贡献 ① 提出了个开源写架列为识别库
② 统比对了几种主流方法
③ ST-GCN++ { spatial module 添加动差链接
的分改 temporal connet.

而使:
① 使用该仓库
② references 中
没看过的论文
看一下

PYSKL: Towards Good Practices for Skeleton Action Recognition

Haodong Duan[1]   Jiaqi Wang[2]   Kai Chen[2]   Dahua Lin[1,2]
[1]The Chinese University of HongKong   [2]Shanghai AI Laboratory

Abstract

We present PYSKL: an open-source toolbox for skeleton-based action recognition based on PyTorch. The toolbox supports a wide variety of skeleton action recognition algorithms, including approaches based on GCN and CNN. In contrast to existing open-source skeleton action recognition projects that include only one or two algorithms, PYSKL implements six different algorithms under a unified framework with both the latest and original good practices to ease the comparison of efficacy and efficiency. We also provide an original GCN-based skeleton action recognition model named ST-GCN++, which achieves competitive recognition performance without any complicated attention schemes, serving as a strong baseline. Meanwhile, PYSKL supports the training and testing of nine skeleton-based action recognition benchmarks and achieves state-of-the-art recognition performance on eight of them. To facilitate future research on skeleton action recognition, we also provide a large number of trained models and detailed benchmark results to give some insights. PYSKL is released at https://github.com/kennymckormick/pyskl and is actively maintained. We will update this report when we add new features or benchmarks. The current version corresponds to PYSKL v0.2.

1. Introduction

Skeleton-based action recognition focuses on performing action recognition and video understanding based on human skeleton sequences. Compared to other modalities (like RGB / Flow), skeleton data (2D / 3D human joint coordinates) are compact yet informative and robust to illumination changes or scene variations. Due to these good properties, skeleton action recognition has attracted increasing attention in recent years. Various algorithms are developed to perform skeleton-based action recognition, which can be mainly categorized as GCN-based approaches and CNN-based approaches.

Since ST-GCN [23] first proposed to use Graph Convolutional Networks for skeleton processing, GCN-based approaches soonly became the most popular paradigm in skeleton-based action recognition. ST-GCN directly takes the sequence of joint coordinates as inputs and models skeleton data with a GCN backbone. The GCN backbone consists of alternating spatial graph convolutions and temporal convolutions for spatial and temporal modeling. Following works [3, 4, 10, 13, 17, 18, 24, 25] inherited its basic design and made different improvements by: 1) developing better graph topologies, either manual [13] or learnable [3, 17, 24, 25]; 2) training skeleton-based action recognition jointly with other auxiliary tasks [10]; 3) adopting better data pre-processing, training, and testing strategies [3, 17, 18]. With these improvements, there comes great progress in recognition performance. For instance, on NTURGB+D-Xsub [15] benchmark, the improvement of Top-1 Acc is over 10%: from 81.5% (ST-GCN [23], 2018) to 92.4% (CTR-GCN [3], 2021).

Despite the considerable improvements, the settings of different GCN approaches do not align well. For example, ST-GCN only reports the recognition performance with a single joint-stream, while most following works report the performance with an ensemble of joint / bone-stream (first proposed by 2s-AGCN [17]) or even four streams (first proposed by MS-AAGCN [18]). Besides, the pre-processing techniques (skeleton alignments, temporal padding, denoising, *etc.*) also differ a lot. However, existing open-source repositories [3, 13, 18, 23] only implement a single algorithm with their own practices. To the best of our knowledge, none of them had ever compared these architectures under a unified setting. Therefore, we developed PYSKL, which includes implementations of representative GCN approaches under a unified framework. We trained and tested each algorithm with all the latest and original good practices on multiple benchmarks. Surprisingly, we find that the recognition performance of different GCNs does not vary a lot: the extreme deviations of Top-1 Acc are less than 2% on all four NTURGB+D benchmarks [12, 15]. Especially, on NTURGB+D XView, the current state-of-the-art CTR-GCN [3] only outperforms the original ST-GCN [23] by 0.5%.

