Event Stream-based Sign Language Translation: A High-Definition Benchmark Dataset and A Novel Baseline

Shiao Wang[1], Xiao Wang[1⋆], Duoqing Yang[1], Yao Rong[1], Fuling Wang[1], Jianing Li[2], Lin Zhu[3], and Bo Jiang[1⋆]

[1] School of Computer Science and Technology, Anhui University, Hefei, China
{e24101001,e125221163,e23201049}@stu.ahu.edu.cn
{xiaowang,jiangbo}@ahu.edu.cn, rytaptap@163.com
[2] Qiyuan Laboratory, Beijing, China
lijianing@pku.edu.cn
[3] Beijing Institute of Technology, Beijing, China
linzhu@pku.edu.cn

Abstract. Sign Language Translation (SLT) is a core task in the field of AI-assisted disability. Traditional SLT methods are typically based on visible light videos, which are easily affected by factors such as lighting variations, rapid hand movements, and privacy concerns. This paper proposes the use of bio-inspired event cameras to alleviate the aforementioned issues. Specifically, we introduce a new high-definition event-based sign language dataset, termed Event-CSL, which effectively addresses the data scarcity in this research area. The dataset comprises 14,827 videos, 6,097 glosses, and 2,544 Chinese words in the text vocabulary. These samples are collected across diverse indoor and outdoor scenes, covering multiple viewpoints, lighting conditions, and camera motions. We have also benchmarked existing mainstream SLT methods on this dataset to facilitate fair comparisons in future research. Furthermore, we propose a novel event-based, gloss-free sign language translation framework, termed EvSLT. The framework first segments continuous video features into clips and employs a Mamba-based memory aggregation module to compress and aggregate spatial detail features at the clip level. Subsequently, these spatial features, along with temporal representations obtained from temporal convolution, are then fused by a graph-guided spatiotemporal fusion module. Extensive experiments on Event-CSL, as well as other publicly available datasets, demonstrate the superior performance of our method. The dataset and source code are publicly available at https://github.com/Event-AHU/OpenESL/tree/main/EvSLT.

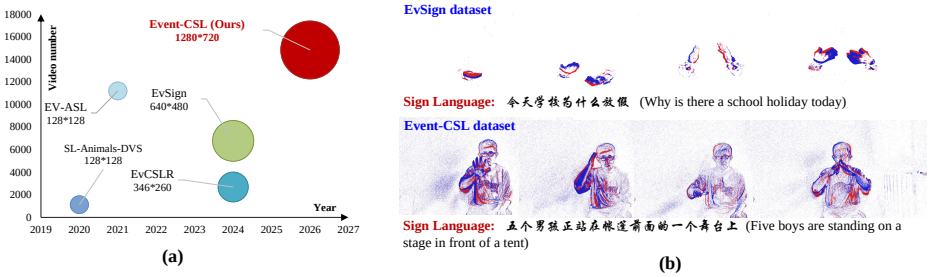
Keywords: Sign Language Translation · Event Camera · Mamba Networks · Spatio-temporal Learning

1 Introduction

With the rapid development of deep learning [22], *AI (Artificial Intelligence) for good* has received widespread attention [34, 37, 45, 46], among which, Sign Language

⋆ Corresponding authors: Xiao Wang and Bo Jiang.

Translation (SLT) [12, 40, 51, 53, 55] is increasingly being emphasized. It facilitates communication between deaf individuals and non-signers by translating sign language videos into natural language text, thereby significantly enhancing social participation and quality of life for the deaf community. However, the performance of the SLT task is still limited due to the usage of traditional RGB cameras, as it is easily influenced by limited frame rate, illumination, motion blur, etc. In addition, the deployment of RGB-based SLT models raises potential ethical concerns, as it inherently involves issues related to human privacy.



(a) (b)

Fig. 1: (a) Comparison of scales between existing event-based sign language datasets and our newly proposed Event-CSL benchmark dataset (the bubble size represents the resolution). (b) Comparison of sample examples between the existing event-based SLT dataset (EvSign) and our newly proposed Event-CSL dataset.

Recently, biologically inspired event cameras (e.g., DVS346, Prophesee, CeleX) have drawn more and more attention [6, 42, 43] due to their unique advantages in *high dynamic range*, *low energy cost*, *sparse spatial but dense temporal resolution*, and *low latency*, etc. [8]. In terms of the imaging mechanism, each pixel in an event camera operates independently and asynchronously, which fundamentally differs from RGB cameras that capture synchronized frame-based outputs. A binary pulse signal is recorded only when the change in brightness for each pixel exceeds a specific threshold. Some computer vision tasks have involved the event camera for event-based or event-enhanced learning, such as object detection [9], tracking [28, 41, 43], action recognition [44, 47], and also the sign language translation [54] task discussed in this paper.

Although research in this area remains limited, several studies have recently begun to explore event-based sign language translation (SLT) and sign language recognition (SLR), as illustrated in Fig. 1 (a). Early approaches are typically developed using simulated event data generated from existing RGB datasets, such as PHOENIX-2014 [21] and CSL-Daily [56]. However, such simulated data fail to fully capture the essential characteristics of real event streams. Recently, four real event-based datasets have been introduced, namely SL-Animal-DVS [38], EV-ASL [48], EvCSLR [17], and EvSign [54]. The first three datasets are designed for the SLR task and suffer from low spatial resolutions (128×128 and 346×260), while EvSign [54] removes the actors' head regions for stronger privacy protection, as illustrated in Fig. 1 (b). We argue that this operation may distort the event signals, thereby negatively impacting the

recognition and analysis of gestures involving head movements. In addition, to reduce the substantial computational complexity introduced by spatial modeling, most existing SLT methods [54, 55] primarily focus on temporal modeling, while overlooking the crucial role of spatial detail features in gesture understanding.

In this paper, we propose a new sign language translation dataset, termed Event-CSL, which is the largest, high-definition (1280×720) real-event dataset collected using the Prophesee EVK4-HD event camera. It contains 14,827 videos, 6,097 glosses, and 2,544 Chinese words in the text vocabulary. Our dataset is collected across diverse indoor and outdoor scenes, covering multiple viewpoints, lighting conditions, and camera motions. More importantly, we provide the most primitive event stream data, which lays the groundwork for accurately recognizing the meanings conveyed by gestures. The inherent sensing characteristics of event cameras offer a certain degree of privacy protection. In our experiments, we split these videos into training, validation, and testing subsets, which contain 12,602, 741, and 1,484 samples, respectively. To build a more comprehensive benchmark, we retrain and evaluate recently released SLT models for future comparison.

