Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.078089

ARTICLE

NestLipGNN: A Hierarchical Graph Neural Network Framework with Nested Multi-Granularity Learning for Robust Visual Speech Recognition

Vinh Truong Hoang*, Nghia Dinh, Luu Quang Phuong, Kiet Tran-Trung, Ha Duong Thi Hong, Bay Nguyen Van, Hau Nguyen Trung and Thien Ho Huong

AI Lab, Faculty of Information Technology, Ho Chi Minh City Open University, 35-37 Ho Hao Hon Street, Co Giang Ward, District 1, Ho Chi Minh City, Vietnam

*Corresponding Author: Vinh Truong Hoang. Email: vinh.th@ou.edu.vn

Received: 23 December 2025; Accepted: 25 March 2026; Published: 08 May 2026

ABSTRACT: Visual speech recognition (VSR) aims to infer spoken content from visual observations of articulatory movements. Despite significant progress, it remains a challenging task in computer vision and speech processing. Its difficulty arises from pronounced speaker-to-speaker variability, the presence of homophenes (phonemes that are visually indistinguishable), changes in illumination, and the intrinsically high-dimensional nature of spatiotemporal lip dynamics. In this work, we propose NestLipGNN, a graph-based framework that integrates Graph Neural Networks (GNNs) with a nested multi-granularity learning strategy for visual speech recognition. We construct dynamic lip graphs from facial landmarks to model both spatial relationships between lip regions and their temporal motion during speech articulation. The proposed nested learning architecture supports hierarchical feature extraction across several levels of linguistic abstraction, spanning phoneme-level articulatory units, viseme-level visual speech categories, and word-level semantic representations. We further introduce a Temporal Graph Attention mechanism (T-GAT) that adaptively reweights the importance of distinct lip regions over time. We also introduce a graph-based contrastive learning objective to improve the discrimination of visually similar speech patterns, directly confronting the challenge of homophene resolution. Experiments on the LRW, LRS2, LRS3, and GRID datasets show that NestLipGNN improves recognition accuracy compared with existing methods, obtaining 92.3% word-level accuracy on LRW and delivering a 2.1% absolute performance gain over prior methods. Comprehensive ablation analyses confirm the contribution of each architectural component.

KEYWORDS: Visual speech recognition; graph neural networks; nested optimization; hierarchical representation learning; spatiotemporal modeling; contrastive learning; lip reading

1 Introduction

Visual speech recognition (VSR), commonly referred to as lip reading, involves computationally inferring spoken language content solely from visual observations of orofacial articulatory motion, without relying on acoustic information [1,2]. This line of research has attracted considerable interest within the scientific community due to its broad range of practical uses, such as silent speech interfaces for people who have undergone laryngectomy [3], assistive tools for individuals with hearing impairments [4], multimodal speech recognition systems designed for acoustically challenging conditions [5], and security and forensic surveillance applications [6].

The core difficulty in visual speech recognition arises from the inherent ambiguity of visual speech cues: many phonetically different utterances produce lip movements that are visually indistinguishable, a phenomenon known as homopheny in phonetic studies [4,7]. For example, the bilabial consonants /p/, /b/, and /m/ share nearly identical visible articulations, making it extremely challenging to differentiate them using only visual input. This intrinsic visual ambiguity, further exacerbated by inter-speaker morphological differences, variations in head pose, and changes in lighting conditions, makes VSR a significantly more demanding computational task than conventional acoustic speech recognition [6].

Recent progress in deep learning techniques has led to notable gains in the performance of lip reading systems. Convolutional Neural Networks (CNNs) are widely employed to extract spatial features from lip-region images [8,9], whereas Recurrent Neural Networks (RNNs) and their gated extensions are used to model temporal dependencies across sequences of frames [10]. More recently, three-dimensional CNNs [9], and Transformer-based models [11] achieved competitive performance by jointly capturing spatiotemporal patterns. Recent advances including global-local integrated frameworks [12], language model-enhanced approaches and viseme-guided generation methods [13] continue to push performance boundaries. However, these approaches largely treat lip images as conventional Euclidean grids and therefore overlook the intrinsic anatomical structure of different lip regions and the topological constraints that shape articulatory motion.

