Sign Language Translation with Hierarchical Spatio-Temporal Graph Neural Network

Jichao Kan[1,2,*], Kun Hu[1,†], Markus Hagenbuchner[3], Ah Chung Tsoi[3],
Mohammed Bennamoun[4], Zhiyong Wang[1]
[1]School of Computer Science, The University of Sydney
[2]Data Science Institute, University of Technology Sydney
[3]School of Computing and Information Technology, University of Wollongong
[4]Department of Computer Science and Software Engineering, The University of Western Australia
jichao.kan@student.uts.edu.au, kuhu6123@uni.sydney.edu.au, {markus,act}@uow.edu.au,
mohammed.bennamoun@uwa.edu.au, zhiyong.wang@sydney.edu.au

Abstract

Sign language translation (SLT), which generates text in a spoken language from visual content in a sign language, is important to assist the hard-of-hearing community for their communications. Inspired by neural machine translation (NMT), most existing SLT studies adopted a general sequence to sequence learning strategy. However, SLT is significantly different from general NMT tasks since sign languages convey messages through multiple visual-manual aspects. Therefore, in this paper, these unique characteristics of sign languages are formulated as hierarchical spatio-temporal graph representations, including high-level and fine-level graphs of which a vertex characterizes a specified body part and an edge represents their interactions. Particularly, high-level graphs represent the patterns in the regions such as hands and face, and fine-level graphs consider the joints of hands and landmarks of facial regions. To learn these graph patterns, a novel deep learning architecture, namely hierarchical spatio-temporal graph neural network (HST-GNN), is proposed. Graph convolutions and graph self-attentions with neighborhood context are proposed to characterize both the local and the global graph properties. Experimental results on benchmark datasets demonstrated the effectiveness of the proposed method.

1. Introduction

Sign languages, which engage visual-manual modalities to convey meanings, are the primary communication tools for the deaf and hard-of-hearing community. However, it is still an open research problem to reduce the gap of the communications between sign language users and spoken language users who have limited sign language knowledge. Therefore, researchers have utilized various methods to convert sign language to spoken language for a better communication between the users of different languages.

Early efforts fell into the category of sign language recognition (SLR). At the beginning, SLR methods aimed to recognize an isolated gloss from a sign language video [11, 34], while the continuous nature of languages was ignored. Therefore, continuous SLR was proposed to generate a sequence of pre-defined glosses (i.e., the written words interpreting signing poses) [8, 23]. With the recent success of deep learning techniques in many applications, SLR can be regarded as a neural machine translation (NMT) tasks following an encoder-decoder framework, where the *source* is a *video* and the *target* is a corresponding spoken sentence. Early NMT based studies for sign language understanding conducted a SLR task (e.g., [8, 11, 22]). Researchers adopted an encoder to extract latent representations from sign language videos and a decoder to perform the process of generating gloss scripts [4]. Similar to the development of general NMT studies, attention mechanisms [1, 35] have been introduced for SLR to focus on the most relevant video frames when generating a specified gloss [43].

It is worth noting that sign languages convey meaning from multiple aspects: the manual articulations and the non-manual elements such as postures and movements of different body parts contribute to the meaning as well as the lexical distinction, grammatical structure, adjectival or adverbial content, and discourse functions. It is anticipated that such unique domain knowledge could be beneficial for advancing sign language understanding. As a result, a num-

*Work was done during Master of Philosophy Study at The University of Sydney.

†Corresponding Author, supported by Australian Research Council (ARC) Grant DP210102674.

ber of studies were proposed to explore such knowledge by characterizing the local patterns individually of several key regions of a signer's body [4, 18, 20, 22, 26, 33]. Recently, the interactions between these local regions have been explored by focusing only on the spatial relations between the regions or the temporal relations across the same region [23, 24, 43, 15]. Nonetheless, these SLR methods are not ideal for sign language understanding as the gloss-level recognition is still different from spoken languages. Therefore, it is attractive to devise sign language translation (SLT) methods to reduce the communication barrier. SLT takes a sign language sentence performed by a signer as input to produce text scripts of the signing sentence in a spoken language. As an early attempt, [7] formalized the SLT problem and released an SLT dataset PHONEIX-Weather-2014-T. After that, more MNT based methods were investigated for SLT tasks (e.g., [13, 39]).

However, existing methods on either SLR and SLT have not fully explored the interactions between those local regions at a fine-grained level to best utilize the unique aspects of sign languages, which demands a better representation of signing poses. Therefore, in this paper, to better represent the spatio-temporal relations at a finer-grained level for SLT, a novel graph-based sign language representation and a hierarchical graph neural network architecture are proposed. As illustrated in Figure 1, a sign language sentence can be characterized with appearance, motion and pose representations. The proposed method exploits these representations from a fine-level to a high-level. A fine-level spatio-temporal graph is based on the joints within a human body region (e.g., the left-hand). A high-level spatio-temporal graph is to formulate the relations between the human body regions, which can be based on appearance, motion, or pose features. To this end, a hierarchical spatio-temporal graph neural network (HST-GNN) to learn the hierarchical graph patterns with both high-level and fine-level graphs for SLT. HST-GNN introduces a connection between graph convolutions and graph self-attentions using the neighborhood context, which helps to formulate graph patterns from multiple perspectives. Comprehensive experiments on datasets demonstrated the performance of our proposed method.

In summary, the key contributions of this study are as follows:

- A novel deep architecture, namely hierarchical spatio-temporal graph neural network (HST-GNN), is devised for SLT.
- Multiple spatio-temporal graphs with hierarchical structures are constructed to represent signing poses.
- Graph convolution and graph self-attention with connections based on their neighborhood context are studied to learn graph representations from multiple perspectives.

2. Related Work

In this section, relevant studies are reviewed from two aspects: existing SLR and SLT methods, and graph neural networks which are relevant to our proposed method.

2.1. Sign Language Recognition & Translation

Vision-based sing language recognition and translation aim to understand visual contents of sign languages to generate glosses and spoken language text scripts, respectively.

