Multi-view Integration with View-specific Self-attention for Sign Language Recognition

Hoang Quang Huy and Tran Anh Vu *

Department of Electronic Engineering School of Electrical and Electronic Engineering (SEEE), Hanoi University of Science and Technology (HUST), Hanoi, Vietnam
Email: huy.hoangquang@hust.edu.vn (H.Q.H.); vu.trananh@hust.edu.vn (T.A.V.)
*Corresponding author

***Abstract*—Sign Language Recognition (SLR) plays a pivotal role in mitigating communication barriers between the deaf community and broader society. Recently, the Video Transformer Network (VTN), an extension of the transformer architecture incorporating a multi-head self-attention mechanism, has demonstrated efficacy in video processing tasks and SLR. However, relying solely on Red Green Blue (RGB) frame information may render the model susceptible to redundant data and environmental factors such as lighting variations and complex backgrounds. Additionally, most existing SLR datasets provide only frontal viewpoints, which constrain the model's generalizability, particularly in real-world settings where diverse perspectives are prevalent. In this study, we propose Video Transformer Network with 3 Graph Convolutional Network (VTN3GCN), a multi-view and multi-stream framework that integrates RGB data, skeleton coordinates, and pose flow from three distinct viewpoints: left, right, and center. This architecture enhances VTN by incorporating Graph Convolutional Networks (GCN) to learn skeletal frame features and employs an early fusion mechanism between the RGB and skeleton streams. Experiments conducted on the Multi-VSL200 dataset—a newly curated dataset for Vietnamese Sign Language (VSL) featuring three viewpoints per video demonstrate that the VTN3GCN framework achieves a top-1 accuracy of up to 92.92%, achieving state-of-the-art accuracy. Our code and data are available at: https://github.com/fossbk/MultiView-ISLR/tree/main/VTN3GCN.**

***Keywords*—Vietnamese sign language, sign language recognition, deep learning, video transformer network**

I. Introduction

Sign language is a modality of communication that combines hand gestures and body movements, instead of spoken sounds, and is predominantly utilized by individuals who are deaf or hard of hearing. Mastery of sign language demands significant time and dedication, which not everyone can commit to, thereby creating an invisible barrier in communication between deaf individuals and hearing individuals. With the remarkable advancements in scientific research, particularly in machine learning, deep learning, and computer vision, Sign Language Recognition (SLR) systems have been developed to address this issue, facilitating more effective communication between the deaf community and the broader society. However, constructing robust SLR systems remains a complex challenge due to the inherent variability in sign language gestures, regional dialects, and the diversity in individual signing styles.

Traditional methods in SLR primarily rely on handcrafted feature extraction and the utilization of classical machine learning algorithms [1–3]. In recent years, the field of SLR and action recognition, in general, has undergone significant advancements through the application of state-of-the-art deep learning techniques [4–10]. Modern researches predominantly focus on enhancing accuracy, practicality, and scalability by leveraging deep learning and emerging methodologies to process both single-channel data—such as Red Green Blue (RGB) frames and skeleton—and multi-channel data. RGB frame-based approaches employ video data to extract spatial and temporal features, offering robust visual image analysis capabilities [5, 7, 11]. However, this approach is constrained by factors such as lighting conditions and viewing angles, struggles in complex background settings, and is susceptible to redundant information. Conversely, skeleton-based methods utilize skeletal frame data to model kinematics and the relationships between joints [6, 12, 13]. This approach is more resilient to variations in lighting and color but is vulnerable to noise and loss in key point coordinates during extraction, resulting in unreliable predictions. Currently, hybrid methods that integrate RGB data, skeleton data, and additional data modalities such as optical flow and depth images have emerged to capitalize on the strengths of each approach, thereby improving generalizability and recognition performance [4, 10, 14].

However, a common limitation in most current SLR studies is the exclusive focus on data from frontal viewpoints. Relying solely on a single viewpoint does not adequately represent real-world scenarios, where perspectives can vary significantly, thereby restricting recognition accuracy [15]. Utilizing multi-view data enables models to access information from multiple

Manuscript received July 29, 2025; revised August 22, 2025; accepted November 5, 2025; published July 23, 2026.

angles, thereby enhancing generalizability and recognition accuracy. Consequently, employing multi-view data constitutes a crucial advancement in improving the generalizability and accuracy of SLR systems. Nevertheless, processing multi-view data demands greater computational resources. Moreover, creating rich and accurately labeled multi-view datasets remains a significant challenge.

Although deep learning-based models have made significant advancements in SLR, the majority of existing work still concentrates on single-frontal viewpoint data. The research gap lies in effectively leveraging multi-view information and the integration of multiple data streams, such as RGB frames, skeleton coordinates, and pose flow, to enhance both the accuracy and generalization capabilities of SLR systems. In this study, we aim to develop a multi-view, multi-stream integrated approach for the SLR task. Main contributions include:

- Proposition of the Video Transformer Network 3 Graph Convolutional Network (VTN3GCN) multi-view-multi-stream framework: We extend the VTN architecture by concurrently integrating three viewpoints-left, right, and center-and incorporating GCN to exploit skeletal frame features, alongside pose flow to capture comprehensive motion information. This approach enables the framework to more fully represent sign language expressions from different observational angles. To the best of our knowledge, this is the first method to utilize three streams and three viewpoints for Vietnamese Sign Language (VSL) recognition.
- Evaluation on the Multi-VSL dataset: The VTN3GCN framework was trained and analyzed on Multi-VSL200, a newly collected dataset for VSL featuring three distinct viewpoints per video [16]. Quantitative results demonstrate that the proposed framework significantly improves recognition accuracy compared to VTN or single-view methods and their three-view variants, thereby affirming the potential of multi-channel and multi-view research directions in SLR.
- Self-attention encoder analysis: Independent self-attention modules were qualitatively analyzed to assess their capability to capture important gestures in sign language, as well as their robustness against asynchronous input data from different views within the same time frame.
- Robustness analysis of VTN3GCN against missing data: We conducted experiments simulating data loss scenarios on one or two of the three viewpoints, thereby evaluating the impact on the framework's accuracy. The results highlight the importance of maintaining at least one observational viewpoint and suggest the development of data recovery techniques to enhance system robustness.

II. Related Work

A. Sign Language Recognition

SLR is a complex and significant research area within the computer vision community. Traditional methods in SLR employ Support Vector Machines (SVM) [17], Hidden Markov Models (HMM) [1, 3], or Histogram of Oriented Gradients (HOG) [2]. However, these methods present several limitations, such as failing to fully capture the dynamic and spatial features of gestures, being constrained in their ability to process complex data, and being unable to exploit the latent information inherent in large datasets. Deep learning models, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), GCN, and transformers, have been demonstrated to be effective in improving the accuracy of SLR specifically and action recognition in general [4, 12–14, 18–23].

Skeleton-based methods have become popular due to their computational efficiency and independence from variations in subjects and backgrounds. Skeleton data provides a detailed geometric structural representation of the human body through joints and bones, allowing for the extraction of rich structural information. Deep learning architectures such as RNN, CNN, and Graph Convolutional Network have been applied to effectively exploit spatiotemporal information from skeleton data, thereby significantly enhancing performance in skeleton-based action recognition [12–14, 18, 19]. For instance, studies utilize LSTM, a type of RNN, to extract features from skeleton data [12, 13]. GCN is employed to learn spatial features, while Bidirectional Encoder Representations from Transformers (BERT) is used to capture temporal features [19]. Meanwhile, Jiang [14] propose Sign Language Graph Convolution Network (SLGCN) and Separable Spatial-Temporal Convolution Network (SSTCN) to explore features from hand and whole-body key points, in combination with 3D-CNNs and additional data modalities such as RGB and optical flow.

However, skeleton data is susceptible to noise or incompleteness during extraction, posing a significant challenge that adversely affects recognition accuracy [24]. Recent advancements in RGB-based SLR have shown promising results [4, 20–23]. Methods utilizing Motion History Images (MHI) combined with 3D CNN have demonstrated high efficacy, wherein RGB-MHI is integrated as a spatial attention module or through late fusion techniques [20]. Furthermore, focusing on specific body regions such as hands, face, and the upper body has contributed to a 19% improvement in accuracy by training separate CNN models and fusing their scores [23]. Attention mechanisms, such as the Top-Down (TD) attention model TD-SLR, have enhanced recognition by prioritizing critical regions of video sequences through the integration of RGB, optical flow, and attention data [21]. Additionally, integrating RGB data with other modalities such as depth images and skeleton data has also demonstrated significant effectiveness. Other notable studies include StepNet [22], a network focusing on part-level spatiotemporal modeling without relying on key points, and the Two-Stream Network [4], which simultaneously processes raw video and key point sequences with frame-level automatic distillation.

