

Article

Adaptive Transformer-Based Deep Learning Framework for Continuous Sign Language Recognition and Translation

Yahia Said ^{1,2,*} , Sahbi Boubaker ³ , Saleh M. Altowaijri ⁴ , Ahmed A. Alsheikhy ⁵  and Mohamed Atri ⁶¹ Center for Scientific Research and Entrepreneurship, Northern Border University, Arar 73213, Saudi Arabia² King Salman Center for Disability Research, Riyadh 11614, Saudi Arabia³ Department of Computer & Network Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah 21959, Saudi Arabia; sboubaker@uj.edu.sa⁴ Department of Information Systems, Faculty of Computing and Information Technology, Northern Border University, Rafha 91911, Saudi Arabia; saleh.altowaijri@nbu.edu.sa⁵ Department of Electrical Engineering, College of Engineering, Northern Border University, Arar 91431, Saudi Arabia; aalsheikhy@nbu.edu.sa⁶ College of Computer Sciences, King Khalid University, Abha 62529, Saudi Arabia; matri@kku.edu.sa

* Correspondence: yahia.said@nbu.edu.sa

Abstract: Sign language recognition and translation remain pivotal for facilitating communication among the deaf and hearing communities. However, end-to-end sign language translation (SLT) faces major challenges, including weak temporal correspondence between sign language (SL) video frames and gloss annotations and the complexity of sequence alignment between long SL videos and natural language sentences. In this paper, we propose an Adaptive Transformer (ADTR)-based deep learning framework that enhances SL video processing for robust and efficient SLT. The proposed model incorporates three novel modules: Adaptive Masking (AM), Local Clip Self-Attention (LCSA), and Adaptive Fusion (AF) to optimize feature representation. The AM module dynamically removes redundant video frame representations, improving temporal alignment, while the LCSA module learns hierarchical representations at both local clip and full-video levels using a refined self-attention mechanism. Additionally, the AF module fuses multi-scale temporal and spatial features to enhance model robustness. Unlike conventional SLT models, our framework eliminates the reliance on gloss annotations, enabling direct translation from SL video sequences to spoken language text. The proposed method was evaluated using the ArabSign dataset, demonstrating state-of-the-art performance in translation accuracy, processing efficiency, and real-time applicability. The achieved results confirm that ADTR is a highly effective and scalable deep learning solution for continuous sign language recognition, positioning it as a promising AI-driven approach for real-world assistive applications.

Keywords: sign language translation (SLT); deep learning; adaptive transformer; self-attention mechanism; multimodal feature fusion; computer vision; natural language processing (NLP); deaf communication assistance

MSC: 68T07; 68T50; 68T10



Academic Editors: Juan Manuel Rendón-Mancha, Edgar Roman-Rangel and Junlin Hu

Received: 2 February 2025

Revised: 26 February 2025

Accepted: 4 March 2025

Published: 8 March 2025

Citation: Said, Y.; Boubaker, S.; Altowaijri, S.M.; Alsheikhy, A.A.; Atri, M. Adaptive Transformer-Based Deep Learning Framework for Continuous Sign Language Recognition and Translation. *Mathematics* **2025**, *13*, 909. <https://doi.org/10.3390/math13060909>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The majority of the world's deaf population uses sign language (SL) as their primary means of communication. It differs from spoken language in its distinctive qualities and its ability to transmit visual information via the coordinated use of many organs (such as the

hands, body, lips, and facial expressions). The hearing-challenged are severely limited in their social opportunities due to the difficulty of non-disabled individuals understanding this language due to its intricate body motions and grammatical reasoning. Therefore, if SL can be automatically translated into normal language, it can certainly lessen the communication load on the hearing impaired in their everyday life. Generally, there are two approaches to executing SL translation (SLT): one is the end-to-end method, and the other is the two-stage method with continuous SL recognition (CSLR) and gloss to text (G2T).

Continuous SL films may be identified as gloss sequences by CSLR and then translated into SL sentences using the G2T. Additionally, it is capable of producing SL sentences and glosses in tandem using CSLR and SLT multi-task joint learning. Nearly all recent SL research has been on CSLR and SLT with glosses, whereas very little has focused on end-to-end SLT that does not use glosses [1–5]. The most important factor is that the sign motions in the SL video match the gloss sequences. Learning the temporal correspondence between SL videos and gloss sequences can help CSLR and SLT models with the syntactic alignment problem and improve SLT performance through two-stage or multi-task joint learning. Having said that, gloss annotations are exclusively the domain of SL specialists. On the other hand, end-to-end SLT eliminates the need for glosses and can immediately convert continuous SL films into sentences.

Furthermore, end-to-end SLT primarily needs to improve the overall mapping relationship between SL videos and sentences, as opposed to CSLR's emphasis on explicitly aligning SL video frames with glosses [6] and enhancing short-term temporal information [7]. Since end-to-end SLT is easily transferable to other SL datasets, it offers a broader variety of applications. But end-to-end SLT is not easy for a few reasons: (i) SL videos have longer frames and more complicated corpus information than gloss sequences; (ii) SL videos and gloss sequences use different grammar rules for natural language sentences; and (iii) end-to-end SLT is a weakly supervised problem because continuous SL videos do not have boundary annotations for SL action transitions.

This paper focuses on end-to-end SLT with the goal of improving the translation performance of continuous SL videos without using glosses in training, while also addressing the challenges of obtaining gloss annotations and the strong migration of end-to-end SLT methods [8]. Before introducing the Transformer network [9] to SLT, Camgöz et al. [10] showed good performance in order to increase the translation quality of SLT. In the time after, the Transformer was used as the SLT backbone network in numerous studies, which enhanced it in various ways. These included attention mechanisms, gloss-text joint learning, network pre-training, various data inputs (such as SL video clips, multi-cue images, etc.), and network pre-training. However, while some of the aforementioned approaches in [2–5] perform end-to-end SLT without glosses, the majority of them concentrate on improving SLT networks that rely on them.

In addition, the majority of approaches only take one kind of data into account, which means they do not address issues like data convergence across sources or the generalization problem that arises when working with extended sequences and gloss erosion. Because of the importance of SL and many semantic organs working together to transmit visual information, the relative position information and temporal consistency of SL semantic organs are often compromised when using multi-cue images (e.g., images of hands, faces, or bodies) or conventional data enhancement techniques (e.g., random cropping, flipping, or scaling).

The solution to improve end-to-end SLT performance without glosses lies in enhancing network generalization and making greater use of SL movies. This provides motivation to investigate video representation learning (VRL) techniques, which contain a wealth of

literature on video processing, feature and model generalization augmentation, and have been the subject of substantial study [11–14]. By improving the video feature representation, these approaches not only learn all visual information using full-size frames, but they also increase performance on a range of downstream tasks. If you want to use a big batch size for your data inputs, though, most VRL algorithms would need a lot of processing power. Long SL videos also have a big frame size, which makes training more challenging. In order to increase the translation impact by boosting its generalization potential, we attempt to simplify certain efficient VRL algorithms and include them into the SLT network.

Our proposal for learning robust continuous SL video feature representations is an adaptive video representation enhanced Transformer (ADTR). Adaptive fusion (AF), Local Clip Self-Attention (LCSA), and Adaptive Masking (AM) are three more modules additionally proposed to enhance model capability. We begin by presenting an AM module that is based on the VideoMoCo Generator [13]; this module can give a unique mask for adaptively removing the feature representation from video frames, which will make SL films more feature resilient. To improve the video's quality, we use the BiLSTM [15] of the AM module to extract the time-related information from SL movies, and then we use the mask it generates to remove frames that are not crucial to the story's progression. The AM module can be integrated either before or after the encoder and decoder, depending on preference, because the Transformer encoder does not ruin the input feature's dimensions.

It is worth noting that there are no matching annotations for supervised training in the AM module. We imitate the Discriminator of VideoMoCo by passing the temporal feature of the AM module to the Transformer decoder; this eliminates semantic ambiguity, which may arise from raising the dropout thresholds of video frame representations. Next, we use spoken translation sentences as pseudo-labeling to provide a weakly supervised loss constraint for the decoding results of the AM module, which stabilizes the dropout effect. We incorporate an LCSA into the Transformer encoder to improve the network's capacity to learn local semantics from SL videos, as opposed to the common video clip partition (CCP) in [3,6] and the utilization of short-term nearby frames [7]. It is able to extend keyframe characteristics from several adjacent clips after clip-level splitting of continuous video features.

After that, in order to improve the local information of each clip, intercross attention (ICA) is set. The encoder may learn both the local and global information of SL films through the relationship between LCSA and disguised self-attention, given that the AM module's temporal feature and the encoder's output feature are both decodable, and that these two features differ in their temporal and spatial learning. Instead of relying just on simple concatenation, as in [7,14], we provide an AF module that utilizes GRF [16] and AFA [17] to adaptively combine the two features and produce a strong feature representation. Our approach improves the end-to-end SLT effect without employing gloss annotations in training by allowing the Transformer to learn more spatio-temporal information, thanks to the more robust feature representation.

