposed sparse ST-GCNs can achieve comparable performance to their dense components. Even with 95% fewer parameters, the sparse ST-GCNs exhibit a degradation of < 1% in top-1 accuracy. Meanwhile, the multi-level sparsity ST-GCNs, which require only 66% of the parameters of the dense ST-GCNs, demonstrate an improvement of > 1% in top-1 accuracy. The code is available at <https://github.com/davelailai/Sparse-ST-GCN>.

1. Introduction

Human action recognition (HAR) is an essential topic in video understanding and has a wide range of applications in intelligent surveillance systems[26], and human-computer interaction[27]. Recently, spatial-temporal graph convolution networks (ST-GCNs) [10, 11, 33, 40, 43, 51, 53] have gained significant popularity. Compared with other meth-

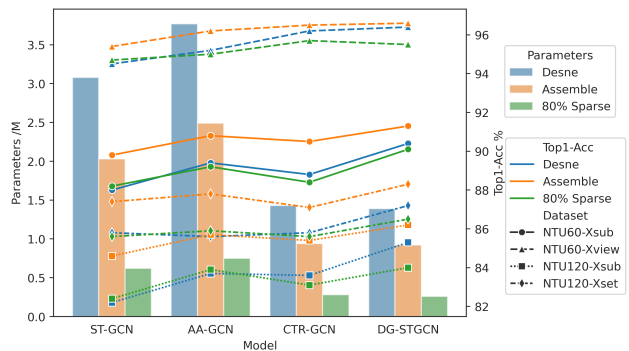


Figure 1. Model comparisons across the NTURGB+D benchmarks. ‘Dense’ means dense backbone, the deviations of Top-1 Acc across all the four NTURGB+D benchmarks remain below 2%, and the results are independent of the model size. ‘80% Sparse’ means 80% parameters are masked out, and the sparsity model shows a slight or no degradation in Top-1 Acc when compared with the corresponding dense model. ‘Assemble’ means multi-level sparsity ST-GCNs by incorporating the sparse structure at different sparsity levels, the results show a significant improvement compared with the corresponding dense model.

ods [3, 16, 21, 28, 31, 46, 48, 54] that rely on RGB image sequence [5] or optical flow analysis [44], the ST-GCNs use body pose and movement information directly, and can effectively capture the interactions between body joints, making it more robust against variations of camera viewpoint and video appearance.

Although there are many ST-GCN variants with their own merits, their recognition performance does not vary much after aligning the model setting [20]. As depicted in Figure 1, the deviations of top-1 Acc across all four NTURGB+D benchmarks [32, 37] remain below 2% when considering four different backbones [10, 18, 43, 51], and the result is independent of the model size. On the other hand, the model assembling has demonstrated promising improvements in HAR performance [1, 13–15, 29, 42, 47, 47]. However, these approaches typically yield larger mod-

*Corresponding Author

els with more parameters, posing challenges for deployment on devices with constrained hardware resources.

Observing these, we hypothesize that: 1) *ST-GCNs are potentially over-parameterized for HAR*. 2) *Sparse ST-GCNs may enhance the practical application of assembling models by reducing overall model size*. We then substantiated and extended these hypotheses through practical experimentation, and generated a multi-level sparsity ST-GCNs model. This model integrates identical backbones at multiple sparsity levels and demonstrates improved HAR performance without increasing parameters.

Inspired by lottery ticket hypothesis (LTH) [12, 22, 23, 36, 55], which states that an independent trainable sub-network exists in a randomly initialized dense model, whose performance is compared to the fully-trained network, we substantiate these hypotheses through practical experimentation in **Section 4.2**. Our findings reveal that sparse ST-GCNs extracted from fully-trained dense networks achieve comparable performance, even after masking 80% of the parameters. Additionally, we uncover two notable drawbacks of the sparse ST-GCNs obtained directly from the original LTH [23]: (1) the performance heavily relies on the initial network weights, and (2) a mask characterized by a high sparse ratio leads to a substantial loss of information, resulting in performance collapse.

Thus, to address the above drawbacks and obtain stable sparse ST-GCNs from random initial weights, we proposed a sparse ST-GCNs generator for the sub-networks extraction from dense ST-GCNs. Firstly, to reduce the reliance on the initial states, the masks in our method are randomly selected or predefined and then kept fixed, while the convolution kernel weights are subject to learning during training. This process enables the conversion of parameters within the randomly selected sub-network into a trainable state. Secondly, to alleviate the loss of information and retain the advantage of the dense network structure, an information compression loss based on the group lasso penalty [34] was proposed. This penalty loss was utilized to facilitate the information transformation from weights that have been masked to the intended sparse structures.

Furthermore, based on the proposed sparse ST-GCNs generator, we generated multi-level sparsity ST-GCNs by integrating the same models at various sparsity levels. The assembled models have demonstrated a substantial enhancement in HAR performance without parameter increases. This introduces a novel way to improve the overall performance of the assembling ST-GCNs while addressing the challenges of model size. By leveraging sparsity, this approach offers flexibility in model design and implementation, allowing for deployment in diverse scenarios and hardware constraints without compromising efficiency or accuracy.

We summarize our contributions as follows:

- We successfully substantiated that ST-GCNs exhibit a significant degree of over-parameterization by experiments based on the LTH. To the best of our knowledge, this is the first time to demonstrate a notable degree of over-parameterization in ST-GCNs.
- We proposed a sparse ST-GCNs generator, which can train stable sparse ST-GCNs from their randomly initialed dense network. Thorough experiments have verified its efficiency across various ST-GCN backbones. The obtained sparse ST-GCNs demonstrate comparable performance to their fully trained dense counterparts.
- Based on the sparse ST-GCNs generator, we demonstrated that multi-level sparsity ST-GCNs, incorporating backbones at different sparsity levels, enhance HAR performance while reducing parameters, offering a novel approach to optimizing assembling ST-GCNs' overall performance and addressing model size concerns.