B. Multi-Stream Approach

The multi-stream approach has been extensively utilized in action recognition [5, 7, 25–28]. For instance, the two-stream ConvNet employs RGB frames and optical flow frames as its inputs [7]. The SlowFast network leverages both slow and fast frame rates for RGB inputs [5], whereas SF-GCN is applied to key point coordinates [26]. Beyond the two-stream configuration, Wang *et al*. [28] incorporates Motion Stacked Difference Images (MSDI) in addition to RGB frames and optical flow as inputs to the model. Similarly, in SLR, analogous concepts have been implemented [4, 7, 8–10, 14]. For example, the Spatial Multi-Cue Module (STMC) divides the RGB input into four components, including the full RGB frame, face, hands, and pose, which serve as inputs for subsequent stages of the model [9]. In this work, we utilize early fusion to combine the feature maps of RGB and key point coordinates, thereby enhancing the model's robustness against key point noise and mitigating the presence of redundant information within the RGB frames [29].

C. Multi-View Approach

With multi-view data, methods employing view-invariant representations have demonstrated promising results [30–40]. For example, Dividing and Aggregating Network (DA-Net) utilizes view-specific CNN branches to exploit multi-view relationships [35], while Shah *et al.* [36] employ contrastive learning to capture multi-view features. Multi-view features are derived from horizontal and vertical gradients [40]. Studies leverage global Gaussian Mixture Models (GMM) or multi-dimensional Grassmann and Stiefel manifolds to construct multi-view spaces [30, 31], whereas some studies adopt different approaches to learn a global codebook [32, 33]. Although these methods have been extensively explored in action recognition, they remain limited in SLR due to the predominance of frontal views in most multi-view datasets. This limitation may hinder these approaches from achieving satisfactory recognition performance on multi-view datasets such as the Multi-VSL200 dataset in Fig. 1. In this work, we focus on developing a model that integrates CNN, GCN, and a self-attention decoder tailored to each viewpoint.

Fig. 1. Example from the Multi-VSL dataset. Different performers with three viewpoints: Left, centre, and right. The performers' faces have been hidden to ensure privacy.

More recently, the field has seen the emergence of end-to-end transformer architectures and explorations into diffusion-based models for sign language production and recognition. While our work builds upon the established VTN, these newer approaches represent an exciting frontier. A comprehensive comparison with such models constitutes a valuable direction for future research to further benchmark the performance of multi-view integration against the latest architectural innovations.

III. Methods

A. Dataset

The Multi-VSL200 dataset includes 199 glosses from the VSL library. Each video captures a volunteer performing a single gloss from one of three distinct viewpoints: left (~45°), center (~0°), and right (~45°). The average duration of each video is 2.67 s, recorded at a frame rate of 30 Frames Per Second (FPS), and the minimum video resolution is 1280 × 720 pixels.

To ensure high representativeness, each video was recorded using three cameras corresponding to three distinct viewpoints. A total of 18,981 videos were captured for this dataset, equivalent to 5663 instances across all viewpoints. This tri-view structure is a crucial factor in enhancing the generalization capability of SLR models, as it reflects the realistic perspectives that signers may encounter in real-world environments. The Multi-VSL200 dataset offers a comprehensive and diverse approach for the research and development of VSL recognition models.

The dataset was collected from 30 different volunteers (5 teachers and 25 students, aged 10–50), including both native signers (24 deaf signers) and trained performers (4 hearing teachers) to ensure diversity in signing styles. The 199 glosses are distributed across various categories, including nouns, verbs, adjectives, and common phrases. Recordings were conducted in a controlled indoor environment with consistent lighting against a simple classroom background (green, blue screen or plain wall) to minimize environmental noise. Three synchronized cameras were used to capture the viewpoints simultaneously.

B. VTN3GCN

We introduce the VTN3GCN framework for the SLR task. We employ Hand Crop (HC) and PoseFlow (PF), which have been demonstrated to provide relevant information exclusively from RGB video and body pose movements [41]. The primary pipeline of the framework is illustrated in Fig. 2.

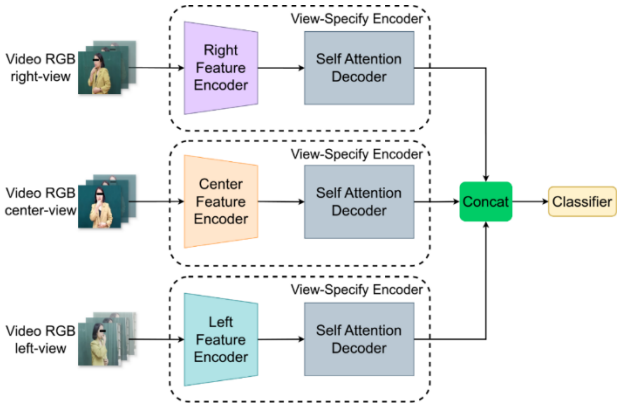

Fig. 2. Our framework pipeline.

The proposed VTN3GCN framework, as illustrated in Figs. 3–6, comprises five main components: two hands RGB feature extractor—CNN, key points feature extractor—GCN, fusion, self-attention decoder, 3-views concatenation, and linear classifier.

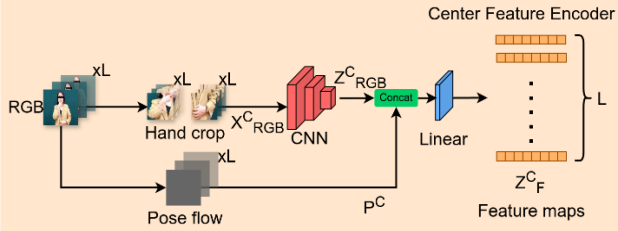

Fig. 3. Centre feature encoder.

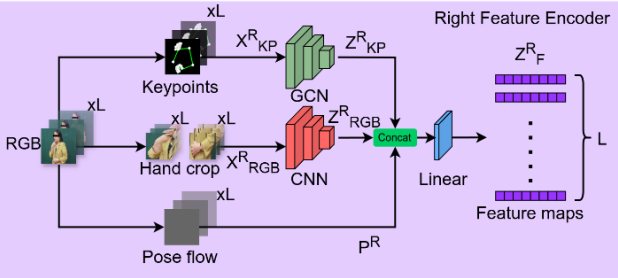

Fig. 4. Right feature encoder.

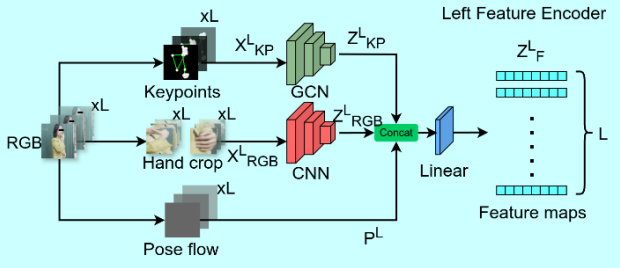

Fig. 5. Left feature encoder.

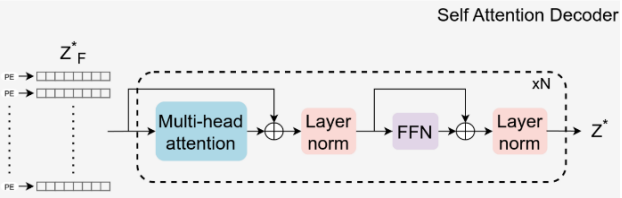

Fig. 6. Self-attention decoder.

Two hands RGB feature extractor employs a 2D CNN, specifically ResNet34, a model that utilizes residual blocks with the objective of constructing a deeper architecture, enhancing image feature extraction, and simultaneously mitigating the effects of the vanishing gradient phenomenon [42]. Assuming that the input to the CNN is X_{RGB}^{L}, X_{RGB}^{C}, $X_{RGB}^{R} \in \mathbb{R}^{L\times 2\times C_{RGB}\times H\times W}$, where X_{RGB}^{L}, X_{RGB}^{C}, X_{RGB}^{R} respectively represent the left, center, and right-angle sequences of RGB frames. L is the number of frames, 2 denotes the two hands, C is the number of channels, and H and W are the height and width of the frames, respectively. The feature maps of CNN are as Eq. (1):

$$Z_{RGB}^{L}=CNN\left(X_{RGB}^{L}\right),\ Z_{RGB}^{C}=CNN\left(X_{RGB}^{C}\right),\ Z_{RGB}^{R}=CNN\left(X_{RGB}^{R}\right) \quad (1)$$

where Z_{RGB}^{L}, Z_{RGB}^{C}, $Z_{RGB}^{R} \in \mathbb{R}^{L\times 2D_{RGB}}$.