Sign language recognition is a task taking the visual content performed by sign language signers to produce the associated glosses. Early studies on SLR focused on recognizing isolated signs or gestures to produce word-level or phrase-level outputs, which followed a pattern recognition pipeline: various hand-crafted visual features such as SIFT and SURF were obtained from an input signing video and a trained classifier took these features to produce signing labels [11, 34]. With the advances of deep learning, convolution neural networks (CNN) and recurrent neural networks (RNN) were also adopted for isolated SLR [44]. However, recognizing isolated signs provides limited understanding of a complete sign language sentence. Therefore, continuous SLR has been investigated at the sentence level by treating continuous signings as a sequence of signing poses. Recently, various deep architectures have been proposed to perform continuous SLR task [12, 13]. To consider the domain knowledge of sign languages, fine-level regional patterns were investigated for accurate SLR in an independent manner [7, 9, 22, 28, 30, 39]. To explore the interactions between the local regions of a human body in a signing pose, graph-based neural networks were proposed to formulate the spatial relations between the regions or the temporal relations within a region across frames [20, 23, 24, 25, 43]. However, the spatio-temporal relations have been seldom explored with hierarchical structures, which could miss important sign language patterns.

Although there have been impressive SLR results, the gap between glosses and spoken language sentences still exist. To address this problem, sign language translation has been studied to take the rich grammatical structures in spoken languages into consideration. It aims to generate spoken language sentences rather than a sequence of glosses [29]. For example, RNN [7] and Hierarchical LSTM [20] were adopted to extract visual information and to generate spoken language sentences. Moreover, the above-mentioned deep learning based SLR methods were also explored for SLT [39, 40, 43]. However, domain knowledge of sign languages such as the interactions between human body regions has not been adequately investigated yet. Missing such fine-grained patterns could result in less accurate sign language representations, and thus negatively impacts the quality of the generated spoken language scripts.

2.2. Graph Neural Network

While conventional neural networks have been utilized to process vectorized data, there are numerous applications involving data in non-Euclidean forms such as graphs. Graph neural network (GNN) was first proposed to address the learning tasks with graph inputs [16, 17, 32, 38]. Graph convolution network (GCN) extended convolution filters for graphs to help construct deep graph representations [2]. Graph attention network (GAT) was further proposed to estimate adjacency weights [36]. Recently, various methods have been proposed to discover special graph properties, which have been ignored by conventional GCNs, such as CPNGNN [31] which exploits the local ordering of the nodes to increase the representation capacity of graph networks, and DimNet [19] which introduces directed message passing to improve the capacity of graph representation. [3] applied the GCN in the sign language segmentation.

3. Methodology

The proposed HST-GNN adopts an encoder-decoder scheme as illustrated in Figure 1. The encoder represents an input video of a sign language sentence to a latent space, which includes graph construction, graph convolution and graph self-attention mechanism. Two levels of graphs are constructed including high-level graphs between the key body regions (i.e., the left and right hands and the face) with appearance and optical flow vertex features and fine-level graphs of the key regions with appearance features (i.e., left-hand graph, right-hand graph and facial graph). The graph convolution and the graph self-attention with neighborhood context are introduced to formulate the local and the global graph properties, respectively. A hierarchical graph pooling mechanism is adopted to fuse these graphs as the final encoded latent vector for an input sign language sentence.

The decoder generates text scripts in a spoken language using the encoded representation, which adopts a two stage recognition scheme with two LSTMs containing attention mechanisms: the first one translates the fused vector to glosses (i.e., written words corresponding to individual signing poses) and the second one translates the glosses to text scripts in a spoken language. In this section, the details of the proposed methods are introduced.

3.1. Hierarchical Spatio-temporal Graphs

As illustrated in Figure 1, two levels of graphs including high-level graphs and fine-level graphs are constructed to characterize the three key-regions of a signer's body and their interactions in hierarchical structures.

A high-level graph characterizes the spatio-temporal relationships between the three key regions of a human body in video frames. That is, a high-level graph can be constructed with three vertices which denote facial region, left-hand region, and right-hand region, respectively. The three key regions can be obtained by pose estimation algorithms (e.g., HRNet [37]). To characterize each vertex, a bounding box is used to extract an associated frame sub-patch and the vertex-level feature can be computed using the image patch with a pre-trained CNN. Note that the frame used to extract the visual features can be in the modality of appearance (RGB) or motion (e.g., optical flow[41]) in this paper. As a result, two high-level graphs are constructed, one for each modality.

In more detail, denote the vertex-level features for high-level graphs (Level 1 graphs) derived from the t-th frame of the input video as $\dot{\mathbf{V}}_t^{m,1} = \{\dot{\mathbf{v}}_{i,t}^{m,1} \in \mathbb{R}^d\}$, where d is the dimension of the feature vectors of modality $m \in \{\text{appearance(a)}, \text{opticalflow(o)}\}$ and i indicates the i-th body region in the frame. To involve additional temporal relationship from neighbouring frames, a slide window W is defined and the set of the vertices of the t-th graph is defined as follows:

$$\mathbf{V}_t^{m,1} = \bigcup_{w=-W}^{W} \dot{\mathbf{V}}_{t+w}^{m,1} := \{\mathbf{v}_{i,t}^{m,1} \in \mathbb{R}^d | i = 1, ..., n_1\}, \quad (1)$$

which contains n_1 vertices.