2. Related Works

2.1. ST-GCNs for skeleton-based HAR

GCNs have been widely used for skeleton-based HAR [10, 11, 18, 33, 40, 43, 49–51, 53]. Yan *et al.* [51] introduced a pre-defined skeleton graph according to the human body's natural link and proposed the ST-GCN to capture the spatial and temporal patterns from the graph structure. Upon this baseline, some spatial adaptive graph generation methods based on no-local mechanisms were proposed to increase the flexibility of the skeleton graph structure [10, 11, 18, 40, 53]. Instead of only applying the fixed graph structure, these methods learned other adaptive graphs to boost the GCNs' representation ability. For instance, the 2S-AGCN [40] learned a data-driven adaptive graph for all feature channels, and CTR-GCN citechen2021channel learned an adaptive graph for each individual feature channel. Meanwhile, the multi-scale and shift GCN were proposed [11, 33] to address the over-smooth problem in graph long-distance transfer. In the temporal pattern, multi-scale temporal convolution was proposed to boost the information aggregation in temporal space [10, 18].

2.2. Lottery Ticket Hypothesis

First proposed by Frankle and Carbin [23], the lottery ticket hypothesis (LTH) states that there exists a sparse sub-network (called a winning ticket) in a randomly initialized dense network, whose performance is comparable to the fully-trained dense network. Furthermore, several extensions [2, 12, 24, 36, 55] have been proposed to enhance its effectiveness. These techniques have observed the existence of untrained sub-networks within randomly initialized convolutional neural networks (CNNs). Remarkably, these sub-networks can achieve comparable accuracy to a fully trained network, even without any updates to the net-

work’s weights. Chen *et al.* [9] firstly expended the LTH to the graph content. They introduced a unified GNN sparsification (UGS) framework, demonstrating the effectiveness of LTH across various GNN architectures. While LTH has found applications in various fields [6–9, 25, 35], our work is the first attempt to formulate the LTH in ST-GCNs perspective. Moreover, we extend the LTH to be specifically tailored for ST-GCNs while preserving the advantages of dense network architecture.

3. Method

3.1. Preliminaries

Notations. The human skeleton in a video is represented as a spatial-temporal graph and is denoted as $\mathcal{G} = (V, E_s, E_t, X)$, where V is the set of joints, X is the joints feature, E_s and E_t is the spatial and temporal graph respectively. Consider $V = \{v_{ti} | t = 1, \dots, T, i = 1, \dots, N\}$, where N is the number of body joints in each frame, and T is the number of video frames. Here, v_{ti} represents the n^{th} body joints in t^{th} frames. Let $X \in \mathbb{R}^{N \times T \times d}$ represent the joint coordinates as the node feature, where d is the feature dimension. As for E_s , an adjacent matrix $A \in \mathbb{R}^{N \times N}$ was defined to describe the overall spatial graph topology, and the edges are formulated as the intro-body connection. The E_t is constructed by connecting the same joints along consecutive frames. ST-GCNs can be divided into two parts: a spatial-GCN (S-GCN) works on the spatial graph $\mathcal{G}_s = (V, E_s, X)$, and a temporal-GCN (T-GCN) works on the $\mathcal{G}_t = (V, E_t, X)$.

Spatial-GCN. The S-GCN works on a spatial graph \mathcal{G}_s , and the main operation is to update the node features by aggregating information from its neighborhood within the same frame as Equation 1.

$$X_S = \Phi(X, E_s, \theta), \quad (1)$$

where X_S represents the outputs of S-GCN, and θ the parameters of the mapping function. To achieve this, rewriting the input features as $X = \{X^t \in \mathbb{R}^{N \times d} | t = 1, \dots, T\}$, and outputs $X_S = \{X_S^t \in \mathbb{R}^{N \times C} | t = 1, \dots, T\}$ of S-GCN can be formulated as Equation 2, where C is output feature channels.

$$X_S^t = f(AX^t, \theta), t = 1, \dots, T, \quad (2)$$

where f is an updating function with learnable parameters θ .

Temporal-GCN. The T-GCN works on the \mathcal{G}_t , and the key idea is to update the joints’ features at the current frame by aggregating the features from its K -neighbor frames as Equation 3.

$$X_T = \Psi(X, E_t, \omega), \quad (3)$$

where X_T represents the outputs of T-GCN, ω is the parameters of the mapping function Ψ . To achieve this, rewriting

$X = \{X^n \in \mathbb{R}^{T \times d} | n = 1, \dots, N\}$ as the input feature of T-GCN, the output $X_T = \{X_T^n \in \mathbb{R}^{T \times d} | n = 1, \dots, N\}$ of T-GCN can be formulated as Equation 4.

$$X_{T_t}^n = \sum_{k=-l}^l \omega_k * X_{t+k}^n, n = 1, \dots, N, t = 1, \dots, T, \quad (4)$$

where $X_{T_t}^n$ represent n^{th} point in the t^{th} frame, l is the window size of temporal convolution; $\omega \in \mathbb{R}^{2l-1}$ is the learnable weights for feature aggregating.

Spatial-Temporal GCN. A spatial-temporal GCN is generated by operating an S-GCN and a T-GCN sequentially as Equation 5.

$$X' = \Psi(X_S, E_t, \omega) = \Psi(\Phi(X, E_s, \theta), E_t, \omega). \quad (5)$$

During the dense ST-GCNs training, let $D = (x, y)$ be the training data, where x is the samples, and y is the corresponding labels. A dense ST-GCN is represented as $\mathcal{F}(x, \theta, \omega)$ with parameters (θ, ω) , and θ and ω are updated based on the Equation 6.

$$\theta', \omega' = \arg \min_{\theta, \omega} \mathcal{L}(\mathcal{F}(x, \theta, \omega), y), \quad (6)$$

where (θ', ω') is the optimized parameters for the networks and \mathcal{L} is the classification loss function.

