HyperSign: Saliency-Aware Spatial Graphs and Temporal Hypergraphs for Continuous Sign Language Recognition

Weiyi Ye[1], Xu-Hua Yang[1,2*], Dong Wei[1], Gang-Feng Ma[3], Yujiao Huang[1], Xiao-Xin Li[1]

[1] College of Computer Science and Technology, Zhejiang University of Technology
[2] Zhejiang Key Laboratory of Visual Information Intelligent Processing
[3] School of Computer Science and Technology, Zhejiang Normal University
{ywy23, xhyang, vddong, gf_ma, huangyujiao, mordekai}@zjut.edu.cn

Abstract

Continuous sign language recognition (CSLR) technology enables social communication for the hearing-impaired by converting sign language videos into text. However, due to the limited receptive fields of convolutional networks and inefficient long-range dependency modeling in temporal modules, current methods find it difficult to capture cross-regional and high-order dynamic semantics in complex gestures. To address these limitations, we propose a dynamic spatiotemporal hypergraph network named HyperSign, which optimizes feature learning through innovative graph architectures. For single-frame spatial modeling, we propose a saliency-aware spatial graph construction strategy that dynamically quantifies semantic saliency by integrating feature complexity and motion intensity information from patches. This strategy can adaptively adjust node connectivity based on the computed saliency, thereby enabling the graph structure to focus on information-dense regions such as hands and faces. For temporal dependency modeling, we abandon the conventional pairwise frame interactions and propose a temporal hypergraph construction method. This method employs a learnable clustering algorithm to aggregate semantically correlated nodes within temporal windows into hyperedges, thereby explicitly capturing high-order associations within individual gesture actions that span multiple frames. Extensive experiments on the PHOENIX14, PHOENIX14-T, and CSL-Daily datasets demonstrate that HyperSign outperforms the state-of-the-art (SOTA) approaches in CSLR without any additional annotation information, establishing a new feature learning paradigm for the CSLR task.

Introduction

As a visual language, sign language relies on the coordination of multimodal elements such as hand movements, facial expressions, and body postures. Continuous Sign Language Recognition (CSLR) aims to convert sequential sign language video streams into their corresponding gloss sequences, the atomic lexical units of sign language (Min et al. 2021; Hu et al. 2023b). Current CSLR methods (Cui, Liu, and Zhang 2019; Zheng et al. 2023) predominantly utilize CNNs as feature extraction backbones (Gan et al. 2021; Lu, Salah, and Poppe 2024), augment temporal receptive fields through adjacent frame concatenation (Hu et al. 2023b; Yang, Hu, and Lin 2025), and subsequently employ 1D CNNs and BiLSTM for local and global temporal modeling (Xu et al. 2025; Yu et al. 2025a). However, this architecture has significant limitations. As shown in Figure 1, traditional CNNs primarily extract holistic spatial features from single frames, failing to effectively distinguish the salient regions for sign expression, such as the hands and face. Furthermore, as the semantic meaning in sign language often spans multiple timesteps, the sole reliance on the limited temporal information between adjacent frames is insufficient for capturing complex cross-frame dynamic patterns.

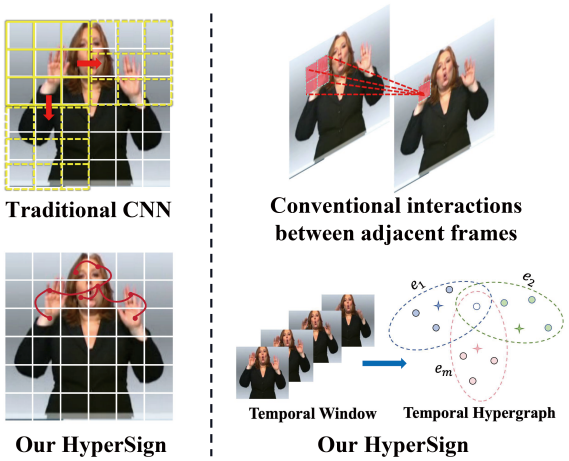

Figure 1: (Left) Comparison of spatial feature extraction between traditional CNNs and our HyperSign. (Right) Comparison of temporal feature extraction between methods based on conventional interactions and our HyperSign.

Researchers have sought to overcome the limitations of existing CSLR frameworks by fusing domain knowledge, such as skeleton information (Hu et al. 2023a; Yang, Min, and Chen 2024), text information (Zheng et al. 2023; Guo et al. 2023), and other visual features like keypoints and optical flow (Zuo and Mak 2022; Chen et al. 2022). While these multimodal methods improve recognition accuracy, they also increase computational complexity and render models

*Corresponding Author

more susceptible to annotation noise. Recent research (Gan et al. 2024) introduced the Visual Graph Neural Network (ViG) (Han et al. 2022) to achieve non-local feature modeling through the construction of explicit topological relationships. Nevertheless, ViG's fixed neighborhood strategy treats all nodes equally, ignoring regional importance variations in sign expression. This uniform approach cannot handle CSLR's information imbalance, limiting representation capacity and generalization.