An adjacency matrix $\mathbf{A}_t^{m,1}$ is defined to represent the relationships between the vertices in $\mathbf{V}_t^{m,1}$. Instead of using a pre-defined $\mathbf{A}_t^{m,1}$ empirically, in this study, $\mathbf{A}_t^{m,1}$ is learned in an unsupervised way. The element $a_{ij,t}^{m,1}$ of $\mathbf{A}_t^{m,1}$ in the i-th row and j-th column denotes the interaction between the vertices $\mathbf{v}_{i,t}^{m,1}$ and $\mathbf{v}_{j,t}^{m,1}$. The following computation characterizes the interaction $a_{ij,t}^{m,1}$ by considering the relevant vertex-level features:

$$a_{ij,t}^{m,1} = \sigma({\mathbf{v}_{i,t}^{m,1}}^T \mathbf{M}^{m,1} \mathbf{v}_{j,t}^{m,1}), \quad (2)$$

where $\mathbf{M}^{m,1} \in \mathbb{R}^{d \times d}$ is the matrix of a bilinear transform containing trainable parameters and σ is an non-linear function to increase the capability to represent complex patterns. Furthermore, to reduce the number of the parameters in $\mathbf{M}^{m,1}$ and the model complexity, a low-rank decomposition of $\mathbf{M}^{m,1}$ is introduced:

$$\mathbf{M}^{m,1} = \mathbf{M}_1^{m,1} {\mathbf{M}_2^{m,1}}^T, \quad (3)$$

where $\mathbf{M}_1^{m,1} \in \mathbb{R}^{d \times p}$, $\mathbf{M}_2^{m,1} \in \mathbb{R}^{d \times p}$, and $p << d$.

In this paper, two additional constraints are applied to the adjacency matrix $\mathbf{A}_t^{m,1}$. The first one is for symmetry:

$$\dot{\mathbf{A}}_t^{m,1} = {\mathbf{A}_t^{m,1}}^T \mathbf{A}_t^{m,1}. \quad (4)$$

The second one is for the normalization, which alleviates the scale difference between the vertices. In detail, $\dot{\mathbf{A}}$ is normalized by its matrix norm:

$$\ddot{\mathbf{A}}_t^{m,1} = \frac{\dot{\mathbf{A}}_t^{m,p}}{||\dot{\mathbf{A}}_t^{m,1}||}. \quad (5)$$

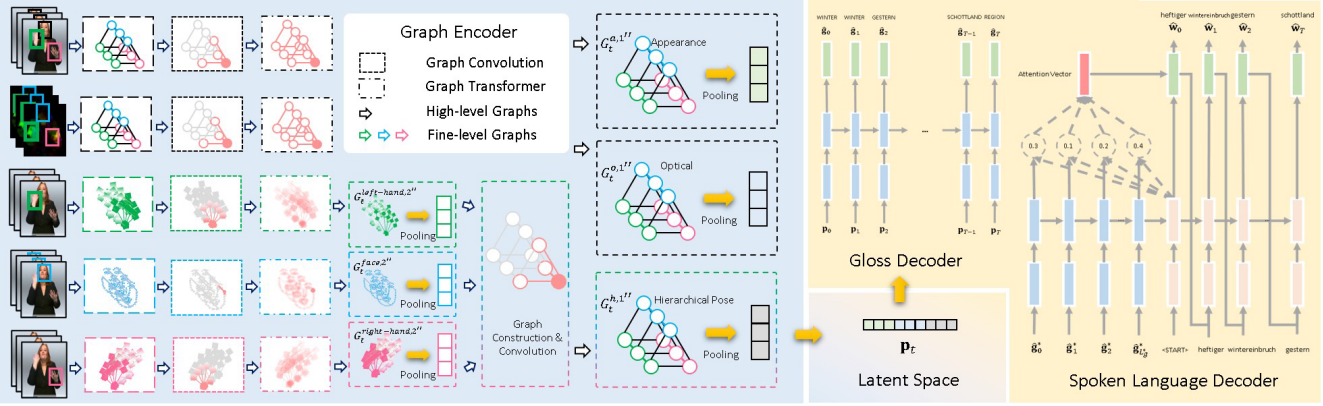

Figure 1. Illustration of the proposed HST-GNN architecture for SLT.

For convenience, in the following discussion, $\mathbf{A}_t^{m,1}$, $\dot{\mathbf{A}}_t^{m,1}$ and $\ddot{\mathbf{A}}_t^{m,1}$ are not particularly distinguished. In summary, for a particular modality m, a high-level graph sequence can be obtained as $\{\mathbf{G}_t^{m,1} = \{\mathbf{V}_t^{m,1}, \mathbf{A}_t^{m,1}\}\}$.

For the fine-level graphs, key points (e.g., the joints for hands and the landmarks for face) are further identified to construct individual graphs (Level 2 graphs) for each body region (i.e., left hand, right hand, and face). Similar to the high-level graphs, the regions near the joints can be used to characterize these key points and their relationships. For computational efficiency, only appearance features are used for the fine-level graphs and $\{\mathbf{G}_t^{r,2} = \{\mathbf{V}_t^{r,2}, \mathbf{A}_t^{r,2}\}\}$ can be obtained, where $r \in \{\text{left hand, right hand, face}\}$ indicates the body regions under consideration.

The proposed encoder of HST-GNN encodes these graphs to a latent graph space for SLR by considering the local and the global graph properties. For convenience, the superscripts are omitted in the following discussion of the graph convolution and the graph self-attention.

3.2. Graph Convolution

Graph convolution neural networks generalize the convolution filters for graph inputs of arbitrary structures, of which the input and the output are graphs. In detail, it can be viewed as the message passing through the neighbors of each vertices in a graph in line with its vertex-level features and adjacency patterns. By stacking multiple graph convolution layers, a deep graph neural network can be obtained for graph-based deep representations. Formally, the computations for the t-th graph of a video in the l-th graph convolution layer can be formulated as:

$$\mathbf{H}_t^{l+1} = f(\mathbf{H}_t^l, \mathbf{A}_t; \mathbf{W}^l) = f(\mathbf{A}_t \mathbf{H}_t^l \mathbf{W}^l), \tag{6}$$

where $\mathbf{H}_t^l \in \mathbb{R}^{p^l}$ is the input of the layer, $\mathbf{H}_t^{l+1} \in \mathbb{R}^{p^{l+1}}$ is the output of the layer, $\mathbf{W}^l \in \mathbb{R}^{p^l \times p^{l+1}}$ contains the learnable parameters as a linear transform, and f is a non-linear vertex-wise activation function. Note that the multiplication of $\mathbf{A}_t$ and $\mathbf{H}_t^l$ implements the message passing through the vertices, which helps each vertex to obtain proper patterns from its neighbouring body regions or key points for aggregating and modeling the graph context. In this way, the graph convolutions mainly focus on the local properties for each vertex. In particular, $\mathbf{H}_t^0 = [\mathbf{v}_1^t, \mathbf{v}_2^t, ..., \mathbf{v}_n^t]^T$ and p^0 is the dimension of the vertex-level feature vectors. The output graph from the last graph convolution layer is denoted as $\mathbf{G}_t' = \{\mathbf{V}_t', \mathbf{A}_t'\}$.