3.2. Lottery Ticket Perspective of Sparse ST-GCNs

As illustrated in Figure 2 (a), a learnable mask $m = (m_s, m_t)$ is introduced, where m_s is the mask for the parameters θ in S-GCN, and m_t is the mask for the parameters ω in T-GCN. The process of sparse ST-GCNs can be formulated as Equation 7.

$$X' = \Psi\left(\Phi(X, E_s, \mathcal{S}(m_s) \odot \theta), E_t, \mathcal{S}(m_t) \odot \omega\right), \quad (7)$$

where \odot represents element-wise multiplication. \mathcal{S} is the binary operation that controls the sparsity level, $\mathcal{S}(m_s)$ and $\mathcal{S}(m_t)$ represent the binary mask that is utilized for sparse ST-GCNs generation. Thus, the sparse ST-GCNs can be trained following Equation 8

$$m'_s, m'_t = \arg \min_{m_s, m_t} \mathcal{L}\left(\mathcal{F}(x, \mathcal{S}(m_s) \odot \theta, \mathcal{S}(m_t) \odot \omega), y\right), \quad (8)$$

the threshold of \mathcal{S} is determined as the S^{th} smallest element within $m = [m_s, m_t]$, based on the degree of sparsity.

3.3. Sparse ST-GCNs Generator

Despite the feasibility of obtaining sparse ST-GCNs through the algorithm introduced in Section 3.2, two drawbacks remain apparent (see Section 4.2): (1) the performance heavily relies on the initial network weight, as the

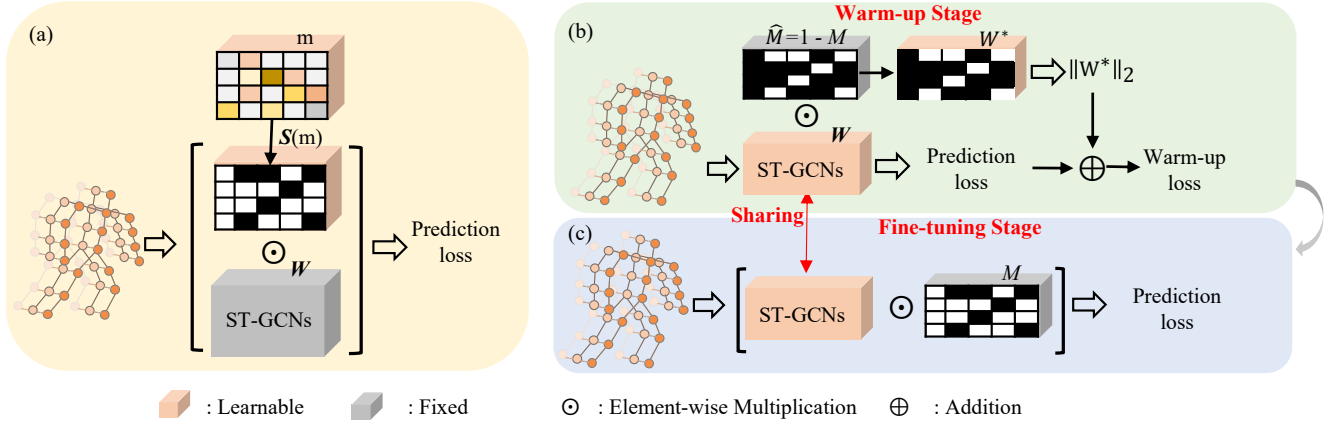


Figure 2. The framework of sparse ST-GCNs generator. (a) represents the Lottery Ticket Perspective of Sparse ST-GCNs, where the weights W in ST-GCNs are fixed, and a learnable mask m is learned. The $S(m)$ represents the binary operation. (b) and (c) represent the two stages of sparse ST-GCNs generator, where the weights W in ST-GCNs are learnable and the mask is pre-defined and kept fixed. During the Warm-up Stage in (b), all the parameters W are involved in training, and the masked parameters W^* with $M = 0$ are constrained by penalty loss. During the Fine-Tuning Stage in (c), only the parameters with $M = 1$ are engaged in training.

sparse ST-GCNs derived from randomly initialised exhibit considerable performance degradation in comparison to those extracted from pre-trained networks; (2) a mask characterized by a high sparse ratio leads to a loss of information, resulting in performance collapse.

In order to solve these two limitations, we proposed a sparse ST-GCNs generator. Different from Section 3.2, where the sparse ST-GCN is achieved by training masks for the network’s weights, our proposed sparse ST-GCNs generator maintains a static mask while allowing the network weights to be trainable. At the onset of the proposed sparse ST-GCNs generator training, a mask is randomly initialized and then kept constant, while the network’s weights are adapted to align with the intended sparse structure. Additionally, to leverage the advantages of the dense architecture, we divided the whole training process into two stages: warm-up and fine-tuning, and proposed an information compression penalty loss, denoted as L_c . During the warm-up stage, all network parameters (θ, ω) are optimized and the L_c is incorporated to extract relevant information from the masked portion and then direct it toward the designated sparse network. During the fine-tuning stage, only the parameters selected by the predefined mask are optimized.