To address the aforementioned limitations, we propose HyperSign, a novel model that leverages dynamic high-order graph structures for efficient sign language feature learning. For spatial feature modeling, we propose a semantic saliency-aware spatial graph construction strategy, which quantifies node-level semantic importance by jointly evaluating feature complexity and motion intensity, thereby dynamically allocating graph connectivity to enable adaptive focus on critical regions. Regarding cross-frame temporal modeling, we propose a temporal hypergraph construction method that aggregates semantically related nodes within temporal windows into hyperedges, explicitly capturing high-order dynamic patterns across multiple frames. The main contributions of this work are summarized as follows:

- We propose the **Saliency-Aware Spatial Graph Module (SASGM)**, a dynamic graph construction strategy that adjusts node connectivity based on semantic importance, enabling adaptive focus on information-dense regions while suppressing background noise.
- We introduce the **Temporal Hypergraph Module (THM)** that captures high-order temporal dependencies across multiple frames through learnable clustering, explicitly modeling complex gesture dynamics that span beyond pairwise frame interactions.
- We present **HyperSign**, a hierarchical architecture that cascades spatial graph and temporal hypergraph modules to achieve multi-scale spatio-temporal feature learning, significantly advancing CSLR performance.
- The extensive experiments on three widely used datasets demonstrate that our proposed HyperSign model outperforms the state-of-the-art (SOTA) models without relying on any additional domain knowledge or external annotations.

Related Work

Continuous Sign Language Recognition

Existing CSLR methods can be broadly categorized into two paradigms: single-cue methods and multi-cue methods. Single-cue methods directly decode gloss sequences from RGB videos. For instance, VAC (Min et al. 2021) regards the visual encoder as a student and the gloss decoder as a teacher, leveraging knowledge distillation to align video sequences with textual gloss modalities. CVT-SLR (Zheng et al. 2023) uses a pre-trained variational autoencoder to achieve cross-modal alignment of vision and gloss, while CAP-SLR (Wei et al. 2025) adopts a multi-scale dilated convolutional attention module to enhance the modeling of key regions in sign language. Nevertheless, existing single-cue models have difficulty in adequately capturing the inherent spatiotemporal semantic dependencies in sign language videos, constraining their accuracy and robustness.

Multi-cue methods, conversely, enhance CSLR accuracy by incorporating additional pre-processed cues. Examples include C^2ST (Zhang et al. 2023), which integrates knowledge of gloss sequences into video representation learning and sequence transduction, and C^2SLR (Zuo and Mak 2022), which leverages pose heatmaps to supervise and optimize visual representations. Research also explores multi-stream architectures, such as SlowFastSign (Ahn, Jang, and Chung 2024), which fuses spatiotemporal features across multi-temporal resolutions via dual-path networks, and TwoStream-SLR (Chen et al. 2022), which progressively integrates RGB videos with keypoint heatmaps to achieve complementarity. However, these multi-cue methods typically require elevated training costs, more intricate architectures, and greater inference latency, which limit their practicality in actual CSLR applications.

Vision Graph Neural Network

Graph Neural Networks (GNNs) (Jiang et al. 2019) provide a new paradigm for computer vision that breaks through the receptive field limitations of traditional CNNs. As a pioneering work in this domain, ViG (Han et al. 2022) segments images into patches as nodes and employs the K-Nearest Neighbours (K-NN) algorithm to construct graph topologies, effectively capturing long-range dependencies within images. However, the inherent high computational complexity of this algorithm constrains its practical application. To address this limitation, researchers have proposed various optimization strategies aimed at reducing computational overhead. Specifically, MobileViG (Munir, Avery, and Marculescu 2023) incorporates a graph-based sparse attention mechanism, while GreedyViG (Munir et al. 2024) adopts a dynamic axial graph construction strategy to limit edge connectivity. Additionally, WiGNet (Spadaro et al. 2025) introduces a window-constrained K-NN method that confines graph construction to local regions. To further enhance the graph representation capability, recent research has explored hypergraph structures. For instance, Vision HGNN (Han et al. 2023) pioneered the encoding of images into hypergraphs, utilizing hyperedges to connect multiple nodes and capturing high-order features through hypergraph convolution. On this basis, DVHGNN (Li et al. 2025) introduced a dilated convolution mechanism to efficiently extract multi-scale and high-order visual information by setting various dilation rates, marking a further advancement in this research direction.

Although existing ViG research has made some progress, it still faces challenges in dealing with complex spatiotemporal dynamic tasks such as CSLR. Existing ViG architectures typically rely on graph construction strategies with a fixed number of neighbours and low-order interaction modeling (Gan et al. 2024), which makes it difficult to flexibly adapt to the dynamic changes in sign language information density and effectively capture the high-order semantic correlations that span across multiple frames within gestures.