3.3. Graph Transformer

The transformer with the self-attention mechanism and the positional encoding was proposed for sequentially organized data [35], which processes the sequential patterns globally in a parallel manner. As graph convolutions mainly formulate the local graph properties, inspired by the transformer, we propose a novel graph transformer to formulate the global graph properties with a shortcut connection of the graph convolutions by involving the neighboring patterns.

The self-attention mechanism can be regarded as a querying process with queries, keys and values. For an input graph $\mathbf{G}_t' = \{\mathbf{V}_t', \mathbf{A}_t'\}$, the computations are as follows:

$$\mathbf{Q}_t = \mathbf{V}_t' \mathbf{W}^{Query}, \mathbf{K}_t = \mathbf{V}_t' \mathbf{W}^{Key}, \mathbf{L}_t = \mathbf{V}_t' \mathbf{W}^{Value}, \tag{7}$$

where $\mathbf{W}^{Query}$, $\mathbf{W}^{Key}$ and $\mathbf{W}^{Value}$ are learnable parameters of three linear projections. With these computations, for a particular vertex $\mathbf{v}_i^{t'} \in \mathbf{V}_t'$, the associated query $\mathbf{q}_i^t$, which is the i-th row of $\mathbf{Q}_t$, is used to query a context from other vertices. More specifically, the inner product of $\mathbf{q}_i^t$ and $\mathbf{k}_j^t$ (the j-th row of $\mathbf{K}_t$) is computed as a score s_{ij} to measure the extent of collecting patterns from the values $\mathbf{l}_j^t$ (the j-th row of $\mathbf{L}_t$) of the vertex $\mathbf{v}_j^{t'}$ to the vertex $\mathbf{v}_i^{t'}$. Formally, the score matrix computations can be written as:

$$\mathbf{S}_t = \mathbf{Q}_t \mathbf{K}_t{}^T. \tag{8}$$

To eliminate the effects in terms of the variable graph nodes,

a normalization step is introduced:

$$\dot{\mathbf{S}}_t = softmax(\frac{\mathbf{S}_t}{\sqrt{n}}), \tag{9}$$

where n is the number of the vertices of an input graph. In order to incorporate the guidance of the local graph properties, a shortcut for the neighborhood context is established with the estimated adjacency matrix $\mathbf{A}_t$ that is used for graph convolutions. The score matrix is computed as follows:

$$\ddot{s}_{ij} = \frac{a_{ij} exp(\dot{s_{ij}})}{\sum_k a_{ik} exp(\dot{s}_{ik})}. \tag{10}$$

For the sake of convenience, $\mathbf{S}_t$, $\dot{\mathbf{S}}_t$ and $\ddot{\mathbf{S}}_t$ are all represented by $\mathbf{S}_t$ in the following discussion. With the score matrix, the vertex-level features with contextual information can be computed as:

$$\dot{\mathbf{V}}''_t = \mathbf{S}_t \mathbf{L}_t. \tag{11}$$

Therefore, with the above computations, the self-attention mechanism helps each vertex to collect information globally compared with graph convolutions which focus only on local neighborhoods.

In addition, to take different perspectives for the graph modelling, multiple independent self-attention heads can be computed as $\mathbf{V}''_{t,k}$, where $k \in \{1, ..., K_{head}\}$ and K_{head} is the number of the independent heads. A fully connected layer can be used to summarize these concatenated attention heads as:

$$\dot{\mathbf{V}}''_t = [\dot{\mathbf{V}}''_{t,1}, \dot{\mathbf{V}}''_{t,2}, ..., \dot{\mathbf{V}}''_{t,n}]\mathbf{W}^{Multihead}, \tag{12}$$

where $\mathbf{W}^{Multihead}$ is the parameter matrix of a linear projection.

Lastly, a vertex-level feed-forward network (FFN) is applied to $\dot{\mathbf{V}}''_t$, which is constructed by fully connected layers. The computations in the transformer are computed in a vertex-wise manner, so the outputs are still in graphs. In addition, the graph transformer can also be stacked to construct deep architectures: each of them takes the inputs from the outputs of its previous layer. The output of the graph transformer is denoted as $\mathbf{G}''_t = \{\mathbf{V}''_t, \mathbf{A}''_t\}$.

3.4. Hierarchical Graph Pooling

With the above discussions, the graphs at the two levels, $\mathbf{G}_t^{m,1}$ and $\mathbf{G}_t^{r,2}$, have been encoded to a latent space by using graph convolutions and graph transformers. Particularly, in the latent space, $\mathbf{G}_t^{a,1\prime\prime}$, $\mathbf{G}_t^{o,1\prime\prime}$, $\mathbf{G}_t^{left-hand,2\prime\prime}$, $\mathbf{G}_t^{right-hand,2\prime\prime}$ and $\mathbf{G}_t^{face,2\prime\prime}$ can be obtained as encoded graph representations for high-level appearance, high-level motion, fine-level left hand, fine-level right hand and fine-level face, respectively. To use these graphs for decoding, a hierarchical pooling strategy is introduced as illustrated in Figure 1. First, average pooling is applied to each fine-level graph individually and a pooled feature vector can be obtained. These feature vectors are further used to construct another high-level graph denoted as $\mathbf{G}_t^{h,1\prime\prime}$. Next, the high-level graphs are pooled individually and the vectors obtained can be concatenated as a fused latent vector $\mathbf{p}_t$ to represent the t-th video frame and a sequence $\mathbf{p} = \{\mathbf{p}_0, ..., \mathbf{p}_T\}$ is used to represent the entire video sequence.