Graph Neural Networks (GNNs) offer a mathematical framework for learning from non-Euclidean structured data [14–16]. The human lip system can be naturally modeled as a graph, where nodes represent anatomical landmark locations and edges encode spatial relations that characterize lip shape and deformation patterns. This modeling strategy allows explicit incorporation of lip structural attributes during speech production, yielding robustness to certain geometric transformations while retaining critical articulatory information [17]. Graph-based representations of facial landmarks and temporal attention mechanisms have been explored in related domains such as skeleton-based action recognition and facial expression analysis. Recent work demonstrates the efficacy of redundancy-aware learning with symmetric view modeling [18] and global-local integrated frameworks [12] for capturing complex lip dynamics. Our contribution lies not in the individual components themselves, but in their principled integration and adaptation for the specific challenges of visual speech recognition, particularly homophene disambiguation.

The nested learning paradigm, rooted in hierarchical optimization and bilevel programming theory [19], also provides a systematic approach for learning representations across multiple levels of abstraction. Speech is intrinsically organized in a nested hierarchy: phonemes (basic acoustic components) combine to form visemes (visual speech units), which in turn compose words, phrases, and sentences [20]. Recent advances in viseme-guided generation and language model integration [13] demonstrate the importance of multi-level linguistic representation in visual speech recognition. This linguistic structure implies that effective VSR systems should learn representations at several granularities, where lower-level features support and shape higher-level abstractions via structured knowledge transfer.

In this paper, we present NestLipGNN (Fig. 1), a framework that combines graph neural networks with nested multi-granularity learning for visual speech recognition. The main contributions of this work are as follows. First, we introduce a principled approach for building dynamic lip graphs that encode both spatial lip geometry and temporal deformation dynamics using learnable adjacency matrices, which adapt to speaker-specific articulatory patterns. Second, we propose a hierarchical learning scheme that jointly optimizes representations at the phoneme, viseme, and word levels through coupled hierarchical loss functions, facilitating bidirectional information flow across different linguistic abstraction layers. Third, we report state-of-the-art performance on four standard benchmark datasets and complement these results with detailed ablation experiments analyzing the framework's components.

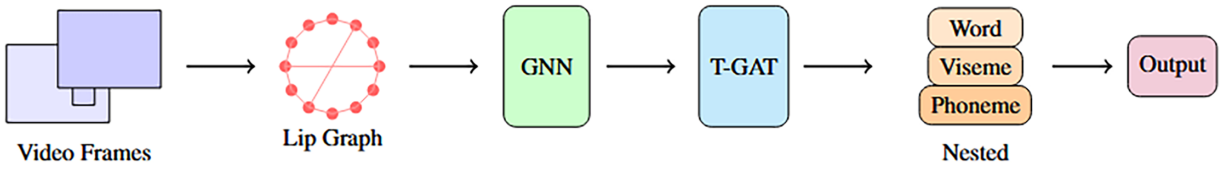

Figure 1: Architectural overview of the proposed NestLipGNN framework.

The rest of the paper is structured as follows. Section 2 reviews related literature. Section 3 details the proposed NestLipGNN framework. Section 4 outlines the experimental setup and reports empirical results. Section 5 analyzes the implications of our findings, and Section 6 closes with concluding remarks and potential avenues for future work.

2 Related Work

2.1 Visual Speech Recognition: Historical Evolution and Modern Methodologies

The development of visual speech recognition (VSR) systems mirrors broader progress in pattern recognition and machine learning. Early computational methods relied on manually designed feature representations, such as discrete cosine transform (DCT) coefficients [21], active appearance models (AAM) [22], and Hidden Markov Models (HMMs) for modeling temporal dynamics [23]. Although these approaches laid essential theoretical and practical foundations, their effectiveness was inherently limited by the expressive power of hand-crafted features and their restricted capacity to represent the complex nonlinear dependencies characteristic of visual speech signals.

The emergence of deep learning led to a major transformation in VSR research. Stafylakis and Tzimiropoulos investigated a variety of architectural designs, including three-dimensional CNNs for joint spatiotemporal feature learning [9], and conformer architectures that integrate convolutional modules with self-attention [24]. More recently, transformer-based approaches have set new performance benchmarks. Afouras et al. [5] adopted transformer encoders for temporal modeling, reporting significant performance improvements. Ma et al. [25] further showed that large-scale multilingual pretraining enables effective cross-lingual transfer. Self-supervised pretraining techniques [11,26], which exploit vast collections of unlabeled video data, have yielded further accuracy gains. The AV-HuBERT framework [26] demonstrated that masked prediction objectives can learn powerful audio-visual representations, while recent work on LP-Conformer [27] has shown the benefits of combining local and global context modeling. Nevertheless, most current methods still rely on grid-based representations, which restrict their ability to explicitly characterize the structural constraints underlying lip articulation.