Based on the newly proposed Event-CSL dataset, we further propose a novel event-based, gloss-free sign language translation framework, termed EvSLT. As shown in Fig. 2, we first feed the event video frames into a visual encoder, which is composed of residual networks [13], to extract visual features. The resulting video features are then segmented into consecutive video clip features of a predefined length, following the temporal order of the frames. Features from these fixed-length video clips are subsequently compressed and aggregated using a Mamba-based memory aggregation module to capture fine-grained spatial details. Together with temporal features extracted via an average pooling layer and temporal convolution, the compressed clip features and aggregated spatial features are fed into a graph-guided spatiotemporal fusion module, where hypergraphs constructed from the compressed clip features serve as a bridge to enable effective interactions between spatial and temporal representations. Finally, the spatiotemporally fused features are projected into language model embedding space through a sign embedding layer, and then fed into a language model (mBART [29]) for decoding into sign language text.

To sum up, the main contributions of this paper can be summarized as the following three aspects:

1). We propose a new large-scale, high-definition event-based sign language translation dataset, termed Event-CSL. It contains 14,827 videos, 6,097 glosses, and 2,544 Chinese words recorded in both indoor and outdoor scenarios.

2). We propose a novel event-based, gloss-free sign language translation framework, termed EvSLT. It employs a Mamba-based memory aggregation module to compress and aggregate spatial features, and introduces a graph-guided spatiotemporal fusion module for spatiotemporal interaction.

3). Extensive experiments conducted on two event-based SLT datasets, EvSign and our proposed Event-CSL, demonstrate the effectiveness of our framework.

2 Related Work

2.1 Sign Language Translation

Sign language translation aims to transform sign language videos into spoken text, thereby facilitating communication for deaf individuals. Traditional gloss-based methods [3, 5, 51] first translate sign language videos into glosses and then convert those glosses into spoken text. Joint-SLT [3] proposes a Transformer-based framework that jointly learns continuous sign language recognition and translation in an end-to-end manner. Recently, gloss-free SLT methods [27, 49, 53, 55] have gained increasing attention for removing the need for costly gloss annotations. For example, Zhou et al. [55] propose a two-stage approach, which bridges the semantic gap between visual and text representation and restores masking sentences. Sign2GPT [49] leverages LLM for gloss-free SLT. EvSign [54] introduces the first event-based SLT dataset, with participants' heads removed to enhance privacy protection. Different from previous works, we introduce a larger-scale high-definition event-based SLT dataset, and explore the SLT task by leveraging the unique advantages of event cameras.

2.2 State Space Model

State space models (SSMs) [19] have recently gained attention due to their linear complexity. Gu et al. [11] propose the Structured State Space sequence model, which reparameterizes the SSM, stabilizes the state matrix via low-rank correction, and simplifies computation through a Cauchy kernel formulation. Mamba [10] is a sequence modeling approach with linear complexity that addresses the low computational efficiency of traditional methods on long sequences. In computer vision, VMamba [30] extends the visual Mamba with four-directional scanning while retaining the benefits of a global receptive field and dynamic weights. Vim [57] further introduces a bidirectional vision Mamba block that serves as a universal backbone for vision tasks. Following the design of vision Mamba, many recent visual models [15, 24, 35, 50] adopt similar strategies to achieve efficient spatial modeling. Building upon these advances, this paper employs the vision Mamba block to aggregate spatial details from video clips, thereby effectively enhancing sign language translation performance.

3 Methodology

3.1 Overview

As illustrated in Fig. 2, we propose a unified framework for event-based sign language translation. Input event streams are first stacked into event frames and passed through a visual encoder to extract features. Temporal dynamics are modeled using average pooling followed by temporal convolution, while a Mamba-based memory aggregation module compresses and aggregates clip-level spatial representations. A graph-guided spatiotemporal fusion module then constructs relational graphs from the compressed clip features, enabling effective interaction between enhanced spatial details and temporal dependencies. Finally, the fused spatiotemporal features are projected via a sign embedding module and decoded by a language model (mBART [29]) to generate accurate Chinese sentences.

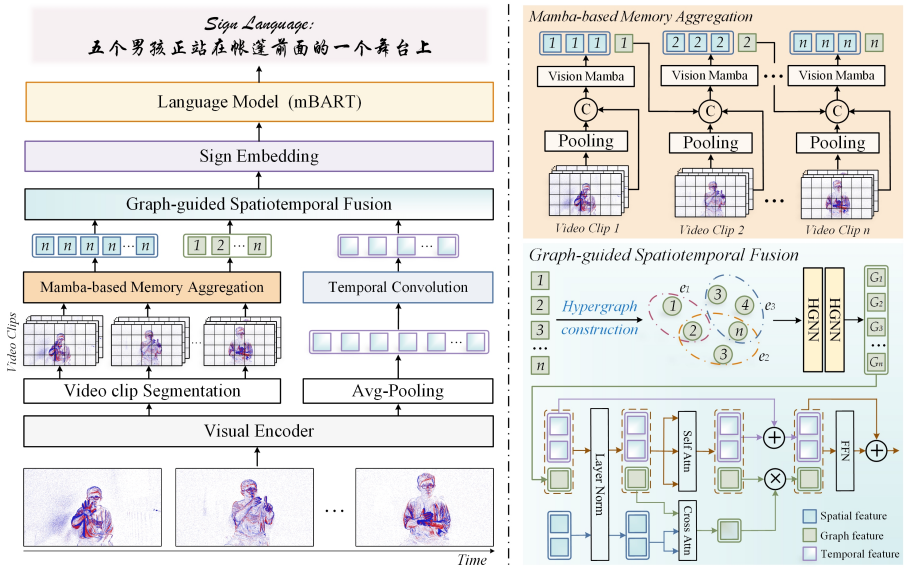


Fig. 2: An overview of the proposed event-based sign language translation framework. The framework segments continuous video features into clip features and employs a Mamba-based memory aggregation module to compress and aggregate spatial details. The resulting spatial and compressed clip features, together with temporal representations from temporal convolution, are fused in a graph-guided spatiotemporal fusion module. Hypergraphs built from compressed clip features facilitate interaction between memory-enhanced spatial and temporal representations. Finally, a Transformer-based language model generates the corresponding Chinese sentences.

3.2 Input Representation

Given event streams $\mathcal{E}^p = \{e_1, e_2, ..., e_M\}$, where M denotes the number of event points, each point $e_i|i \in (1, M)$ is a quadruple $\{x, y, t, p\}$, here the (x, y) denotes the spatial coordinates, t and p denotes the time stamp and polarity, respectively. To make full use of existing deep neural networks, we first stack the event streams of each video into a sequence of event frames using non-overlapping fixed temporal windows, yielding $\mathcal{E}^T = \{E_1, E_2, \ldots, E_T\}$, where T denotes the number of stacked frames. Each event frame is then resized to a fixed resolution, i.e., $E_i \in \mathbb{R}^{C\times H\times W}$. Here, $\{C, H, W\}$ denotes the dimensions of the channel, height, and width of an event frame, respectively. During training, we randomly sample a mini-batch of B event video samples as input, with dimensions $B \times T \times C \times H \times W$.