Key points feature extractor employs a GCN, specifically the Adaptive Graph Convolutional Network (AGCN), which comprises Adaptive Graph Convolutional Blocks (AGCB) [6]. Each AGCB consists of a spatial GCN (Convs) and a temporal GCN (Convt), accompanied by Batch Normalization (BN) [43], an intermediate dropout layer [44], and a Rectified Linear Unit (ReLU) activation function at the end. The AGCN architecture is chosen due to its demonstrated ability to capture stable temporal-agnostic information, as evidenced in action recognition tasks. We have removed the final SoftMax layer, retaining the remaining components for the purpose of extracting features from the input key points. Assuming that the GCN inputs for the left and right are X_{KP}^{L}, $X_{KP}^{R} \in \mathbb{R}^{C_{KP}\times L\times V}$, where X_{KP}^{L} and X_{KP}^{R} respectively represent the sequences of keypoints from the left and right view. C_{KP} is the number of keypoint channels. L is the number of frames, and V is the number of vertices. Feature maps of GCN are as Eq. (2):

$$Z_{KP}^{\prime L}=GCN\left(X_{KP}^{L}\right),\ Z_{KP}^{\prime R}=GCN\left(X_{KP}^{R}\right) \quad (2)$$

where $Z_{KP}^{\prime L}$, $Z_{KP}^{\prime R} \in \mathbb{R}^{D_{KP}\times L^{\prime}\times V}$. D_{KP} is the feature dimension of key points, $L^{\prime}=L/4$.

To facilitate concatenation with the features extracted by the CNN, we performed dimensionality transformation and interpolation as Eq. (3):

$$Z_{KP}^{L}=Interpolate\left(Z_{KP}^{\prime L}\right),\ Z_{KP}^{R}=Interpolate\left(Z_{KP}^{\prime R}\right) \quad (3)$$

where Z_{KP}^{L}, $Z_{KP}^{R} \in \mathbb{R}^{L\times D_{KP}}$.

The fusion component concatenates the features extracted from the RGB video and key points, along with PF, and then passes them through a linear layer. Specifically, the extracted features from the left and right angles include Z_{RGB}^{L}, Z_{RGB}^{R}, Z_{KP}^{L}, and Z_{KP}^{R}, along with PF from the left and right views, denoted as P^{L}, $P^{R} \in \mathbb{R}^{L\times K\times 2}$. Here, K is the number of selected points for body pose computation, and 2 represents the angle and magnitude of the motion vector. They are concatenated as Eqs. (4) and (5):

$$Z_{F}^{L}=Concat\left(Z_{RGB}^{L},Z_{KP}^{L},P^{L}\right),\ Z_{F}^{R}=Concat\left(Z_{RGB}^{R},Z_{KP}^{R},P^{R}\right) \quad (4)$$

$$Z_{F}^{L}=Z_{F}^{L}W_{F}^{L},\ Z_{F}^{R}=Z_{F}^{R}W_{F}^{R} \quad (5)$$

where Z_{F}^{L}, $Z_{F}^{R} \in \mathbb{R}^{L\times D}$. W_{F}^{L} and W_{F}^{R} are left and right linear weights, respectively, $D=2D_{RGB}$.

For the center view, since there are no key point features, Z_{RGB}^{C} and PF of center view is $P^{C} \in \mathbb{R}^{L\times K\times 2}$ are concatenated:

$$Z_F^C = Concat\left(Z_{RGB}^C, P^C\right) \tag{6}$$

$$Z_F^C = Z_F^C W_F^C \tag{7}$$

where $Z_F^C \in \mathbb{R}^{L\times D}$. W_F^C is center linear weight.

Self-Attention Decoder comprises N self-attention blocks [45], constructed from residual multi-head dot-product attention layers and residual Feed-Forward Networks (FFN), with layer normalization [46] applied between and at the end of these components. The multi-head attention mechanism computes attention using queries, keys, and values derived from the same input sequence. Positional encoding for each element in the sequence is added to the feature vectors prior to the first multi-head self-attention layer in the decoder. VTN3GCN utilizes fixed positional encoding from the original transformer architecture [45]. This encoding is essential to provide the network with information about the sequence order. Instead of using linear layers, the FFN employs Convolutional layers with a kernel size of 1 (Conv1D). The self-attention decoder is utilized to decode the features from each view Z_F^L, Z_F^C, Z_F^R. Each view has its own self-attention decoder with similar operations. Therefore, we represent all three decoders collectively as follows. Assume that the input to the self-attention network is Z_F^*, where $* \in \{L;C;R\}$. The input Z_F^* is transformed into queries Q, keys K, and values V through trainable linear transformations.

$$Q^* = Z_F^* W_Q^*,\ K^* = Z_F^* W_K^*,\ V^* = Z_F^* W_V^* \tag{8}$$

Each head i among the h heads computes attention on a portion of the queries, keys, and values.

$$Q_i^* = Q^* W_{Q-i}^*,\ K_i^* = K^* W_{K-i}^*,\ V_i^* = V^* W_{V-i}^* \tag{9}$$

$$head_i^* = soft\max\left(\frac{Q_i^* K_i^{*T}}{\sqrt{d_k}}\right)V_i^* \tag{10}$$

where the weight matrices W_{Q-i}^*, $W_{K-i}^* \in \mathbb{R}^{d_{model}\times d_k}$, and $W_{V-i}^* \in \mathbb{R}^{d_{model}\times d_k}$. Here, $d_{model} = D$, $d_k = d_v = \frac{d_{model}}{h}$, and h is the number of heads.

These matrices are used to project the queries, keys, and values into lower-dimensional subspaces. The outputs of all heads are concatenated and projected back to the original dimension.

$$MultiHead^* = Concat\left(head_1^*, ..., head_h^*\right)W^* \tag{11}$$

where $W^* \in \mathbb{R}^{hd_v\times d_{model}}$.

This concatenated output is then passed through a residual feed-forward network consisting of two convolutional layers with a kernel size of 1 and a residual connection:

$$Z^* = FeedForward\left(MultiHead^*\right) \tag{12}$$

where $Z^* \in \mathbb{R}^{L\times D}$, and specifically Z^L, Z^C, $Z^R \in \mathbb{R}^{L\times D}$.

3-views Concatenation: The feature representations from the three views, Z^L, Z^C, Z^R, are concatenated to form a unified output as Eq. (13):

$$Z = Concat\left(Z^L, Z^C, Z^R\right) \tag{13}$$

where $Z \in \mathbb{R}^{L\times 3D}$.

This concatenation aggregates features extracted from different viewpoints, resulting in a comprehensive representation that encapsulates essential characteristics from all three perspectives. Subsequently, Z is passed through a linear classifier, which consists of a linear layer and an additional dropout layer, to predict the input class.

C. Implementation Details

We utilize the AUTSL dataset to initialize the weights for the VTN3GCN framework and subsequently fine-tune it using the Multi-VSL200 dataset [47]. Specifically, we first initialize the weights of the AGCN and VTN models with the AUTSL data, and remove the final layers of the AGCN as described in Section III. B, and freeze the first three AGCB blocks along with five CNN layers in the VTN to prevent overfitting. These models are then fine-tuned on the Multi-VSL200 dataset. The weights of the AGCN and VTN are subsequently employed as the initial weights for the VTN3GCN framework. We employ the Adam optimizer with a weight decay of $1e^{-4}$, a learning rate of $1e^{-4}$, and a learning rate decay factor of 0.8 every 10 epochs [48]. Training is conducted until the validation loss does not decrease for 30 consecutive epochs, after which the weights corresponding to the lowest validation loss are selected as the final model. One-hot labels are utilized in conjunction with the cross-entropy loss function. The CNN is pretrained on ImageNet with $d_{model} = D = 2D_{RGB} = 1024$ [49]. The GCN employed is AGCN with $D_{kp} = 256$. For the self-attention decoder block, we use $N = 4$ self-attention blocks with $h = 8$, resulting in $d_k = d_v = d_{model}/h = 128$. An increase in N leads to less accurate predictions from the framework and a concomitant rise in computational resource requirements. All experiments are conducted using the PyTorch framework [50].

D. Evaluation Metrics

We employ top-1 and top-5 classification accuracy metrics on the test set. Each experiment is conducted with five random seeds, and we report the mean values along with the standard deviations of the metrics for each experiment. Top-1 accuracy serves as a measure of the

classification model's performance, quantifying the absolute correct prediction rate, i.e., the proportion of samples where the highest-probability label predicted by the model matches the true label. The Eq. (14) for top-1 accuracy is expressed as follows.

$$Top-1Accuracy=\frac{\sum_{i=1}^{N}1\left(y_i=y_i^{1\Delta}\right)}{N} \tag{14}$$

where N is the total number of test samples. y_i is the true label of sample i. $y_i^{1\Delta}$ is the label with the highest predicted probability for sample i, and $1(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise.

In contrast, $Top-5Accuracy$ is a metric that evaluates the frequency with which the true label appears within the top five predicted labels by the model. The formula for top-5 accuracy is as Eq. (15).

$$Top-5Accuracy=\frac{\sum_{i=1}^{N}1\left(y_i\in y_i^{5\Delta}\right)}{N} \tag{15}$$

where N, $1(\cdot)$, and y_i are defined similarly to top-1 accuracy. $y_i^{5\Delta}$ is the list of five labels with the highest predicted probabilities for sample i.