Specifically, let $M = (M_s, M_t)$ be a randomly selected mask where M_s is the mask for parameters θ in S-GCN, and M_t is the mask for parameter ω in T-GCN, and the parameters masked by $M = 0$ as $W^* = (\theta^*, \omega^*)$, and those masked by $M = 1$ as $\hat{W}^* = (\hat{\theta}^*, \hat{\omega}^*)$. During the warm-up stage, L_c is applied to shrink W^* close to zero, signifying that these parameters have a limited impact on the network and can be eliminated to obtain the final sparse network. Considering there are multi-layers in the whole ST-GCNs

Algorithm 1 Sparse ST-GCN Generator

Require: (x, y) , $M = (M_s, M_t)$, warm-up, epoch, $W = (\theta, \omega)$

Ensure: M_s and M_t keep fixed

if epoch < warm-up **then**

$$W^* = W \odot (1 - M),$$

$$\theta', \omega' = \arg \min_{\theta, \omega} \left(\mathcal{L}(F(x, \theta, \omega), y) + \lambda \sum_{i=1}^n \|W_i^*\|_2 \right)$$

else

$$\theta', \omega' = \arg \min_{\theta, \omega} \mathcal{L}(F(x, M_s \odot \theta, M_t \odot \omega), y)$$

end if

structural (normally ten layers), with $W^* = (W_1^*, \dots, W_n^*)$ where n is the number of ST-GCN layers, a group lasso penalty over the entire set of W^* can be calculated by Equation 9

$$L_c = \|W^*\|_2 = \sum_{i=1}^n \|W_i^*\|_2 \quad (9)$$

Thus during the warm-up stage, the ST-GCNs can be trained following Equation 10

$$\theta', \omega' = \arg \min_{\theta, \omega} \left(\mathcal{L}(F(x, \theta, \omega), y) + \lambda \sum_{i=1}^n \|W_i^*\|_2 \right), \quad (10)$$

where λ is a hyper-parameter to balance the two loss items and set as 1 in our experiments.

During the fine-tuning, only the parameters in the intended sparse structure are involved in training, and the

sparse ST-GCNs can be trained by following Equation 11.

$$\theta', \omega' = \arg \min_{\theta, \omega} \mathcal{L}(\mathcal{F}(x, M_s \odot \theta, M_t \odot \omega), y), \quad (11)$$

where M_s and M_t are randomly initiated and kept fixed during the training. Thus, the overall pipeline of the sparse ST-GCN generator can be illustrated as Algorithm 1

4. Experiments

In this section, joint coordinates were utilized as input features. By extracting the subnet from the fully pre-trained dense ST-GCNs and verifying that the subnet yields comparable results to the original ST-GCNs, we confirmed *Hypotheses 1* regarding the over-parameterization of ST-GCNs. Then we verified the advantages of the proposed sparse ST-GCNs generator on four backbones [10, 19, 43, 51] using the NTU-RGBD dataset [32, 37]. Finally, we proved that multi-level sparsity ST-GCNs can enhance HAR performance with fewer parameters. Training details can be found in *Supplementary*

4.1. Datasets

Four commonly-used datasets were utilized: NTU RGB+D 60 [37], NTU RGC+D 120 [32], Kinetics-400 [5], and FineGYM [38]. The details can be found in *Supplementary*.

4.2. Evidence of Over-parameterized in ST-GCNs

To prove that the dense ST-GCNs are over-parameterized for HAR, we employed the LTH in fully trained ST-GCN [51] and CTR-GCN [10]. By extracting the subnet from the pre-trained dense network, we observed that the sparse network achieves a comparable result to the full dense network. As shown in Figure 3, even when masking out 80% of the parameters from the pre-trained dense network, there is a slight or no degradation in performance.

Let’s look at the two drawbacks of sparse ST-GCNs based on LTH. From Figure 3, it can be observed that the ST-GCN with the randomly initiated shows a significant degradation when compared with that pre-trained. On the other hand, As illustrated in Figure 3, both ST-GCN and CTR-GCN show a significant degradation in performance when the sparsity ratio exceeds 95%. These two drawbacks are effectively tackled by our sparse ST-GCNs generator.

4.3. Experiments for Sparse ST-GCNs Generator

Effectiveness of Sparse ST-GCNs Generator. To validate the effectiveness of the proposed sparse ST-GCNs generator, four different backbones [10, 19, 43, 51] were employed, considering two datasets [32, 37]. The networks were initialized randomly, and sparsity levels of 0.6, 0.8, 0.95, and 0.99 were applied. The performance of the sparse ST-GCNs made by the proposed generator is shown in Table 1. Across all four backbones, the sparse network with

80% of parameters masked out exhibits only a slight or no degradation in performance (< 1%).

When evaluating the performance of sparse ST-GCNs under model-size constraints, it is clear that the optimal sparsity level varies for each backbone. As illustrated in Table 1, the performance degradation in sparse structures is associated with the model size in a way. Specifically, as the model size increases, the optimal sparsity level also increases. Comparing the AA-GCN (model size: 3.47M) and DG-STGCN (model size: 1.31M), it can be observed that the performance shows a smaller decline in the AA-GCN than in the DG-STGCN when setting the sparse level as the same. For instance, at 99% sparsity, AA-GCN’s performance degradation in sparse structure is less than 2%, which is lower than DG-STGCN’s degradation (> 3%).

Comparing the performance of sparse ST-GCNs in each dataset. It is clear that the ideal sparsity level relies on the size of the dataset. As shown in Table 1, at the same sparse level, NTU60 demonstrates less performance degradation compared to NTU120 for all four backbones. Taking the CTR-GCN as an example, when setting the sparse level to 0.99, the performance on NTU60-Xsub experiences a 1.1% degradation, while both NTU120 witness a degradation of over 4%. This suggests that the optimal sparsity level tends to be lower in larger datasets.