3.3 Network Architecture

• **Visual Encoder and Temporal Branch.** Given an input of size $B \times T \times C \times H \times W$, we first employ a residual network as the visual encoder to extract visual feature representations for video frames. Specifically, we reshape the input by merging the batch and temporal dimensions into $B \times T \rightarrow B * T$, resulting in a tensor of size

$B * T \times C \times H \times W$, and then fed into ResNet-18 [13] to extract visual features, yielding video feature maps $\mathbf{F} \in \mathbb{R}^{B\times T\times C'\times H'\times W'}$ ($H' = H/32$, $W' = W/32$). Here, C', H', and W' denote the channel dimension, height, and width of the feature maps obtained after the hierarchical downsampling of ResNet-18.

In the temporal branch, we perform global average pooling over the spatial dimensions of the extracted features to obtain the temporal features $\mathbf{F}_t \in \mathbb{R}^{B\times T\times C'}$. After that, we employ two temporal convolution blocks to aggregate temporal information and obtain the aggregated temporal features $\mathbf{F}'_t \in \mathbb{R}^{B\times T'\times C'}$. It can be formulated as:

$$\begin{aligned} \mathbf{H}^{(1)} &= \text{MaxPool}\Big(\sigma\Big(\text{BN}\Big(\text{Conv1d}_5^{(1)}(\mathbf{F}_t)\Big)\Big)\Big), \\ \mathbf{F}'_t &= \text{MaxPool}\Big(\sigma\Big(\text{BN}\Big(\text{Conv1d}_5^{(2)}\Big(\mathbf{H}^{(1)}\Big)\Big)\Big)\Big), \end{aligned} \tag{1}$$

where $\text{Conv1d}_5^{(i)}(\cdot)$ denotes the 1D convolution with kernel size 5 in the i-th temporal convolution block, MaxPool represents the max pooling operation, and $\sigma(\cdot)$ denotes the ReLU activation function. Temporal convolution enables the model to capture continuous hand movements and establish coherent temporal dependencies, thereby improving contextual understanding and translation accuracy.

• **Mamba-based Memory Aggregation.** To preserve spatial detail while reducing computational cost, we introduce a Mamba-based memory aggregation module to compress and aggregate spatial features. Specifically, we first divide the continuous video features $\mathbf{F} \in \mathbb{R}^{B\times T\times C'\times H'\times W'}$ into fixed-length video clip features along the temporal dimension, resulting in n clip features, denoted as $\mathbf{F}_c = \{\mathbf{F}_c^1, \mathbf{F}_c^2, \ldots, \mathbf{F}_c^n\}$. Consequently, the spatial feature of each video clip can be represented as $\mathbf{F}_c^i \in \mathbb{R}^{B\times \frac{T}{n}\times C'\times H'\times W'}$, which can be further reshaped into $\mathbf{F}_c^i \in \mathbb{R}^{B\times \frac{T\cdot H'\cdot W'}{n}\times C'}$. Next, we apply global average pooling to the spatial feature of the i-th video clip to compress them and obtain a compressed clip feature representation $\mathbf{F}_g^i \in \mathbb{R}^{B\times 1\times C'}$. The resulting compressed clip feature is then concatenated with the spatial features of the corresponding video clip and fed into the visual Mamba network for spatial feature aggregation.

Following vision Mamba [57], the concatenated vision features are first normalized through a normalization layer, and then projected into features *x* and *z* using two linear layers. Specifically, we also employ both forward and backward scanning to process the input. For each direction, we obtain x' using *the 1D convolution* layer and *SiLU* activation function:

$$x' = SiLU(Conv(x)). \tag{2}$$

After that, the output x' is projected to yield the input matrix $\mathbf{B} \in \mathbb{R}^{N\times T}$, output matrix $\mathbf{C} \in \mathbb{R}^{T\times N}$, and also the timescale parameter Δ:

$$\mathbf{B}, \mathbf{C}, \Delta = Linear(x'), \tag{3}$$

Here, the timescale parameter Δ is used for discretization, as the raw SSM is designed for a continuous system, but the data we process are discretized visual tokens. The Zero-Order Hold (ZOH) rule is adopted to discretize the continuous state space matrices:

$$\overline{\mathbf{A}}, \overline{\mathbf{B}} = \text{ZOH}(\mathbf{A}, \mathbf{B}, \Delta), \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the state matrix. Therefore, the formula of the discretized SSM can be denoted as:

$$h' = \overline{\mathbf{A}}h + \overline{\mathbf{B}}x', \quad y = \mathbf{C}h', \tag{5}$$

where h and h' represent the hidden state, x' and y are the inputs and output. The forward and backward SSM branches are applied for bidirectional context awareness to produce y_{for}, y_{back}. After that, y_{for}, y_{back} are gated by z and the result of their addition is linearly projected to y', which can be written as:

$$y' = Linear(y_{for} \odot SiLU(z) + y_{back} \odot SiLU(z)). \tag{6}$$

For each video clip feature $\mathbf{F}_c^i$, we apply the aforementioned series of operations, and the resulting compressed clip features $\mathbf{F}_g^{(i)'}$ are pass to the $(i+1)$-th video clip feature to enable inter-clip information exchange and memory aggregation. Mathematically, this can be expressed as:

$$\mathbf{F}_g^{(i+1)'}, \mathbf{F}_c^{(i+1)'} = Mamba([\mathbf{F}_g^{(i)'}, \mathbf{F}_g^{i+1}, \mathbf{F}_c^{i+1}]), \tag{7}$$

where $[\cdot]$ denotes the concatenation operation, and the compressed feature of the previous clip is discarded.

In this way, we progressively aggregate the spatial features of each clip in a forward manner, analogous to how the human brain sequentially remember each video clip from beginning to end, resulting in the final aggregated spatial features $\mathbf{F_s} = \mathbf{F}_c^{(n)'}$ along with the compressed clip features $\mathbf{F}_g = [\mathbf{F}_g^{(1)'}, \mathbf{F}_g^{(2)'}, ..., \mathbf{F}_g^{(n)'}]$. Next, the memory-enhanced spatial features $\mathbf{F_s}$, the compressed clip features $\mathbf{F}_g$, and the temporal features $\mathbf{F}'_t$ are jointly fed into the graph-guided spatiotemporal fusion module.