3.5. Language Decoder

Following a two-stage scheme, the language decoder aims to generate a translation in a spoken language by using the fused latent vector. Based on the latent vector sequences $\mathbf{p}$, the first stage - *feats2gloss* - outputs an estimated gloss $\hat{\mathbf{g}}_i$ for each video frame (i.e., latent vector) and a sequence can be obtained as $\hat{\mathbf{g}} = \{\hat{\mathbf{g}}_0, ..., \hat{\mathbf{g}}_T\}$. The sequence $\hat{\mathbf{g}}$ can be viewed as an alignment path to video frames of an estimated gloss sequence $\hat{\mathbf{g}}^* = \{\hat{\mathbf{g}}_0^*, ..., \hat{\mathbf{g}}_{L_g^*}^*\}$ of the ground truth $\mathbf{g}^* = \{\mathbf{g}_0^*, ..., \mathbf{g}_{L_{g^*}}^*\}$, where L_{g^*} is the length of the gloss sequence. The second stage - *gloss2text* - outputs the sentence $\hat{\mathbf{w}} = \{\hat{\mathbf{w}}_0, ..., \hat{\mathbf{w}}_{L_w}\}$ as an estimation of the ground truth $\mathbf{w} = \{\mathbf{w}_0, ..., \mathbf{w}_{L_w}\}$ in spoken language using the estimated gloss sequence $\hat{\mathbf{g}}^*$, where L_w is the sentence length.

The *feats2gloss* stage involves an LSTM network for sequential to sequential recognition. Formally, this stage formulates the following conditional probability:

$$p(\mathbf{g}_t | \hat{\mathbf{g}}_0, ..., \hat{\mathbf{g}}_{t-1}, \mathbf{p}_0, ..., \mathbf{p}_{t-1}), \tag{13}$$

which is the probability that the t-th generated gloss is $\mathbf{g}_t$ by considering the previously generated glosses $\hat{\mathbf{g}}_0, ..., \hat{\mathbf{g}}_{t-1}$ and the encoded vectors $\mathbf{p}_0, ..., \mathbf{p}_{t-1}$. In particular, denote the output of this LSTM network as $\mathbf{Y}^g = (y_{ij}^g)$, in which the element y_{ij}^g indicates the probability that the i-th gloss in the output sequence is associated with the j-th encoded latent vector.

The *gloss2text* stage is based on another LSTM network with a general attention mechanism. It adopts the generated gloss sequence to formulate the following probability:

$$p(\mathbf{w}_l | \hat{\mathbf{w}}_0, ..., \hat{\mathbf{w}}_{l-1}, \hat{\mathbf{g}}_0^*, ..., \hat{\mathbf{g}}_{L_g^*}), \tag{14}$$

where the estimation $\hat{\mathbf{w}}_l$ of the l-th word $\mathbf{w}_l$ is obtained according to the previously estimated words $\hat{\mathbf{w}}_0, ..., \hat{\mathbf{w}}_{l-1}$ and the gloss sequence $\hat{\mathbf{g}}^*$ estimated by decoding stage 1. The generation starts from $\mathbf{w}_{start}$, which is a start signal, and ends with a stopping signal $\mathbf{w}_{end}$.

3.6. Optimization Loss

The translation error is measured by considering the error of the generated glosses and the generated words, associated with the two decoding stages - *feats2gloss* and *gloss2text*, respectively.

Following a conventional practice in SLR, a connectionist temporal classification (CTC) loss [10] is adopted for the glosses, which helps to obtain the unknown alignment between the encoded vector sequence and the gloss sequence. In detail, given the encoded vector sequence $\mathbf{p}$ and the gloss annotation $\mathbf{g}^*$ of the corresponding video, the CTC loss is defined as:

$$L_{ctc} = -\log p_{ctc}(\mathbf{g}^*|\mathbf{p}), \tag{15}$$

where p_{ctc} is a probability to generate the given gloss sequence with the condition of the given encoded vector sequence.

There are many different potential paths to align the encoded vectors with the given gloss sequence. Denote $\mathcal{M}^{-1}(\mathbf{g}^*)$ as the set of all these paths. For a particular path $\mathbf{z} = \{\mathbf{z}_0, ..., \mathbf{z}_T\} \in \mathcal{M}^{-1}(\mathbf{g}^*)$, the probability to obtain this path is in line with the probability computed in $\mathbf{Y}^g$:

$$p(\mathbf{z}|\mathbf{p}) = \prod_t y^g_{z_t t}. \tag{16}$$

Hence, the probability for all potential paths, which is exactly the probability p_{ctc}, can be computed as:

$$p_{ctc}(\mathbf{g}|\mathbf{p}) = \sum_{\mathbf{z} \in \mathcal{M}^{-1}(\mathbf{P})} p(\mathbf{z}|\mathbf{p}). \tag{17}$$

For the generated words in a spoken language, the alignment between the words and the glosses is often not required due to the lack of proper orders. Therefore, a general cross-entropy loss is used to measure the error of each word in a sequence. In detail, the loss can be written as:

$$L_{ce} = -\log \prod_l p(\mathbf{w}_l|\hat{\mathbf{w}}_0, ..., \hat{\mathbf{w}}_{l-1}, \hat{\mathbf{g}}_0, ..., \hat{\mathbf{g}}_T). \tag{18}$$

Note that the softmax function to compute the probability is embedded in the computations of the output of the LSTM in the stage 2 decoder.