Given the pilot experiments, we find that good practices contribute more to achieving strong recognition performance rather than a complicated architectural design. This

report presents all the good practices we adopted for training GCN-based models for skeleton-based action recognition. The practices include different aspects, including data pre-processing, spatial / temporal data augmentations, and hyper-parameter settings. Besides that, we also propose an original GCN algorithm named ST-GCN++. With simple modifications on top of the ST-GCN, we achieve strong recognition performance comparable with the state-of-the-art without any complicated attention mechanism. ST-GCN++ can serve as a strong baseline for future research on skeleton-based action recognition.

Another paradigm for skeleton-based action recognition leverages Convolutional Neural Networks to process skeleton data. These approaches [5,6,22] represent human joints as Gaussian maps and aggregate them as pseudo images or video clips. The generated inputs are processed by 2D-CNN or 3D-CNN. We implement a recent state-of-the-art 3D-CNN based approach PoseC3D [6] in PYSKL. PoseC3D can achieve strong recognition performance on skeleton-based action recognition benchmarks and has unique advantages (like robustness, scalability, interoperability) compared to GCNs. However, it is much heavier than most of the existing GCN approaches due to its 3D-CNN backbone.

To summarize, PYSKL implements six representative skeleton-based action recognition approaches and supports nine different benchmarks. It provides extensive benchmark results for five GCNs on skeleton-based action recognition, including four NTURGB+D benchmarks, four skeleton modalities, and two annotation types (3D / 2D skeleton). PYSKL is released at https://github.com/kennymckormick/pyskl under the Apache-2.0 License. The repository contains all the source code, a large-scale model zoo, detailed instructions for installation, dataset preparation, and (distributed) training and testing. PYSKL also provides tools for visualizing 2D / 3D skeletons and performing skeleton-based action recognition on custom datasets with no skeleton information available.

2. GCN-based approaches

2.1. Good Practices for GCN-based approaches

2.1.1 Data Pre-processing

Skeleton sequences of different videos may have different temporal lengths, different numbers of persons, and may be captured by sensors from different views or with different setups. For 3D skeletons [12, 15] captured with Kinect sensors [26], various pre-processing approaches are adopted. ST-GCN uses no extra pre-processing and pads all sequences to a maximum length with zero padding. 2s-AGCN, however, performs pre-normalization by 1) aligning the center point of the person in the 1_{st} frame with the origin of the 3D-Cartesian coordinate system; 2) rotating all

skeletons so that the spine of the person in the first frame is parallel with the z-axis in the 3D-Cartesian coordinate system. Besides, 2s-AGCN pads skeleton sequences to a maximum length with loop padding. CTR-GCN follows the spatial pre-processing used by 2s-AGCN. However, it keeps the original length of each skeleton sequence and uses different criteria for manual denoising. PYSKL follows the pre-processing approach of CTR-GCN. For 2D skeletons predicted by pose estimators [1,20], we pre-normalize them into a fixed range (like [0, 1] or [-1, 1]) following [23]. We also perform simple pose-based tracking to form 1 or 2 skeleton sequences for NTURGB+D data.

2.1.2 Temporal Augmentations

Most GCN works do not use any temporal augmentations. Among representative GCN approaches, CTR-GCN adopts random cropping as temporal augmentations. It crops a substring from the entire skeleton sequence (substring length ratio may vary from 50% to 100%) and resize the substring to a fixed length of 64 with bilinear interpolation. Inspired by [6], we use Uniform Sampling as the temporal augmentation strategy. To generate a skeleton sequence of length M (M=100 in PYSKL), we divide the original sequence uniformly into M splits with equal lengths and randomly sample one frame per split. The sampled skeletons will be joined again and form the input sequence. With Uniform Sampling, we can generate numerous data samples with similar distribution to the source data (no interpolation used).