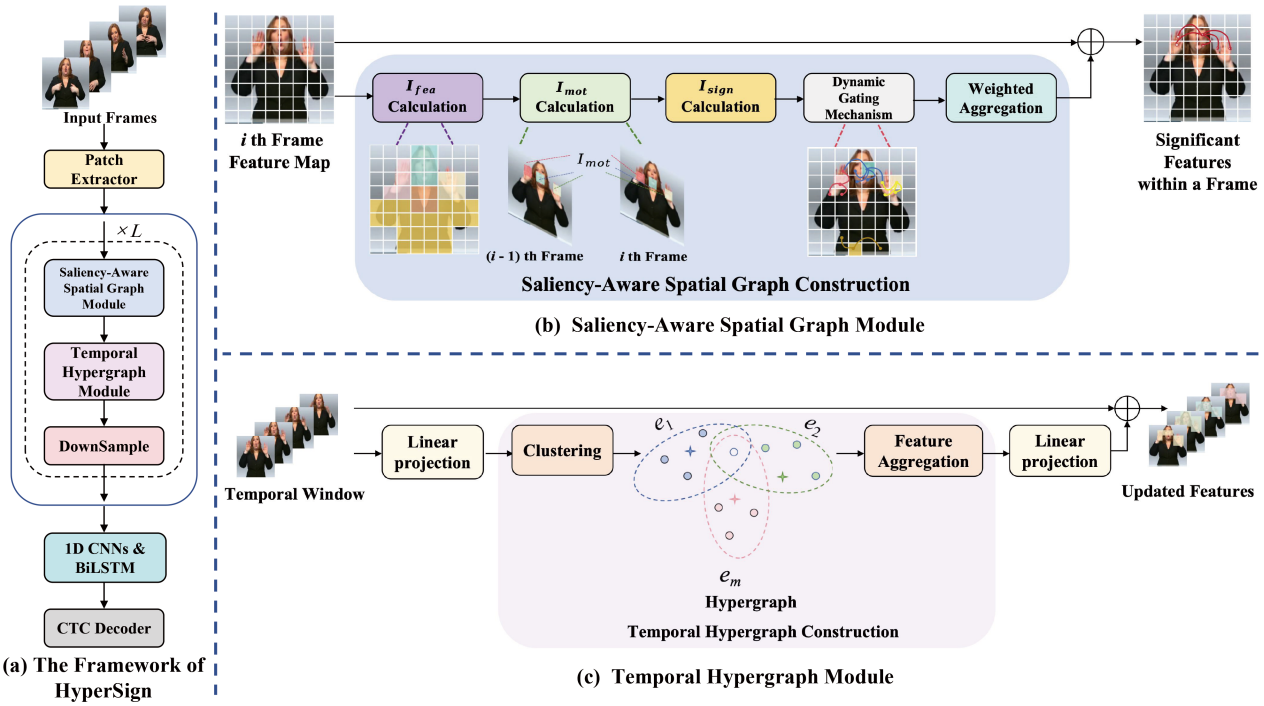

Figure 2: (a) The Framework of **HyperSign**, comprising a patch extractor followed by two core modules: a saliency-aware spatial graph module and a temporal hypergraph module. (b) **Saliency-Aware Spatial Graph Module**, which consists of saliency calculation steps and a dynamic gating mechanism to integrate them for spatial feature enhancement adaptively. (c) **Temporal Hypergraph Module**, which constructs a hypergraph via clustering and then performs feature aggregation to update the temporal features.

Methodology

Overall Framework

As illustrated in Figure 2(a), the proposed HyperSign model employs a hierarchical spatio-temporal architecture. Given an input sign language video sequence $X = \{X_i\}_{i=1}^{T} \in \mathbb{R}^{T\times C\times H\times W}$, where T denotes frame count, C represents the number of channels, and $H \times W$ indicates the spatial dimensions. The sequence is first processed by the Patch Extractor module, which partitions each frame into N non-overlapping patches and projects them into a feature sequence $F \in \mathbb{R}^{T\times N\times D}$ where $N = (H/P) \times (W/P)$, P is the patch size, and D represents the feature dimension.

The feature sequence is then processed by a core network comprising L cascaded modules. As depicted in Figures 2(b) and 2(c), each module contains two core components: the **Saliency-Aware Spatial Graph Module (SASGM)** and **Temporal Hypergraph Module (THM)**. The SASGM captures long-range dependencies between critical spatial regions through dynamic graph construction within individual frames, while the THM models high-order temporal correlations across frames utilizing hypergraph structures. Between consecutive modules, N non-overlapping patches $p_i = \{p_{ij}\}_{j=1}^{N}$ are progressively merged into larger patches $\hat{p}_i = \{\hat{p}_{ij}\}_{j=1}^{N/4}$. This downsampling is achieved by merging the feature representations within 2×2 blocks of spatially adjacent patches. This progressive feature integration enlarges the receptive field and constructs multi-scale feature representations, effectively capturing hierarchical information from local details to global dynamics.

Consistent with prior CSLR models (Hu et al. 2023b; Gan et al. 2024), the resultant features are aggregated through 1D CNNs and BiLSTM to model global context following multi-level feature extraction. Final outputs are classified and aligned with target gloss sequences using Connectionist Temporal Classification (CTC) loss optimization (Graves et al. 2006).

Saliency-Aware Spatial Graph Module

Given a feature map $F_i \in \mathbb{R}^{N\times D}$ extracted from the i-th frame, we quantify the semantic saliency of each node by evaluating its static feature complexity and dynamic motion intensity. Specifically, the *feature complexity* I_{fea} measures information density through the reconstruction error of downsampling-upsampling operations:

$$I_{\text{fea}}(v_{ij}) = \sum_{d=1}^{D} \left| F_i^{(d)}(v_{ij}) - \text{Up}\left(\text{Down}\left(F_i^{(d)}(v_{ij})\right)\right)\right| \tag{1}$$

where $F_i^{(d)}(v_{ij})$ denotes the feature value of node v_{ij} along dimension d, with $\text{Up}(\cdot)$ and $\text{Down}(\cdot)$ represent bilinear upsampling and average pooling, respectively.

Recognizing that static features alone are insufficient to capture the dynamic nature of sign language, we also quantify motion intensity using a frame-differencing approach. The *motion saliency*, I_{mot}, is defined as the L1 norm of the difference between the feature vectors of patches at the same spatial location in adjacent frames:

$$I_{\text{mot}}(v_{ij}) = \begin{cases} \|F_i(v_{ij}) - F_{i-1}(v_{ij})\|_1 & i > 1 \\ 0 & i = 1 \end{cases} \tag{2}$$

These two metrics are then integrated into a unified *semantic saliency score* I_{sign}:

$$I_{\text{sign}}(v_{ij}) = \lambda \cdot \widetilde{I_{\text{fea}}}(v_{ij}) + (1 - \lambda) \cdot \widetilde{I_{\text{mot}}}(v_{ij}) \tag{3}$$

where $\lambda \in [0, 1]$ is a hyperparameter, and $\widetilde{(\cdot)}$ denotes min-max normalization applied per-frame across all N nodes.