IV. Experimental Results

A. Multi-Model Data Preparation

1) RGB frames

In this study, all video frames within the dataset were standardized to a resolution of 224 × 224 pixels and converted to the RGB color space. Each video was divided into 28 segments of equal length, with one frame selected from each segment to serve as input to the framework. Each selected frame was processed to compute HC as described by Coster *et al*. [41]. To enhance the diversity of the input data and mitigate overfitting, we employed a series of data augmentation techniques prior to training. Specifically, the training videos were subjected to a random combination of methods, including random cropping, adjustments to brightness, contrast, saturation, and hue parameters. These augmentations were applied to expand and enrich the original dataset, thereby improving the model's generalization capabilities.

2) Keypoints graph

We employed RTMPose to extract and construct the keypoints graph [51]. Specifically, 46 keypoints were extracted from each video frame, including 21 keypoints for each hand, 2 keypoints for the left and right shoulders, and 2 keypoints for the left and right elbows. These keypoints were normalized according to the human body structure. From these keypoints, the keypoints graph was constructed similarly to that illustrated in Fig. 7.

However, some videos were recorded at high speeds over short durations, resulting in keypoints not being extracted for certain frames, which posed challenges for graph construction. To address this issue, we utilized the keypoints reconstruction method [52]. This method is based on bilinear interpolation, assuming that f^k represents the normalized hand keypoints at the k^{th} frame.

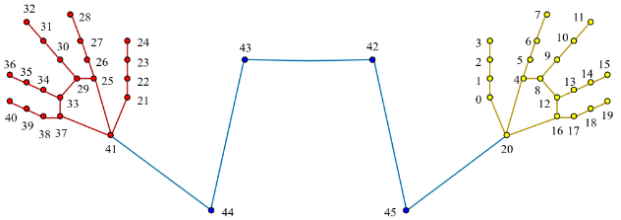

Fig. 7. Key points graph.

The bilinear interpolation equation is defined as Eq. (16):

$$f_{interpolated}^{k}=\frac{\beta f_{k-\alpha}+\alpha f_{k+\beta}}{\alpha+\beta},\ if\ f^k=0 \tag{16}$$

where α and β are the smallest integers such that $f_{k-\alpha}$ and $f_{k+\beta}$ are non-zero, i.e., $f_{k-\alpha}, f_{k+\beta}\neq 0$. If the initial frames lack extracted keypoints, they are omitted, which constitutes a limitation of this approach.

Additionally, we applied graph rotation data augmentation in a counterclockwise direction, with the center of rotation being the graph's centroid. The rotation is performed according to Eq. (17).

$$f_{rotate}=\left((x-0.5)\cos\theta-(y-0.5)\sin\theta,(y-0.5)\cos\theta+(x-0.5)\sin\theta+0.5\right) \tag{17}$$

where x and y are the coordinates of the normalized keypoints, and θ is the rotation angle.

3) Pose flow

We extracted 133 key points from RGB videos using RTMPose [51], retaining 53 key points for subsequent processing. The PF was then calculated in accordance with the methodology [41]. Each PF is stored as a NumPy file for each frame and is concatenated with the corresponding RGB frame and selected key points graph.

B. State-of-the-Art Comparison

We evaluated the performance of the VTN3GCN framework against State-of-the-Art (SOTA) models in the domains of action recognition and SLR. The selected models for comparison include I3D [25], VTN with HC-PF [40], Video Swin Transformer [53], and MViTv2 [54]. As these models are originally designed for single-view data, we adopted a methodology similar to that depicted in Fig. 2, substituting the backbones of the SOTA models within the view-specific encoder to create 3-view model variants. The single-view models were trained using comprehensive multi-view data, whereas the 3-view models were trained with data corresponding to their respective view-specific encoders. The results, presented in Table I, demonstrate that VTN3GCN achieves a top-1 accuracy that outperforms the single-view models by at

least 11.28%. When compared to the 3-view SOTA models, VTN3GCN surpasses them with a top-1 accuracy improvement ranging from a minimum of 4.93% to a maximum of 26.57%.

Upon analyzing the mispredictions of the VTN model that are not present in the VTN3GCN framework specifically the confusion between certain pairs of glosses such as glosses 0 and 132, and glosses 41 and 150, we observed that these glosses exhibit highly similar representations as shown in Fig. 8. For instance, glosses 0 and 132 appear nearly identical when viewed from the center perspective, differing only in that gloss 132 involves shaking the right hand, whereas gloss 0 maintains the right hand in a steady position. In both glosses, the left hand remains static. However, distinct differences become apparent when observed from the left and right angles: in gloss 132, the left hand moves smoothly from a low to a high position in synchronization with the right hand, whereas in gloss 0, the left hand remains unchanged.

TABLE I. THE COMPARISONS OF VTN3GCN AND SOTA MODELS

Method		Top-1 Accuracy (%)	Improve	Top-5 Accuracy (%)	Improve
I3D [25]	1-view	56.62 ± 0.0055	36.30	83.99 ± 0.0063	14.26
	3-view	66.35 ± 0.0066	26.57	90.14 ± 0.0050	8.11
Swin Trans [53]	1-view	77.29 ± 0.0083	15.63	91.81 ± 0.0064	6.44
	3-view	82.33 ± 0.0034	10.59	94.03 ± 0.0032	4.22
MViTv2 [54]	1-view	81.57 ± 0.0038	11.35	92.36 ± 0.0029	5.89
	3-view	86.45 ± 0.0026	6.47	95.55 ± 0.0031	2.70
VTN [41]	1-view	81.64 ± 0.0052	11.28	92.29 ± 0.0037	5.96
	3-view	87.99 ± 0.0051	4.93	95.50 ± 0.0017	2.75
VTN3GCN (Ours)	3-view	92.92 ± 0.0009	–	98.25 ± 0.0011	–

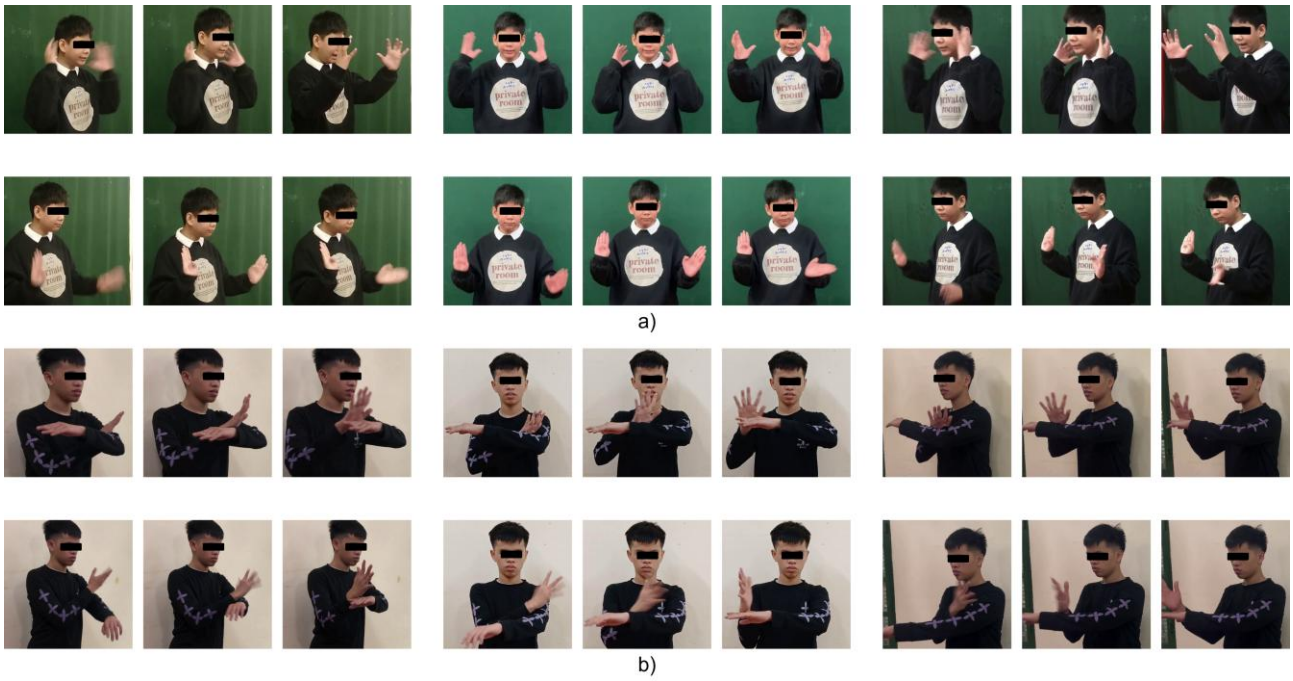

Fig. 8. Certain pairs of glosses convey different meanings but exhibit similar gestures when viewed solely from the centre perspective: a) Gloss pair number 41 (top row) and 150 (bottom row) of signer 10; b) Gloss pair number 0 (top row) and 132 (bottom row) of signer 05.

A similar pattern is observed with gloss pairs 41 and 150, where the differences are more pronounced from the left and right views rather than the center view.

From a linguistic perspective, these pairs often share core phonological parameters, such as handshape and location, differing only in a single parameter like movement (e.g., a steady hand vs. a shaking motion). Such minimal pairs are notoriously challenging for SLR systems, especially from a single viewpoint where the distinguishing movement feature may be subtle or occluded.