Therefore, the total loss is a linear combination of the two loss functions L_{ctc} and L_{ce} with an additional regularization term for the parameters Θ of the proposed architecture,

$$L = \lambda_{ctc} L_{ctc} + \lambda_{ce} L_{ce} + \lambda_r ||\Theta||, \tag{19}$$

where λ_{ctc}, λ_{ce} and λ_r are the weights associated with the three losses and can be tuned as hyper-parameters during the optimization.

4. Experimental Results

4.1. Datasets & Evaluation Metrics

The proposed method was evaluated on two widely used benchmark sources: PHOENIX-2014, PHOENIX-2014-T [4] and Chinese Sign Language Recognition (CSL) [18, 30, 42]. PHOENIX-2014 contains videos from PHOENIX TV station, which includes the weather forecast content featured with signers over a period of three years. The videos were collected with a resolution of 210 by 260 at 25 frames per second (fps) using a stationary color camera. The dataset was annotated with sign language glosses and texts in German spoken language. The vocabulary size is 1,115 for sign glosses and 3,000 for German. The CSL dataset contains two subsets: Split I and Split II. In this study, Split II was adopted, which contains 100 sign language sentences related to the daily life with a vocabulary size of 178. Each sentence in the split was performed by 50 signers each of whom repeated the signing for 5 times. These sentences were recorded in RGB videos with a spatial resolution 1280 by 720 at 30 fps. Among the 100 sentences, 94 sentences were in the training set and 6 sentences were in the test set.

In terms of the evaluation metrics to evaluate the performance, two metrics are adopted: word error rate (WER) and bilingual evaluation understudy (BLEU) score, which are widely used for natural language processing (NLP) [27]. WER measures the recognition performance at the gloss level, whilst BLEU scores measure the performance of the translation. BLEU was first proposed to measure the performance of machine translation by comparing the recall and the precision of n-grams.

4.2. Implementation Details

To construct input graphs, human skeletons were extracted by two algorithms, HRNet [37] and OpenPose [6]. In detail, the coordinates can be extracted for the key points and the key body regions that are used in this study. For high-level graphs, key body regions of the face, the left hand and the right hand were extracted using a window of size 24 by 24 of which the center was located on the corresponding detected key point. Appearance and motion features for these regions were further computed to represent the vertices by using ResNet-152 [14] and TVL1-flow [41], respectively. The dimension of these feature vectors is 1,024. For fine-level graphs, 29 landmarks and 21 joints were obtained for face and hands, respectively. The coordinates detected by the skeleton detection algorithms were adopted as vertex-level features directly.

Our proposed method was implemented with PyTorch. An Adam optimizer with an initial leaning rate 0.001, and 30 epochs were used to train the model. At the validation stage, hyper-parameters λ_{ctc} and λ_{ce} were set to 0.5 and 0.5, respectively; the temporal window size to construct the spatio-temporal graphs was set to 3 and further discussions are provided in Section 4.5.

4.3. Overall Performance

To demonstrate the effectiveness of the proposed method, a number of recently proposed methods were com-

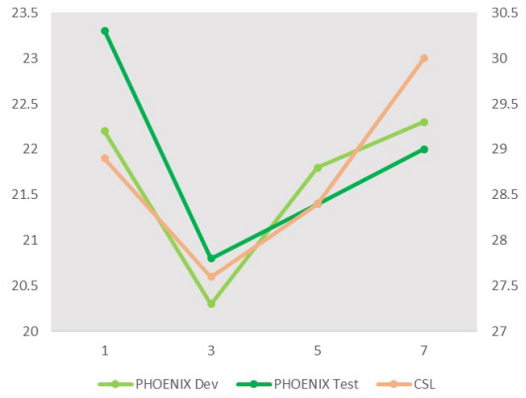

Figure 2. Effect of the windows size on recognition performance (lower is better). WER scores on PHONIX (left y-axis) WER scores on CSL (right y-axis) vs. window size (x-axis).

Table 1. Recognition Performance on PHOENIX-2014. Metric: WER: Lower is better.

Dataset	Phoenix		CSL
Subset	Test	Dev	-
IAN [30]	36.7	37.1	32.7
DenseTCN [12]	36.5	35.9	44.7
CNN-LSTM-HMM [21]	26.0	26.0	-
DNF [9]	24.4	23.8	-
STMC [43]	20.7	21.1	28.6
HLSTM [13]	-	-	48.7
Ours	**19.8**	**19.5**	**27.6**

pared as shown in Table 2: iterative alignment network (IAN) [30], which adopts an iterative alignment network to reduce the gap between videos and generated glosses, enabling a better correspondence between glosses and frames; DenseTCN [12], which introduced temporal convolutions to efficiently explore temporal patterns; CNN-LSTM-HMM [21], which combined LSTM and HMM in language modeling for the construction of gloss sequences and an intermediate synchronization constraints, respectively. Deep neural frame (DNF) [9], which incorporates RGB and motion features of body regions to explore fine-level sign language patterns; Spatial-temporal multi-cue network (STMC) [43], which involved a dense network to characterize the spatial and the temporal dependencies to improve the recognition quality; Hierarchical LSTM (HLSTM) [13], which was proposed to extract multiple levels of attention with adaptive online key clip mining; Neural sign language translation (NSLT) [7], which was the first study using CNN-LSTM sign language translation in an end-to-end manner; Neural language translation with transformer (SLTT) [39], which utilized transformers for sign language translation based on STMC [43].

It can be observed that the methods with fine-level regional body patterns (e.g., DNF and STMC) achieved better performance compared to IAN. The introduction of the relationships between these body regions further improved the performance as STMC was better than DNF. The explicit inclusion of the temporal clues could also help to improve the SLR performance, for example, CNN-LSTM-HMM was superior to IAN. Moreover, the adoption of the transformer helped to increase the recognition performance, such as NSLT vs SLTT, which suggested that proper transformer designs could be beneficial for the sign language modelling. In terms of WER, our method achieved 19.8 on the Phoenix-2014-T dev set, 19.5 on Phoenix-2014-T test set and 27.6 on the CSL dataset; for BLEU-1 score, our method achieved 45.2 on the Phoenix-2014-T dev set, 46.1 on Phoenix-2014-T test set and 49.1 on the CSL dataset. This indicates that our method achieved the state-of-the-art performance on the two benchmark datasets for sign language recognition and translation.