Based on this saliency score, we introduce a dynamic gating mechanism to modulate the connectivity strength between any node pair (v_{ij}, v_{ik}). The gating value $g_{jk} \in (0, 1)$ is computed as:

$$g_{jk} = \sigma\left(\mathbf{W}_g^\top \left[I_{\text{sign}}(v_{ij}), I_{\text{sign}}(v_{ik})\right] + b_g\right) \tag{4}$$

where $\mathbf{W}_g \in \mathbb{R}^2$ and $b_g \in \mathbb{R}$ are learnable parameters, and $\sigma(\cdot)$ is the sigmoid function. This mechanism allows the model to adaptively allocate more attention to salient regions, such as hands and face, which are critical for understanding signs.

Finally, the node features are updated through a saliency-weighted aggregation process. The features are first linearly transformed, then aggregated based on the gating values, and finally updated with a residual connection:

$$f'_{ik} = f_{ik}\mathbf{W}_l \tag{5}$$

$$f_{ij}^{\text{new}} = \text{ReLU}\left(\sum_{k=1}^{N} g_{jk} \cdot f'_{ik}\right) \tag{6}$$

$$f_{ij}^{\text{final}} = f_{ij}^{\text{new}}\mathbf{W}'_l + f_{ij} \tag{7}$$

where $\mathbf{W}_l \in \mathbb{R}^{D \times D'}$ and $\mathbf{W}'_l \in \mathbb{R}^{D' \times D}$ is learnable projection matrices to ensure dimensional consistency. The residual connection helps to preserve original features and stabilize the training process.

Unlike methods that rely on fixed-neighbourhood graphs, our approach dynamically adjusts graph connectivity based on input-dependent saliency. This prioritizes information-dense regions while suppressing background noise, enabling efficient cross-region feature learning that is critical for capturing sign language semantics.

Temporal Hypergraph Module

Following spatial feature extraction, we obtain a node feature sequence $\mathbf{F} \in \mathbb{R}^{T \times N \times D}$, where T, N, and D denote the temporal length, number of nodes per frame, and feature dimension, respectively. To capture high-order temporal dependencies, we employ a sliding window approach with window size K and stride 1. For each temporal window starting at time t, we concatenate the node features:

$$\mathbf{F}_t^{\text{win}} = \text{concat}(\mathbf{F}_t, \mathbf{F}_{t+1}, \ldots, \mathbf{F}_{t+K-1}) \in \mathbb{R}^{(K \times N) \times D} \tag{8}$$

To facilitate efficient semantic clustering, we project the concatenated features to a lower-dimensional embedding space:

$$\mathbf{Z}_t = \mathbf{F}_t^{\text{win}}\mathbf{W}_p \in \mathbb{R}^{(K \times N) \times d} \tag{9}$$

where $\mathbf{W}_p \in \mathbb{R}^{D \times d}$ is a learnable projection matrix and $d \ll D$. We then introduce M learnable cluster centers $\mathcal{C} = \{c_m\}_{m=1}^{M}$, where each $c_m \in \mathbb{R}^d$ adapts dynamically during training. For each node embedding z_j, we compute its similarity to all cluster centers using cosine similarity:

$$S_{jm} = \frac{z_j \cdot c_m^T}{\|z_j\| \cdot \|c_m\|} \tag{10}$$

Based on similarity scores, each cluster center c_m defines a hyperedge e_m containing its most similar nodes:

$$e_m = \{v_j \in V \mid m = \arg\max_k(S_{jk})\} \tag{11}$$

where V represents all nodes within the temporal window. A straight-through estimator (STE) is used to pass gradients through this non-differentiable assignment. The hypergraph topology is encoded by an incidence matrix $H \in \{0, 1\}^{(K \times N) \times M}$, where $H_{jm} = 1$ if node $v_j \in e_m$.

For each hyperedge, we compute its aggregated representation as a weighted combination of member nodes:

$$h_m = \sum_{j:H_{jm}=1} w_{jm} z_j \tag{12}$$

where w_{jm} results from applying Softmax to S_{jm} scores, normalized across all nodes j within the hyperedge e_m. Each node then aggregates information from its associated hyperedge:

$$\mathbf{Z}_j^{\text{agg}} = \sum_{m:H_{jm}=1} h_m \tag{13}$$

To map the aggregated features back to the original feature space, we apply a linear transformation followed by residual connection:

$$\mathbf{F}_j^{\text{new}} = \alpha\mathbf{F}_j + (1 - \alpha) \cdot (\mathbf{Z}_j^{\text{agg}}\mathbf{W}_q) \tag{14}$$

where $\mathbf{W}_q \in \mathbb{R}^{d \times D}$ is a learnable projection matrix and $\alpha \in [0, 1]$ is a balancing hyperparameter. We denote the set of N features corresponding to frame i within window l, updated via Eq. (14), as $\mathbf{F}_{i,l}^{\text{new}}$.

Due to the overlapping nature of sliding windows, a single frame may belong to multiple windows. Therefore, the final representation of the characteristic of frame i is obtained by averaging the outputs of all windows containing it:

$$\mathbf{F}_i^{\text{final}} = \frac{1}{|\mathcal{W}_i|} \sum_{l \in \mathcal{W}_i} \mathbf{F}_{i,l}^{\text{new}} \tag{15}$$

where $\mathcal{W}_i$ denotes the set of window indices containing frame i.