• **Graph-guided Spatiotemporal Fusion.** To capture the complex relationships among different video clips, we treat the compressed feature vectors of video clips as nodes to construct hypergraphs. Following [7], we adopt K-nearest neighbors (K-NN, where the maximum number of neighboring nodes is set to 10 in our work) based on Euclidean distance to generate the hypergraph structure $\mathcal{H}$. It can be formulated as:

$$\mathcal{H} = \{V, \mathcal{E}, \mathbf{H}\}, \quad where\ \mathbf{H}_{ji} = \begin{cases} 1, & \text{if } v_j \in e_i, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

$$\mathbf{G} = \mathbf{D}_v^{-\frac{1}{2}}\,\mathbf{H}\,\mathbf{D}_e^{-1}\,\mathbf{H}^\top\,\mathbf{D}_v^{-\frac{1}{2}}, \tag{9}$$

where $\mathbf{H}$ is the node–hyperedge incidence matrix, V and $\mathcal{E}$ denote the vertex and hyperedge set. $\mathbf{D}_v$ and $\mathbf{D}_e$ are the node degree matrix and hyperedge degree matrix, respectively. $\mathbf{G}$ denotes the normalized hypergraph propagation matrix. Subsequently, we feed $\mathbf{G}$ into Hypergraph Neural Network (HGNN) [7] to facilitate effective interaction among video clips:

$$\mathbf{X}' = \mathbf{G}(\mathbf{X}\mathbf{W} + \mathbf{b}), \tag{10}$$

where the $\mathbf{X}$ and $\mathbf{X}'$ denote the input and output feature vectors, i.e., $\mathbf{F}_g$ and $\mathbf{F}'_g$, respectively. $\mathbf{W}$ is the learnable weight matrix and $\mathbf{b}$ is the bias.

To enable effective interaction between spatial and temporal features, i.e., $\mathbf{F}_s$ and $\mathbf{F}'_t$, we propose to use graph features $\mathbf{F}'_g$ derived from the compressed clip representations as a bridge to enable efficient spatiotemporal fusion. Specifically, we first pass the spatial, temporal, and graph features through a shared layer normalization layer. This is followed by two parallel attention networks that process the features concurrently. Let SA and CA denote the self-attention and cross-attention mechanisms, respectively. The temporal and graph features are concatenated and input to the SA module, whereas the graph and spatial features serve as the query and context, respectively, for the CA module. It can be expressed by the following formulas:

$$\left[\mathbf{F}_t^{\text{SA}}, \mathbf{F}_g^{\text{SA}}\right] = \text{SA}\big([\mathbf{F}'_t, \mathbf{F}'_g]\big), \quad \mathbf{F}_g^{\text{CA}} = \text{CA}\big(\mathbf{F}'_g, \mathbf{F}_s\big), \tag{11}$$

where the $\mathbf{F}_g^{\text{SA}}$ and $\mathbf{F}_g^{\text{CA}}$ denote the output graph features of SA and CA, respectively. Subsequently, the two enhanced graph features are combined via element-wise multiplication, together with the SA-enhanced temporal features, fed into a feedforward neural network. The feedforward network aggregates and transforms the attention-enhanced features, producing the final spatiotemporally enhanced fused representation. Since the detailed spatial features consist of a large number of tokens, whereas the temporal and graph features contain relatively few, this design provides two key advantages. First, the graph features act as a bridge, facilitating effective interactions between the spatial and temporal features. Second, it substantially reduces the computational burden of the attention mechanism, since the spatial detail features are only used as the keys and values in the cross-attention operation.

• **Sign Language Decoder.** Once the spatiotemporally enhanced features are obtained, they are fed into a sign embedding module, which consists of a linear layer and a ReLU activation function. Finally, a language decoder is adapted to generate sign language sentences. In our practical implementation, we use the pre-trained parameters of *mBART* [29] to initialize the language model, which consists of 3 Transformer encoder layers and 3 Transformer decoder layers. Note that we directly output the language based on given input event streams, without the help of gloss annotations, i.e., our model is a gloss-free SLT framework.

3.4 Loss Function

The language decoder sequentially generates the logits of sign language sentence S, which can be expressed as:

$$S = \{p(<bos>), p(w_1), p(w_2), ..., p(w_U), p(<eos>)\}. \tag{12}$$

The first special word <bos> marks the beginning of the sentence, the last special word <eos> marks the end of the sentence, and $p(w)$ represents the logits of the generated words. In this work, we adopt the widely used *cross-entropy loss* function to calculate the distance between the annotated sign language sentence and the generated sentence, which can be formulated as:

$$\mathcal{L} = -\log \prod_{u=1}^{U} p(s_u|w_u). \tag{13}$$

Table 1: Comparison of the datasets for sign language recognition and translation.

Dataset	Year	Language	#Videos	Resolution	Gloss	Text	Source	Raw Data	Scene	Continuous	SLT	Event
SIGNUM [39]	2010	DGS	15,075	776 × 578	455		Lab	-	-	✓	✗	✗
DEVISIGN-G [4]	2015	CSL	432	640 × 480	36	-	Lab	-	-	✗	✗	✗
DEVISIGN-D [4]	2015	CSL	6,000	640 × 480	500	-	Lab	-	-	✗	✗	✗
DEVISIGN-L [4]	2015	CSL	24,000	640 × 480	2,000	-	Lab	-	-	✗	✗	✗
PHOENIX-2014 [21]	2015	DGS	6,841	210 × 260	1,081	-	TV	-	-	✓	✗	✗
PHOENIX-2014T [2]	2018	DGS	8,257	210 × 260	1,066	2,887	TV	-	-	✓	✓	✗
MSASL [18]	2018	ASL	25,513	-	1,000	-	Web	-	-	✗	✗	✗
INCLUDE [36]	2020	ISL	4,287	1920 × 1080	263	-	Lab	-	-	✗	✗	✗
CSL-Daily [56]	2021	CSL	20,654	1920 × 1080	2,000	2,343	Lab	-	-	✓	✓	✗
SL-Animals-DVS [38]	2020	SSL	1,121	128 × 128	19	-	Lab	✓	Indoor	✗	✗	✓
EV-ASL [48]	2021	ASL	11,200	128 × 128	56	-	Lab	-	-	✗	✗	✓
EvCSLR [17]	2024	CSL	2,685	346 × 260	423	-	Lab	✓	Indoor	✓	✗	✓
EvSign [54]	2024	CSL	6,773	640 × 480	1,387	1,947	Lab	✓	Indoor	✓	✓	✓
Event-CSL (Ours)	2026	CSL	**14,827**	**1280 × 720**	**6,097**	**2,544**	Lab	✓	Indoor, Outdoor	✓	✓	✓