4.4. Effect of Temporal Window Size

It is anticipated that the temporal window size impacts the performance of our proposed HST-GNN as a larger temporal window implies that longer time dependencies can be captured. Nonetheless, a long temporal perception could increase the model complexity and introduce unnecessary historical information. Therefore, the trade-off to select a proper window size was investigated. The results are shown in Figure 2 in terms of the WER and BLEU scores on the two benchmarks. It can be observed that the recognition performance increased when the window size was changed from 1 (i.e., only the current state without any historical patterns) to 3. However, if the window size further increased, the performance was negatively impacted for the two datasets. Hence, a window size of 3 was used as a proper choice in line with the experimental results.

4.5. Ablation Study

To further explore how the proposed mechanisms work for SLR under different configurations, ablation studies were conducted for the spatial, temporal and hierarchical modules in HST-GNN. A baseline model denoted as Model I was introduced without involving any graph patterns. Next, a number of architectures involving parts of these mechanisms were investigated on top of the baseline model: Model II involves temporal graphs; Model III involves spatial graphs; Model IV involves both spatial and temporal graphs; and Model V considers spatial, temporal, and hierarchical graphs all, which is the full version of the proposed HST-GNN. The results are shown in Table 3 and Table 4. It can be observed that using the spatial or the temporal graph patterns individually improved the performance compared to the baseline model in terms of WER and BLEU scores on both benchmark datasets. Introducing both the spatial and temporal graph patterns clearly en-

Table 2. Translation Performance on PHOENIX-2014-T (Phoenix) and CSL. BLEU: Higher is better.

Metric	BLEU-1			BLEU-2			BLEU-3			BLEU-4		
Dataset	Phoenix		CSL	Phoenix		CSL	Phoenix		CSL	Phoenix		CSL
Subset	Test	Dev	-	Test	Dev	-	Test	Dev	-	Test	Dev	-
NSLT [7]	43.3	42.9	-	30.4	30.3	-	22.8	22.02	-	18.1	18.4	-
SLTT[39]	44.95	48.27	-	36.53	35.20	-	29.30	27.47	-	24.00	22.47	-
JSL[5]	46.61	47.26	-	33.73	34.40	-	26.19	27.05	-	21.32	22.38	-
Ours	**45.2**	**46.1**	**49.1**	**34.7**	**33.4**	**33.1**	**27.1**	**27.5**	**22.7**	**22.3**	**22.6**	**17.8**

Table 3. Ablation studies on PHOENIX-2014-T (Phoenix) and CSL. WER: Lower is better, BLEU: Higher is better. ✓ (or ✗) indicates the inclusion (or exclusion) of a specific mechanism (S: spatial graph, T: temporal graph, H: hierarchical graph).

Model	Method			WER			BLEU-1			BLEU-2			BLEU-3			BLEU-4		
	Dataset			Phoenix		CSL	Phoenix		CSL	Phoenix		CSL	Phoenix		CSL	Phoenix		CSL
	S	T	H	Test	Dev	-	Test	Dev	-	Test	Dev	-	Test	Dev	-	Test	Dev	-
I	✗	✗	✗	35.8	35.4	32.1	43.2	43.3	48.1	30.2	30.1	28.1	22.1	21.9	16.6	18.0	18.2	14.3
II	✗	✓	✗	23.4	23.2	30.8	43.4	43.4	48.4	30.9	30.7	29.1	23.2	23.1	17.1	18.5	18.8	14.6
III	✓	✗	✗	22.4	22.5	29.3	43.6	43.5	48.8	31.7	31.2	31.3	24.7	25.2	19.9	19.2	19.7	15.9
IV	✓	✓	✗	20.7	20.3	28.6	43.7	43.8	49.1	32.1	32.5	32.1	25.9	26.1	21.1	20.8	21.3	17.4
V	✓	✓	✓	**19.8**	**19.5**	**27.6**	**45.2**	**46.1**	**49.1**	**34.7**	**33.4**	**33.1**	**27.1**	**27.5**	**22.7**	**22.3**	**22.6**	**17.8**

Table 4. Ablation Study on PHOENIX-2014T: Different Features.

Dev					
HST-GNN	WER	B1	B2	B3	B4
w/o appearance	23.8	44.1	31.3	26.2	20.7
w/o motion	20.1	42.3	32.2	24.9	19.1
w/o pose	22.3	43.7	32.8	25.1	20.3
Test					
HST-GNN	WER	B1	B2	B3	B4
w/o appearance	23.7	44.8	31.8	26.7	20.1
w/o motion	20.3	42.1	32.9	24.5	17.9
w/o pose	21.9	40.1	33.1	23.9	19.3

hanced the performance further. Lastly, hierarchical modelling also showed its effectiveness for SLR.

4.6. Qualitative Analysis

To understand the proposed methods, two examples, which were featured by two different signers, were illustrated in Figure 3. Video frames, the frame-level gloss predictions of the five models and the ground truth of the gloss sequence are presented. From the first example, it can be observed that the *Baseline* method (Model I) missed the glosses DONNERSTAG and WEITER. With the temporal graphs (Model II), the gloss WEITER can be detected, whilst the gloss DONNERSTAG was still missed and a new insertion error occurred. Similarly, for the case only with the spatial graphs (Model III), three glosses were detected and one was missed with an insertion error. By introducing the spatial and the temporal mechanisms simultaneously (Model IV), all glosses were detected correctly without insertion errors. The hierarchical mechanism further improved the gloss predictions, by which the number of frames with glosses were increased significantly. The present of both graph can successfully detect the all of the glosses.