Ablation of The Penalty Loss. In order to validate the influence of the penalty loss in the warm-up stage, the sparse ST-GCN was trained in two settings, with penalty loss and without penalty loss. The result is shown in Figure 4, it is evident that the inclusion of the information compression penalty loss in the sparse ST-GCNs generator consistently led to improved performance when compared to training without it. This effect was especially notable at higher sparse levels, where the performance gap became quite substantial. Certainly, considering the performance in NTU120-Xset, the difference in performance between training with the penalty and without it exceeds 10%. **Ablation of The Warm-up Stage.** To verify the importance of the warm-up stage, we trained the sparse ST-GCN generator using two strategies: train the sparse model without a warm-up stage and train the sparse model with a warm-up stage. The results are shown in Figure 5. It can be observed that the warm-up stage plays an important role in the sparse ST-GCNs generator, particularly at higher sparse levels. At the sparse levels below 0.8, the sparse ST-GCNs generator operates stably even without the warm-up stage, but there is a noticeable degradation in performance when compared to the configurations with the warm-up stage. Meanwhile, as sparsity levels increase significantly, some backbones (AA-GCN, CTR-GCN, and DG-STGCN) without the warm-up stage may suffer performance collapse, while this issue can be mitigated with the warm-up stage.

Based on the definition in CTR-GCN [10], four back-

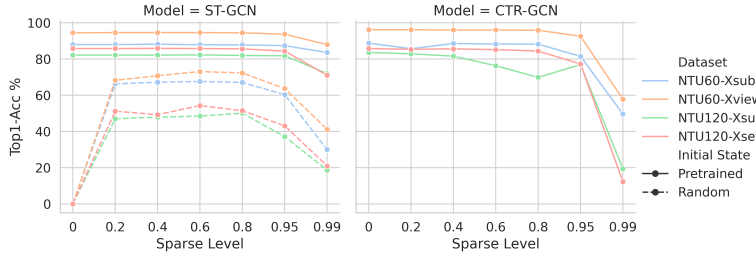


Figure 3. Results for LTH in sparse ST-GCNs. Sparse level means the percentage of masked parameters. The sparse ST-GCNs extracted from pre-trained dense networks can achieve results comparable to the fully trained dense network, which indicates the over-parametrization of the ST-GCNs. However, a notable degradation in the top-1 accuracy is observed at high sparse levels, and the sparse ST-GCN fails to compare favourably with the baseline in the case of the randomly initiated.

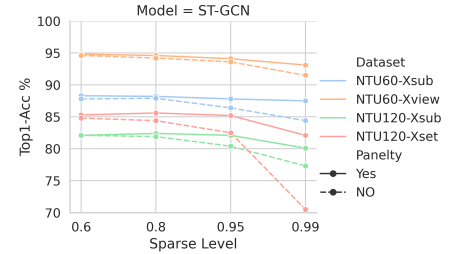


Figure 4. Ablation study of the penalty loss. ‘Yes’ means training the warm-up stage with the information compression penalty, and ‘No’ means without. It is obvious that the penalty loss had a positive influence on the performance of sparse ST-GCNs.

Table 1. Performance of the sparse ST-GCNs generator. The names of datasets are represented as NTU60-Xsub(60-sub), NTU60-Xview (60-view), NTU120-Xsub (120-sub), NTU120-Xset (120-set).

Model	ST-GCN (Paras:3.10M)					AA-GCN (Paras: 3.47M)				
	Base	0.6	0.8	0.95	0.99	Base	0.6	0.8	0.95	0.99
60-sub	88.0	88.3 (+0.3)	88.2 (+0.2)	87.8(-0.2)	87.5(-0.5)	89.4	89.3(-0.1)	89.3(-0.1)	89.5 (+0.1)	88.6(-0.8)
60-view	94.5	94.8 (+0.3)	94.6 (+0.1)	94.1(-0.4)	93.1(-1.4)	95.1	95.4 (+0.4)	95.1(+0.0)	95.0(-0.1)	94.1(-1.0)
120-sub	82.2	82.1(-0.1)	82.4 (+0.2)	82.1(-0.1)	80.1(-2.1)	83.7	83.0(-0.7)	83.9 (+0.2)	83.2(-0.5)	82.2(-1.5)
120-set	85.8	85.3(-0.5)	85.6(-0.2)	85.2(-0.6)	82.1(-3.7)	85.6	86.0 (+0.4)	85.8 (+0.2)	85.4(-0.2)	84.1(-1.5)
Model	CTR-GCN (Paras: 1.45M)					DG-STGCN (Paras: 1.31M)				
	Base	0.6	0.8	0.95	0.99	Base	0.6	0.8	0.95	0.99
60-sub	88.8	89.4 (+0.6)	88.4(-0.4)	88.7(-0.1)	86.7(-1.1)	90.4	90.5 (+0.1)	90.1(-0.3)	89.4(-1.0)	87.4(-3.0)
60-view	96.2	95.7(-0.5)	95.7(-0.5)	95.4(-0.8)	92.8(-3.4)	96.4	95.8(-0.6)	95.5(-0.9)	94.9(-1.5)	91.4(-5.0)
120-sub	83.6	83.4(-0.2)	83.1(-0.5)	83.1(-0.5)	79.2(-4.4)	85.3	84.7(-0.6)	84.0(-1.3)	83.2(-3.2)	80.7(-4.6)
120-set	85.8	84.9(-0.9)	85.3(-0.5)	85.2(-0.6)	80.7(-5.1)	87.2	86.1(-1.1)	86.5(-0.7)	86.2(-1.0)	82.3(-4.9)

bones can be divided into two types: the graph adaptive-based method (AA-GCN, CTR-GCN, and DG-STGCN) where the skeleton graph is adaptable based on the no-local mechanism, and the graph fixed-based method (ST-GCN), where the skeleton graph is predefined and fixed. Considering performance in the context of model type, it is notable that without a warm-up stage, models employing adaptive skeleton graphs are prone to performance collapse under high sparsity levels. For instance, both sparse CTR-GCN and sparse DG-STGCN encounter failure when the sparsity level surpasses 0.8. However, with the warm-up stage, these sparse ST-GCNs achieve results comparable to those of the fully trained dense network, even with 95% parameters masked out. Meanwhile, in the case of the graph fixed-based method (ST-GCN), the introduction of a warm-up stage resulted in a consistent increase in performance compared to training without a warm-up stage. This effect was particularly pronounced at higher sparse levels, where the performance gap became substantial.