2.1.3 Hyper Parameter Setting

The hyper parameter settings differ a lot in previous works for skeleton-based action recognition using GCN. In PYSKL, we use the same hyper parameter setting to train all GCN models. We set the initial learning rate to 0.1, batch size to 128, and train each model for 80 epochs with the CosineAnnealing LR scheduler. For the optimizer, we set the momentum to 0.9, weight decay to 5×10^{-4}, and use the Nesterov momentum. We find that for most GCN networks, the new hyper parameter setting leads to better recognition performance than previous settings that use the MultiStep LR scheduler.

2.2. The Design of ST-GCN++

We also propose an original GCN model named ST-GCN++. With only simple modifications to the original ST-GCN, ST-GCN++ achieves strong recognition performance comparable with the state-of-the-art approach with without a complicated attention mechanism. Meanwhile, the computational overhead is also reduced. ST-GCN++ modifies the design of the interleaving spatial modules (spatial graph

Table 1. **Benchmarking GCN skeleton-based action recognition algorithms on the NTURGB+D benchmark.** Inputs are 3D skeletons with 25 joints. We set the input length to 100, input person number to 2, and apply all good practices introduced in Sec 2.1.

Model	NTURGB+D XSub				NTURGB+D XView				Computational Efficiency	
	Joint	Bone	2s	4s	Joint	Bone	2s	4s	GFLOPs	MParams
ST-GCN [23]	87.8	88.6	90.0	90.7	95.5	95.0	96.2	96.5	5.34	3.08
AAGCN [18]	89.0	89.2	90.8	91.5	95.7	95.2	96.4	96.7	6.07	3.77
MS-G3D [13]	**89.6**	89.3	91.0	91.7	**95.9**	95.0	96.4	96.9	10.27	3.17
CTR-GCN [3]	**89.6**	90.0	**91.5**	**92.1**	95.6	95.4	96.6	**97.0**	2.82	1.43
ST-GCN++	89.3	**90.1**	91.4	**92.1**	95.6	**95.5**	**96.7**	**97.0**	**2.80**	**1.39**

Table 2. **Benchmarking GCN skeleton-based action recognition algorithms on the NTURGB+D 120 benchmark.** Inputs are 3D skeletons with 25 joints. We set the input length to 100, input person number to 2, and apply all good practices introduced in Sec 2.1.

Model	NTURGB+D 120 XSub				NTURGB+D 120 XSet				Computational Efficiency	
	Joint	Bone	2s	4s	Joint	Bone	2s	4s	GFLOPs	MParams
ST-GCN [23]	82.1	83.7	85.6	86.2	84.5	85.8	87.5	88.4	5.34	3.08
AAGCN [18]	82.8	84.7	86.3	86.9	84.8	86.2	88.1	88.8	6.07	3.77
MS-G3D [13]	**84.0**	85.3	86.9	87.8	**86.0**	87.3	88.9	89.6	10.27	3.17
CTR-GCN [3]	**84.0**	**85.9**	**87.5**	**88.1**	85.9	87.4	**89.2**	**89.9**	2.82	1.43
ST-GCN++	83.2	85.6	87.0	87.5	85.6	**87.5**	89.1	89.8	**2.80**	**1.39**

convolutions) and temporal modules (temporal 1D convolutions).

2.2.1 Spatial Module Design

In ST-GCN, pre-defined sparse coefficient matrices are used for fusing features of different joints belonging to the same person, while the coefficient matrices are derived from a pre-defined joint topology. Meanwhile, ST-GCN also re-weights each element in coefficient matrices with a set of learnable weights. However, in ST-GCN++, we only use the pre-defined joint topology to initialize the coefficient matrices. We update the coefficient matrices iteratively with gradient descent during training without any sparse constraints. Besides, we also add a residual link in the spatial module, which further improves the spatial modeling capability.