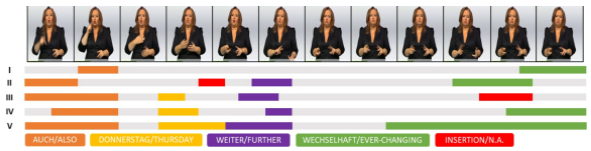

Figure 3. Sample results of our proposed HST-GNN model under different configurations. CTC objectives are illustrated to indicate the associations of the gloss (with English translation) and the video frames. *INSERTION* means a wrong gloss prediction outside the ground truth gloss annotation.

5. Conclusion

In this paper, a novel neural network, namely HST-GNN, is presented for sign language understanding. Hierarchical graphs are introduced to characterize visual signing content, which include high-level graphs and fine-level graphs associated with key body regions. HST-GNN follows an encoder-decoder framework: the encoder adopts graph convolutions and graph transformers with an adjacency matrix based connection to explore both the global and local graph properties; the decoder reconstructs the glosses and the spoken language script in line with the latent embedding. Experiments on two widely used dataset including PHOENIX-2014-T and CSL were conducted and the results clearly demonstrated the effectiveness of the proposed HST-GNN. In our future work, we will investigate additional graph properties to improve the performance of SLT.

References

[1] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *International Conference on Learning Representations*, 2014.

[3] Hannah Bull, Michèle Gouiffès, and Annelies Braffort. Automatic segmentation of sign language into subtitle-units. In *European Conference on Computer Vision*, pages 186–198. Springer, 2020.

[4] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In *IEEE International Conference on Computer Vision*, 2017.

[5] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2021.

[7] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[8] Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13(1):2205–2231, 2012.

[9] Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, 21(7):1880–1891, 2019.

[10] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine Learning*, 2006.

[11] Kirsti Grobel and Marcell Assan. Isolated sign language recognition using hidden markov models. In *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, 1997.

[12] Dan Guo, Shuo Wang, Qi Tian, and Meng Wang. Dense temporal convolution network for sign language translation. In *International Joint Conference on Artificial Intelligence*, 2019.

[13] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Hierarchical LSTM for sign language translation. In *AAAI Conference on Artificial Intelligence*, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] Al Amin Hosain, Panneer Selvam Santhalingam, Parth Pathak, Huzefa Rangwala, and Jana Košecká. Hand pose guided 3d pooling for word-level sign language recognition. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[16] Kun Hu, Zhiyong Wang, Kaylena A Ehgoetz Martens, Markus Hagenbuchner, Mohammed Bennamoun, Ah Chung Tsoi, and Simon JG Lewis. Graph fusion network-based multimodal learning for freezing of gait detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[17] Kun Hu, Zhiyong Wang, Wei Wang, Kaylena A.S Ehgoetz Martens, Liang Wang, Tieniu Tan, Simon JG Lewis, and David Dagan Feng. Graph sequence recurrent neural network for vision-based freezing of gait detection. *IEEE Transactions on Image Processing*, 29:1890–1901, 2019.

[18] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *AAAI Conference on Artificial Intelligence*, 2018.

[19] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.

[20] SangKi Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. Neural sign language translation based on human keypoint estimation. *Applied Sciences (Switzerland)*, 9(13):2683, 2018.

[21] Oscar Koller, Cihan Camgoz, Hermann Ney, and Richard Bowden. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[22] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 2015.

[23] Oscar Koller, Hermann Ney, and Richard Bowden. Deep Hand: how to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.

[24] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep Sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. *International Journal of Computer Vision*, 2018.

[25] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *The IEEE Winter Conference on Applications of Computer Vision*, 2020.

[26] Zhipeng Liu, Xiujuan Chai, Zhuang Liu, and Xilin Chen. Continuous gesture recognition with hand-oriented spatiotemporal feature. In *IEEE International Conference on Computer Vision Workshops*, 2017.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Annual Meeting on Association for Computational Linguistics*, 2002.

[28] Junfu Pu, Wengang Zhou, Hezhen Hu, and Houqiang Li. Boosting continuous sign language recognition via cross modality augmentation. In *ACM International Conference on Multimedia*, 2020.

[29] Junfu Pu, Wengang Zhou, Hezhen Hu, and Houqiang Li. Boosting continuous sign language recognition via cross modality augmentation. In *ACM International Conference on Multimedia*, 2020.

[30] Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative alignment network for continuous sign language recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[31] Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Approximation ratios of graph neural networks for combinatorial problems. In *Advances in Neural Information Processing Systems*, 2019.

[32] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[33] Advaith Sridhar, Rohith Gandhi Ganesan, Pratyush Kumar, and Mitesh Khapra. Include: A large scale dataset for Indian sign language recognition. In *ACM International Conference on Multimedia*, 2020.

[34] Dominique Uebersax, Juergen Gall, Michael Van den Bergh, and Luc Van Gool. Real-time sign language letter and word recognition from depth data. In *IEEE International Conference on Computer Vision Workshops*. IEEE, 2011.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Transformer: Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[37] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[38] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[39] Kayo Yin. Sign language translation with transformers. *arXiv preprint arXiv:2004.00588*, 2020.

[40] Kayo Yin and Jesse Read. Better sign language translation with STMC-transformer. In *International Conference on Computational Linguistics*, pages 5975–5989, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics.

[41] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007.

[42] Hao Zhou, Wengang Zhou, and Houqiang Li. Dynamic pseudo label decoding for continuous sign language recognition. In *2019 IEEE International Conference on Multimedia and Expo*, 2019.

[43] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. Spatial-Temporal Multi-Cue Network for Continuous Sign Language Recognition. In *AAAI Conference on Artificial Intelligence*, 2020.

[44] Guangming Zhu, Liang Zhang, Peiyi Shen, Juan Song, Syed Afaq Ali Shah, and Mohammed Bennamoun. Continuous gesture segmentation and recognition using 3DCNN and convolutional lstm. *IEEE Transactions on Multimedia*, 2018.