This hypergraph-based mechanism transforms each node's feature from isolated single-frame data into a comprehensive representation incorporating high-order temporal dynamics. Unlike conventional pairwise frame interactions, it explicitly models complex multi-frame collaborations through dynamic hypergraphs, capturing long-range dependencies in CSLR.

Methods	Backbone	PHOENIX14				PHOENIX14-T	
		Dev(%)		Test(%)		Dev(%)	Test(%)
		del/ins	WER ↓	del/ins	WER ↓	WER ↓	WER ↓
VAC (Min et al. 2021)	ResNet18	7.9/2.5	21.2	8.4/2.6	22.3	-	-
SMKD (Hao, Min, and Chen 2021)	ResNet18	6.8/2.5	20.8	6.3/2.3	21.0	20.8	22.4
CoSign-2s† (Jiao et al. 2023)	ST-GCN(GCN)	-	19.7	-	20.1	19.5	20.1
TLP (Hu et al. 2022)	ResNet18	6.3/2.8	19.7	6.1/2.9	20.8	19.4	21.2
AdaBrowse (Hu et al. 2023d)	ResNet18	6.0/2.5	19.6	5.9/2.6	20.7	19.5	20.6
SEN (Hu et al. 2023c)	ResNet18	5.8/2.6	19.5	7.3/4.0	21.0	19.3	20.7
CorrNet (Hu et al. 2023b)	ResNet18	5.6/2.8	18.8	5.7/2.3	19.4	18.9	20.5
SignGraph† (Gan et al. 2024)	Customized(GCN)	4.9/2.0	18.2	5.3/1.9	19.1	17.8	19.1
CSGC (Rao et al. 2024)	ResNet18	-	18.1	-	19.0	17.2	19.5
TCNet (Lu, Salah, and Poppe 2024)	ResNet18	5.5/2.4	18.1	5.4/2.0	18.9	18.3	19.4
CVSign (Yu et al. 2025a)	ResNet34	-	17.8	-	18.0	17.4	18.6
HSTENet (Xu et al. 2025)	ResNet34	-	17.5	-	17.7	17.4	18.7
OLMD (Yu et al. 2025b)	ResNet34	4.8/2.4	17.1	4.7/2.3	17.2	17.1	18.4
STMC* (Zhou et al. 2020)	VGG11	7.7/3.4	21.1	7.4/2.6	20.7	19.6	21.0
C^2SLR* (Zuo and Mak 2022)	VGG11	-	20.5	-	20.4	20.2	20.4
HST-GNN†* (Kan et al. 2022)	Customized(CNN+GCN)	-	19.5	-	19.8	19.5	19.8
TwoStream* (Chen et al. 2022)	S3D	-	18.4	-	18.8	17.7	19.3
HyperSign† (Ours)	Customized(GCN)	**4.8/2.0**	**16.9**	**4.6/1.8**	**17.1**	**16.6**	**18.2**

Table 1: Comparison of CSLR performance on PHOENIX14 and PHOENIX14-T, † represents the adoption of the GNN methodology, * indicates the introduction of additional visual cues.

Experiments

Datasets

We conduct our experiments on three public datasets.

PHOENIX14(Koller, Forster, and Ney 2015) is a corpus derived from German weather forecasts, featuring nine signers recorded against a high-contrast background at a resolution of 210×260 pixels. The dataset comprises a vocabulary of 1,295 glosses distributed across 6,841 sentences. The standard data split consists of 5,672 samples for training, 540 for development, and 629 for testing.

PHOENIX14-T(Camgoz et al. 2018) serves as an extension to the PHOENIX14 dataset. It contains a vocabulary of 1,085 sign glosses within a total of 8,247 sentences, which are partitioned into 7,096 training, 519 development, and 642 test samples.

CSL-Daily(Zhou et al. 2021) is a comprehensive dataset featuring content related to daily life. The videos were recorded indoors with ten signers at a frame rate of 30 fps. It encompasses a total of 20,654 sentences, split into 18,401 for training, 1,077 for development, and 1,176 for testing.

Implementation Details

This section elaborates on the implementation of HyperSign.

Network Architecture. For fair comparison with recent SOTAs(Wei et al. 2025; Yu et al. 2025b; Xu et al. 2025), we use ResNet34 (He et al. 2016) pre-trained on ImageNet as the Patch Extractor module. Our core network uses $L = 4$ cascaded modules, and 32×32 was selected as the default patch size for all subsequent experiments. For 1D CNNs, we adopt SOTA configurations {K5, P2, K5, P2}, where K5 represents a convolution with a kernel size of 5, and P2 represents pooling with a kernel size of 2. We adopt the pooling method from TLP (Hu et al. 2022), setting the hidden layers of the BiLSTM to 1024. In the THM, the embedding dimension d (Eq. 9) was set to 128.

Training and Inference Strategy. We trained the model for 100 epochs, with the initial learning rate of 0.001, and reduced it to 20% at epochs 30 and 60. The default Adam (Kingma and Ba 2014) optimizer is used with a weight decay of 0.001, and the batch size is 4. During training, all input frames are first resized to 256×256 pixels and then randomly cropped to 224×224 pixels, with a 50% chance of horizontal flipping and a 20% probability of temporal rescaling. For inference, we use a central crop of 224×224 pixels. All experiments are implemented on an Intel(R) Xeon(R) Gold 6326 CPU platform and an NVIDIA RTX A6000 48G GPU.

Evaluation Metric

We use word error rate (WER) as the performance metric for our model. It measures the minimum substitutions (#sub), insertions (#ins), and deletions (#del) required to match a predicted sentence with the reference (#ref):

$$WER = \frac{\#sub + \#ins + \#del}{\#ref} \tag{16}$$

Note that the **lower** the WER, the **better** the accuracy.