4 Event-CSL Dataset

4.1 Protocols

When collecting the Event-CSL dataset, we adhere to the following collection protocols: *1). Diversity of views*: We capture different perspectives of sign language, including slightly downward, horizontal, and slightly upward views. *2). Camera motions*: Our dataset collection takes into account a variety of motion patterns, including fast movement, slow movement, and moderate movement. *3). Diversity of scenes*: In order to conform to the daily use of sign language, we consider different usage scenarios when recording videos. Specifically, these scenarios comprise office environments, daily life, and outdoor scenes (such as streets, parking lots, and open spaces). *4). Intensity of light*: In environments with strong or weak light intensity (high and low exposure), event cameras can still record information about the movement of objects due to the presence of pixel changes. We record videos under different lighting conditions, including strong, low, and dim light. *5). Continuity of action*: Keep the movement of sign language flowing and coherent when recording sign language, reducing useless information and redundant video frames. *6). Diversity of sign language content*: To better reflect real-world applications, sign language videos should cover diverse content. We randomly extract captions from the COCO-CN [25] dataset, encompassing daily life, outdoor activities, animals and plants, sports, character introductions, and landscapes.

4.2 Statistical Analysis

Our proposed dataset is the first large-scale, high-definition event-based sign language benchmark. All videos are recorded using the Prophesee EVK4-HD event camera at a resolution of 1280×720. In total, the dataset contains 14,827 sign language videos, 6,097 gloss annotations, and a vocabulary of 2,544 Chinese words. We randomly divide the dataset into training, validation, and test sets with 12,602, 741, and 1,484 videos, respectively. The text corpus includes sentences ranging from 8 to 75 characters, with an average length of 18. As shown in Table 1, our dataset significantly surpasses existing event-based sign language datasets in terms of video quantity, spatial resolution, gloss

annotations, and vocabulary diversity. For detailed annotation and attribute statistics, please refer to the *Supplementary Material*.

4.3 Benchmark Baselines

To construct a comprehensive benchmark dataset for the event-based SLT task, we retrain and report the performance of the following SLT algorithms as the baselines on the Event-CSL dataset for future works to compare: **1). Gloss-based SLT:** Joint-SLT [3], Chen et al. [5], Sign-XmDA [51]; **2). Gloss-free SLT:** TSPNet [23], GASLT [53], GFSLT [55], SignCL [52], Sign2GPT [49], MMSLT [20], and SpaMo [16].

5 Experiment

5.1 Dataset and Evaluation Metric

Our experiments are conducted on two event-based SLT datasets, including the newly proposed **Event-CSL**, and the public **EvSign** [54] datasets. We adopt BLEU [32] and ROUGE-L [26] metrics to assess the performance of sign language translation. The introduction to these datasets can be found in the *Supplementary Material*.

5.2 Implementation Details

To balance performance and efficiency, all images are resized to 224×224 during training and testing before being input to the network. Each segmented video clip contains 8 event frames. In the Mamba-based memory aggregation module, we stack two randomly initialized vision Mamba [57] layers to achieve efficient spatial feature aggregation, with the hidden dimension set to 512. Meanwhile, in the graph-guided spatiotemporal fusion module, two HGNN [7] layers are employed as the hypergraph learning network. During training, the batch size is set to 6. We use the SGD [1] optimizer with an initial learning rate of 0.01, together with a cosine annealing learning rate scheduler over 200 epochs. Our code is implemented using PyTorch [33] based on Python. All the experiments are conducted on a server with A800 GPUs. More implementation details can be found in our source code.

5.3 Comparison on Public SLT Datasets

• **Results on Event-CSL Dataset.** As shown in Table 2, we compare our method with other state-of-the-art SLT approaches on the Event-CSL dataset. Our model achieves ROUGE-L and BLEU-4 scores of 69.80 and 50.76, respectively. Compared with gloss-based methods, our proposed gloss-free approach outperforms Joint-SLT [3] and Sign-XmDA [51], while still exhibiting a performance gap compared to Chen et al. [5]. Furthermore, compared with recent gloss-free approaches such as Sign2GPT [49] and MMSLT [20], which leverage large language models for effective sign language translation, our model achieves superior accuracy. These results clearly demonstrate the effectiveness of our approach on the Event-CSL dataset.

Table 2: Experimental results on our Event-CSL dataset.

No.	Algorithm	Publish	Backbone	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
01	Joint-SLT [3]	CVPR2020	ViT	60.89	61.24	55.38	47.41	42.30
02	Chen et al. [5]	CVPR2022	S3D	75.60	77.26	68.47	62.31	55.68
03	Sign-XmDA [51]	EMNLP2023	ViT	65.81	66.96	60.18	52.54	48.15
04	GASLT [53]	CVPR2023	ViT	31.69	34.27	23.41	16.73	12.64
05	TSPNet [23]	NeurIPS2020	I3D	26.63	30.91	17.84	10.89	7.25
06	GFSLT [55]	ICCV2023	CNN	67.23	69.00	60.62	53.77	48.20
07	SignCL [52]	NeurIPS2024	CNN	67.76	69.12	60.88	54.18	48.52
08	Sign2GPT [49]	ICLR2024	ViT	68.74	70.35	61.98	55.13	49.23
09	MMSLT [20]	ICCV2025	CNN, BERT	68.95	70.70	62.18	55.42	49.95
10	SpaMo [16]	NAACL2025	ViT	67.78	68.59	61.27	55.12	49.08
11	Ours	-	CNN, Mamba	69.80	71.36	63.01	56.28	50.76

• **Results on EvSign Dataset.** As shown in Table 3, we further evaluate our method on the EvSign [54] dataset. We compare our results with those reported in the EvSign benchmark and additionally include the classical GFSLT method [55] as well as the recent MMSLT approach [20] to ensure a comprehensive evaluation. Our method achieves the best overall performance among all competing approaches. Notably, by effectively integrating spatial detail representations, it surpasses GFSLT with a +2.32% improvement in BLEU-4, further demonstrating its robustness and generalization capability.

Table 3: Comparison between our model and other SOTA algorithms on the EvSign dataset.