4.4. Experiments for Multi-level Sparsity ST-GCNs

Inspired by the multi-stream fusion framework [42], we constructed a multi-level sparsity ST-GCNs incorporating the backbones in multiple sparsity levels. Specifically, we trained four backbones with sparse levels set at 0.6, 0.8, 0.95, and 0.99, and the results for multi-level sparsity ST-GCNs were obtained by aggregating predictions from all sparse structures, as outlined in Table 2. It is clear that, for each backbone, the assembled module, yields a significant improvement, but only requires only 66% of the parameters compared to the dense networks. This presents a novel approach to enhance performance without increasing the model parameters.

To delve into classification performance, we conducted a detailed analysis of the classification distribution across the entire dataset. Utilizing the maximum prediction probability for each sample, rather than the predicted label, we generated violin plots to depict the distribution of classification results. In the first row of Figure 6, it’s evident

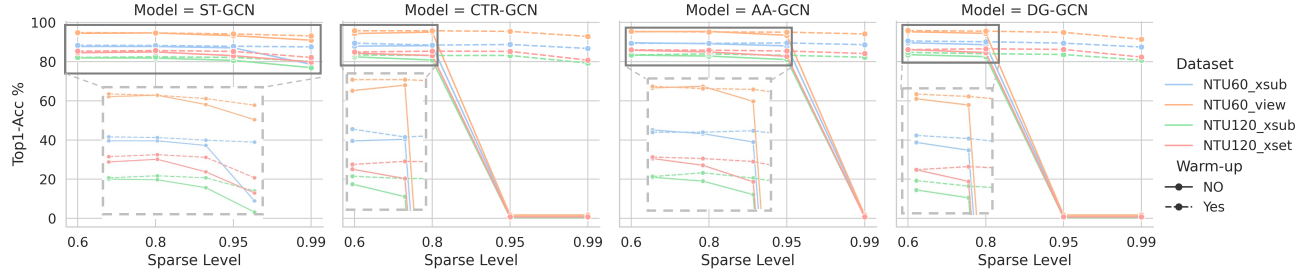


Figure 5. Ablation experiment of the warm-up stage. ‘Yes’ means training the sparse ST-GCNs with the warm-up stage, and ‘No’ means training the sparse ST-GCNs without the warm-up stage, Four backbones were utilized on the NTU RGB+D dataset.

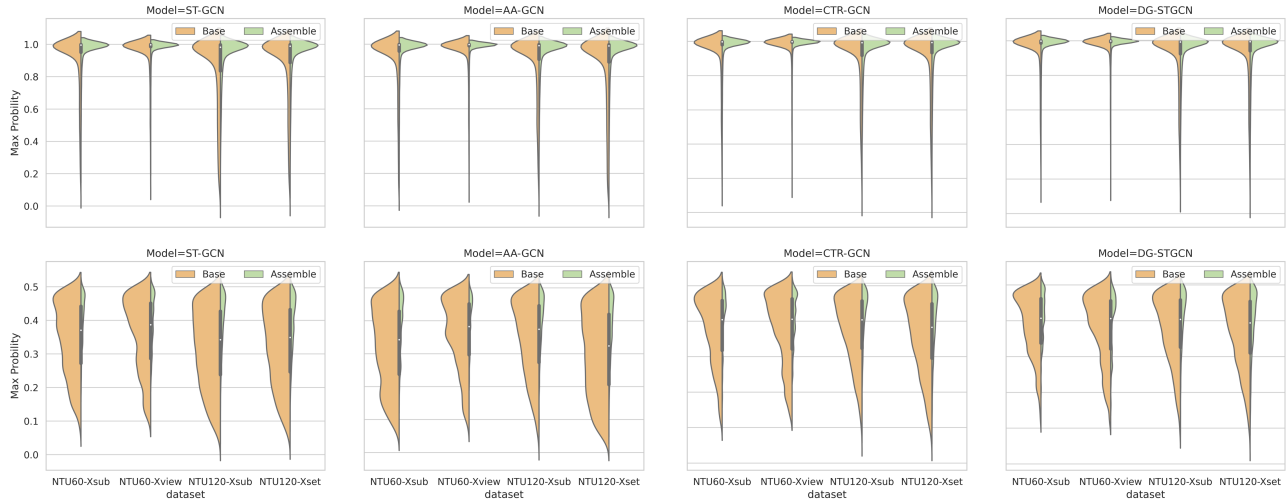


Figure 6. Analysis of the classification distribution for each backbone. ‘Base’ means the backbone, and ‘Assemble’ represents the multi-level sparsity structure. The max probability of each sample was utilized as the final result, in the first row, the samples with max probability ranging from 0 to 1 were analyzed. In the second row, the samples with a max probability lower than 0.5 were analyzed. From left to right, the performance of the four backbones is analyzed.

that the distribution for the proposed assemble models is more concentrated around 1, indicating the superiority of the multi-level sparsity ST-GCNs. Moreover, to ensure the results’ validity, we set a threshold of 0.5 to define the confidence range. Samples with a maximum prediction probability lower than 0.5 were then utilized to generate another set of violin plots. In the second row of Figure 6, we observe a significant reduction in the number of samples with a maximum prediction probability lower than 0.5 in multi-level sparsity ST-GCNs compared to their backbones. This suggests that the proposed algorithm can confidently classify indistinguishable samples more effectively.