Specifically, for the pair “bãi cỏ” (lawn, gloss 0) and “công viên” (park, gloss 132): Both signs share the same location (chest), handshape (a straight, closed left hand and a spread right hand), and palm orientation. The sole distinction lies in the movement: “bãi cỏ” involves a smooth sliding motion (“miết”), whereas “công viên” features a wrist-shaking motion (“lắc cổ tay”). Similarly, for the pair “chính tả” (spelling, gloss 41) and “dầu ăn” (cooking oil, gloss 150): Both signs are performed at the same location (the right corner of the mouth) and use the same handshape (the tips of the index finger and thumb touching). The subtle differences are found in the movement (a small wavy path versus a diagonal outward motion) and palm orientation (inward versus outward).

These observations may explain why the VTN3GCN model, along with other 3-view variants of SOTA models, achieves higher accuracy compared to the single-view

models. Although the single-view models are trained with comprehensive multi-view data, they do not capture the motion features of sign language as effectively as the 3-view variants and VTN3GCN. Achieving the highest accuracy among all compared models indicates that the VTN3GCN framework effectively leverages multi-view information, particularly through the integration of GCN, CNN, PF, and Self-Attention Decoders tailored to each view. This integration enables the framework to more accurately distinguish between sign gestures that exhibit only minor differences when viewed from the center perspective, as illustrated by the gloss pairs 0–132 and 41–150. These results further confirm that the design of view-specific architectures and the flexible integration of the aforementioned components play crucial roles in enhancing the model's ability to learn and differentiate distinctive motion features in sign language.

C. *Self Attention Encoder Analysis*

We selected two pairs of glosses from two signers to perform a qualitative analysis of the self-attention encoder within the VTN3GCN framework.

The attention maps were utilized to visualize the attention weights, specifically extracting all eight heads in the final attention layer of the self-attention encoder corresponding to each input data view, as shown in Figs. 9 and 10. The two attention maps are displayed sequentially for the two glosses with frames arranged in temporal order. Each row in the attention map represents the weights assigned by the self-attention module to each frame, progressing from the beginning to the end of the video sequence for each head. Warm colors (green, yellow) indicate high weights, while purple or violet colors represent lower weights. Generally, brightly colored regions (yellow, green) tend to appear during prominent motion phases, such as when the signer changes hand positions, transitions arm states, or performs emphasizing actions. The presence of extended purple (or dark violet) areas indicates that the self-attention module does not focus extensively on frames with minimal gesture variations, where the signer maintains a consistent posture across multiple consecutive frames.

Within the same view, the heads have learned different aspects of motion, and similarly, the self-attention modules in different views have learned distinct aspects.

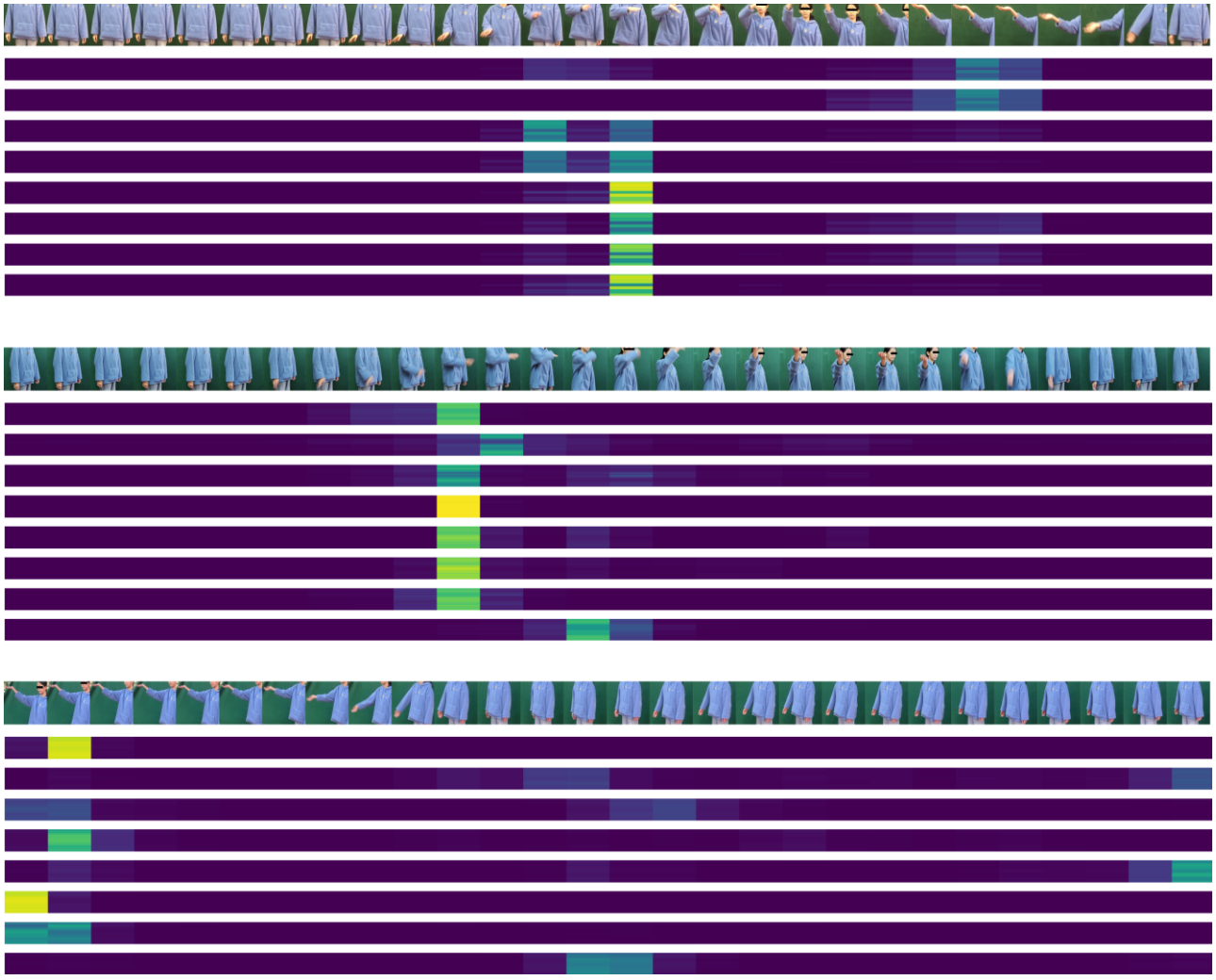

Fig. 9. Attention maps of the three data views—centre-view, left-view, and right-view from top to bottom-for gloss 0 of signer 05 in the final attention layer, encompassing all 8 heads. Full RGB frames are utilized to simplify interpretation. High weight values are represented by warm colors, whereas low weight values are depicted by cool colours.