2.2 Graph Neural Networks: Theoretical Foundations and Applications

Graph Neural Networks (GNNs) have become highly effective computational frameworks for learning from graph-structured data. Spatial methods [14,28] implement convolutions by aggregating information from local neighborhoods via message-passing schemes. Graph Attention Networks (GATs) [15] further extend this paradigm by introducing attention mechanisms to adaptively weight neighboring nodes, thereby enhancing both expressive power and interpretability.

Within human body analysis, GNNs have been widely adopted for skeleton-based action recognition [17], leveraging the inherent graph structure formed by anatomical joint connections. In facial analysis, graph-based techniques have been applied for expression recognition, face alignment, and 3D face reconstruction. Recent advances have also explored dynamic graph construction for temporal sequences

and multi-scale graph representations. In contrast, the targeted and systematic use of GNNs for lip reading has received comparatively little attention, which motivates our graph-theoretic formulation of visual speech recognition.

2.3 Hierarchical and Nested Optimization

Hierarchical learning has long-standing theoretical foundations in machine learning, ranging from hierarchical clustering techniques to deep neural networks with multiple layers of abstraction. Nested optimization, often referred to as bilevel programming, has seen a resurgence of interest in areas such as meta-learning [29], hyperparameter optimization [19,30], and neural architecture search [31].

The key idea is that many learning problems naturally admit a hierarchical organization, where higher-level representations are defined in terms of lower-level features via nested optimization loops, each with its own convergence behavior. This framework has been effectively employed in few-shot learning, domain adaptation, and multi-task learning. In speech recognition, hierarchical formulations have captured the phoneme-word-sentence structure [32,33], although commonly using cascaded pipelines rather than fully nested optimization schemes.

2.4 Positioning of This Work

We distinguish this work from related approaches as follows. Table 1 summarizes the key differences between NestLipGNN and related graph-based and hierarchical methods. Graph-based landmark modeling has been explored in skeleton action recognition and facial expression analysis [17], but these methods do not address the specific challenges of visual speech recognition, particularly the fine-grained temporal dynamics of articulation and homophene disambiguation. Temporal attention mechanisms [34] have been explored in video understanding, but our T-GAT mechanism specifically incorporates graph structure similarity into the attention computation. Hierarchical supervision has been used in acoustic speech recognition [33], but the specific phoneme-viseme-word hierarchy aligned with visual speech units is novel to this work. The key contribution of this work is the integration of graph modeling, temporal attention, and hierarchical supervision into a single VSR architecture, along with the graph-based contrastive learning objective for homophene resolution.

Table 1: Comparison of NestLipGNN with related approaches.

Method	Graph Lip	Temporal Attention	Hierarchical Supervision	Homophene Contrast	VSR Focus
ST-GCN [17]	✓				
GAT [15]	✓				
Transformer [34]		✓			
CTC/Attention [33]			✓		
Ma et al. [25]		✓			✓
NestLipGNN (Ours)	✓	✓	✓	✓	✓

2.5 Contrastive Representation Learning

Contrastive learning has become a prominent self-supervised framework [35], in which informative representations are acquired by distinguishing positive pairs from negative examples within a learned embedding space. In speech processing, contrastive approaches have been utilized for acoustic representation

learning [36], speaker verification [37], and audio-visual correspondence modeling [38]. Building on this paradigm, we introduce graph-structured contrastive objectives to tackle the homophene disambiguation challenge in visual speech recognition.

3 Proposed Methodology

Fig. 2 presents the complete architectural specification of NestLipGNN. Our methodology comprises five principal components: (1) Dynamic Lip Graph Construction, (2) Spatial Graph Encoder, (3) Temporal Graph Attention, (4) Nested Multi-Granularity Decoder, and (5) Graph Contrastive Learning.