Additionally, we conduct an in-depth investigation and evaluate the experimental outcomes of several end-to-end SLT approaches on the Arab sign dataset. This paper's main contributions are as follows:

- To increase translation performance and relieve the poorly guided difficulty of end-to-end SLT, we suggest an adaptive video representation enhanced Transformer (AVRET), which is both simple and effective. The Transformer-based network may accommodate three more AVRET modules at no additional cost;
- We built the first CSL video dataset based on a news corpus, CSL-FocusOn, and describe and offer a method for collecting CSL videos. It is easy to extend and has a vast corpus of contents.

The following is the outline for the remainder of the paper. Section 2 discusses relevant research on SLT and VRL. Section 3 details the ADTR model architecture and the steps used to build the model with proposed additional modules. We describe the model's implementation and ablation analysis in Section 4, and then we show the experimental findings and compare them to other models that were used for the same purpose. Section 5 serves as the paper's conclusion.

2. Related Works

Systems for recognizing and understanding sign language have been the focus of much research. Algorithms with sign recognition and classification capabilities are required for this purpose. Machine learning and deep learning are two methods that scholars have investigated as potential solutions to this problem of sign language recognition. Sign language recognition has made use of both deep learning and more conventional machine learning techniques. To accomplish recognition tasks, SURF, SIFT, PCA, LDA, and HOG have been integrated with traditional ML algorithms including random forests, k-nearest neighbors, and support vector machines. Also showing promise in sign language recognition are learning approaches like transformer models, recurrent neural networks (RNNs), and convolutional neural networks (CNNs). CNNs are great at capturing hierarchies and correlations between picture features, and RNNs are great at handling dynamic data.

Attention network designs based on RNNs were the mainstay of early SLT research [1,2]. Due to its capacity to effectively resolve the long-term dependence problem of RNN models and further improve their context learning capabilities, researchers have been using BiLSTM to SLT [18,19] since its introduction [15]. A Transformer, which is widely used in NLP, dramatically enhances the efficiency and quality of different sequence translation tasks. The Transformer network is based only on attention mechanisms and feed-forward layers.

Furthermore, an increasing number of studies are enhancing and applying it to computer vision (CV) tasks, with promising results in areas such as online anomaly detection [20], video captioning [14], image captioning [17], oriented object recognition in remote sensing images [21], and more. So, to prove that cooperatively learning works, Camgöz et al. [10] used the Transformer network on SLT first, and they obtained good translation results. A PiSLTRc model, as suggested by Xie et al. [22], enhances the Transformer network by disentangled relative position encoding and content and position-aware temporal convolution. From the vantage point of low-latency simultaneous SLT, Yin et al. [23] suggested a boundary predictor to mimic the relationship between SL videos and words.

In order to carry out video-based SLT utilizing multi-cue images, Zhou et al. [19] suggested a spatio-temporal multi-cue network (STMC-T). By employing multi-modality transfer learning (MMTLB), Chen et al. [24] conducted pre-training on several datasets. The semantic information of SL was learned by Kan et al. [25] using a hierarchical spatio-temporal graph neural network (HST-GNN) that represented the SL semantic organs. An SLT contrastive framework at the token level (ConSLT) was developed by Fu et al. [26]. Zhou et al. [27] used the SignBT approach and parallel data to train SLTs with enormous amounts of spoken text. To achieve gloss-free end-to-end SLT, Li et al. [3] suggested a TSPNet that, with a fixed clip frame size, may acquire discriminative SL video characteristics through the semantic hierarchy between clips.

To aid the model's understanding of SL videos through knowledge transfer, Yin et al. [4] presented a gloss attention GASLT. This technique enables the attention to concentrate on video clips with shared local semantics. By merging contrastive language-image pre-training (CLIP) with masked self-supervised learning, GFSLT VLP [5] is able to pre-train a model and then import its previous knowledge into an SLT framework, enhancing the SLT impact. The majority of SLT approaches increase performance; however, this is due

to glosses or model pre-training. Due to the lack of correspondence gloss restriction and previous knowledge, the network's generalization and translation performance are reduced when it is not participating in training.

In addition, the aforementioned techniques seldom incorporate research into specific frame selection and data augmentation methodologies; instead, they utilize all video frames and a singular data input approach. On the other hand, we streamline and include VRL techniques to boost translation performance through expanded model generalizability and improved feature robustness. In their study on the video captioning assignment, Yan et al. [14] developed a global-local framework (GLR) that can encode video clips at multiple ranges to enhance linguistic expression. This framework may be used to evaluate how various data inputs can be utilized with SLT. We include their study concept with SL features and transfer it to our approach as SLT is also part of the video captioning work.

Hence, we provide the Transformer encoder with the capability to learn both local and global video data by adding a clip-level feature learning module, LCSA. Down order to zero, down on the temporal features of consecutive video frames, and video representation learning (VRL) approaches rely heavily on unsupervised learning. In their study, Pan et al. [13] used a VideoMoCo that could adaptively remove a number of crucial frames from the initial video sequence using a Generator (G). Then, the Discriminator (D) would be given both the full frame video and the video of the dropped frames in order to learn feature representations that are comparable. Using sample rate order prediction, Huang et al. [28] investigated information about the temporal context. Speed, random, periodic, and warp were the four video temporal modification methods examined and assessed by Jenni et al. [29]. By assigning positive and negative clip samples within the same video and negative clip samples on different videos, Tao et al. [30] created an inter-intra contrastive framework that relies on self-supervised contrastive learning. This allows the model to learn more discriminative temporal information. Additionally, by including multi-modal input (such as optical flow [31], audio [32], and text [33]), certain multi-modal based learning algorithms may learn additional video information.

To enhance language expressiveness, VRL has several particular applications. For video captioning, Yan et al. [14] suggested a GLR framework that can encode video representations at multiple ranges (e.g., long-range, short-range, and local-keyframe). We streamline a few techniques from VideoMoCo and GLR, taking into account the SL visual properties and the original VRL approaches' complexity. In contrast to VideoMoCo, we use an adaptive mask generation process to remove video frames and their feature representations from frames. To mimic VideoMoCo's D, we employ the Transformer decoder. In addition, we merge the short-range clips with local-keyframe to improve the neighbor information, instead of GLR's method of extracting video representations from three ranges and fusing them via concatenation. Furthermore, by including extra LCSA and AF modules, we may achieve feature representations that are more robust.

3. Proposed Approach

3.1. Overview

The SL Transformer is an encoder-decoder network that is mostly used for sequence-to-sequence SLT activities. The encoder has the ability to transform the n -frame SL video sequences $(x_1, \dots, x_n)$ into a sequential feature representation $\gamma \in \mathbb{R}^{n \times dm}$, where dm is the feature dimension. In an auto-regressive fashion, the decoder can produce a sentence sequence $S = \{w_u\}_{u=1}^U$ with a conditional probability $p(S|\gamma)$ after acquiring γ . The

stackable masked self-attention (MSA) is the central component of the SL Transformer. The usual MSA formula is presented in Equation (1).

$$MSA(Q, K, V) = \text{concat}(h_1, \dots, h_h) W \quad (1)$$

$$h_i = A(QW_i^Q, KW_i^K, VW_i^V)$$

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{dm}}\right)V$$

where $W \in \mathbb{R}^{dm \times dm}$ and $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{dm \times \frac{dm}{h}}$ are the projection parameters matrices, and h_i is head attention.

For a more in-depth explanation of the Transformer network, readers can refer to [10]. We omit this description since our work is focused on enhancing the network with additional modules to provide more reliable SL video representations. In addition to the three main modules indicated in Figure 1, our model also includes an AM, LCSA, and AF module. First, when given an n-frame SL video sequence ($x_1, \dots, x_n$), the spatial representation is extracted via sign embedding and then each frame is joined to a continuous video feature. Following this, the video feature's temporal information is captured by the AM module's BiLSTM. The resulting temporal feature and mask are then sent to the AF module and Transformer encoder, respectively. To top it all off, an encoder that has both LCSA and MSA can pick up both local and global data all at once. Next, the AF module adaptively fuses the encoder's spatial feature with the AM module's temporal feature, and the second AM module masks the fused feature. The Transformer decoder then uses the masked feature and sentence sequences for inference and sequence learning, resulting in the SL sentence.