Comparison with State-of-the-Art Methods

We evaluate the performance of HyperSign on three widely used public datasets.

Methods	Dev(%)	Test(%)
SignBT (Zhou et al. 2021)	33.2	33.2
AdaBrowse (Hu et al. 2023d)	31.2	30.7
SEN (Hu et al. 2023c)	31.1	30.7
CorrNet (Hu et al. 2023b)	30.6	30.1
TCNet (Lu, Salah, and Poppe 2024)	29.7	29.3
CoSign-2s† (Jiao et al. 2023)	28.1	27.2
SignGraph† (Gan et al. 2024)	27.3	26.4
OLMD (Yu et al. 2025b)	25.8	24.7
TwoStream* (Chen et al. 2022)	25.4	25.3
HyperSign† (Ours)	**25.1**	**24.2**

Table 2: Comparison of CSLR performance on CSL-Daily, † represents the adoption of the GNN methodology, * indicates the introduction of additional visual cues.

SASGM	THM	Dev(%)	Test(%)
-	-	21.3	21.1
✓	-	17.5	17.7
-	✓	17.9	18.3
✓	✓	**16.9**	**17.1**

Table 3: Ablations for SASGM, THM on PHOENIX14.

On PHOENIX14 and PHOENIX14-T. As shown in Table 1, we compare our model with other SOTA models on PHOENIX14 and PHOENIX14-T datasets. All models are categorized into two groups: single-cue models and multi-cue models. Notably, HyperSign outperforms all single-cue and multi-cue models. For instance, our WER on PHOENIX14 is reduced by 0.2% (Dev) and 0.1% (Test) compared to the single-cue SOTA OLMD. On PHOENIX14-T, the WER is still reduced by 0.5% (Dev) and 0.2% (Test) compared to OLMD. Multi-Cue SOTA Two-stream relies on auxiliary keypoint sequences and a large network structure to improve recognition. However, HyperSign outperforms it with single-cue inputs and a lightweight structure. Remarkably, the WER on PHOENIX14 is reduced by 1.5% (Dev) and 1.7% (Test). On PHOENIX14-T, the WER is still reduced by 1.1% (Dev) and 1.1% (Test). These results underscore the effectiveness of HyperSign in capturing and modeling the intricate motion features in sign language, resulting in significant performance improvements.

On CSL-Daily. CSL-Daily is a challenging dataset with the largest vocabulary and the largest number of sign language videos among all publicly available CSLR datasets. As shown in Table 2, HyperSign demonstrates superior performance, outperforming all single-cue and multi-cue counterparts. Compared to the single-cue method SOTA OLMD, HyperSign achieves a significant WER reduction of 0.7% on the dev set and 0.5% on the test set. Moreover, it surpasses the multi-cue SOTA model Two-stream, with WER reductions of 0.3% (dev) and 1.1% (test). These outcomes validate the advanced design of HyperSign and its robust scalability to more extensive and complex datasets.

Operation	Dev(%)	Test(%)
Only I_{fea}+Gating	17.5	17.8
Only I_{fea}+K-NN	17.7	18.3
Only I_{fea}+Fixed threshold	17.8	18.4
Only I_{mot}+Gating	17.9	18.4
Only I_{mot}+K-NN	18.1	18.7
Only I_{mot}+Fixed threshold	18.4	19.1
I_{sign}+K-NN	17.2	17.5
I_{sign}+Fixed threshold	17.5	17.8
Only K-NN	18.4	19.2
random K	19.8	20.7
Fully connected	20.8	21.4
HyperSign (Ours)	**16.9**	**17.1**

Table 4: Ablations for different graph construction strategies in SASGM on PHOENIX14.

Ablation Studies

We evaluate the effectiveness of each component via ablation studies on PHOENIX14.

Ablations for the Saliency-Aware Spatial Graph Module (SASGM) and Temporal Hypergraph Module (THM). Table 3 presents an ablation study of two key components: SASGM and THM. When applied individually, SASGM achieves substantial improvements of 3.8% on Dev and 3.4% on Test, while THM delivers more modest gains of 3.4% and 2.8%, respectively. The combination of both modules achieves the highest improvements of 4.4% on Dev and 4.0% on Test, which validates the complementary benefits of integrating spatial and temporal feature modeling.

Ablations for different graph construction strategies in SASGM. Table 4 demonstrates the effectiveness of our saliency-aware graph construction approach, which combines saliency information (I_{sign}) with an adaptive gating mechanism. Our method significantly outperforms all other variants, notably K-NN and fixed-threshold methods using different saliency cues (I_{fea}, I_{mot}), as well as simplistic baselines without saliency (standard K-NN and fully-connected graphs). This confirms the key contributions of both the adaptive connection mechanism and saliency information.

Ablations for different hyperparameters λ in SASGM. We optimize λ, which weights the contribution of static feature complexity and dynamic motion information in SASGM. As shown in Figure 3, experiments across $\lambda \in [0, 1]$ reveal peak performance at $\lambda = 0.6$, achieving the lowest WER. This indicates that a slight emphasis on feature complexity yields optimal feature representations.