No.	Algorithm	Publish	Backbone	Modality	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
01	Joint-SLT. [3]	CVPR2020	ViT	RGB	40.05	39.84	23.54	15.60	10.63
02	VAC+TH [31]	ICCV2021	CNN,BiLSTM	RGB	39.08	38.74	23.90	15.88	10.19
03	CorrNet+TH [14]	CVPR2023	CNN,BiLSTM	RGB	39.41	39.45	23.74	15.68	10.57
04	Joint-SLT [3]	CVPR2020	ViT	Event	41.54	40.13	24.36	16.04	10.87
05	VAC+TH [31]	ICCV2021	CNN,BiLSTM	Event	39.48	39.22	24.11	15.94	10.01
06	CorrNet+TH [14]	CVPR2023	CNN,BiLSTM	Event	41.23	40.85	25.34	16.95	11.83
07	GFSLT [55]	ICCV2023	CNN	Event	51.98	54.26	35.91	24.97	17.88
08	Zhang et al. [54]	ECCV2024	CNN,BiLSTM,ViT	Event	42.43	41.44	25.61	17.55	12.37
09	MMSLT [20]	ICCV2025	CNN, BERT	Event	52.14	54.29	37.68	26.09	19.12
10	Ours	-	CNN, Mamba	Event	53.43	55.45	38.45	27.41	20.20

5.4 Ablation Study

• **Component Analysis on Event-CSL.** To demonstrate the effectiveness of our method, we evaluate the contribution of each module on the Event-CSL dataset, as shown in Table 4. We first compare the temporal and spatial branches ((a) and (b)), and find that temporal features outperform spatial ones by a large margin, as the SLT task relies heavily on temporal continuity. Combining spatial details with temporal representations allows the model to better capture fine-grained motion and contextual cues. We then examine

the impact of the Mamba layer in the spatial branch. Replacing it with simple MLPs results in a 0.68% drop in BLEU-4 (comparing (c) with (e)), confirming the benefit of the Mamba-based design. Finally, the fourth experiment ((d)) validates the contribution of the proposed Graph-guided Spatiotemporal Fusion (GSTF) module. Overall, these results demonstrate the effectiveness of all components in our framework.

Table 4: Component analysis results on the Event-CSL dataset.

No.	Temporal	Spatial	Mamba	GSTF	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Params
(a)	✓	✗	✗	✗	67.23	69.00	60.62	53.77	48.20	115.5M
(b)	✗	✓	✓	✗	56.31	57.15	47.77	40.63	35.12	119.8M
(c)	✓	✓	✗	✓	69.14	70.83	62.78	55.81	50.08	134.4M
(d)	✓	✓	✓	✗	69.02	70.47	62.65	55.69	49.94	130.7M
(e)	✓	✓	✓	✓	**69.80**	**71.36**	**63.01**	**56.28**	**50.76**	138.2M

• **Analysis of the Number of Input Event Frames.** The number of input frames in sign language videos significantly affects both the model's performance and its computational resource usage. To investigate this, we set four different frame sampling ratios: L/8, L/4, L/2, and L, where L denotes the total number of frames in each sign language video. As shown in Table 5, we compare the model's performance under these four frame settings on the Event-CSL dataset. The results demonstrate that as the number of frames increases, the model performance consistently improves. When all the video frames (L) are used, the performance is several times higher than that obtained using only L/8 frames. This indicates that increasing the number of input event frames enriches the temporal context, thereby improving the translation quality.

Table 5: Analysis of the **Number of Input Event Frames**, **Different Fusion Methods**, and the **Model Architectures**.

# Input Frames	**ROUGE-L**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
L / 8	29.93	33.36	22.01	14.89	10.62
L / 4	48.26	50.20	40.37	33.23	27.92
L / 2	59.69	61.67	52.52	45.44	39.80
L	**69.80**	**71.36**	**63.01**	**56.28**	**50.76**
# Fusion Methods	**ROUGE-L**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
Concatenate	65.79	67.32	59.09	52.43	47.00
Cross-Attention	69.02	70.47	62.65	55.69	49.94
Ours *w/o* Graph	69.08	70.81	62.72	55.76	50.17
Ours (GSTF)	**69.80**	**71.36**	**63.01**	**56.28**	**50.76**
# Model Architectures	**ROUGE-L**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
BiLSTM	68.07	69.90	61.27	54.34	48.68
Transformer	69.16	70.51	62.11	55.31	49.74
Mamba (Ours)	**69.80**	**71.36**	**63.01**	**56.28**	**50.76**

• **Analysis of Different Fusion Methods.** The graph-guided spatiotemporal fusion (GSTF) module is a key component of our framework. In Table 5, we compare it with other feature-level fusion methods. The results show that our spatiotemporal fusion approach outperforms alternative fusion strategies. Removing the hypergraph modeling strategy (Ours *w/o* Graph) leads to a clear performance drop, demonstrating the effectiveness of employing a hypergraph to capture the relationships among video clips.
• **Ablations against Mamba alternatives.** As shown in Table 5, the Mamba-based architecture consistently outperforms the BiLSTM and Transformer variants across all evaluation metrics. Compared with the Transformer baseline, our design yields gains of +0.64% and +1.02% in ROUGE-L and BLEU-4, respectively. These improvements suggest that the selective state-space modeling mechanism provides a more effective way to capture sequential information. These results further validate the effectiveness of the proposed architecture in modeling long-range dependencies.

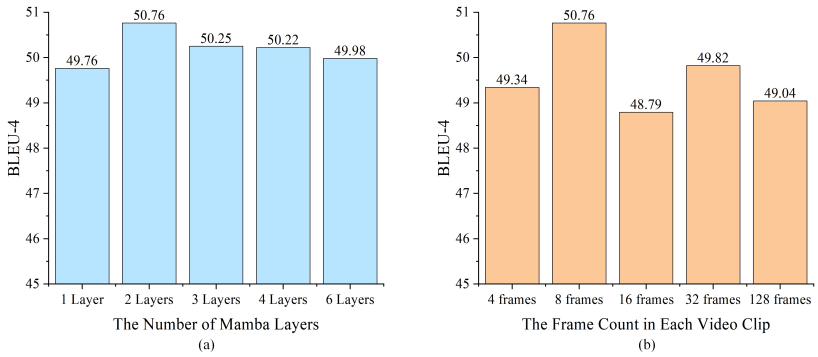


Fig. 3: (a) Comparison with different numbers of Mamba layers. (b) Comparison with different frame counts in each video clip.

• **Analysis of the Depth of Mamba Network.** As shown in Fig. 3 (a), we compare the effect of different numbers of Mamba layers on the final performance. The results show that two stacked Mamba layers yield the optimal configuration. When only a single Mamba layer is used, the model accuracy decreases by 1%, indicating that the model has limited capacity to capture the complex spatiotemporal dependencies across video clips. In contrast, further increasing the number of Mamba layers also leads to performance degradation. Therefore, a two-layer Mamba provides a good trade-off between accuracy and efficiency, resulting in the best overall performance.
• **Analysis of the Frame Count in Each Video Clip.** As shown in Fig. 3 (b), we analyze the number of frames contained in each video clip of the Event-CSL dataset. It is clear that both too few and too many frames result in performance degradation. When too few frames are provided as input, the model lacks sufficient contextual information to capture the semantic context of sign language. In contrast, an excessive number of frames introduces redundant visual information, increasing the model's learning burden

and even leading to overfitting. Therefore, we choose to include a number of 8 frames per video clip to achieve the best performance.