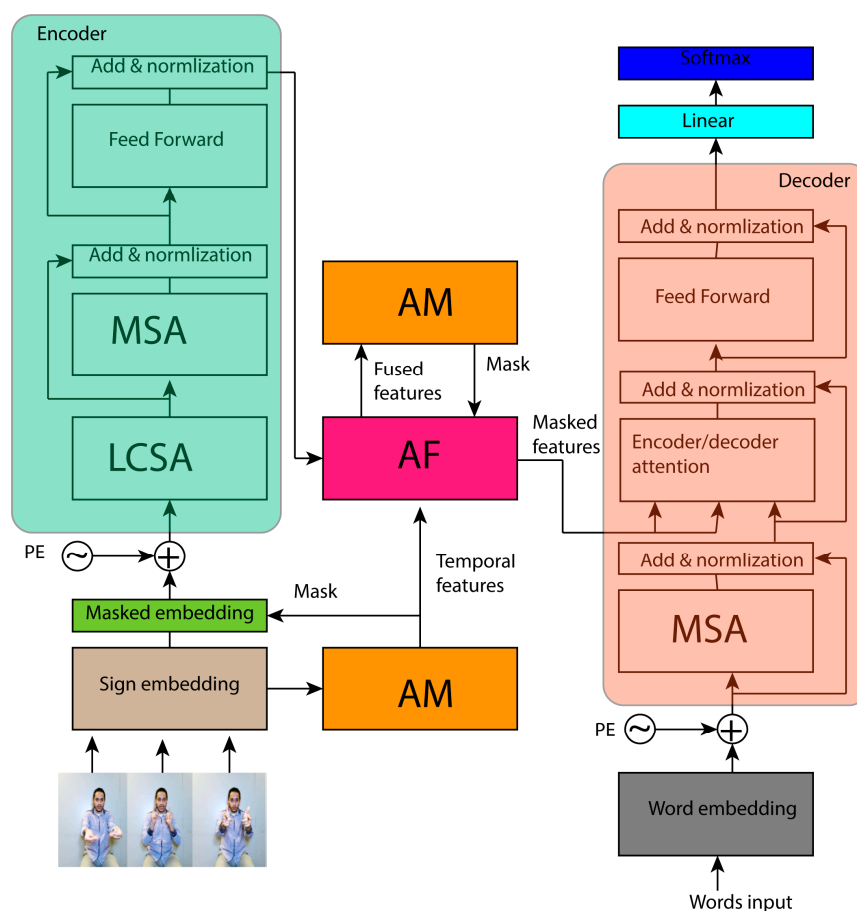


Figure 1. Overview of the proposed ADTR.

The Adaptive Fusion (AF) module receives the temporal feature and a mask as output from the bottom Adaptive Masking (AM) module. The mask can remove video frame representations. The AM module's temporal feature and the encoder's output feature can be adaptively fused by the AF module. After that, the decoder receives a masked version of the fused feature that was created by the top AM module, after which it is transmitted on. Positional Encoding is abbreviated as PE.

3.2. Adaptive Masking

The AM module must analyze the SL video features $F_v \in \mathbb{R}^{n \times dm}$ that were retrieved by sign embedding before they can be sent to the encoder, as illustrated in Figure 2. Generating a unique mask for F_v to remove the frame feature representations is performed in order to provide a continuous temporal feature $F_{ct} \in \mathbb{R}^{n \times dm}$. We begin by introducing the AM module's temporal feature extraction layer, and then we go into depth about how to acquire crucial frame sequences using the temporal feature. This feature allows us to adaptively remove representations of frame features that are not relevant to the current time. Incorporating VideoMoCo into the Transformer model will significantly increase model parameters due to the two-stage feature extraction that it requires. Hence, we use the pre-trained SL video feature representation F_v in place of the sign embedding layer.

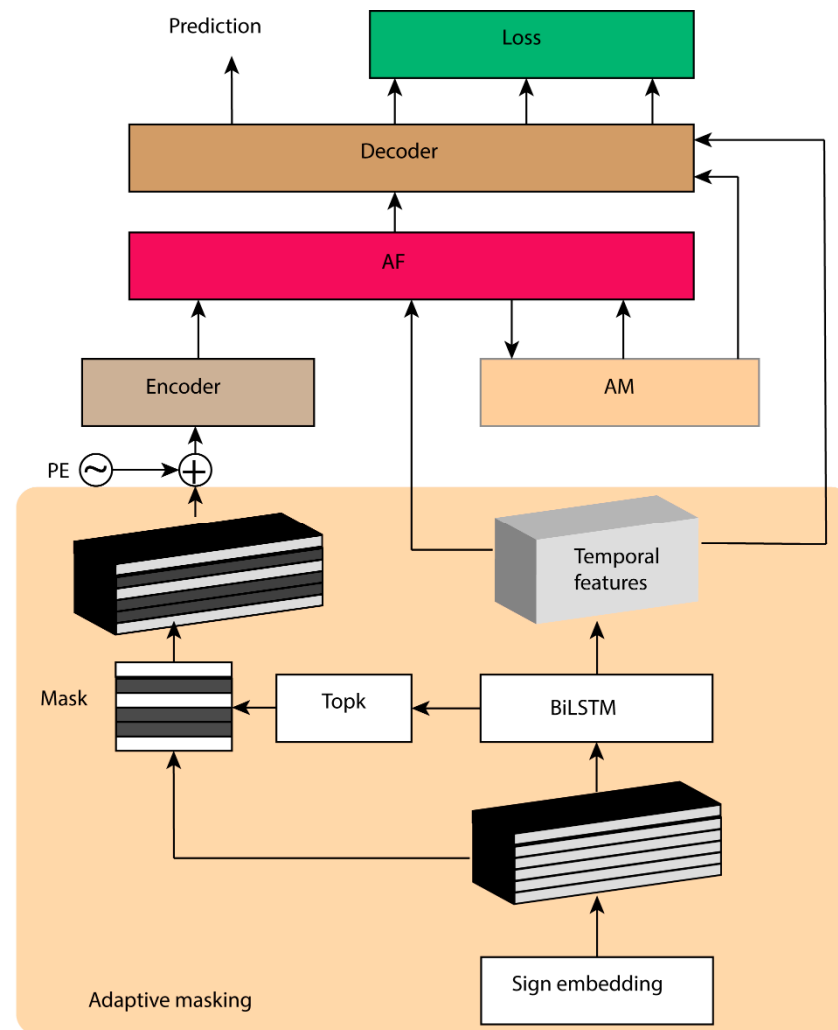


Figure 2. Adaptive masking module.

The sign embedding is passed through the BiLSTM to generate temporal features. The *topk* function is used to filter features for concatenation with the masked embeddings. The position embedding (PE) is used to pass the features on to the encoder.

We see that BiLSTM has been successful in learning the long-term relationships and temporal semantic information of video sequences, and that the original input's feature dimension is unaffected by the continuous temporal feature that it extracts in its temporal feature extraction layer. Hence, its probability distribution can be used to obtain the index numbers of masked video frames. In order to construct a continuous temporal feature for sign embedding, we initially employ the BiLSTM. The relevant probability distribution for each video frame representation can be generated by using a linear layer to map the dimension of the temporal feature to one dimension. The linear map features $F_{lm} \in \mathbb{R}^{n \times 1}$ can be computed as in Equation (2).

$$F_{lm} = \text{linear}(\text{BiLSTM}(F_v)) \quad (2)$$

This step is pivotal because it allows the model to dynamically select frames based on their temporal semantic information, ensuring that less informative or redundant frames are masked out. One of the primary advantages of this approach is that it preserves the original feature dimensions while efficiently leveraging long-term dependencies captured by the BiLSTM. This helps maintain the integrity of the continuous temporal features, enabling more robust sign embedding and improved downstream translation performance. Additionally, this targeted masking mechanism reduces noise and redundancy in the input, leading to enhanced generalization and more precise learning of both local and global temporal characteristics.

After generating the linear feature map, the softmax function was applied to compute the probability distribution $P_{dl} \in \mathbb{R}^n$ as in Equation (3).

$$P_{dl} = \text{softmax}(F_{lm}) \quad (3)$$

This plays a crucial role by transforming the raw linear feature map into a normalized probability distribution. This conversion ensures that the unbounded scores derived from the BiLSTM and linear mapping are scaled into probabilities that sum to one, providing each video frame with a relative importance score. Such normalization is essential for robust frame masking, as it allows the model to systematically identify and select the most informative frames while mitigating the influence of any extreme values. Moreover, the differentiable nature of the softmax function enables smooth gradient propagation during backpropagation, which is vital for effective end-to-end training. Overall, this equation enhances the stability and interpretability of the temporal features, ultimately leading to more precise and discriminative sign language translation performance.

Lastly, in order to construct the mask embedding $F_{me} \in \mathbb{R}^{n \times dm}$, the original mask is modified based on the index numbers where the top *k* biggest values in P_{dl} are located. The mask embedding is computed as in Equation (4).

$$F_{me} = \text{mask}(I_{me}, F_v) \quad (4)$$

$$I_{me} = \text{topk}(P_{dl}, k)$$

The mask embedding plays a crucial role in refining the input video features by constructing a mask embedding that emphasizes the most informative frames. This equation first identifies the indices corresponding to the top *k* highest values in the probability distribution (which quantifies the importance of each frame). Then it uses these indices to selectively mask the original feature representation. The key advantages of this approach

include its ability to dynamically and adaptively focus on the most relevant temporal information, thereby reducing noise and redundancy from less significant frames. This targeted selection not only enhances the discriminative power of the temporal features but also improves the robustness and efficiency of the overall feature representation, leading to better downstream performance in tasks such as sign language translation.

To find the k greatest values and related index numbers of the input matrix, the $\text{topk}(\text{input}, k)$ function was used. Any tensor $I_{me} \in \mathbb{R}^k$ that contains k index numbers is represented by k . The $\text{mask}(\text{index}, \text{input})$ function can specify which SL video inputs to mask. With the exception of manually setting the value of k in advance, the updated mask can be trained to achieve a stable state by adaptively adjusting its parameters. In addition, studies have used BiLSTM as a foundational network for SLT decoding and inference [9,13]. Hence, BiLSTM's temporal characteristic can be decoded. Consequently, following the BiLSTM layer, we incorporate a normalization layer. The AF module and decoder are fed by this layer's output. The continuous temporal feature $F_{ct} \in \mathbb{R}^{n \times dm}$ can be computed as in Equation (5).

$$F_{ct} = \text{layer normalization} (\text{BiLSTM}(F_v)) \quad (5)$$

Although it is important to note that the output feature of the Transformer encoder does not destroy the feature dimension of the original input, it is important to maintain this fact. On account of the fact that this is the case, the AM module can be positioned in the middle of the encoder and the decoder in order to exclude particular feature representations from frames before sending them on to the decoder.