Ablations for different hyperparameters M and α in THM. THM has two key hyperparameters: the number of cluster centers (M) and the balancing factor (α) between original and aggregated hypergraph features. Since exhaustive grid search is computationally expensive, we optimized each parameter individually while fixing the other at a reasonable value. As shown in Figure 4, performance peaks at M = 5 for both Dev and Test sets. With M fixed at 5, we then tuned α and found optimal performance at α = 0.2. There-

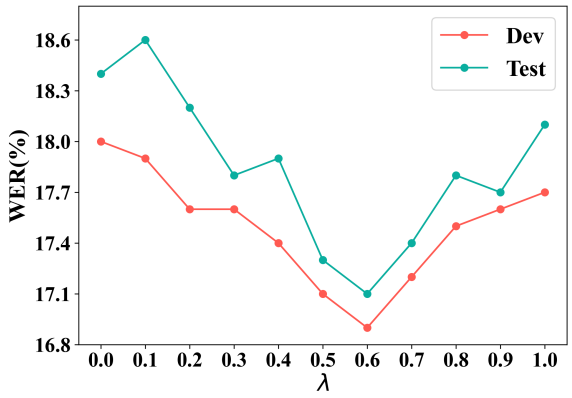

Figure 3: Ablations for different λ in SASGM.

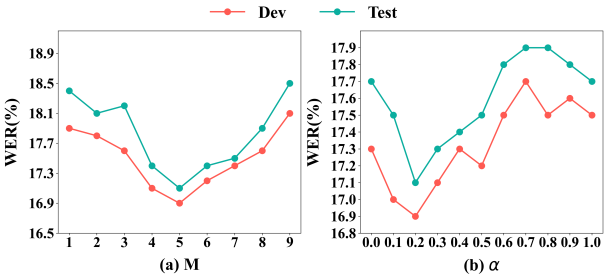

Figure 4: Ablations for different M and α in THM.

fore, we set the final hyperparameter configuration as M = 5 and $\alpha = 0.2$.

Ablations for different window sizes (K) in THM. Table 5 shows that a 4-frame window achieves optimal performance. Smaller windows (2 frames) fail to capture complete temporal context, while larger windows (6-12 frames) introduce noise from unrelated gestures, causing performance degradation.

Ablations for different Stages. Table 6 demonstrates that combining features from all four stages yields optimal performance. This hierarchical approach enables the model to leverage both fine-grained spatial details from early stages and abstract semantic information from later stages, creating robust multi-scale representations.

Window Sizes (K)	Dev(%)	Test(%)
2	17.3	17.7
4	**16.9**	**17.1**
6	17.2	17.4
8	17.7	18.1
10	18.1	18.7
12	18.5	19.4

Table 5: Ablations for different window sizes (K) in THM on PHOENIX14.

Stage 2	Stage 3	Stage 4	Dev(%)	Test(%)
-	-	-	18.3	19.6
✓	-	-	17.5	18.3
-	✓	-	17.7	18.7
-	-	✓	17.9	19.0
✓	✓	-	17.4	18.1
✓	-	✓	17.3	17.9
-	✓	✓	17.2	18.0
✓	✓	✓	**16.9**	**17.1**

Table 6: Ablations for different stages output in HyperSign on PHOENIX14.

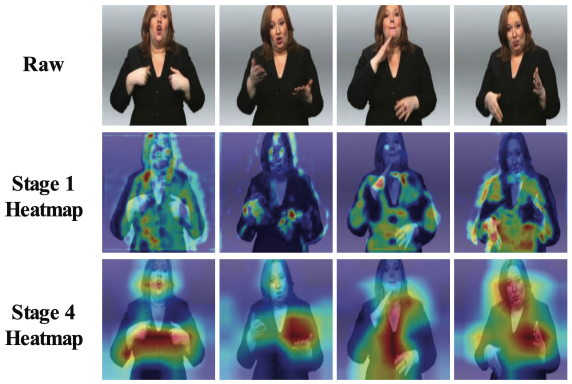

Figure 5: Visualizations of heatmaps by Grad-CAM. HyperSign could generally focus on the human body (light yellow areas) and highlight distinct spatial granularities across stages.

Visualization for HyperSign

Figure 5 visualizes the effect of applying HyperSign to sign language videos using Grad-CAM (Selvaraju et al. 2017). The top, middle, and bottom rows show the raw frames, Stage 1 output feature heatmaps, and Stage 4 output feature heatmaps, respectively, demonstrating HyperSign's effectiveness in capturing changes in moving and facial regions while suppressing background attention.

Conclusion

In this work, we introduce HyperSign, a multi-scale, spatio-temporal hypergraph network for CSLR. Its spatial module constructs dynamic graphs via a saliency-gating mechanism that fuses static and dynamic cues to focus on key regions. Its temporal module uses a clustering-based hypergraph to capture high-order, multi-frame correlations, overcoming the pairwise limitations of traditional models. This hierarchical architecture, which aggregates features at varying granularities, demonstrates SOTA effectiveness and stability in extensive experiments on three public datasets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 62176236, and the Natural Science Foundation of Zhejiang Province under Grant LZ24F030011.

References

Ahn, J.; Jang, Y.; and Chung, J. S. 2024. Slowfast network for continuous sign language recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3920–3924. IEEE.

Camgoz, N. C.; Hadfield, S.; Koller, O.; Ney, H.; and Bowden, R. 2018. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7784–7793.

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

Cui, R.; Liu, H.; and Zhang, C. 2019. A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, 21(7): 1880–1891.

Gan, S.; Yin, Y.; Jiang, Z.; Wen, H.; Xie, L.; and Lu, S. 2024. Signgraph: A sign sequence is worth graphs of nodes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13470–13479.

Gan, S.; Yin, Y.; Jiang, Z.; Xie, L.; and Lu, S. 2021. Skeleton-aware neural sign language translation. In *Proceedings of the 29th ACM International Conference on Multimedia*, 4353–4361.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376.

Guo, L.; Xue, W.; Guo, Q.; Liu, B.; Zhang, K.; Yuan, T.; and Chen, S. 2023. Distilling cross-temporal contexts for continuous sign language recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10771–10780.