Table 6: Efficiency Analysis on Event-CSL dataset.

Algorithm	# Params	# FLOPs	# Speed
SignCL [52]	115.5M	266.5G	175 ms/video
Sign2GPT [49]	1771.1M	490.5G	247 ms/video
MMSLT [20]	641.9M	2308.9G	1382 ms/video
EvSLT (Ours)	138.2M	252.0G	86 ms/video

5.5 Efficiency Analysis

Table 4 presents a component-wise performance analysis and the associated model parameters. The results show that our framework achieves notable performance gains with only a marginal increase in model parameters. By integrating spatial feature aggregation and spatiotemporal fusion, it achieves a balanced trade-off between complexity and accuracy. Furthermore, as shown in Table 6, we compare our model with recent methods on parameters, FLOPs, and speed. Our model maintains low computational cost and delivers the fastest translation speed (approximately 86 ms per video). Notably, our method adopts a unified single-stage framework, achieving higher translation accuracy while maintaining computational efficiency.

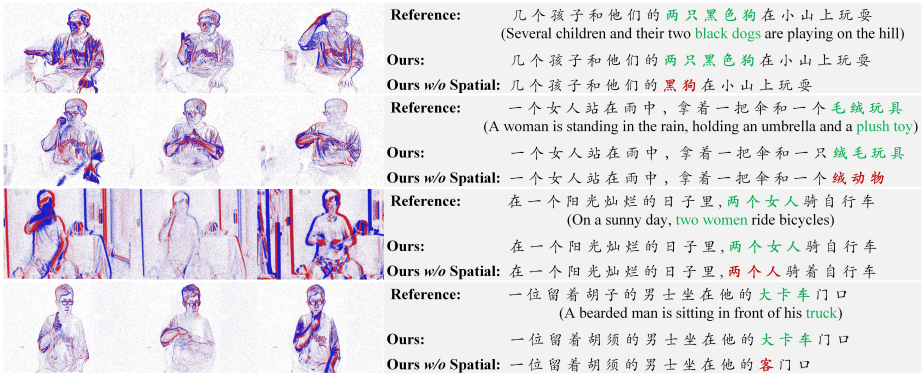


Fig. 4: A qualitative comparison of the generated sign language translation texts.

5.6 Visualization

In this section, we present a qualitative analysis of the SLT results. As shown in the Fig. 4, **Reference** denotes the ground-truth translation sentence corresponding to

the video, **Ours** represents the translation result generated by our model, and **Ours *w/o* spatial** indicates the result obtained using only the temporal branch, without spatial detail features. The red text highlights incorrectly translated keywords, while the green text shows the correct translations of those keywords. It can be observed that, after removing spatial detail features, the model loses its ability to perceive and translate fine-grained sign details, e.g., "两只黑色狗 (two black dogs)" instead of simply "黑狗 (black dog)". This demonstrates that our proposed spatiotemporal fusion method effectively enhances the model's ability to translate detailed components.

6 Limitation Analysis

Although the proposed SLT framework works well, we believe it can be further improved in the following aspects: 1). We transform the event streams into event frames and resize them into 224×224. Designing novel architectures to better leverage high-definition event data remains an interesting direction for future research. 2). Stacking event streams into frames enables fair comparisons with frame-based baselines and the use of pretrained models, but may obscure their asynchronous temporal information. We will explore event-driven frameworks that directly encode continuous event streams to better capture fine-grained temporal dynamics in the temporal branch.

7 Conclusion

In this paper, we propose Event-CSL, the first large-scale, high-definition sign language translation dataset captured using an event camera, aiming to bridge the data scarcity gap in this field. The dataset consists of 14,827 high-definition sign language videos with a resolution of 1280×720, along with 6,097 glosses and 2,544 Chinese words in the text vocabulary. We believe that Event-CSL will serve as a valuable resource for advancing research in event-based sign language translation. Furthermore, we also propose a unified spatiotemporal fusion framework for the sign language translation task. To enhance spatial detail representation, we first introduce a Mamba-based memory aggregation module to compress and aggregate video clip features. Then, a graph-guided spatiotemporal fusion module leverages the compressed clip features as a bridge to effectively fuse spatial features with temporal representations, enabling more precise capture of sign language details. We hope that the proposed benchmark dataset and the novel baseline method will contribute to the advancement of this field and enable deaf individuals to communicate more freely.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grants 62102205 and 62576004, the Natural Science Foundation of Anhui Province under Grants 2408085Y032 and 2408085J037. The authors acknowledge the High-performance Computing Platform of Anhui University for providing computing resources.

References

1. Bottou, L.: Stochastic gradient descent tricks. In: Neural Networks: Tricks of the Trade: Second Edition, pp. 421–436. Springer (2012)
2. Camgoz, N.C., Hadfield, S., Koller, O., Ney, H., Bowden, R.: Neural sign language translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7784–7793 (2018)
3. 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. pp. 10023–10033 (2020)
4. Chai, X., Wanga, H., Zhoub, M., Wub, G., Lic, H., Chena, X.: Devisign: dataset and evaluation for 3d sign language recognition. Technical report, Beijing, Tech. Rep (2015)
5. 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 conference on computer vision and pattern recognition. pp. 5120–5130 (2022)
6. Cheng, Y., Knoll, A., Cao, H.: Urnet: uncertainty-aware refinement network for event-based stereo depth estimation. Visual Intelligence **3**(1), 18 (2025)
7. Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y.: Hypergraph neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 3558–3565 (2019)
8. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., et al.: Event-based vision: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(1), 154–180 (2020)
9. Gehrig, D., Scaramuzza, D.: Low-latency automotive vision with event cameras. Nature **629**(8014), 1034–1040 (2024)
10. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. In: First conference on language modeling (2024)
11. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. In: International Conference on Learning Representations (2021)
12. Guo, D., Wang, S., Tian, Q., Wang, M.: Dense temporal convolution network for sign language translation. In: IJCAI. pp. 744–750 (2019)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Hu, L., Gao, L., Liu, Z., Feng, W.: Continuous sign language recognition with correlation network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2529–2539 (2023)
15. Huang, J., Wang, S., Wang, S., Wu, Z., Wang, X., Jiang, B.: Mamba-fetrack: Frame-event tracking via state space model. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV). pp. 3–18. Springer (2024)
16. Hwang, E.J., Cho, S., Lee, J., Park, J.C.: An efficient gloss-free sign language translation using spatial configurations and motion dynamics with llms. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 3901–3920 (2025)
17. Jiang, Y., Wang, Y., Li, S., Zhang, Y., Guo, Q., Chu, Q., Gao, Y.: Evcslr: Event-guided continuous sign language recognition and benchmark. IEEE Transactions on Multimedia **27**, 1349–1361 (2024)
18. Joze, H.R.V., Koller, O.: Ms-asl: A large-scale data set and benchmark for understanding american sign language. In: British Machine Vision Conference. p. 100 (2018)
19. Kalman, R.E.: A new approach to linear filtering and prediction problems (1960)