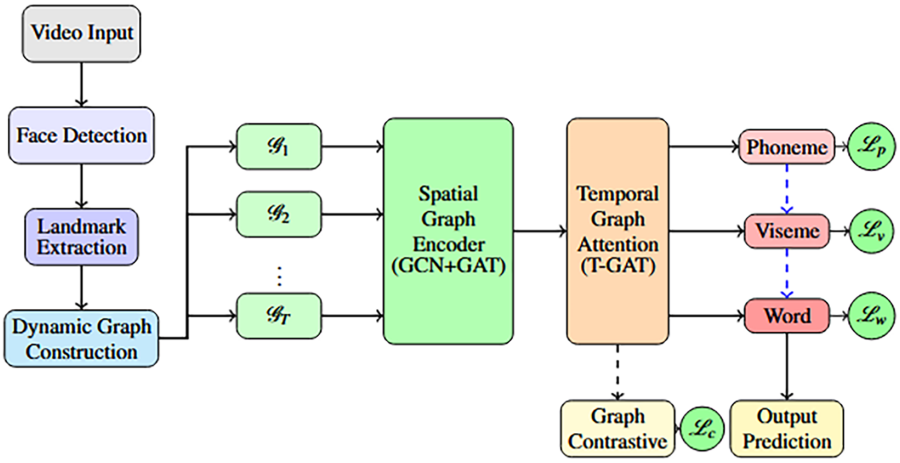

Figure 2: Comprehensive architectural diagram of NestLipGNN. The framework processes video frames to construct dynamic lip graphs, encodes spatial features through GCN and GAT layers, captures temporal dynamics via T-GAT, and generates predictions through a nested learning hierarchy with phoneme ($\mathcal{L}_p$), viseme ($\mathcal{L}_v$), and word-level ($\mathcal{L}_w$) supervision. A graph contrastive learning module ($\mathcal{L}_c$) enhances discriminative features for homophene disambiguation.

3.1 Problem Formulation

Let $\mathbf{V} = \{I_1, I_2, \ldots, I_T\} \in \mathbb{R}^{T\times H\times W\times 3}$ denote a video sequence with T frames, where each frame I_t has spatial resolution $H \times W$ and three color channels. The goal is to learn a mapping function $f_\Theta : \mathbb{R}^{T\times H\times W\times 3} \to \mathcal{Y}$, parameterized by Θ, that converts the visual input into a linguistic output $\mathbf{y} \in \mathcal{Y}$. Here, $\mathcal{Y}$ may correspond to either a set of discrete word labels for classification problems or a sequence of characters for recognition tasks.

Our key idea is to factorize this mapping into a sequence of graph-based transformations:

$$f_\Theta(\mathbf{V}) = f_d \circ f_t \circ f_s \circ f_g(\mathbf{V}) \tag{1}$$

where $f_g : \mathbb{R}^{T\times H\times W\times 3} \to \{\mathcal{G}_t\}_{t=1}^{T}$ is the graph construction function, f_s is the spatial graph encoder, f_t denotes the temporal graph attention module, and f_d is the nested hierarchical decoder. Each function $f_\cdot$ transforms the data into a more abstract representation: f_g converts raw frames into a sequence of dynamic graphs, f_s extracts structural features from each graph, f_t models temporal evolution across graph frames, and f_d integrates hierarchical information to produce the final linguistic output.

3.2 Dynamic Lip Graph Construction

3.2.1 Anatomical Landmark Extraction

We use a pretrained facial landmark detector based on the Face Alignment Network (FAN) [39] to obtain 68 anatomical facial landmarks for each video frame. From this full set, we retain $N = 20$ landmarks (Fig. 3) associated with the perioral area: 12 points outlining the external lip contour (vermilion border) and 8 points specifying the internal lip contour (including the labial commissures and the oral aperture boundary). Let $\mathbf{P}_t = \{p_1^t, p_2^t, \ldots, p_N^t\} \subset \mathbb{R}^2$ denote the lip landmark set at time index t, where each $p_i^t = (x_i^t, y_i^t)^\top$ corresponds to the two-dimensional coordinates of landmark i. These coordinates are extracted pixel-wise from the video frame I_t and represent salient anatomical points critical for modeling articulatory movement.

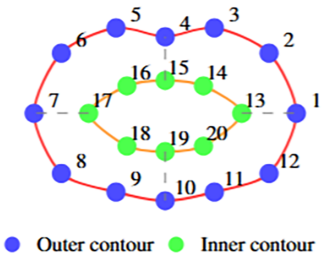

Figure 3: Lip graph topology comprising 20 anatomical landmarks. Blue nodes correspond to outer lip contour points (vermilion border, 12 vertices), and green nodes correspond to inner lip contour points (8 vertices). Solid edges indicate anatomically defined connections, while dashed edges illustrate learned cross-contour links.