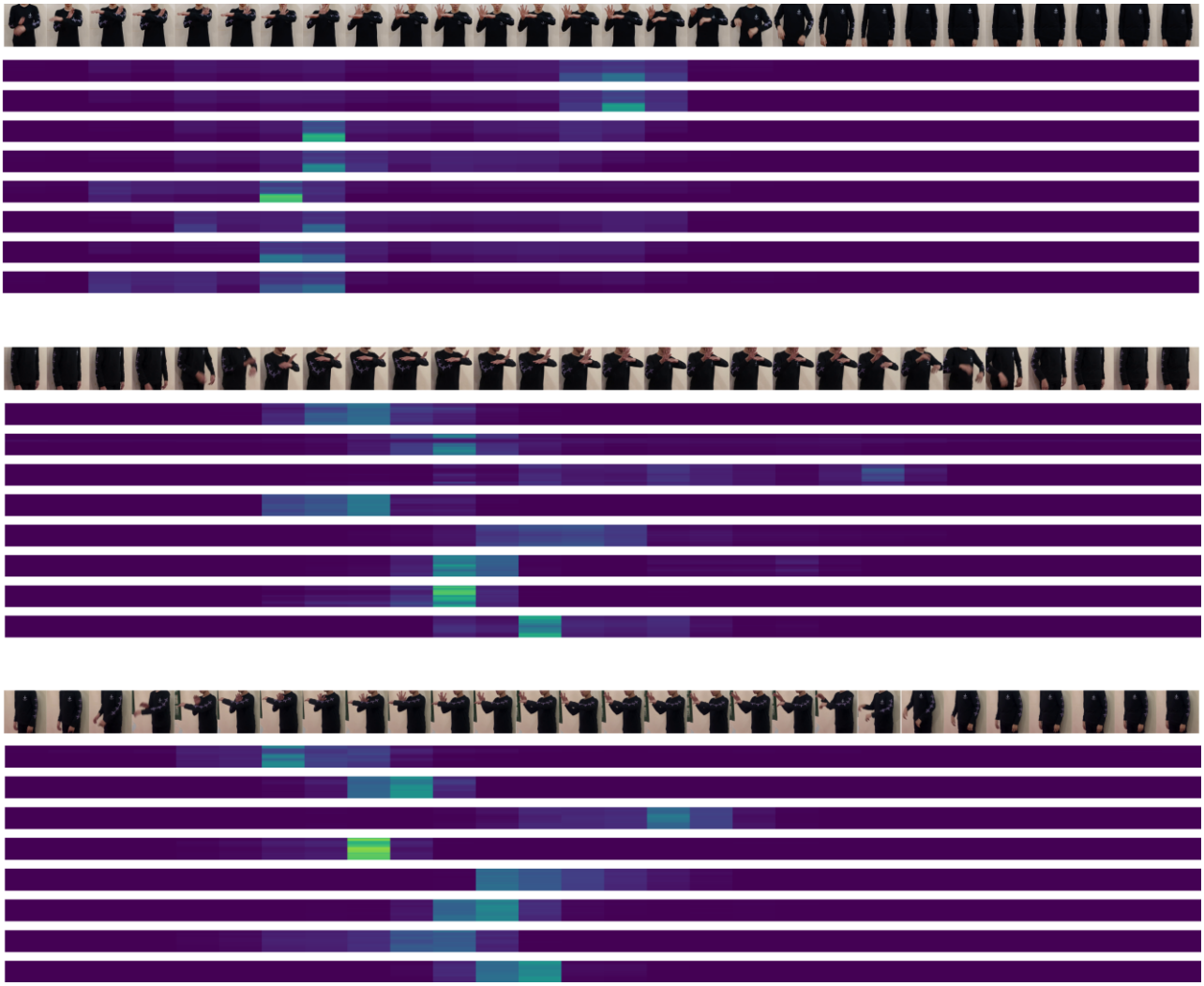

Fig. 10. Attention maps of the three data views for gloss 22 from signer 20, ordered from top to bottom as centre-view, left-view, and right-view.

Specifically, in Fig. 9, since the timelines across the three views are not significantly offset, the points assigned high weights by the three self-attention modules are concentrated at the beginning to the middle of the video. The color layout appears as blocks of green/blue/violet corresponding to adjacent frame groups. This reveals that the gloss may contain multiple similar segments, requiring the self-attention module to identify subtle distinguishing features between segments. Several frames exhibit green hues (high weights), typically located near the center of each segment, indicating that the model considers these frames as containing the most critical motion features. This suggests that the gloss is complex, as the model must track multiple arm movement phases, such as moving forward, crossing towards the chest, and extending the arm horizontally. The attention heads clearly distribute tasks, with some heads focusing on left-hand gestures, others on the right-hand crossing phase, and so forth.

In Fig. 10, it is evident that the temporal frames of the data from the three views are misaligned. Specifically, the right-view data lacks the segment where the right hand is raised and only includes the phase where the hand moves horizontally and then downward. Consequently, the self-attention module for the right view concentrates on the beginning portion of the data, while the self-attention modules for the center and left views focus on the middle portion. Some rows corresponding to certain heads show frames gradually darkening from left to right, indicating that the attention modules are tracking continuous actions rather than focusing on a single fixed moment. Other heads abruptly focus on one or two specific frames, which may coincide with the transition phase of the gloss (for example, when the signer changes the dominant hand).

Since each view is equipped with its own self-attention module, they can independently learn to detect key frames even if the timing of prominent actions does not perfectly align across videos. This means that if an important action occurs at frame 10 in the left view but at frame 12 in the right view, the two attention modules can still capture these frames independently. If a frame in one view is noisy or temporally misaligned, the corresponding attention module is only locally affected. The remaining views can still accurately identify the critical moments of the action, thereby preserving the overall information. As a result, minor desynchronizations do not degrade the system's overall accuracy.

After each self-attention module extracts features for its respective view, these feature vectors are concatenated in

the subsequent step. This allows the model to integrate all important stages from each view, even if there is a slight temporal misalignment within each view's sequence. Thanks to the independent self-attention mechanism and the final concatenation process, the model maintains consistent accuracy instead of being disrupted by frames that are not perfectly aligned. This explains why VTN3GCN continues to perform well even in real-world recording scenarios where synchronization between cameras is imperfect.

D. Missing Data Study on VTN3GCN

In this experiment, we explored the capabilities of the VTN3GCN framework under data loss scenarios involving the omission of one or two views simultaneously, with missing rates ranging from 10% to 50% in increments of 10%. We randomly replaced RGB frames, poseflow, and keypoints (where available) with zero vectors according to the specified missing rates, and the results are presented in Table II. The data from the center view emerged as the most critical, as the absence of center-view data led to a significant reduction in the framework's accuracy—specifically, a decrease of 10.27% with missing center-view data, 31.5% with both center-view and left-view data missing, and 36.89% with both center-view and right-view data missing at a 50% missing rate. The right-view data appeared to be more important than the left-view data, as missing right-view data resulted in poorer word recognition by the model. Specifically, at a 50% missing rate, the absence of right-view data caused the framework's predictions to decrease by 0.61% compared to the absence of left-view data, whereas the additional loss of center-view data resulted in a 5.39% decrease. When only one data channel was randomly missing—such as the left-view, center-view, or right-view—the framework maintained a relatively high top-1 accuracy, approximately between 89% and 92% for missing rates of 10–30%. Even with a 50% data loss rate, VTN3GCN still achieved accuracy levels above 80–90%, demonstrating the framework's robustness against single-view data loss.

In configurations LC, LR, and CR, the decline in top-1 accuracy became more pronounced as the missing data rate increased. With data loss ranging from 10% to 50%, accuracy decreased to 61.15% for LC and 55.76% for CR at a 50% missing rate. This reflects the necessity for the model to leverage complementary information from multiple views to comprehensively capture motion features. When two views were simultaneously replaced with zero vectors, the resulting information deficit created a larger "gap" compared to the loss of a single view's data. Missing data in both the left-view and right-view led to lower accuracy in the VTN3GCN model compared to the VTN with HC-PF 1-view model, specifically by 3.28%, 6.79%, 10.29%, and 13.55% at missing rates of 20%, 30%, 40%, and 50%, respectively. This can be attributed to the fact that after the features pass through the self-attention decoder, they are concatenated to generate the final prediction. The loss of information during concatenation adversely affects the linear classifier, resulting in less accurate predictions. These observations suggest that, in practical deployment scenarios, it is essential to prioritize the preservation or restoration of at least one view's data to maintain acceptable performance. Furthermore, developing models or algorithms with enhanced capabilities to integrate missing information will improve the system's resilience against significant data loss.

TABLE II. RESULTS ILLUSTRATING THE IMPACT OF DATA LOSS ACROSS DIFFERENT VIEWS ON VTN3GCN

Missing Rates (%)	Top-1 Accuracy (%)					
	L	C	R	LC	LR	CR
10	92.12 ± 0.0050	90.62 ± 0.0060	92.06 ± 0.0052	86.14 ± 0.0069	88.51 ± 0.0086	88.50 ± 0.0085
20	91.61 ± 0.0059	88.48 ± 0.0079	91.34 ± 0.0042	79.68 ± 0.0099	84.71 ± 0.0094	77.60 ± 0.0118
30	91.25 ± 0.0053	86.38 ± 0.0038	90.77 ± 0.0045	73.49 ± 0.0121	81.20 ± 0.0090	70.10 ± 0.0101
40	90.76 ± 0.0036	84.27 ± 0.0061	90.14 ± 0.0045	66.99 ± 0.0135	77.70 ± 0.0092	62.64 ± 0.0150
50	90.32 ± 0.0041	82.38 ± 0.0071	89.71 ± 0.0055	61.15 ± 0.0155	74.44 ± 0.0126	55.76 ± 0.0186

Abbreviations used in the table: L—left-view data, C—center-view data, R—right-view data.

E. Ablation Study

1) Effect of hand crop and pose flow

In this experiment, we conducted an ablation study focusing on the HC and PF components. The results, presented in Table III, demonstrate that removing the HC component from the framework leads to a substantial decrease in top-1 accuracy, dropping from 92.92% to 81.21%. This reduction of 11.71% underscores the critical importance of concentrating on the hand region to capture the intricate details essential for accurately representing sign language. This finding indicates that the inclusion of the HC mechanism is necessary for the framework to extract more precise features.

TABLE III. IMPACT OF INCLUDING/EXCLUDING HC-PF IN VTN3GCN

VTN3GCN variations	Top-1 (%)	Improve	Top-5 (%)	Improve
w/o HC	81.21 ± 0.0055	11.71	95.90 ± 0.0055	2.35
w/o PF	92.62 ± 0.0011	0.30	98.20 ± 0.0014	0.05
w/o HC-PF	84.40 ± 0.0060	8.52	96.79 ± 0.0028	1.46
w/ HC-PF	92.92 ± 0.0009	-	98.25 ± 0.0011	-

Legend: w/o – "without", w/ – "with".

When the PF component is excluded from the framework, the top-1 accuracy remains relatively high at 92.62%, only 0.3% lower than the full model. Although this decline is minimal, it still highlights that PF contributes to the framework by providing a clearer depiction of the motion trajectory or the overall posture of the signer. Compared to the removal of HC, the absence of PF has a smaller impact on accuracy but still offers a beneficial advantage.