3.3. Local Clip Self-Attention

In order to improve the SL video features at the clip-level, we implement a Local Clip Self-Attention (LCSA) to learn local semantic information, as demonstrated in Figure 3. This section begins with an introduction to the clip partition (CP) approach, which can divide the initial continuous video feature into several clips. Next, we go into depth about inter-cross attention (ICA), a method that lets each clip learn the interrelationships across frames through local sparse feature interactions. Last but not least, the model can use both local and global information to obtain better feature representations through the integration of LCSA with MSA.

To create N distinct clip features, the mask embedding V is clip partitioned. The inter-cross attention module uses the feature split and dimensional transformation (FSDT) technique to create a local clip enhanced feature from each clip feature X , which comprises S frames. Last but not least, the clip reverse and concatenate method returns all clip features to continuous feature state, allowing for further global learning.

Numerous studies [5,8,14,25] commonly use a non-overlapping sliding window as data input to split a full movie into numerous chunks. But finding the right window size for dividing video segments is challenging. This is mostly because setting the frame size in a single clip according to SL characteristics is necessary, and a single SL video typically involves numerous complex sign actions. On top of that, end-to-end SLT places an emphasis on overall semantic information, which will be lost if the clips are too sparse. Thus, we substitute long-range clips with original continuous video and fuse several keyframes from neighbor clips with the last frame of short-range clips; this sets our CP approach apart from the common CP (CCP) [5,8,14] and GLR [25].

In particular, we divide the video representation F_{mk} obtained via sign embedding into N segments using a sliding window size of w and a stride s . We proceed by expanding the feature representations of m keyframes for every clip. With a stride size of t and an extension frame size of m , the first and last clips are backward and forward extended, respectively. With an extension frame size of $m/2$ and a stride size of t , the central clips are

extended to both sides. Consequently, there are m extended frames in each clip's frame size, $S = w$. It adds neighbor information and alleviates semantic ambiguity problems caused by incorrect clip size and boundary by supplementing each clip with sparse keyframes from neighbor clips. In order to learn global contextual information, the standard MSA permits spatial mixing of the complete input sequence of geographical locations. However, the features in the sequence do not interact with one another in any meaningful way.

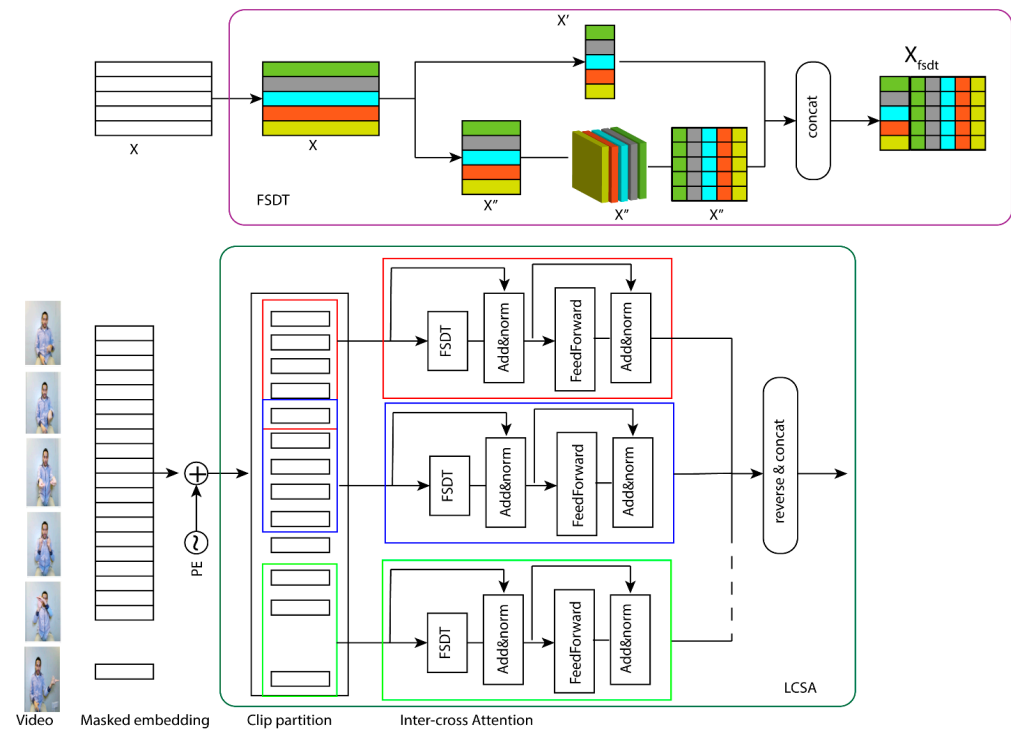


Figure 3. Local clip self-attention. The color-coded boxes represent distinct processing streams in the Inter-Cross Attention (ICA) module, where each stream consists of an FSDT block, followed by Add & Norm, a FeedForward layer, and another Add & Norm operation.

In order to address this problem, we present an MSA-based ICA that decomposes the feature axis, splits the full-size feature into two sparse-size features, and, following dimensional modification and concatenation, produces an information-interactive augmented feature. Both MSA and the feature split and dimensional transformation (FSDT) method make up ICA, as illustrated in Figure 3. To be more precise, the feature axis is defined as the last dimension dm of each clip feature $X \in \mathbb{R}^{s \times dm}$, with S representing the size of the clip frame. After that, we divide X into $X' \in \mathbb{R}^{s \times p}$ and $X'' \in \mathbb{R}^{s \times h}$ on the feature axis dm . Then, we decompose the dimension of X'' to $X'' \in \mathbb{R}^{s \times d \times \frac{h}{d}}$ on the feature axis H , where dm is equal to $(P + H)$ and D is equal to S . Sparse feature size is represented by D . Afterwards, X'' is changed to be a dimension that belongs to the set $\mathbb{R}^{D \times H \times D \times S}$. In conclusion, the final $X_{fsdt} \in \mathbb{R}^{s \times dm}$, where $dm = (p + \frac{h}{d}s)$, is obtained by applying the concatenation function to X' and X'' . X_{fsdt} can be obtained by computing the FSDT as in Equation (6).

$$X_{fsdt}(s \times dm) = X_{fsdt}\left(s, p + \frac{h}{d} \times s\right) \quad (6)$$

$$X_{fsdt}\left(S, P + \frac{h}{d} \times S\right) = \text{concat}(X', X'')$$

$$X''(s, h) \rightarrow X''\left(s, d, \frac{h}{d}\right) \rightarrow X''\left(d, \frac{h}{d} \times s\right)$$

$$X(s, d_m) \rightarrow X''(s, h) \& X'(s, p)$$

The modified clip feature X'' is then passed into the MSA. We enrich the features within each clip locally by applying ICA to them. In order to learn both local and global information, we mix LCSA and MSA to build the Transformer encoder. This is performed after reversing and concatenating the clip features. The final result of the proposed process can be computed as in Equation (7).

$$f_{enc} = MSA(CRC(LCSA(CP(PE(F_{me})))))) \quad (7)$$

The output features of the decoder is denoted by f_{enc} and the clip reverse and concatenation are denoted by CRC.

3.4. Adaptive Fusion

By integrating the BiLSTM and Transformer, Yin et al. [18] improved the model's ability to learn the semantic information and long-term dependencies of SL video sequences, leading to superior translation outcomes. In contrast, without the use of glosses during training, even the most basic techniques for these two networks—network connection, feature concatenation [25], matrix-vector addition [8], etc.—do not result in improved SLT. Not to mention that the AM module's temporal feature and the encoder's output feature are both decodable, but they're distinct in terms of the spatial and temporal learning. Hence, a fresh approach would be to let the model measure the crucial feature information of both networks independently and then merge them so that the model can learn both spatial and temporal information better at the same time. We provide an Adaptive Fusion (AF) module that is built upon GRF [27] and AFA [13]. We streamline GRF's memory mechanism and adapt it to the requirements of spatio-temporal feature fusion by taking into account its gate mechanism, computational complexity, and extensibility.

The improved spatio-temporal characteristics F_{enc} are produced by the encoder using its LCSA and MSA mechanism, after which the F_{me} produced by the AM module is passed to it (Figure 3). The AF module then takes F_{enc} and F_{ct} , a temporal characteristic, and uses them for Adaptive Fusion. Additionally, in contrast to the adaptive gate mentioned in reference [13], we create adaptive weights, $\alpha, \beta \in [0, 1]$, for every feature in order to amplify its crucial data. The fusion features F_{ff} can be computed as in Equation (8).

$$F_{ff} = \alpha \odot F_{enc} + \beta \odot F_{ct} \quad (8)$$

$$\alpha, \beta = \sigma(F_{enc}W_{enc} + F_{ct}W_{ct})$$

where σ is the sigmoid function, $\odot$ is the elementwise function, and W_{enc} and W_{ct} are the parameters to learn.