2.2.2 Temporal Module Design

A vanilla ST-GCN uses a single 1D convolution on the temporal dimension with kernel size 9 for temporal modeling. The large kernel covers a wide temporal receptive field. However, this design lacks flexibility and results in redundant computations and parameters. Inspired by [3, 13], we use a multi-branch temporal ConvNet (TCN) to replace the single branch design. The adopted multi-branch TCN consists of six branches: a '1x1' Conv branch, a Max-Pooling branch, and four temporal 1D Conv branches with kernel size 3 and dilations from 1 to 4. It first transforms features with '1x1' Conv and divides them into six groups with equal

Table 3. **Benchmarking spatial augmentations using ST-GCN++ on two NTURGB+D 120 benchmarks.** Random rotation works for 3D skeletons, while random scaling works for both 2D and 3D skeletons.

Spatial Augs	Anno	NTURGB+D 120 XSub			NTURGB+D 120 XSet		
		Joint	Bone	2s	Joint	Bone	2s
None	3D	83.2	85.6	87.0	85.6	87.5	89.1
Rot	3D	83.8	**86.0**	87.7	**86.8**	**88.0**	**89.9**
Scale	3D	84.0	85.9	87.7	86.3	87.3	89.2
Rot + Scale	3D	**84.7**	**86.0**	**87.9**	86.6	**88.0**	89.8
None	2D	84.4	84.8	86.4	88.1	88.5	90.0
Scale	2D	**85.1**	**85.7**	**87.1**	**88.7**	**90.0**	**90.9**

Table 4. **ST-GCN++ trained with good practices and spatial augmenatations surpasses CTR-GCN (official performance) on 3 of 4 NTURGB+D benchmarks.**

Model	NTURGB+D		NTURGB+D 120	
	XSub	XView	XSub	XSet
CTR-GCN	92.4	96.8	**88.9**	90.6
ST-GCN++	**92.6**	**97.4**	88.6	**90.8**

channel width. Then, each feature group is processed with a single branch. The six outputs are concatenated together and processed by another '1x1' Conv to form the output of the multi-branch TCN. The new TCN design not only improves the temporal modeling capabililty, but also saves the computational cost and parameters, due to the reduced channel width for every single branch.

Table 5. **The best performance achieved by PYSKL on nine benchmarks.** On 8 of 9 benchmarks for skeleton action recognition, PYSKL achieves the best recognition accuracy. We report the Top-1 accuracy for all benchmarks except FineGYM (for which we report the mean class accuracy). For Diving48, we use the V2 annotations.

	NTURGB+D (3D)		NTURGB+D 120 (3D)		Datasets with 2D skeleton annotations				
	XSub	XView	XSub	XSet	Kinetics-400	UCF101	HMDB51	FineGYM	Diving48
Previous SOTA	92.4 [3]	96.8 [3]	**88.9** [3]	90.6 [3]	38.6 [14]	69.1 [22]	53.5 [22]	N.A.	N.A.
PYSKL	**92.6**	**97.4**	88.6	**90.8**	**49.1**	**86.9**	**69.4**	**94.1**	**54.5**

2.3. Benchmarking GCN Algorithms

In PYSKL, we benchmark four representative GCN approaches: ST-GCN [23], AAGCN [18], MS-G3D [13], CTR-GCN [3], as well as the original ST-GCN++ on four NTURGB+D benchmarks [12, 15]. For skeleton annotations, we consider 3D skeletons generated with CTR-GCN pre-processing and 2D skeletons estimated by HRNet [20]. We report the Top-1 Accuracy of joint-stream, bone-stream, two-stream fusion (joint + bone), and four-stream fusion (joint + bone + joint motion + bone motion), respectively. This report lists the benchmark results when using 3D skeletons in Table 1, 2. The results for 2D skeletons can be found in the repository.

Unlike the performance originally reported, we find that the accuracy gaps between different GCN approaches are much smaller. On all NTURGB+D benchmarks, the extreme deviation of Top-1 Accuracy is less than 2%. For all algorithms except CTR-GCN, the reproduced results are better than reported due to the adopted good practices[1]. Moreover, our ST-GCN++ achieves strong recognition performance competitive with state-of-the-art GCN approaches, with a much simpler design, fewer parameters, and fewer FLOPs.