To mitigate the impact of head pose changes and to allow consistent comparisons across frames, we normalize the landmarks via Generalized Procrustes Analysis:

$$\hat{\mathbf{P}}_t = \underset{\mathbf{R}\in SO(2), s>0, \mathbf{t}\in\mathbb{R}^2}{\arg\min} \left\| \mathbf{P}_t - (s\mathbf{R}\mathbf{P}_{\text{ref}} + \mathbf{t}\mathbf{1}^\top) \right\|_F^2 \tag{2}$$

where $\mathbf{R} \in SO(2)$ is a 2D rotation matrix that aligns the lip structure with a canonical orientation, $s \in \mathbb{R}^+$ is a global scaling factor ensuring uniform size across subjects, $\mathbf{t} \in \mathbb{R}^2$ is a translation vector that centers the normalized lip shape, $\mathbf{P}_{\text{ref}} \in \mathbb{R}^{N\times 2}$ is the reference shape derived from training data statistics, $\mathbf{1} \in \mathbb{R}^N$ is a column vector of ones that broadcasts translation to all landmarks, and $\|\cdot\|_F$ denotes the Frobenius norm, which quantifies the total squared Euclidean deviation between the observed and reference shapes. This transformation yields pose-invariant landmark configurations $\hat{\mathbf{P}}_t$ by removing global rigid transformations.

3.2.2 Graph-Theoretic Formulation

At each time step t, we define an attributed graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathbf{X}_t, \mathbf{A}_t)$, specified as follows. The vertex set $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ consists of N fixed nodes, each corresponding to one anatomical landmark across all temporal frames. The edge set $\mathcal{E}_t \subseteq \mathcal{V} \times \mathcal{V}$ is dynamic and reflects time-varying spatial relationships among landmarks as the lips articulate. The node feature matrix $\mathbf{X}_t \in \mathbb{R}^{N\times d_0}$ contains d_0-dimensional feature vectors for each node, encoding both geometric and appearance cues. The adjacency matrix $\mathbf{A}_t \in \mathbb{R}^{N\times N}$ is a weighted, asymmetric matrix where each entry $[\mathbf{A}_t]_{ij} \in [0, 1]$ quantifies the strength and direction of the connection from node v_j to node v_i, capturing both anatomical priors and learned dependencies.

3.2.3 Multi-Modal Node Feature Construction

Each node feature vector aggregates positional, appearance-based, textural, and kinematic cues:

$$\mathbf{x}_i^t = \left[\hat{p}_i^t; \phi_{\text{app}}(I_t, p_i^t); \phi_{\text{tex}}(I_t, p_i^t); \Delta p_i^t; \Delta^2 p_i^t\right] \in \mathbb{R}^{d_0} \tag{3}$$

where $\hat{p}_i^t \in \mathbb{R}^2$ is the Procrustes-normalized landmark location, $\phi_{\text{app}}(I_t, p_i^t) \in \mathbb{R}^{d_\phi}$ is a local appearance descriptor, $\phi_{\text{tex}}(I_t, p_i^t) \in \mathbb{R}^{d_\psi}$ is a local texture descriptor, $\Delta p_i^t \in \mathbb{R}^2$ represents the first-order kinematics, and $\Delta^2 p_i^t \in \mathbb{R}^2$ denotes the second-order kinematics. The concatenated feature vector $\mathbf{x}_i^t$ thus captures motion dynamics, local appearance, and geometric structure, with total dimension $d_0 = 2 + d_\phi + d_\psi + 2 + 2$.

The normalized coordinates $\hat{p}_i^t \in \mathbb{R}^2$ are the Procrustes-aligned landmark coordinates, which eliminate global scale, rotation, and translation effects, ensuring that the graph structure is invariant to camera viewpoint while preserving local relative geometry. The local appearance descriptor $\phi_{\text{app}}(I_t, p_i^t) \in \mathbb{R}^{d_\phi}$ is a convolutional feature vector extracted by a compact CNN encoder CNN_ϕ from a $k \times k$ window centered at p_i^t:

$$\phi_{\text{app}}(I_t, p_i^t) = \text{CNN}_\phi\left(\text{Crop}(I_t, p_i^t, k)\right) \tag{4}$$

where $\text{Crop}(I_t, p_i^t, k)$ denotes the pixel region from frame I_t bounded by a square patch of size $k \times k$ centered on landmark p_i^t, and CNN_ϕ is a three-layer convolutional network with ReLU activations and batch normalization to extract hierarchical visual features. The texture descriptor $\phi_{\text{tex}}(I_t, p_i^t) \in \mathbb{R}^{d_\psi}$ is computed using Local Binary Pattern (LBP) features, which quantify the local intensity variation around each landmark using a circular 3×3 neighborhood. The first-order kinematics $\Delta p_i^t = p_i^t - p_i^{t-1} \in \mathbb{R}^2$ encodes the instantaneous velocity of landmark i between consecutive frames t and $t-1$, modeling the rate of spatial change during articulation. The second-order kinematics $\Delta^2 p_i^t = \Delta p_i^t - \Delta p_i^{t-1} \in \mathbb{R}^2$ captures the acceleration, reflecting non-uniform motion dynamics such as rapid lip closure or burst articulation.