4.5. Comparisons with the State-of-the-art

To ascertain the effectiveness of the sparse ST-GCNs generator within the multi-modalities fusion framework [42], we employed the DG-STGCN [18] as the backbone and set the sparse level as 0.6, 0.8, 0.95 and 0.99, the sparse

models are represented as DG-STGCN(S*). Following this, we trained the sparse DG-STGCN on four distinct input modalities: joints (j), joint motion (jm), bone (b), and bone motion (bm). The final result was obtained by aggregating the predictions from all streams. The performance of the sparse DG-STGCN was compared with state-of-the-art methods on NTURGB+D [32, 37], Kinetics-400 [5], and FineGYM [38] in Table 3. It can be found that the sparse DG-STGCN can obtain a comparable result to the SOTAs. The multi-level sparsity DG-STGCN (represented as DG-STGCN(A)), which requires only 66% of the parameters of the dense DG-STGCN, obtained the SOTA in 3 datasets and has comparable results in the other three.

5. Discussion

In this paper, we empirically demonstrated the over-parameterization of ST-GCNs in skeleton-based HAR. We

Table 2. Experiments for multi-level sparsity ST-GCNs. The ‘Baseline’ is the result of fully trained dense networks, and the ‘Assemble’ is the result of the multi-level sparsity ST-GCNs incorporating the backbones in multiple sparsity levels (0.6, 0.8, 0.95, 0.99 here).

Module type	ST-GCN		AA-GCN		CTR-GCN		DG-STGCN	
	Baseline	Assemble	Baseline	Assemble	Baseline	Assemble	Baseline	Assemble
NTU60-Xsub	88.0	89.8	89.4	90.8	88.8	90.5	90.4	91.3
NTU60-Xview	94.5	95.4	95.1	96.2	96.2	96.5	96.4	96.6
NTU120-Xsub	82.2	84.6	83.7	85.7	83.6	85.4	85.3	86.2
NTU120-Xset	85.8	87.4	85.6	87.8	85.8	87.1	87.2	88.3

Table 3. Comparison with SOTAs. Dataset names are abbreviated as NTU60-Xsub (60-sub), NTU60-Xview (60-view), NTU120-Xsub (120-sub), and NTU120-Xset (120-set). DG-STGCN(S*) denotes sparse DG-STGCN with different sparsity levels. DG-STGCN(A_{0.66}) represents the multi-level sparsity version. Results marked with * are reported by [21], and those marked with ° are reported by us.

Module	60-sub	60-view	120-sub	120-set	Kinetics-400	FineGYM
ST-GCN [43]	81.5	88.3	70.7	73.2	30.7	25.2*
SGN [53]	86.6	93.4	-	-	-	-
AS-GCN [30]	86.8	94.2	78.3	79.8	34.8	-
RA-GCN [45]	87.3	93.6	78.3	79.8	34.8	-
2s-GCN [40]	88.5	95.1	-	-	-	-
DGNN [39]	89.9	96.1	-	-	-	-
FGCN [52]	90.2	96.3	85.4	87.4	-	-
ShiftGCN [11]	90.7	96.5	85.9	87.6	-	-
DSTA-Net [41]	91.5	96.4	86.6	89.0	-	-
MS-G3D [33]	91.5	96.2	86.9	88.4	38.0	92.6*
CTR-GCN [10]	92.4	96.8	88.9	90.6	-	-
ST-GCN++ [17]	92.6	97.4	88.6	90.8	49.1	-
PoseConv3D [21]	94.1	97.1	86.9	90.3	47.7	94.3
SKE2GRID [4]	93.8	98.6	87.3	90.8	-	-
DG-STGCN [18]	93.2	97.5	89.6	91.3	40.3	95.1°
DG-STGCN(S _{0.6})	92.7	97.3	89.1	90.8	48.4	95.3
DG-STGCN(S _{0.8})	92.8	97.3	89.0	90.7	47.4	95.1
DG-STGCN(S _{0.95})	92.8	97.1	88.7	90.4	46.8	94.8
DG-STGCN(S _{0.99})	90.5	94.9	85.3	90.4	42.0	92.2
DG-STGCN(A _{0.66})	92.9	97.4	90.4	91.4	47.5	95.3

showed that sub-networks extracted from fully trained dense ST-GCNs perform comparably to their dense counterparts. Leveraging this insight, we proposed a sparse ST-GCNs generator to learn sparse architectures from randomly initialized dense networks. The generator allows for predefined or randomly initiated sparse structures. Extensive ablation experiments across various backbones confirmed the generalization of our approach, with sparse ST-GCNs achieving comparable performance to dense components, even at high sparsity levels. Additionally, we generated multi-level sparsity ST-GCNs by combining backbones at different sparsity levels, resulting in significant HAR performance improvements with fewer parameters.

Notably, our generic generator offers a novel approach for future model applications. However, there are avenues for future work. On one hand, the sparsity level is manually set in the current method. Thus, it’s imperative to develop mechanisms for automatically learning an optimal sparsity level for various structures. On the other hand, the multi-level sparsity model is currently generated by assembling the model after single-model training. In the future, exploring an end-to-end training approach for multi-level sparsity models is warranted.

6. Acknowledgment

This work is supported by the EPSRC (Engineering and Physical Sciences Research Council) Centre for Doctoral Training in Distributed Algorithms (Grant Ref: EP/S023445/1).

