Supplementary of "SignGraph: A Sign Sequence is Worth Graphs of Nodes"

Shiwei Gan[†] Yafeng Yin[†*] Zhiwei Jiang[†] Hongkai Wen[‡] Lei Xie[†] Sanglu Lu[†]

[†] State Key Laboratory for Novel Software Technology, Nanjing University, China

[‡] Department of Computer Science, The University of Warwick, UK

sw@smail.nju.edu.cn {yafeng,jzw,lxie,sanglu}@nju.edu.cn hongkai.wen@warwick.ac.uk

Model	Backbone	FLOPs (G)	Parameters(M)	Inference Speed
TwoStream	S3D	278.11	15.82	6.4 seq/s
VAC	Resnet18	184.20	11.60	11.2 seq/s
SignGraph	GCN	198.01	12.34	11.0 seq/s
MultiSignGraph	GCN	210.83	14.97	10.8 seq/s

Table 6. Efficiency comparison on Phoenix14T dataset.

Model	CSL-Daily		PHOENIX14T	
	WER	Del/Ins	WER	Del/Ins
Using SelfAttention	28.1	8.8/2.3	22.8	6.1/1.6
SignGraph	27.4	8.2/2.1	20.0	5.1/2.0
MultiSignGraph	26.4	7.8/2.1	19.1	4.5/1.8

Table 7. Ablation study on self-attention modules.

Comparision on Computational Complexity

To compare the computational complexity of SignGraph and existing models, we also provide the metrics of FLOPS, the number of parameters and inference speed, calculated with a 3090 GPU over 100 frames, as shown in Table 6. It is worth noting that the FLOPs and parameters for TwoStream do not include the pose estimation stage, thus the actual computational complexity of TwoStream is much higher. Still, as shown in Table 4 and 6, our SignGraph demonstrates lower FLOPs, fewer parameters, and faster inference speed compared wth TwoStream, while achieving competitive performances.

Replacing GCN Layer with Self-Attention Layer

We also show the experimental results by replacing all graph convolutional layers with self-attention layers, which lead to performance decrease, as shown in Table 7. The reason may be that our SignGraph can dynamically establish connections between important regions (*e.g.*, hand areas), rather than combining all regions in self-attention modules. Thus we can focus on SL-related features better.

Analysis on Patterns of Constructed Graphs

As visualized in Figure 6, we select a sign video from Phoenix14T test set and visualize the constructed graph structure of LSG and TSG modules in both two stages. It can be found that the graph structure is formed by multiple connected graphs. In regard to a connected graph, it is formed by the connections of cross regions, including nearby or distant regions in one frame and the same or different regions in adjacent frames, where the region is usually related to hand or facial areas.

Broader Impact and Limitations

In this paper, we propose a simple yet effective graph-based sign language processing model to improve communication between the hearing people and the deaf community. The proposed model is data-driven, thus there may be unpredictable failures and potential negative implications, since that the results predicted by the model can be affected by biases in the data. In addition, the performance of our model will vary slightly with different hyper-parameters K (*i.e.*, the number of edges in the graph). It is difficult to find the optimal K manually by enumerating, thus an effective automated tuning algorithm may be another promising way to improve sign language recognition performance in our framework.

Future Work

Despite the excellent performance of the proposed model, the constructed graphs by the KNN algorithm in LSG and TSG modules are not always satisfactory, thus it is meaningful to explore a better algorithm to construct graphs for exploring correlation of sign-related regions. In addition, to show the effectiveness of SignGraph model, we mainly focus on the CSLR task in our paper. In fact, there exist other sign language tasks, e.g., sign language translation, sign language retrieval. In future, we will try to modify our model and perform more experiments on other sign language tasks.

*Yafeng Yin is the corresponding author.