In the end, we perform a matrix-vector addition with the F_{ff} and add a normalization layer to have a better understanding of F_{ct} and F_{en} . Then, we send it to the second AM module AM2, which uses its masked feature $F_{dec} \in \mathbb{R}^{n \times dm}$ as the decoder input. The decoder features can be computed as in Equation (9).

$$F_{dec} = AM2(F_{ffn}) \quad (9)$$

$$F_{ffn} = layer\ normalization(F_{enc} \oplus F_{ct} \oplus F_{ff})$$

3.5. Loss Function with Joint Design

Considering a sentence $S = \{w_n\}_{n=1}^N$ composed of N words, the positional encoding of the word embedding $\widehat{w}_n$ can be computed as in Equation (10).

$$\widehat{w}_n = WE(w_n) + PE(n) \quad (10)$$

The SLT model that is based on Transformers is an auto-regressive encoder-decoder. Prior to decoding, the target SL sentence S is initially prepended with the special sentence-initial character, “<bog>”. The masked self-attention layer in the decoder is then fed the $S' = \{\widehat{w}_n\}_{n=1}^N$. Until it produces a unique sentence-final token, “<eod>”, the Transformer decoder learns to generate words incrementally during inference by using previously generated words in an auto-regressive fashion. The initial conditional probability at the sentence level, $p(S|V)$ will be broken down into ordered word level conditional probabilities during inference. The probability can be computed as in Equation (11).

$$p(S|V) = \prod_{n=1}^N p(w_n|h_n) \quad (11)$$

$$h_n = \text{decoder} \left(\widehat{w}_{n-1} | \widehat{w}_{1:n-2} \right), F_{dec}$$

So, we can determine the translation loss during training based on the cross-entropy loss that can be computed as in Equation (12).

$$L = L_{tr} = 1 - \prod_{n=1}^N \sum_{m=1}^M p(\widehat{w}_n^m) p(w_n^m|h_n) \quad (12)$$

where the target word $\widehat{w}_n^m$ is associated with the probability $p(\widehat{w}_n^m)$ at step n and size of the sentence vocabulary M .

There is a possibility of a more severe semantic ambiguity issue when raising the dropout threshold of video frame representations due to the AM module’s lack of matching mask annotations for supervised training, which could lead to an unstable dropout impact. So, using Equation (12), we augment the AM module with a weakly supervised loss term L_{am} . The fundamental idea is to transmit the AM module’s temporal feature to the Transformer decoder for decoding, treating it as a decodable feature representation. We model VideoMoCo’s Discriminator after using it for the decoder. It should be noted that while the decoder is shared by both L_{am} and L_{tr} , L_{am} is utilized solely for AM module weakly supervised training and is not engaged in inferring target phrases. In addition, we must configure two loss terms, L_{am1} and L_{am2} , for the two AM modules that we have added to the network. Loss term of the AM module before the encoder is represented by L_{am1} , while loss term of the AM module between the encoder and decoder is represented by L_{am2} . The final loss function L_{fn} is computed as in Equation (13).

$$L_{fn} = L_{tr}\lambda_{tr} + L_{am1}\lambda_{am1} + L_{am2}\lambda_{am2} \quad (13)$$

4. Experiments and Results

4.1. Dataset

Video sentences annotated in both Arabic and English with an ArSL context make up the ArabSign dataset [34]. Multiple recording sessions were conducted by six signers, resulting in a total of ten hours and thirteen minutes of ArSL. Experts in ArSL translation from the Al-Jazeera television network created the tutorial videos that form the basis of this dataset. A total of 50 ArSL sentences are represented by the 9335 samples that make up the dataset. There was a total of six signers for the dataset. At least thirty times throughout separate sessions, each signer was asked to recite each sentence. The signatories are all

men of various ethnicities and skin tones. The signatories are in the age bracket of 21–30. One signer was sporting spectacles, while the rest of the signers are all right-handed. There are 155 signs in the dataset’s phrases and 95 signs in the dataset’s vocabulary. More over 40% of the signals in the dataset appeared less than 5 times, as indicated in the figure. The dataset is suitable for testing real-time recognition systems because it contains many unique indications or indicators that appear several times. About ten hours and thirteen minutes’ worth of sentences have been recorded. A sentence’s length in terms of signs and the signer’s speed determines how long it takes to sign. Sentences typically consist of 3.1 signals on average. The signing speed was used to record the dataset. There was a seamless transition between each sentence’s signing, which led to about 200,000 frames for all sentences executed by a single signer, with a mean of 130.3 frames per phrase. The frames over the sentence level clips for one signer are shown in this figure. The suggested ArabSign dataset’s statistics are shown in Table 1.

Table 1. Statistics of the ArabSign dataset.

RGB resolution	1920 × 1080
Number of signers	6
Depth resolution	512 × 424
Vocabulary size	95
Minimum video duration	1.3 s
Maximum video duration	10.4 s
FPS	30
Average words/sample	3.1
Repetitions/sentence	≥30
Body joints	21
Total hours	10.13
Total Samples	9335

4.2. Implementation Details

For the purpose of extracting SL video features, we employed various networks. When employing S3D, the temporal convolution blocks that take the role of sign embedding are as follows: two Conv1D-BN1D-ReLU-MaxPooling1D layers with a stride size of 1 and a kernel size of 3, and the output size is $(B \times n/4 \times 512)$. With a 0.1 dropout rate, 8 heads, 3 layers, and 2048 feed forward sizes, the Transformer encoder and decoder are identical. There are 16 bits for the clip sliding window, 3 bits for the stride, 4 bits for the extended frame, and 13 bits for the extension stride in the LCSA module. (S, P, H, D) are configured to be (16, 128, 384, 24) and dm is set to 512. We used a value between 0 and 10 for the AM module’s frame dropout threshold $\{k', k''\}$. In the AM module, BiLSTM uses a hidden state size of 512. PyTorch was used to implement the network’s module components.

We initially set the batch size to 16 and used the adamW optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.998$, and a weight decay of 10^{-3} in all of our experiments. The plateau learning rate scheduling was used to lower the learning rate, with a drop factor of 0.7 and a minimum learning rate of 10^{-6} . In order to prepare the model for training, we subject our ADTR network to the corresponding visual-language pre-training (VLP) [21] tasks using the training sets from the Arabsign datasets. In addition, we use visually mask pre-training (VMP) tasks, which are influenced by VLP, to train ADTR to complete pre-trained SLT tasks using SL films that are randomly masked. We have adjusted the mask frame size to 5. It should be noted that model pre-training is exclusively performed when comparing

gloss-free end-to-end SLT approaches with the gloss-free characteristics of the three SL datasets. We use a GTX 960 GPU for all of our SLT research. During both the training and validation phases of our studies, we employ greedy search to decode phrases written in SL. In the inference phase, we decipher the test set by applying the beam search technique and the length penalty. We elaborate on the data augmentation methods applied to the sign language videos, including random cropping, rotation, scaling, and horizontal flipping, to enhance the model's robustness and ensure better generalization across different sign language scenarios. These augmentation techniques provide variations in signer behavior and environmental conditions.

4.3. Results and Comparison Study

In Table 2, we show that our method is effective by reporting the ADTR translation outcomes on the end-to-end SLT task compared to existing models. To make things more equitable, we split the translation results from each dataset into two categories: gloss-free and gloss-based, based on the features that were used. Noting that gloss annotations are utilized by SignBT [31], ConSLT [30], XmDA [35], and MMTLB [24] in order to enhance their SLT training. As demonstrated in the gloss-free findings on ArabSign, GFSLT VLP [5] significantly improves the performance of Transformer-based SLT networks through masked self-supervised pre-training with visual language supervision learning. In order to pre-train and fine-tune features in ADTR, we used the VLP. According to Table 3, our ADTR-VLP outperforms the competition models. Averting the use of glosses during training, ADTR outperforms SLTT-S2T on end-to-end SLT and obtains better BLEU4 scores (+3.57 on the dev set and +4.76 on the test set) in terms of gloss-based results. As far as state-of-the-art SLT approaches are concerned, ADTR performs competitively. It dominates the test set and both sets of data, even beating out HST-GNN and XmDA. We are, however, positioned marginally lower in the training set than SignBT in ROUGE. When it comes to the gloss-free findings, ADTR ranks better than SimulSLT on the dev set and the test set. Plus, thanks to VLP in AVRET, we obtain an additional 9.34 in BLEU 4 on the test set, which is a very big improvement. When it comes to the gloss-based outcomes, the baseline model that trains without sign back-translation is BN-TIN-Transf [12]. Although our method outperforms BN-TIN Transf and has a bigger improvement, it is not as strong as SignBT and MMTLB. We started with the most basic SLT in our model, which is more effective when no additional annotations are involved in the training. In contrast, these techniques rely on gloss and multi-modality pre-training to increase performance.

Table 2. Obtained results on the ArabSign dataset for the proposed ADTR and state-of-the-art models.