References

- [1] Dasom Ahn, Sangwon Kim, Hyunsu Hong, and Byoung Chul Ko. Star-transformer: A spatio-temporal cross attention transformer for human action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3330–3339, 2023.
- [2] Yue Bai, Huan Wang, Zhiqiang Tao, Kunpeng Li, and Yun Fu. Dual lottery ticket hypothesis. *arXiv preprint arXiv:2203.04248*, 2022.
- [3] Carlos Caetano, Jessica Sena, François Brémont, Jeferson A Dos Santos, and William Robson Schwartz. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In *2019 16th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–8. IEEE, 2019.
- [4] Cai et.al. Ske2grid. In *I*, 2023.
- [5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.
- [6] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems*, 33:15834–15846, 2020.
- [7] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Long live the lottery: The existence of winning tickets in lifelong learning. In *International Conference on Learning Representations*, 2020.
- [8] Tianlong Chen, Yu Cheng, Zhe Gan, Jingjing Liu, and Zhangyang Wang. Ultra-data-efficient gan training: Drawing a lottery ticket first, then training it toughly. *arXiv preprint arXiv:2103.00397*, 3, 2021.
- [9] Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. A unified lottery ticket hypothesis for graph neural networks. In *International conference on machine learning*, pages 1695–1706. PMLR, 2021.
- [10] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021.
- [11] Ke Cheng, Yifan Zhang, Xiangyu He, Weihang Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020.
- [12] Daiki Chijiwa, Shin’ya Yamaguchi, Yasutoshi Ida, Kenji Umakoshi, and Tomohiro Inoue. Pruning randomly initialized neural networks with iterative randomization. *Advances in Neural Information Processing Systems*, 34:4503–4513, 2021.
- [13] Srijan Das, Saurav Sharma, Rui Dai, Francois Bremond, and Monique Thonnat. Vpn: Learning video-pose embedding for activities of daily living. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 72–90. Springer, 2020.
- [14] Runwei Ding, Qinqin He, Hong Liu, and Mengyuan Liu. Combining adaptive hierarchical depth motion maps with skeletal joints for human action recognition. *IEEE Access*, 7:5597–5608, 2018.
- [15] Runwei Ding, Yuhang Wen, Jinfu Liu, Nan Dai, Fanyang Meng, and Mengyuan Liu. Integrating human parsing and pose network for human action recognition. In *CAAI International Conference on Artificial Intelligence*, pages 182–194. Springer, 2023.
- [16] Yong Du, Wei Wang, and Liang 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.
- [17] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Pyskl: Towards good practices for skeleton action recognition, 2022.
- [18] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Dg-stgcn: Dynamic spatial-temporal modeling for skeleton-based action recognition, 2022.
- [19] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Dg-stgcn: dynamic spatial-temporal modeling for skeleton-based action recognition. *arXiv preprint arXiv:2210.05895*, 2022.
- [20] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Pyskl: Towards good practices for skeleton action recognition, 2022.
- [21] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022.
- [22] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning*, pages 2943–2952. PMLR, 2020.
- [23] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.
- [24] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020.
- [25] Zhe Gan, Yen-Chun Chen, Linjie Li, Tianlong Chen, Yu Cheng, Shuohang Wang, Jingjing Liu, Lijuan Wang, and Zicheng Liu. Playing lottery tickets with vision and language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 652–660, 2022.
- [26] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury. A “string of feature graphs” model for recognition of complex activities in natural videos. In *2011 International Conference on Computer Vision*, pages 2595–2602, 2011.

- [27] Liang-Yan Gui, Kevin Zhang, Yu-Xiong Wang, Xiaodan Liang, José M. F. Moura, and Manuela Veloso. Teaching robots to predict human motion. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 562–567, 2018.
- [28] Qihong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297, 2017.
- [29] Rahul Kumar and Shailender Kumar. Multi-view multi-modal approach based on 5s-cnn and bilstm using skeleton, depth and rgb data for human activity recognition. *Wireless Personal Communications*, 130(2):1141–1159, 2023.
- [30] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3595–3603, 2019.
- [31] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2017.
- [32] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019.
- [33] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152, 2020.
- [34] Aurélie C Lozano, Naoki Abe, Yan Liu, and Saharon Rosset. Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12):i110–i118, 2009.
- [35] Szymon Jakub Mikler. Comparing rewinding and fine-tuning in neural network pruning. In *ML Reproducibility Challenge 2021 (Fall Edition)*, 2022.
- [36] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What’s hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11893–11902, 2020.
- [37] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.
- [38] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2616–2625, 2020.
- [39] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7912–7921, 2019.
- [40] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019.
- [41] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *Proceedings of the Asian Conference on Computer Vision*, 2020.
- [42] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020.
- [43] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 103–118, 2018.
- [44] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.
- [45] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Richly activated graph convolutional network for robust skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1915–1925, 2020.
- [46] Raviteja Vemulapalli, Felipe Arrate, and Rama 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.
- [47] Pratihtha Verma, Animesh Sah, and Rajeev Srivastava. Deep learning-based multi-modal approach using rgb and skeleton sequences for human activity recognition. *Multi-media Systems*, 26(6):671–685, 2020.
- [48] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):914–927, 2014.
- [49] Jianyang Xie, Yanda Meng, Yitian Zhao, Nguyen Anh, Xiaoyun Yang, and Yalin Zheng. Dynamic semantic-based spatial-temporal graph convolution network for skeleton-based human action recognition. *IEEE Transactions on Image Processing*, 2024.
- [50] Jianyang Xie, Yanda Meng, Yitian Zhao, Anh Nguyen, Xiaoyun Yang, and Yalin Zheng. Dynamic semantic-based spatial graph convolution network for skeleton-based human action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6225–6233, 2024.
- [51] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.
- [52] Hao Yang, Dan Yan, Li Zhang, Yunda Sun, Dong Li, and Stephen J Maybank. Feedback graph convolutional network

for skeleton-based action recognition. *IEEE Transactions on Image Processing*, 31:164–175, 2021.

- [53] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1112–1121, 2020.
- [54] Songyang Zhang, Xiaoming Liu, and Jun Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 148–157. IEEE, 2017.
- [55] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. *Advances in neural information processing systems*, 32, 2019.