2.4. Spatial Augmentations

We also adopt spatial augmentations in skeleton action recognition. We implement three augmentations in PYSKL:

1) Random Rotating: Rotate all skeletons (2D / 3D) with the same random angle θ ($\theta = (\theta_x, \theta_y) \in \mathbf{R}^2$ or $\theta = (\theta_x, \theta_y, \theta_z) \in \mathbf{R}^3$), each element in θ is sampled from a uniform distribution [-0.3, 0.3].

2) Random Scaling: Scale all joint coordinates in a skeleton sequence with the same scale factor r ($r = (r_x, r_y) \in \mathbf{R}^2$ or $r = (r_x, r_y, r_z) \in \mathbf{R}^3$), each element in r is sampled from a uniform distribution [-0.1, 0.1] or [-0.2, 0.2] (for 3D / 2D skeletons).

3) Random Gaussian Noise: Randomly add a small Gaussian noise for each joint. The noise can be frame-specific or frame-agnostic.

Extensive experiments are conducted to validate the effi-

cacy of three spatial augmentations (results in Table 3). We find that among the three augmentations, random rotating works for 3D skeletons; random scaling works for both 2D and 3D skeletons; while random Gaussian noise does not work for any kinds of skeletons. We train ST-GCN++ with 3D skeletons for 120 epochs with random rotating and random scaling. Table 4 shows that on 3 of 4 NTURGB+D benchmarks, ST-GCN++ surpasses the current state-of-the-art CTR-GCN.

3. CNN-based approaches

PYSKL also implements the 3D-CNN based approach PoseC3D [6]. PoseC3D takes 2D human skeletons as inputs. It first generates Gaussian maps given the 2D joint coordinates and then organizes them as a 3D heatmap volume. PoseC3D can use an arbitrary 3D-CNN for 3D heatmap volume processing. In PYSKL, we support three backbones: C3D [21], SlowOnly [8], X3D [7], and release PoseC3D trained on seven different datasets: NTURGB+D [15], NTURGB+D 120 [12], Kinetics-400 [2], UCF101 [19], HMDB51 [9], Fine-GYM [16], and Diving48 [11]. PoseC3D has good spatio-temporal modeling capability and achieves state-of-the-art recognition performance on 6 of 9 benchmarks. However, using 3D-CNN for skeleton processing consumes more computations and is much slower than representative GCN approaches.

4. Conclusion

We have publicly released PYSKL, which has extensive benchmarks for skeleton-based action recognition. PYSKL has implemented six representative algorithms under a unified framework, trained them on nine different skeleton-based action recognition benchmarks, and achieved state-of-the-art recognition performance on eight benchmarks (Table 5). It has offered good practices for training skeleton-based action recognition models and provided extensive benchmarks. Besides, it also introduced a simple and strong baseline named ST-GCN++, which surpasses previous state-of-the-art on the NTURGB+D benchmarks. We hope this repository, along with all the released training configurations and model weights will facilitate future research in this area.

---

[1]For CTR-GCN, the performance dropped a little, since spatial augmentations (like random rotation) are used in the original paper, but not used in this benchmark.

References

[1] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[2] 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. 4

[3] 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. 1, 3, 4

[4] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan 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. 1

[5] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7024–7033, 2018. 2

[6] Haodong Duan, Yue Zhao, Kai Chen, Dian Shao, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. *arXiv preprint arXiv:2104.13586*, 2021. 2, 4

[7] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 4

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 4

[9] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 4

[10] 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. 1

[11] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018. 4

[12] 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, 2020. 1, 2, 4

[13] 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. 1, 3, 4

[14] Yuya Obinata and Takuma Yamamoto. Temporal extension module for skeleton-based action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 534–540. IEEE, 2021. 4

[15] 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. 1, 2, 4

[16] 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. 4

[17] 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. 1

[18] 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. 1, 3, 4

[19] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 4

[20] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 5693–5703, 2019. 2, 4

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 4

[22] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Pose-action 3d machine for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7922–7931, 2019. 2, 4

[23] 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. 1, 2, 3, 4

[24] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 55–63, 2020. 1

[25] 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. 1

[26] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012. 2