3.2.4 Adaptive Adjacency Matrix Formulation

We introduce a trainable adjacency matrix that jointly integrates anatomical priors with data-driven relations:

$$\mathbf{A}_t = \alpha_1 \mathbf{A}_{\text{struct}} + \alpha_2 \mathbf{A}_{\text{learn}}^t + \alpha_3 \mathbf{A}_{\text{dist}}^t \tag{5}$$

where $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}^+$ are learnable weighting coefficients constrained by $\sum_{i=1}^3 \alpha_i = 1$, ensuring that the overall adjacency remains a convex combination of three distinct structural sources.

The motivation for combining these three adjacency components stems from complementary information sources. The structural adjacency $\mathbf{A}_{\text{struct}} \in \{0,1\}^{N \times N}$ encodes anatomically defined lip connectivity based on biological knowledge, ensuring the graph embeds prior anatomical constraints that reflect the physical structure of the lip. For each pair of landmarks (v_i, v_j), $[\mathbf{A}_{\text{struct}}]_{ij} = 1$ if and only if $(v_i, v_j) \in \mathscr{E}_{\text{anat}}$, where $\mathscr{E}_{\text{anat}}$ is the set of known anatomical adjacencies among lip landmarks (e.g., consecutive points on the outer or inner contour). The learned adjacency $\mathbf{A}_{\text{learn}}^t$ models task-dependent interactions that may not follow anatomical connectivity but are important for speech recognition, via a bilinear form:

$$\mathbf{A}_{\text{learn}}^t = \sigma\left(\mathbf{X}_t \mathbf{W}_A \mathbf{X}_t^\top\right) \tag{6}$$

where $\mathbf{X}_t \in \mathbb{R}^{N \times d_0}$ is the node feature matrix at time t, $\mathbf{W}_A \in \mathbb{R}^{d_0 \times d_0}$ is a learnable weight matrix that projects features into a space where pairwise similarity can be computed, and $\sigma(\cdot)$ denotes the sigmoid function. This enables the model to discover non-anatomical interactions based on learned correlations in

articulatory behavior. The distance-based adjacency $\mathbf{A}^t_{\text{dist}}$ captures spatial proximity using a Gaussian kernel, reflecting the intuition that nearby landmarks should have stronger interactions during smooth articulatory deformations:

$$[\mathbf{A}^t_{\text{dist}}]_{ij} = \exp\left(-\frac{\|\hat{p}^t_i - \hat{p}^t_j\|_2^2}{2\tau^2}\right) \tag{7}$$

where $\hat{p}^t_i \in \mathbb{R}^2$ and $\hat{p}^t_j \in \mathbb{R}^2$ are the normalized 2D coordinates of landmarks i and j, $\|\cdot\|_2$ denotes the Euclidean distance, and $\tau \in \mathbb{R}^+$ is a learnable bandwidth parameter that controls the rate at which edge weights decay with physical distance.

3.3 Spatial Graph Encoder

The spatial encoder processes the lip graph of each frame, extracting structural cues through a hierarchical sequence of graph convolution and attention layers.

3.3.1 Spectral Graph Convolutions

We employ spectral graph convolutions derived from the first-order Chebyshev polynomial approximation [14]:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}\right) \tag{8}$$

where $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N\times d_\ell}$ is the node feature matrix at layer ℓ, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacency matrix augmented with self-loops to include node features in aggregation, $\tilde{\mathbf{D}} \in \mathbb{R}^{N\times N}$ is the diagonal degree matrix with $\tilde{D}_{ii} = \sum_{j=1}^{N}\tilde{A}_{ij}$ ensuring row-normalization of the graph Laplacian, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell+1}}$ is a learnable weight matrix transforming the input features to output dimension $d_{\ell+1}$, and $\sigma(\cdot)$ is the GELU activation function.

To enlarge the receptive field and capture information from more distant neighbors, we use a multi-scale aggregation strategy [16]:

$$\mathbf{H}^{(\ell+1)} = \text{Concat}\left[\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}_0; \tilde{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}_1; \tilde{\mathbf{A}}^2\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}_2\right] \tag{9}$$

where $[\cdot;\cdot]$ denotes concatenation along the feature dimension, $\mathbf{W}^{(\ell)}_0$, $\mathbf{W}^{(\ell)}_1$, and $\mathbf{W}^{(\ell)}_2$ are independent linear projection matrices, and $\tilde{\mathbf{A}}^2$ represents two-hop neighbor influence via matrix multiplication.