Model	Gloss Free									
	Dev					Test				
	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4
Multi-channel [36]	44.59				19.51	43.57				18.5
STMC-T [19]	39.76	40.73	29.42	22.61	18.21	39.82	39.07	26.74	21.86	15.74
SimulSLT [23]	36.38	36.21	23.88	17.41	13.57	35.88	37.01	24.70	17.98	14.10
RNN bahdanau [1]	31.80	31.87	19.11	13.16	9.94	31.80	32.24	19.03	12.83	9.58
RNN luong [1]	32.60	31.58	18.98	13.22	10	30.70	29.86	17.52	11.96	9
Multitask-T [2]						36.28	37.22	23.88	17.08	13.25
TSPNet-Joint [3]						34.96	36.10	23.12	16.88	13.41

Table 2. Cont.

Model	Gloss Free									
	Dev					Test				
GASLT [4]						39.86	39.07	26.74	21.86	15.74
ADTR-VLP (ours)	47.92	47.01	35.03	27.84	22.91	46.91	46.89	34.84	27.48	22.73
With gloss										
SLTT-S2T [10]		45.54	32.60	25.30	20.69		45.34	32.31	24.83	20.17
SignBT [27]	50.29	51.11	37.90	29.80	24.45	49.54	50.80	37.75	29.72	24.32
PiSLTRcS2T [22]	47.89	46.51	33.78	26.78	21.48	48.13	46.22	33.56	26.04	21.29
HSTGNN [25]		46.10	33.40	27.50	22.60		45.20	34.70	27.50	22.79
ConSLT [26]							48.73	36.53	29.03	24
XmDA [35]	48.05				22.90	47.33	46.84	34.69	27.50	22.79
MMTLB [34]	45.84	47.31	33.64	25.83	20.76	45.93	47.40	34.30	26.47	21.44
ADTR (ours)	50.28	50.37	37.91	29.84	24.53	50.89	51.49	38.72	30.44	24.93

Table 3. Impact of the AM module with different thresholds and dropouts.

{k1, k2}	Dev	Test
{0, 0}	22.74	22.98
{0, 2}	23.62	23.22
{0, 4}	23.77	23.51
{0, 6}	22.26	22.79
{0, 10}	20.54	20.93
{2, 0}	23.18	23.73
{2, 2}	23.89	24.31
{2, 4}	24.53	24.91
{2, 6}	22.78	23.01
{4, 0}	23.31	23.76
{4, 2}	23.56	23.28
{4, 4}	21.28	21.66
{r1, r2}	Dev	Test
{0, 4}	22.45	22.67
{2, 0}	22.12	21.85
{2, 4}	22.61	22.96
{4, 2}	23.42	22.78

4.4. Discussion

The proposed Adaptive Transformer (ADTR) framework and its ADTR-VLP variant achieve state-of-the-art results in sign language translation (SLT). In the gloss-free SLT setting, ADTR-VLP significantly outperforms existing models, achieving a BLEU4 score of 22.73 on the test set, compared to GASLT (15.74), STMC-T (15.74), and SimulSLT (14.10). This highlights the model's ability to translate sign language videos into spoken language text without relying on gloss annotations. In the gloss-based SLT setting, ADTR attains the highest BLEU4 score of 24.93, surpassing SignBT (24.32), PiSLTRcS2T (21.29), and HSTGNN (22.79), confirming the effectiveness of our Adaptive Transformer architecture in leveraging

gloss annotations for improved translation accuracy. Furthermore, ADTR achieves the highest ROUGE score (50.89), reflecting its superior semantic alignment with reference translations. Figure 4 provides a visualization of the obtained results in comparison with state-of-the-art models.

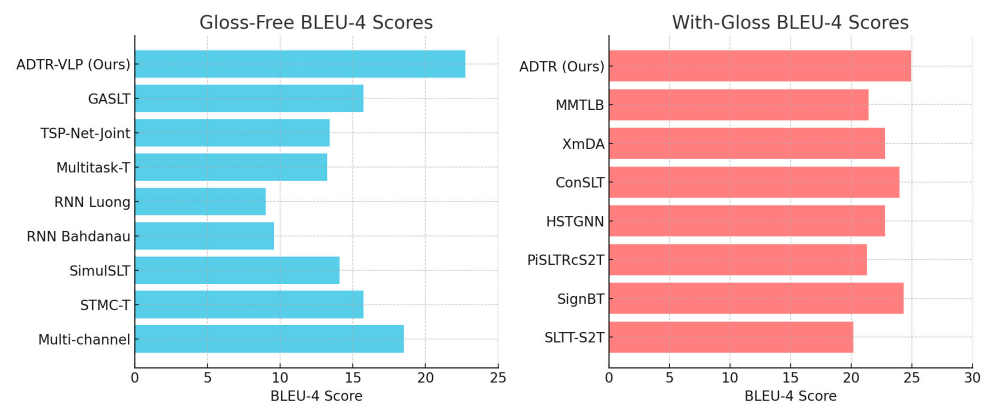


Figure 4. Comparison against state-of-the-art models in terms of BLEU4 score.

Comparative analysis shows that ADTR-VLP leads among gloss-free models, while ADTR outperforms all gloss-based models, demonstrating that our feature optimization modules (Adaptive Masking, Local Clip Self-Attention, and Adaptive Fusion) improve translation quality and feature representation. Unlike previous models that heavily depend on gloss annotations, ADTR excels in direct video-to-text translation, making it more suitable for real-world assistive applications. The gap between ADTR-VLP and ADTR (~2 BLEU4 points) suggests that while gloss annotations enhance translation accuracy, our gloss-free approach remains highly competitive, reinforcing its versatility.

The proposed ADTR framework demonstrates superior performance over existing SLT models, offering high translation accuracy and adaptability in both gloss-free and gloss-based settings. However, further optimizations in computational efficiency, cross-lingual adaptability, environmental robustness, and multimodal integration will be explored to enhance real-world usability and scalability.

Despite these advancements, ADTR has some limitations. Computational complexity for long sign language videos remains a challenge, as self-attention mechanisms in Local Clip Self-Attention (LCSA) demand high computational resources. Future work will explore sparse or memory-efficient transformers to optimize computational efficiency. Another limitation is sensitivity to variations in environmental conditions, such as lighting and camera angles, which may impact translation accuracy. To address this, we plan to augment the training data with diverse scenarios to improve robustness. Finally, while ADTR focuses on video-based sign language translation, some recent SLT models integrate multimodal inputs (video, audio, or text) to enhance context understanding. Incorporating multimodal fusion techniques in future work could further boost performance.

The proposed AM module already contributes to reducing redundant frame representations, thereby lowering processing costs. However, we acknowledge that further refinement of the masking strategy, such as learning-based sparsification, could improve efficiency. On the other hand, self-attention is computationally expensive for long sequences. We are considering alternatives such as sparse attention mechanisms or memory-efficient recurrent strategies to reduce complexity while maintaining translation quality.

To further optimize inference efficiency, we plan to explore knowledge distillation to transfer knowledge from ADTR to a lightweight student model. Additionally, structured pruning techniques can be applied to remove redundant parameters, reducing computational overhead without sacrificing accuracy. Also, the model can be optimized for

resource-constrained devices by leveraging quantization and low-precision computation techniques to accelerate inference while maintaining model effectiveness.

Recent advancements in Sign Language Translation (SLT) have increasingly incorporated multimodal datasets that integrate video, textual gloss annotations, and sometimes audio cues to improve translation accuracy. While these approaches benefit from multiple data sources, they introduce challenges such as higher computational costs, annotation dependencies, and synchronization complexities between different modalities. In contrast, our proposed Adaptive Transformer (ADTR) framework focuses solely on video-based sign language recognition using the ArabSign dataset, offering a lightweight, efficient, and scalable solution in real-world applications where only visual data is available. This approach eliminates the need for gloss annotations, making it particularly useful for under-represented sign languages where labeled data is scarce.

Our experimental results demonstrate that ADTR and ADTR-VLP outperform or remain competitive with multimodal models, despite using only visual features. For instance, our ADTR model achieves a BLEU4 score of 24.93 on the test set, surpassing gloss-free SLT models like SimulSLT (14.10) and STMC-T (15.74). Even compared to multimodal approaches such as SignBT (24.32 BLEU4) and PiSLTRcS2T (21.29 BLEU4), ADTR maintains strong performance, highlighting the effectiveness of our novel feature extraction modules: Adaptive Masking (AM), Local Clip Self-Attention (LCSA), and Adaptive Fusion (AF). These modules enable effective temporal and spatial feature learning, compensating for the absence of supplementary textual or audio input.

Beyond accuracy, our vision-only model offers several practical advantages. Firstly, it ensures independence from gloss annotations, which are often labor-intensive and language-dependent, thus enhancing the model's adaptability across different sign languages. Secondly, the computational efficiency of ADTR makes it suitable for real-time applications in wearable assistive devices, mobile applications, and low-resource settings. Unlike multimodal approaches that require synchronizing multiple input sources, our model processes video streams directly, ensuring faster inference times and reduced dependency on high-quality annotated corpora. Additionally, our method enhances robustness to diverse signers and environments, while multimodal models may suffer from missing modality issues when certain inputs (e.g., speech cues) are unavailable.

Despite its strong performance, our approach has limitations. Some complex sentence structures in sign language rely on facial expressions and spatial references, which could be better captured using depth-aware vision models or motion tracking mechanisms. Additionally, while our current framework operates in a purely visual setting, future extensions could explore lightweight multimodal enhancements, such as hand shape classification and facial emotion recognition, to further refine translation accuracy without significantly increasing computational overhead. Another crucial direction is cross-language generalization, where the model could be fine-tuned on additional datasets from different sign languages (e.g., ASL, BSL, CSL) to further validate its applicability.