When the PF component is excluded from the framework, the top-1 accuracy remains relatively high at 92.62%, only 0.3% lower than the full model. Although this decline is minimal, it still highlights that PF contributes to the framework by providing a clearer depiction of the motion trajectory or the overall posture of the signer. Compared to the removal of HC, the absence of PF has a smaller impact on accuracy but still offers a beneficial advantage.

However, when only the PF component is utilized—equivalent to removing the HC component—the framework experiences a more pronounced decrease in accuracy compared to the simultaneous removal of both HC and PF, with a specific accuracy drop of 3.19%. Khan *et al.* [40] introduced this phenomenon can be explained by the inherent design of PF, which aims to supplement motion information when HC is employed. In scenarios where HC is omitted, both RGB frames and skeleton coordinates independently provide information about body movement, rendering the use of PF redundant. This redundancy prevents the framework from focusing on the core features of the signers' motion, thereby diminishing recognition accuracy.

The distinction between removing only HC and removing both HC and PF illustrates that the framework's performance is influenced not only by the individual components but also by the degree of compatibility and synergy between them. The complete version of the model achieves the highest top-1 accuracy of 92.92%, outperforming all other variants and validating the complementary value of integrating feature information from both HC and PF components. This suggests that to construct an optimal model, it is essential to fully integrate mechanisms that focus on the hand region and effectively describe the body's motion trajectory

2) Effect of GCN

In Table IV, we present the experimental results of the impact of Graph Convolutional Networks (GCN) on the VTN3GCN framework. It is evident that without the use of GCN, the model achieves only 87.99% accuracy, which is significantly lower compared to the performance when GCN is applied. This demonstrates the crucial role of GCN in effectively capturing the spatiotemporal relationships between joints and key points. When GCN is applied across all three views (left, center, and right), the model's top-1 accuracy increases to 92.26%, representing a substantial improvement of 4.27% over the model without GCN. However, compared to the variant that employs GCN only on the left and right views, the accuracy is slightly lower by approximately 0.66%. In certain cases, the center view with GCN may not provide entirely new information compared to the left and right views, while simultaneously introducing additional complexity to the training process. Consequently, integrating GCN for the center view may lead to the dilution of learned weights or require the model to adjust an excessive number of parameters, resulting in less outstanding final performance.

TABLE IV. IMPACT OF INCLUDING/EXCLUDING GCN IN VTN3GCN

VTN3GCN variations	Top-1 (%)	Improve	Top-5 (%)	Improve
w/o GCN	87.99 ± 0.0051	4.93	95.50 ± 0.0017	2.75
w/ GCN in 3 views	92.26 ± 0.0021	0.66	98.56 ± 0.0011	–
w/ GCN in left, right views	92.92 ± 0.0009	–	98.25 ± 0.0011	0.31

Legend: w/o—"without", w/—"with".

On the other hand, regarding top-5 accuracy, using GCN across all three views outperforms the configuration that applies GCN solely to the left and right views by 0.31%. The inclusion of GCN for the center view allows the model to expand its prediction space and encompass a broader range of sign language possibilities. Although the information from the center view's GCN may not be optimized for top-1 predictions, it serves as a supplementary factor that increases the likelihood of the correct sign being among the top five results. The multi-view GCN model benefits from enhanced information coverage, thereby achieving better top-5 accuracy. Conversely, the addition of GCN for the center view may introduce noise or increase the training complexity, leading to top-1 accuracy that is not as high as when only the left and right views are utilized.

This suggests that for frontal (center) perspectives, the features from RGB frames and pose flow may already provide sufficient information for recognition, and adding skeleton graph features might introduce redundant or even noisy signals that slightly hinder the model's ability to pinpoint the top prediction. The additional parameters from the center-view GCN could also increase training complexity, leading to a dilution of learned weights. Future explorations could involve dynamic weighting or gating mechanisms that allow the model to learn the importance of each data stream per view, potentially optimizing the integration of GCN features more effectively.

F. Computational Cost Analysis

The multi-view and multi-stream architecture of VTN3GCN, while effective, introduces considerable computational overhead. The full model comprises approximately 156.77 million parameters as in Table V. Training was conducted on a single NVIDIA RTX 3090. For inference, the model processes a single tri-view video instance in approximately 44.7 ms, achieving a throughput

of ~22.35 FPS. While this is not real-time, it is feasible for many offline applications. Future work could explore model compression techniques, such as knowledge distillation or parameter pruning, to create a more lightweight version suitable for real-time deployment.

TABLE V. COMPUTATIONAL COST ANALYSIS

Model	Architecture	Params (M)	FPS
I3D [19]	3D ConvNet	12.2	122
Swin Transformer [21]	Hierarchical ViT	28.3	115
MViTv2 [22]	Multiscale ViT	35	40
VTN [7]	CNN + MLP + Transformer	29	61.45 ± 2.64
VTNHCPF [41]	2 × CNN + GCN + Transformer	52.05	62
VTN3GCN (3-view)	6 × CNN + Transformer	156.77	22.35

G. Ethical Considerations

The development of SLR technology carries important ethical responsibilities. In this study, we prioritized participant privacy by obtaining informed consent from all volunteers and anonymizing their faces in all published figures and videos. The Multi-VSL200 dataset was collected with the aim of capturing diverse signing styles by including both native and trained signers. However, we acknowledge the potential for algorithmic bias. If a system is trained on a limited demographic, it may not perform equally well for all users. Future work must focus on expanding datasets to be more inclusive of age, gender, ethnicity, and regional dialects within the deaf community. It is essential that the development of SLR technology is done in close collaboration with the deaf community to ensure the resulting tools are respectful, accurate, and truly beneficial.

H. Discussion and Future Work

Our analysis of data loss scenarios provided critical insights into the framework's operational dynamics. The results revealed a notable dependency on the information maintained by the central viewpoint for achieving maximum accuracy. While the VTN3GCN framework demonstrates general robustness against the absence of one or two viewpoints, the significant performance drop when center-view data is completely missing highlights a key challenge for real-world applications where this specific view may not always be reliably available.

These observations inform our future work, which will focus on mitigating this dependence and enhancing the system's resilience. Specifically, future research could explore advanced techniques such as view-invariant feature learning, adversarial training aimed at reducing reliance on any single view, or the development of more sophisticated data synthesis mechanisms to reconstruct missing viewpoints. Furthermore, we plan to further enhance the model by expanding its input capabilities to incorporate additional data modalities, such as depth images, to improve the overall generalizability and accuracy of the SLR system across unconstrained environments.

V. CONCLUSION

In this study, we successfully introduced the VTN3GCN framework, a novel multi-view and multi-stream approach for SLR. This framework was designed to jointly exploit and integrate RGB data, skeleton data, and pose flow captured from three distinct viewpoints: left, right, and center. Experimental results on the Multi-VSL200 dataset robustly demonstrate that our multi-view integration strategy significantly improves performance, elevating top-1 accuracy to 92.92%, which substantially surpasses the original VTN architecture and various single-view baseline methods. These findings underscore the immense potential of combining multi-channel data and multi-view strategies within the SLR domain, thereby broadening the applicability of such systems in real-world environments where viewpoints and environmental conditions are frequently variable. Although this study focused on VSL, the proposed VTN3GCN framework is inherently language-agnostic. The core components-CNN for visual features, GCN for skeletal structure, and self-attention for temporal dynamics-are generalizable and not specific to VSL, suggesting that our multi-view, multi-stream approach offers a robust and generalizable solution for the broader SLR field.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Hoang Quang Huy conceived the idea of the paper, conducted the research, developed the software and wrote the paper; Tran Anh Vu developed the methodology, performed the formal analysis and examined the data. All authors had approved the final version.

ACKNOWLEDGMENTS

This research is funded by the Ministry of Education and Training (MOET) under grant number B2026.VKG.09: "Research on the application of technology to support teaching Vietnamese sign language to primary school students with students with deaf or hard of hearing".

REFERENCES

[1] S. Auephanwiriyakul, S. Phitakwinai, W. Suttapak, P. Chanda, and N. Theera-Umpon, "Thai sign language translation using scale invariant feature transform and hidden markov models," *Pattern Recognit. Lett.*, vol. 34, no. 11, pp. 1291–1298, 2013. doi: 10.1016/j.patrec.2013.04.017

[2] J. Lin and Y. Ding, "A temporal hand gesture recognition system based on hog and motion trajectory," *Optik*, vol. 124, no. 24, pp. 6795–6798, 2013. doi: 10.1016/j.ijleo.2013.05.097

[3] T. E. Starner, "Visual recognition of American sign language using hidden Markov models," *Program in Media Arts and Sciences*, 1995.