3.3.2 Graph Attention Mechanism

Following the Graph Attention Network paradigm [15], we incorporate an attention mechanism that assigns adaptive importance to neighboring nodes. The unnormalized attention scores are given by:

$$e^{(\ell)}_{ij} = \text{LeakyReLU}\left(\mathbf{a}^{(\ell)\top}\left[\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell)}_i \| \mathbf{W}^{(\ell)}\mathbf{h}^{(\ell)}_j\right]\right) \tag{10}$$

where $\mathbf{h}^{(\ell)}_i$ and $\mathbf{h}^{(\ell)}_j$ are the transformed node features at layer ℓ, $\|$ denotes vector concatenation, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell+1}}$ is a linear transformation, and $\mathbf{a}^{(\ell)} \in \mathbb{R}^{2d_{\ell+1}}$ is a learnable attention weight vector.

The normalized attention coefficients are obtained with a softmax over the neighborhood of node i:

$$\alpha^{(\ell)}_{ij} = \frac{\exp(e^{(\ell)}_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e^{(\ell)}_{ik})} \tag{11}$$

where $\mathcal{N}_i = \{j \in \mathcal{V} \mid (v_i, v_j) \in \mathcal{E}_t\}$ is the set of neighbors of node v_i in graph $\mathcal{G}_t$.

The node features are updated by aggregating neighbor embeddings weighted by these attention coefficients:

$$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(\ell)} \mathbf{W}^{(\ell)} \mathbf{h}_j^{(\ell)}\right) \tag{12}$$

To capture heterogeneous interaction patterns, we adopt multi-head attention with $K = 8$ parallel heads:

$$\mathbf{h}_i^{(\ell+1)} = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(\ell,k)} \mathbf{W}^{(\ell,k)} \mathbf{h}_j^{(\ell)}\right) \tag{13}$$

where the vertical bar $\|$ denotes concatenation across $K = 8$ attention heads.

3.3.3 Graph-Level Aggregation

After L layers of message passing, we compute a single embedding for the entire graph by combining mean and max pooling over the final node representations:

$$\mathbf{h}_{\mathcal{G}_t} = \text{MLP}\left(\frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_i^{(L)} \Big\| \max_{i \in \mathcal{V}} \mathbf{h}_i^{(L)}\right) \tag{14}$$

where $\mathbf{h}_i^{(L)} \in \mathbb{R}^{K \cdot d_{L+1}}$ is the final node embedding at the L-th layer.

3.4 Temporal Graph Attention (T-GAT)

To capture how lip graphs evolve over time, we propose a Temporal Graph Attention mechanism (Fig. 4) that extends self-attention to graph-structured temporal sequences. This mechanism computes attention over all pairs of temporal frames, resulting in $O(T^2 \cdot N \cdot d)$ complexity where T is the sequence length, N is the number of nodes, and d is the feature dimension. This quadratic scaling in T is a limitation for very long sequences, though for typical VSR scenarios with $T \leq 100$ frames, the computational cost remains tractable.

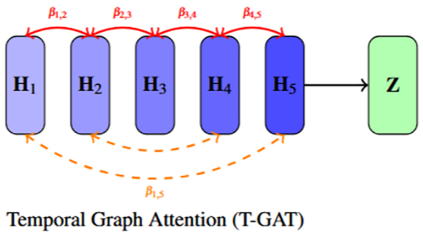

Figure 4: Temporal Graph Attention captures both short-range temporal consistency (solid red arrows) and long-range interactions (dashed orange arrows) over frame-level graph features.