This work demonstrates that a well-optimized, video-only SLT model can achieve state-of-the-art performance, rivaling multimodal approaches while remaining computationally efficient and scalable. The elimination of gloss annotations and other linguistic dependencies makes our ADTR framework an ideal candidate for real-world assistive applications. Future research will focus on enhancing spatial representation learning, exploring lightweight multimodal extensions, and improving generalization across diverse sign languages, ensuring that the model continues to evolve as a robust, AI-driven communication tool for the deaf and hearing communities.

4.5. Ablation Study

To evaluate the effectiveness of the proposed modules, an ablation study was conducted on the ArabSign dataset using the BLEU 4 metric.

Table 3 shows the results of our analysis of the efficacy of several video frame representation dropout methods on ArabSign. These methods include random dropout, two AM modules in ADTR, and others. The AM dropout threshold $\{k1, k2\}$ and the random dropout threshold $\{r2, r2\}$ are the experimental variables. Here, $k1$ and $r1$ mark the dropout threshold before the feature is sent to the encoder, and $k2$ and $r2$ mark the dropout threshold before the feature is sent to the decoder. According to the average frame size in all videos, we established the range of $\{k1, k2\}$ from 0 to 10, with 0 indicating that the AM module is not employed. The BLEU4 score on the ArabSign dataset shows a consistent upward trend as $\{k1, k2\}$ values range from 0 to 4, with the best SLT outcomes obtained at $\{2, 4\}$. The BLEU4 score begins to drop sharply, though, as $k1$ is raised to even higher values. Setting $\{k1, k2\}$ to $\{0, 10\}$ and $\{4, 4\}$ respectively results in a smaller SLT impact compared to $\{0, 0\}$. Both AM modules have a pivotal role in determining the dropout threshold, according to the experimental results. If the values of $\{k1, k2\}$ are appropriate, SLT performance can be enhanced, but if they are very big, vital information can be lost, negatively impacting the ultimate SLT effect. The optimal $\{k1, k2\}$ on ArabSign, according to our findings, is $\{2, 2\}$.

The Adaptive Masking (AM) module significantly influences the training of the ADTR model by selectively dropping video frame features at different processing stages. This study evaluates its impact on sign language translation (SLT) performance using different dropout thresholds $\{k1, k2\}$, where $k1$ controls feature dropout before encoding and $k2$ before decoding. The results show that in the early training phase, the model struggles to extract meaningful representations if the dropout threshold is too high ($k1 > 4$). Excessive feature masking at this stage prevents the model from learning robust patterns, leading to poor generalization and unstable convergence. However, with a moderate dropout threshold ($\{2, 2\}$), the model effectively learns to focus on essential gesture features while avoiding overfitting to redundant information, resulting in a steady increase in BLEU4 scores.

During the mid-training phase, the model starts to generalize better and distinguish between relevant and irrelevant visual features. The BLEU4 score improves consistently for dropout thresholds in the range of $\{0, 4\}$, with the highest performance observed at $\{2, 4\}$, where slightly higher dropout before decoding ($k2$) helps refine representations. Compared to random dropout methods, the structured AM dropout leads to better feature selection and smoother translations, as it forces the model to rely on critical rather than redundant frames. However, in the late training phase, excessive dropout negatively impacts SLT performance. When $k1$ is increased beyond 4 (e.g., $\{4, 4\}$ or $\{0, 10\}$), the BLEU4 score drops sharply due to the loss of vital contextual information. The degradation is particularly evident in the $\{0, 10\}$ configuration, where the translation quality is even worse than when no dropout is applied ($\{0, 0\}$), proving that overly aggressive masking can strip away essential motion features.

These findings highlight the importance of properly tuning the AM dropout thresholds. In the early phase, a low to moderate threshold ($\{0, 2\}$) helps establish robust feature learning. In the mid-phase, the optimal dropout configuration ($\{2, 4\}$) enhances translation quality by improving generalization and feature refinement. In the late phase, excessive dropout ($\{4, 4\}$ or higher) leads to performance degradation due to the removal of critical visual cues. Overall, the study confirms that AM dropout outperforms random dropout techniques by preserving key frame representations while reducing redundancy. The best configuration for AM on the ArabSign dataset is $\{2, 2\}$, as it provides the highest BLEU4 score while maintaining a balance between accuracy and generalization.

We improve the information interaction inside each clip by introducing a clip partition (CP) approach and inter-cross attention (ICA) for clip-level inputs in our proposed LCSA. Various combinations of attention modules in the ADTR encoder are tested in Table 4 to confirm the efficacy of the LCSA and clip-level inputs. Using masked self-attention (MSA) on the initial inputs is the baseline model for notation. MAE with ICA denotes MAE with ICA. It stands for “common clip partition” (CCP) and “multi-level clip analysis” (MSA) applied to inputs at the clip level. The terms CP + MSA and CP + LCSA refer to the use of MSA and LCSA, respectively, on clip-level inputs. The encoder uses continuous features, which are a combination of clip-level features, as its output. By connecting MSA after the original attention combinations, CP + MSA + MSA and CP + LCSA + MSA increase global information learning. Table 4 shows that when the original inputs are used, there is a noticeable performance reduction in the MSA that is equipped with ICA. This is because ICA influences the discriminative information across different movements when processing long SL films with several sign gestures, effectively crossing features together. In addition, the consequences of various attention combinations change after processing the original inputs into clip-level. When CCP and clip-level inputs are the only things performed, MSA’s improvement is not as strong as LCSA’s, and LCSA offers a bigger improvement than MSA with ICA. By enhancing the encoder’s capacity to learn continuous spatiotemporal characteristics and global information, clip-level attention combinations related to the MSA further enhance SLT performance. In comparison to other methods, our LCSA outperforms them all, and it even outperforms the combined local and global versions.

Table 4. Impact of the attention method in the encoder.

Encoder Attention	Dev	Test
MSA	22.45	22.89
MSA/ICA	21.68	22.28
CCP/MSA	22.83	23.35
CP/MSA	23.34	23.76
CP/LCSA	23.66	24.27
CP/MSA/MSA	23.78	24.46
CP/LCSA/MSA	24.52	24.92

During the early stages of training, when the model is still learning to extract and represent local spatiotemporal features, the introduction of the LCSA module—with its clip partition (CP) and inter-cross attention (ICA) components—helps organize the raw input into meaningful clip-level segments. This initial partitioning mitigates the confusion that arises from processing long sequences of sign gestures directly, thus allowing the model to focus on more coherent and context-rich sub-sequences. However, when using the original inputs without clip-level processing, incorporating ICA into masked self-attention (MSA) can actually degrade performance by overly mixing discriminative features from distinct movements, which indicates that proper segmentation is essential for the model’s early learning.

As training progresses into the mid-phase, the benefits of LCSA become more pronounced. At this point, the model starts to better differentiate between relevant and redundant information. The use of CP combined with either MSA or LCSA on clip-level inputs begins to improve performance, with LCSA showing a larger enhancement compared to MSA with ICA. This is because LCSA effectively leverages both local and global information by maintaining continuous spatiotemporal characteristics across clips. Additionally, stacking another MSA layer after the initial clip-level attention (forming configurations

like CP + LCSA + MSA) further boosts the learning of global interactions, allowing the model to integrate context across different clips, which is crucial for accurately translating complex sign language sequences.

In the later stages of training, the model refines its ability to capture nuanced movement patterns and temporal dependencies. The structured approach of LCSA, which carefully balances the partitioning of clips and the integration of inter-cross attention, ensures that the model does not lose critical discriminative details while still benefiting from enhanced global information learning. Compared to methods that combine local and global features without such structured attention mechanisms, LCSA consistently outperforms them, indicating that its design not only stabilizes the learning process in the late phase but also leads to superior sign language translation performance overall.

We compare several feature fusion methods in our experiments to ensure the AF module is functional. In Table 5, the output characteristic of the Transformer encoder is denoted as F_{enc} . In the AM module, F_{ct} stands for BiLSTM's temporal feature. F_{enc} and F_{ct} adaptable fusion feature is represented by F_{ff} . We can observe that the second fusion approach improves performance by 0.90 BLEU4 on the ArabSign test set, whereas the third method improves performance by 1.55 BLEU4. Furthermore, it shows that SLT performance may be significantly enhanced by integrating F_{ct} into F_{enc} . Using F_{ff} results in a poorer BLEU 4 score compared to combining F_{enc} and F_{ct} . The reason is that adaptive fusion is a method where the strengthening and enhancement of knowledge coexist. The ultimate SLT effect can be impacted by training-induced erosion of crucial F_{enc} and F_{ct} information. Hence, we combine the three feature representations of F_{enc} , F_{ct} , and F_{ff} in order to make the most of F_{ff} enhancement capabilities while minimizing its weakening effects on the fused features of F_{enc} and F_{ct} . This results in even better performance. Additionally, our fusion method outperforms the control set by 2.24 BLEU, proving its continued validity. In addition, we conduct tests with GRF and basic feature concatenation techniques. Our fusion process is superior, as can be seen from the experimental data shown in Table 5. It is worth mentioning that our AF module outperforms GRF by a small margin, despite being simplified and adapted from it.