[4] Y. Chen, R. Zuo, F. Wei, Y. Wu, S. Liu, and B. Mak, "Two-stream network for sign language recognition and translation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17043–17056, 2022

[5] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6201–6210. doi: 10.1109/ICCV.2019.00630

[6] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12018–12027. doi: 10.1109/CVPR.2019.01230

[7] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[8] M. Vazquez-Enriquez, J. L. Alba-Castro, L. Docio-Fernandez, and E. Rodriguez-Banga, "Isolated sign language recognition with multi-scale spatial-temporal graph convolutional networks," in *Proc. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 3457–3466. doi: 10.1109/CVPRW53098.2021.00385

[9] H. Zhou, W. Zhou, Y. Zhou, and H. Li, "Spatial-temporal multi-cue network for continuous sign language recognition," in *Proc. AAAI Conference on Artificial Intelligence*, 2020, vol. 34, no. 07, pp. 13009–13016. doi: 10.1609/aaai.v34i07.7001

[10] R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training," *IEEE Trans Multimedia*, vol. 21, no. 7, pp. 1880–1891, 2019. doi: 10.1109/TMM.2018.2889563

[11] J. Ahn, Y. Jang, and J. S. Chung, "Slowfast network for continuous sign language recognition," in *Proc. ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 3920–3924. doi: 10.1109/ICASSP48485.2024.10445841

[12] Q. Xiao, M. Qin, and Y. Yin, "Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people," *Neural Networks*, vol. 125, pp. 41–55, 2020. doi: 10.1016/j.neunet.2020.01.030

[13] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *Proc. 2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2871–2875. doi: 10.1109/ICIP.2016.7532884

[14] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, "Skeleton aware multi-modal sign language recognition," in *Proc. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 3408–3418. doi: 10.1109/CVPRW53098.2021.00380

[15] M. Suneetha, M. V. D. Prasad, and P. V. V. Kishore, "Sharable and unshareable within class multi view deep metric latent feature learning for video-based sign language recognition," *Multimed. Tools Appl.*, vol. 81, no. 19, pp. 27247–27273, 2022. doi: 10.1007/s11042-022-12646-0

[16] S. N. Dinh, T. D. Nguyen, D. T. Tran *et al*., "Sign language recognition: A large-scale multi-view dataset and comprehensive evaluation," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025, pp. 7876–7886.

[17] J. L. Raheja, A. Mishra, and A. Chaudhary, "Indian sign language recognition using SVM," *Pattern Recognition and Image Analysis*, vol. 26, no. 2, pp. 434–441, 2016. doi: 10.1134/S1054661816020164

[18] C. C. D. Amorim, D. Macêdo, and C. Zanchettin, "Spatial-temporal graph convolutional networks for sign language recognition," in *Proc. International Conference on Artificial Neural Networks*, 2019, pp. 646–657. doi: 10.1007/978-3-030-30493-5_59

[19] A. Tunga, S. V. Nuthalapati, and J. Wachs, "Pose-based sign language recognition using GCN and BERT," in *Proc. 2021 IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2021, pp. 31–40. doi: 10.1109/WACVW52041.2021.00008

[20] O. Mercanoglu Sincan and H. Y. Keles, "Using motion history images with 3D convolutional networks in isolated sign language recognition," *IEEE Access*, vol. 10, pp. 18608–18618, 2022. doi: 10.1109/ACCESS.2022.3151362

[21] N. Sarhan, C. Wilms, V. Closius, U. Brefeld, and S. Frintrop, "Hands in focus: Sign language recognition via top-down attention," in *Proc. 2023 IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 2555–2559. doi: 10.1109/ICIP49359.2023.10222729

[22] X. Shen, Z. Zheng, and Y. Yang, "StepNet: Spatial-temporal part-aware network for isolated sign language recognition," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 20, no. 7, pp. 1–19, 2024. doi: 10.1145/3656046

[23] Ç. Gökçe, O. Özdemir, A. A. Kındıroğlu, and L. Akarun, "Score-level multi cue fusion for sign language recognition," in *Proc. European Conference on Computer Vision*, 2020, pp. 294–309. doi: 10.1007/978-3-030-66096-3_21

[24] Y. F. Song, Z. Zhang, C. Shan, and L. Wang, "Richly activated graph convolutional network for robust skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1915–1925, 2021. doi: 10.1109/TCSVT.2020.3015051

[25] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733. doi: 10.1109/CVPR.2017.502

[26] Z. Liu, Z. Li, R. Wang, M. Zong, and W. Ji, "Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition," *Neural Comput. Appl.*, vol. 32, no. 18, pp. 14593–14602, 2020. doi: 10.1007/s00521-020-05144-7

[27] N. Sun, L. Leng, J. Liu, and G. Han, "Multi-stream slowfast graph convolutional networks for skeleton-based action recognition," *Image Vis. Comput.*, vol. 109, 104141, 2021. doi: 10.1016/j.imavis.2021.104141

[28] L. Wang, L. Ge, R. Li, and Y. Fang, "Three-stream CNNs for action recognition," *Pattern Recognit. Lett.*, vol. 92, pp. 33–40, 2017. doi: 10.1016/j.patrec.2017.04.004

[29] K. Gadzicki, R. Khamsehashari, and C. Zetzsche, "Early vs late fusion in multimodal convolutional neural networks," in *Proc. 2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 2020, pp. 1–6. doi: 10.23919/FUSION45008.2020.9190246

[30] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *Proc. CVPR 2011*, 2011, pp. 489–496. doi: 10.1109/CVPR.2011.5995624

[31] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, "Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2273–2286, 2011. doi: 10.1109/TPAMI.2011.52

[32] J. Zheng, Z. Jiang, and R. Chellappa, "Cross-view action recognition via transferable dictionary learning," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2542–2556, 2016. doi: 10.1109/TIP.2016.2548242

[33] Y. Kong, Z. Ding, J. Li, and Y. Fu, "Deeply learned view-invariant features for cross-view action recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 3028–3037, 2017. doi: 10.1109/TIP.2017.2696786

[34] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool, "Domain generalization and adaptation using low rank exemplar SVMs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1114–1127, 2018. doi: 10.1109/TPAMI.2017.2704624

[35] D. Wang, W. Ouyang, W. Li, and D. Xu, "Dividing and aggregating network for multi-view action recognition," in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 457–473. doi: 10.1007/978-3-030-01240-3_28

[36] K. Shah, A. Shah, C. P. Lau, C. M. D. Melo, and R. Chellapp, "Multi-view action recognition using contrastive learning," in *Proc. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 3370–3380. doi: 10.1109/WACV56688.2023.00338

[37] M. Kowal, M. Siam, M. A. Islam, N. D. B. Bruce, R. P. Wildes, and K. G. Derpanis, "A deeper dive into what deep spatiotemporal networks encode: Quantifying static vs. dynamic information," in *Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 13979–13989. doi: 10.1109/CVPR52688.2022.01361

[38] S. Vyas, Y. S. Rawat, and M. Shah, "Multi-view action recognition using cross-view video prediction," in *Proc. European Conference on Computer Vision*, 2020, pp. 427–444. doi: 10.1007/978-3-030-58583-9_26

[39] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Unsupervised learning of view-invariant action representations," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[40] M. A. Khan, K. Javed, S. A. Khan *et al.*, "Human action recognition using fusion of multiview and deep features: an application to video surveillance," *Multimed. Tools Appl.*, vol. 83, no. 5, pp. 14885–14911, 2020. doi: 10.1007/s11042-020-08806-9

[41] M. D. Coster, M. Van Herreweghe, and J. Dambre, "Isolated sign recognition from RGB video using pose flow and self-attention," in *Proc. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 3436–3445. doi: 10.1109/CVPRW53098.2021.00383

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90

[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 770–778.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[45] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[46] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv Preprint, arXiv:1607.06450, 2016.

[47] O. M. Sincan and H. Y. Keles, "AUTSL: A large scale multi-modal turkish sign language dataset and baseline methods," *IEEE Access*, vol. 8, pp. 181340–181355, 2020. doi: 10.1109/ACCESS.2020.3028072

[48] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," arXiv Preprint, arXiv:1412.6980, 2014.

[49] J. Deng, W. Dong, R. Socher *et al.*, "ImageNet: A large-scale hierarchical image database," in *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848

[50] A. Paszke, S. Gross, S. Chintala *et al.*, "Automatic differentiation in PyTorch," *NIPS 2017 Workshop Autodiff*., 2017.

[51] T. Jiang, P. Lu, L. Zhang *et al.*, "RTMPose: Real-time multi-person pose estimation based on MMPose," arXiv Preprint, arXiv:2303.07399, 2023.

[52] K. Roh, H. Lee, J. Hwang, S. Cho, and J. C. Park, "Preprocessing mediapipe keypoints with keypoint reconstruction and anchors for isolated sign language recognition," in *Proc. LREC-COLING 2024, 11th Workshop on the Representation and Processing of Sign Languages*, 2024, pp. 323–334.

[53] Z. Liu, J. Ning, Y. Cao *et al.*, "Video Swin Transformer," in *Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3192–3201. doi: 10.1109/CVPR52688.2022.00320

[54] Y. Li, C. Y. Wu, H. Fan *et al.*, "MViTv2: Improved multiscale vision transformers for classification and detection," in *Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4794–4804.