Han, K.; Wang, Y.; Guo, J.; Tang, Y.; and Wu, E. 2022. Vision gnn: An image is worth graph of nodes. *Advances in neural information processing systems*, 35: 8291–8303.

Han, Y.; Wang, P.; Kundu, S.; Ding, Y.; and Wang, Z. 2023. Vision hgnn: An image is more than a graph of nodes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19878–19888.

Hao, A.; Min, Y.; and Chen, X. 2021. Self-mutual distillation learning for continuous sign language recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 11303–11312.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hu, H.; Zhao, W.; Zhou, W.; and Li, H. 2023a. Signbert+: Hand-model-aware self-supervised pre-training for sign language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 11221–11239.

Hu, L.; Gao, L.; Liu, Z.; and Feng, W. 2022. Temporal lift pooling for continuous sign language recognition. In *European conference on computer vision*, 511–527. Springer.

Hu, L.; Gao, L.; Liu, Z.; and Feng, W. 2023b. Continuous sign language recognition with correlation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2529–2539.

Hu, L.; Gao, L.; Liu, Z.; and Feng, W. 2023c. Self-emphasizing network for continuous sign language recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 854–862.

Hu, L.; Gao, L.; Liu, Z.; Pun, C.-M.; and Feng, W. 2023d. Adabrowse: Adaptive video browser for efficient continuous sign language recognition. In *Proceedings of the 31st ACM international conference on multimedia*, 709–718.

Jiang, B.; Zhang, Z.; Lin, D.; Tang, J.; and Luo, B. 2019. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11313–11320.

Jiao, P.; Min, Y.; Li, Y.; Wang, X.; Lei, L.; and Chen, X. 2023. Cosign: Exploring co-occurrence signals in skeleton-based continuous sign language recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 20676–20686.

Kan, J.; Hu, K.; Hagenbuchner, M.; Tsoi, A. C.; Bennamoun, M.; and Wang, Z. 2022. Sign language translation with hierarchical spatio-temporal graph neural network. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 3367–3376.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koller, O.; Forster, J.; and Ney, H. 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141: 108–125.

Li, C.; Li, T.; Hu, X.; Luo, D.; and Jin, T. 2025. DVHGNN: Multi-Scale Dilated Vision HGNN for Efficient Vision Recognition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 20158–20168.

Lu, H.; Salah, A. A.; and Poppe, R. 2024. Tcnet: Continuous sign language recognition from trajectories and correlated regions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3891–3899.

Min, Y.; Hao, A.; Chai, X.; and Chen, X. 2021. Visual alignment constraint for continuous sign language recognition. In *proceedings of the IEEE/CVF international conference on computer vision*, 11542–11551.

Munir, M.; Avery, W.; and Marculescu, R. 2023. Mobilevig: Graph-based sparse attention for mobile vision applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2211–2219.

Munir, M.; Avery, W.; Rahman, M. M.; and Marculescu, R. 2024. Greedyvig: Dynamic axial graph construction for efficient vision gnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6118–6127.

Rao, Q.; Sun, K.; Wang, X.; Wang, Q.; and Zhang, B. 2024. Cross-sentence gloss consistency for continuous sign language recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4650–4658.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Spadaro, G.; Grangetto, M.; Fiandrotti, A.; Tartaglione, E.; and Giraldo, J. H. 2025. Wignet: Windowed vision graph neural network. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 859–868. IEEE.

Wei, D.; Yang, X.-H.; Weng, Y.; Lin, X.; Hu, H.; and Liu, S. 2025. Cross-Modal Adaptive Prototype Learning for Continuous Sign Language Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*.

Xu, M.; Liu, S.; Feng, Y.; Yu, Y.; Jin, Z.; and Yang, X. 2025. Hierarchical Spatial-Temporal Enhancement Network For Continuous Sign Language Recognition. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Yang, X.-H.; Hu, H.-X.; and Lin, X. 2025. ACMC: Adaptive cross-modal multi-grained contrastive learning for continuous sign language recognition. *Image and Vision Computing*, 105622.

Yang, Y.; Min, Y.; and Chen, X. 2024. S2Net: Skeleton-Aware SlowFast Network for Efficient Sign Language Recognition. In *Proceedings of the Asian Conference on Computer Vision*, 319–336.

Yu, Y.; Liu, S.; Feng, Y.; Xu, M.; Jin, Z.; and Yang, X. 2025a. Improving Continuous Sign Language Recognition via Cross-Frame Interactions in Expanded Contextual Spaces. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Yu, Y.; Liu, S.; Feng, Y.; Xu, M.; Jin, Z.; and Yang, X. 2025b. OLMD: Orientation-aware Long-term Motion Decoupling for Continuous Sign Language Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 9707–9715.

Zhang, H.; Guo, Z.; Yang, Y.; Liu, X.; and Hu, D. 2023. C2st: Cross-modal contextualized sequence transduction for continuous sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 21053–21062.

Zheng, J.; Wang, Y.; Tan, C.; Li, S.; Wang, G.; Xia, J.; Chen, Y.; and Li, S. Z. 2023. Cvt-slr: Contrastive visual-textual transformation for sign language recognition with variational alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 23141–23150.

Zhou, H.; Zhou, W.; Qi, W.; Pu, J.; and Li, H. 2021. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1316–1325.

Zhou, H.; Zhou, W.; Zhou, Y.; and Li, H. 2020. Spatial-temporal multi-cue network for continuous sign language recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 13009–13016.

Zuo, R.; and Mak, B. 2022. C2slr: Consistency-enhanced continuous sign language recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5131–5140.