20. Kim, J., Jeon, H., Bae, J., Kim, H.Y.: Leveraging the power of mllms for gloss-free sign language translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21048–21058 (2025)
21. Koller, O., Forster, J., Ney, H.: Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding **141**(1), 108–125 (2015)
22. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
23. 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. Advances in Neural Information Processing Systems **33**, 12034–12045 (2020)
24. Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., Qiao, Y.: Videomamba: State space model for efficient video understanding. In: European conference on computer vision. pp. 237–255. Springer (2024)
25. Li, X., Xu, C., Wang, X., Lan, W., Jia, Z., Yang, G., Xu, J.: Coco-cn for cross-lingual image tagging, captioning and retrieval. IEEE Transactions on Multimedia **21**(9), 2347–2360 (2019)
26. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
27. Lin, K., Wang, X., Zhu, L., Sun, K., Zhang, B., Yang, Y.: Gloss-free end-to-end sign language translation. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12904–12916 (2023)
28. Liu, C., Yuan, Y., Chen, X., Lu, H., Wang, D.: Spatial-temporal initialization dilemma: towards realistic visual tracking. Visual Intelligence **2**(1), 35 (2024)
29. Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., Zettlemoyer, L.: Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics **8**, 726–742 (2020)
30. Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y.: Vmamba: visual state space model. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. pp. 103031–103063 (2024)
31. Min, Y., Hao, A., Chai, X., Chen, X.: Visual alignment constraint for continuous sign language recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11542–11551 (2021)
32. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
34. Peng, S., Zhu, X., Yi, D., Qian, C., Lei, Z.: Formulating facial mesh tracking as a differentiable optimization problem: a backpropagation-based solution. Visual Intelligence **2**(1), 21 (2024)
35. Shi, Y., Xia, B., Jin, X., Wang, X., Zhao, T., Xia, X., Xiao, X., Yang, W.: Vmambair: Visual state space model for image restoration. IEEE Transactions on Circuits and Systems for Video Technology **35**(6), 5560–5574 (2025)
36. Sridhar, A., Ganesan, R.G., Kumar, P., Khapra, M.: Include: A large scale dataset for indian sign language recognition. In: ACM International Conference on Multimedia. pp. 1366–1375 (2020)
37. Tang, S., Xue, F., Wu, J., Wang, S., Hong, R.: Gloss-driven conditional diffusion models for sign language production. ACM Transactions on Multimedia Computing, Communications and Applications **21**(4), 1–17 (2025)

38. Vasudevan, A., Negri, P., Linares-Barranco, B., Serrano-Gotarredona, T.: Introduction and analysis of an event-based sign language dataset. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020). pp. 675–682. IEEE (2020)
39. Von Agris, U., Kraiss, K.F.: Signum database: Video corpus for signer-independent continuous sign language recognition. In: 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies. pp. 243–246 (2010)
40. Wang, S., Guo, D., Zhou, W.g., Zha, Z.j., Wang, M.: Connectionist temporal fusion for sign language translation. In: Proceedings of the 26th ACM international conference on Multimedia. pp. 1483–1491 (2018)
41. Wang, X., Li, J., Zhu, L., Zhang, Z., Chen, Z., Li, X., Wang, Y., Tian, Y., Wu, F.: Visevent: Reliable object tracking via collaboration of frame and event flows. IEEE transactions on cybernetics **54**(3), 1997–2010 (2024)
42. Wang, X., Wang, S., Shao, P., Zhu, L., Jiang, B., Tian, Y.: Event stream based human action recognition: a high-definition benchmark dataset and algorithms. International Journal of Computer Vision **134**(4), 181 (2026)
43. Wang, X., Wang, S., Tang, C., Zhu, L., Jiang, B., Tian, Y., Tang, J.: Event stream-based visual object tracking: A high-resolution benchmark dataset and a novel baseline. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19248–19257 (2024)
44. Wang, X., Wu, Z., Jiang, B., Bao, Z., Zhu, L., Li, G., Wang, Y., Tian, Y.: Hardvs: Revisiting human activity recognition with dynamic vision sensors. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5615–5623 (2024)
45. Wang, X., Tang, S., Cheng, L., Li, F., Wang, S., Hong, R.: Signaligner: Harmonizing complementary pose modalities for coherent sign language generation. arXiv preprint arXiv:2506.11621 (2025)
46. Wang, X., Tang, S., Song, P., Wang, S., Guo, D., Hong, R.: Linguistics-vision monotonic consistent network for sign language production. In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2025)
47. Wang, Y., Zhang, X., Shen, Y., Du, B., Zhao, G., Cui, L., Wen, H.: Event-stream representation for human gaits identification using deep neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3436–3449 (2021)
48. Wang, Y., Zhang, X., Wang, Y., Wang, H., Huang, C., Shen, Y.: Event-based american sign language recognition using dynamic vision sensor. In: Workshop on Applications of Software Agents. pp. 3–10 (2021)
49. Wong, R., Camgoz, N.C., Bowden, R.: Sign2GPT: Leveraging large language models for gloss-free sign language translation. In: The Twelfth International Conference on Learning Representations (2024)
50. Yang, Y., Xing, Z., Yu, L., Huang, C., Fu, H., Zhu, L.: Vivim: A video vision mamba for medical video segmentation. arXiv preprint arXiv:2401.14168 (2024)
51. Ye, J., Jiao, W., Wang, X., Tu, Z., Xiong, H.: Cross-modality data augmentation for end-to-end sign language translation. In: Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 13558–13571 (2023)
52. Ye, J., Wang, X., Jiao, W., Liang, J., Xiong, H.: Improving gloss-free sign language translation by reducing representation density. Advances in Neural Information Processing Systems **37**, 107379–107402 (2024)
53. 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. pp. 2551–2562 (2023)
54. Zhang, P., Yin, H., Wang, Z., Chen, W., Li, S., Wang, D., Lu, H., Jia, X.: Evsign: Sign language recognition and translation with streaming events. In: European conference on computer vision. pp. 335–351. Springer (2024)

55. 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 International Conference on Computer Vision (ICCV). pp. 20871–20881 (October 2023)
56. 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. pp. 1316–1325 (2021)
57. Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. In: International Conference on Machine Learning. pp. 62429–62442 (2024)