Table 5. Impact of the Adaptive Fusion module on the performance.

Fusion	Dev	Test
GRF [16]	23.94	24.58
F_{enc}	21.78	22.47
F_{ff}	22.95	23.58
F_{enc} and F_{ct}	23.73	24.04
F_{enc} and F_{ct} and F_{ff}	24.64	24.98
C (F_{enc} and F_{ct})	23.32	22.97
C(F_{enc} and F_{ct} and F_{ff})	23.84	24.43

5. Conclusions

This paper introduces Adaptive Transformer (ADTR), an advanced deep learning framework tailored for continuous sign language recognition and translation. Our proposed approach effectively addresses key challenges in end-to-end SLT, eliminating the dependency on gloss annotations by integrating three novel modules: Adaptive Masking (AM), Local Clip Self-Attention (LCSA), and Adaptive Fusion (AF). These components collectively optimize temporal feature alignment, multi-scale representation learn-

ing, and context-aware feature fusion, significantly enhancing translation accuracy and model generalization.

The experimental evaluation on the ArabSign dataset demonstrates the effectiveness of ADTR, achieving a high BLEU4 score of 22.73 without gloss annotations and 24.93 when utilizing gloss annotations, surpassing state-of-the-art models by a significant margin. The lightweight architecture and computational efficiency make ADTR well-suited for real-time applications in AI-powered communication systems for the deaf and hard-of-hearing communities. By leveraging deep learning and computer vision techniques, our framework contributes significantly to AI-driven assistive technologies, bridging the gap between sign language and spoken language processing. Beyond the current scope, future work will explore the adaptability of ADTR to other sign language datasets and languages to assess its generalization across diverse linguistic and cultural contexts.

Author Contributions: Conceptualization, Y.S. and S.B.; methodology, Y.S. and M.A.; software, S.M.A. and A.A.A.; validation, S.B.; formal analysis, S.M.A. and M.A.; investigation, A.A.A. resources, S.B.; data curation, S.B. and A.A.A.; writing—original draft preparation, S.B. and Y.S.; writing—review and editing, Y.S. and M.A.; visualization, M.A., S.M.A. and Y.S.; supervision, Y.S. and M.A.; project administration, Y.S. and M.A.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the King Salman Center For Disability Research, through Research Group no KSRG-2024-185.

Data Availability Statement: Data will be made available on request.

Acknowledgments: The authors extend their appreciation to the King Salman Center For Disability Research for funding this work through Research Group no KSRG-2024-185.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Camgoz, N.C.; Hadfield, S.; Koller, O.; Ney, H.; Bowden, R. Neural sign language translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7784–7793.
2. Orbay, A.; Akarun, L. Neural sign language translation by learning tokenization. In Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), Buenos Aires, Argentina, 16–20 November 2020; pp. 222–228.
3. Li, D.; Xu, C.; Yu, X.; Zhang, K.; Swift, B.; Suominen, H.; Li, H. TSPNet: Hierarchical feature learning via temporal semantic pyramid for sign language translation. In Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 12034–12045.
4. Yin, A.; Zhong, T.; Tang, L.; Jin, W.; Jin, T.; Zhao, Z. Gloss attention for gloss-free sign language translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 2551–2562.
5. Zhou, B.; Chen, Z.; Clapés, A.; Wan, J.; Liang, Y.; Escalera, S.; Lei, Z.; Zhang, D. Gloss-free sign language translation: Improving from visual-language pretraining. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 20871–20881.
6. Wei, C.; Zhao, J.; Zhou, W.; Li, H. Semantic boundary detection with reinforcement learning for continuous sign language recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 1138–1149. [[CrossRef](#)]
7. Yin, W.; Hou, Y.; Guo, Z.; Liu, K. Spatial-temporal enhanced network for continuous sign language recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 1684–1695. [[CrossRef](#)]
8. Liu, Z.; Wu, J.; Shen, Z.; Chen, X.; Wu, Q.; Gui, Z.; Senhadji, L.; Shu, H. Improving End-to-end Sign Language Translation with Adaptive Video Representation Enhanced Transformer. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 8327–8342. [[CrossRef](#)]
9. Vaswani, A. Attention is all you need. In Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
10. Camgoz, N.C.; Koller, O.; Hadfield, S.; Bowden, R. Sign language transformers: Joint end-to-end sign language recognition and translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10023–10033.

11. Jing, L.; Tian, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4037–4058. [[CrossRef](#)] [[PubMed](#)]
12. Wei, C.; Fan, H.; Xie, S.; Wu, C.-Y.; Yuille, A.; Feichtenhofer, C. Masked feature prediction for self-supervised visual pre-training. *arXiv* **2021**, arXiv:2112.09133.
13. Pan, T.; Song, Y.; Yang, T.; Jiang, W.; Liu, W. VideoMoCo: Contrastive video representation learning with temporally adversarial examples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 11205–11214.
14. Yan, L.; Ma, S.; Wang, Q.; Chen, Y.; Zhang, X.; Savakis, A.; Liu, D. Video captioning using global–local representation. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 6642–6656. [[CrossRef](#)] [[PubMed](#)]
15. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [[CrossRef](#)] [[PubMed](#)]
16. Fan, C.; Yi, J.; Tao, J.; Tian, Z.; Liu, B.; Wen, Z. Gated recurrent fusion with joint training framework for robust end-to-end speech recognition. *IEEE ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 198–209. [[CrossRef](#)]
17. Jiang, W.; Zhou, W.; Hu, H. Double-stream position learning transformer network for image captioning. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7706–7718. [[CrossRef](#)]
18. Yin, K.; Read, J. Better sign language translation with STMC transformer. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 5975–5989.
19. Zhou, H.; Zhou, W.; Zhou, Y.; Li, H. Spatial–temporal multi-cue network for sign language recognition and translation. *IEEE Trans. Multimed.* **2022**, *24*, 768–779. [[CrossRef](#)]
20. Liu, T.; Zhang, C.; Lam, K.-M.; Kong, J. Decouple and resolve: Transformer-based models for online anomaly detection from weakly labeled videos. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 15–28. [[CrossRef](#)]
21. Zhang, C.; Su, J.; Ju, Y.; Lam, K.-M.; Wang, Q. Efficient inductive vision transformer for oriented object detection in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5616320. [[CrossRef](#)]
22. Xie, P.; Zhao, M.; Hu, X. PiSLTRc: Position-informed sign language transformer with content-aware convolution. *IEEE Trans. Multimed.* **2022**, *24*, 3908–3919. [[CrossRef](#)]
23. Yin, A.; Zhao, Z.; Liu, J.; Jin, W.; Zhang, M.; Zeng, X.; He, X. SimulSLT: End-to-end simultaneous sign language translation. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 4118–4127.
24. Chen, Y.; Wei, F.; Sun, X.; Wu, Z.; Lin, S. A simple multi-modality transfer learning baseline for sign language translation. In Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 5110–5120.
25. Kan, J.; Hu, K.; Hagenbuchner, M.; Tsoi, A.C.; Bennamoun, M.; Wang, Z. Sign language translation with hierarchical spatio-temporal graph neural network. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2022; pp. 2131–2140.
26. Fu, B.; Ye, P.; Zhang, L.; Yu, P.; Hu, C.; Shi, X.; Chen, Y. A token-level contrastive framework for sign language translation. *arXiv* **2022**, arXiv:2204.04916.
27. Zhou, H.; Zhou, W.; Qi, W.; Pu, J.; Li, H. Improving sign language translation with monolingual data by sign back-translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 1316–1325.
28. Huang, J.; Huang, Y.; Wang, Q.; Yang, W.; Meng, H. Self-supervised representation learning for videos by segmenting via sampling rate order prediction. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 3475–3489. [[CrossRef](#)]
29. Jenni, S.; Meishvili, G.; Favaro, P. Video representation learning by recognizing temporal transformations. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 425–442.
30. Tao, L.; Wang, X.; Yamasaki, T. An improved inter-intra contrastive learning framework on self-supervised video representation. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5266–5280. [[CrossRef](#)]
31. Gan, C.; Gong, B.; Liu, K.; Su, H.; Guibas, L.J. Geometry guided convolutional neural networks for self-supervised video representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5589–5597.
32. Korbar, B.; Tran, D.; Torresani, L. Cooperative learning of audio and video models from self-supervised synchronization. In Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS), Montréal, QC, Canada, 2–8 December 2018; pp. 7763–7774.
33. Sun, C.; Myers, A.; Vondrick, C.; Murphy, K.; Schmid, C. VideoBERT: A joint model for video and language representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7464–7473.
34. Camgoz, N.C.; Koller, O.; Hadfield, S.; Bowden, R. Multi-channel transformers for multi-articulatory sign language translation. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 301–319.

35. Ye, J.; Jiao, W.; Wang, X.; Tu, Z.; Xiong, H. Cross-modality data augmentation for end-to-end sign language translation. *arXiv* **2023**, arXiv:2305.11096.
36. Luqman, H. ArabSign: A multi-modality dataset and benchmark for continuous Arabic Sign Language recognition. In Proceedings of the 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG), Waikoloa Beach, HI, USA, 5–8 January 2023; pp. 1–8.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.