Let $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_T] \in \mathbb{R}^{T \times N \times d}$ denote the sequence of spatial graph representations from the spatial encoder, where each $\mathbf{H}_t \in \mathbb{R}^{N \times d}$ contains the final graph embeddings at time t. We first incorporate learnable sinusoidal positional encodings:

$$\tilde{\mathbf{H}}_t = \mathbf{H}_t + \mathbf{PE}_t \tag{15}$$

where the positional encoding $\mathbf{PE}_t \in \mathbb{R}^{N \times d}$ is defined element-wise for position t and dimension i as:

$$\mathrm{PE}(t, 2i) = \sin\left(\frac{t}{10000^{2i/d}}\right) \tag{16}$$

$$\mathrm{PE}(t, 2i+1) = \cos\left(\frac{t}{10000^{2i/d}}\right) \tag{17}$$

T-GAT then obtains query, key, and value tensors through learned linear transformations:

$$\mathbf{Q}_t = \tilde{\mathbf{H}}_t \mathbf{W}_Q, \quad \mathbf{K}_t = \tilde{\mathbf{H}}_t \mathbf{W}_K, \quad \mathbf{V}_t = \tilde{\mathbf{H}}_t \mathbf{W}_V \tag{18}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$ are learnable projection matrices, and $d_k = d/K$ is the key/query dimension per attention head.

Temporal attention scores are determined via a similarity function that is aware of graph structure:

$$\mathrm{sim}(\mathbf{Q}_t, \mathbf{K}_s) = \frac{\mathrm{tr}(\mathbf{Q}_t^{\top} \mathbf{K}_s)}{\sqrt{d_k}} + \lambda_g \cdot \mathrm{GS}(\mathscr{G}_t, \mathscr{G}_s) \tag{19}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator, $\mathrm{GS}(\mathscr{G}_t, \mathscr{G}_s) = \exp(-\|\mathbf{A}_t - \mathbf{A}_s\|_F^2 / \sigma_g^2)$ is the graph structure similarity between frames t and s, $\sigma_g \in \mathbb{R}^+$ is a learnable scale parameter, and $\lambda_g \in \mathbb{R}^+$ is a weight balancing structural and appearance similarity.

These similarity scores are normalized into attention weights using a softmax:

$$\beta_{t,s} = \frac{\exp(\mathrm{sim}(\mathbf{Q}_t, \mathbf{K}_s))}{\sum_{s'=1}^{T} \exp(\mathrm{sim}(\mathbf{Q}_t, \mathbf{K}_{s'}))} \tag{20}$$

The temporally enriched representation at time t is computed as a weighted combination of the value tensors:

$$\mathbf{Z}_t = \sum_{s=1}^{T} \beta_{t,s} \mathbf{V}_s \tag{21}$$

We stack M T-GAT layers and apply residual connections together with layer normalization to facilitate stable training:

$$\mathbf{Z}^{(m+1)} = \mathrm{LayerNorm}\left(\mathbf{Z}^{(m)} + \text{T-GAT}(\mathbf{Z}^{(m)})\right) \tag{22}$$

3.5 Nested Multi-Granularity Learning Framework

The proposed nested framework enables hierarchical representation learning by simultaneously optimizing multiple layers of linguistic abstraction (Fig. 5).

3.5.1 Hierarchical Representation Levels

We define three representation layers aligned with the linguistic hierarchy.

The phoneme level ($\mathbf{Z}^p \in \mathbb{R}^{T \times d_p}$) encodes fine-grained articulatory units that capture elementary phonetic details, modeling the atomic building blocks of speech such as /p/, /b/, /t/, /d/ with high temporal resolution. It is derived directly from the T-GAT output $\mathbf{Z}$ via a transformer decoder block with cross-attention over time and learnable phoneme queries $\mathbf{E}^p \in \mathbb{R}^{T \times d_p}$:

$$\mathbf{Z}^p = \mathrm{TransformerDecoder}_p(\mathbf{Z}, \mathbf{E}^p; \theta_p) \tag{23}$$

where θ_p are learnable parameters of the decoder, $\mathbf{Z}$ is the context from T-GAT, and $\mathbf{E}^p$ acts as a learnable set of reference embeddings to decode phoneme sequences. The query embeddings $\mathbf{E}^p$, $\mathbf{E}^v$, and $\mathbf{E}^w$ are initialized from a normal distribution $\mathcal{N}(0, 0.02)$ and updated through backpropagation during training.

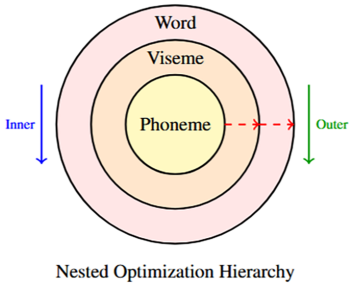

Figure 5: Schematic of the nested learning hierarchy. The inner loop refines phoneme-level encodings, the middle loop estimates viseme-level representations conditioned on phonemes, and the outer loop optimizes word-level embeddings conditioned on both subordinate layers. Dashed arrows indicate gradient propagation across hierarchy levels.

The viseme level ($\mathbf{Z}^v \in \mathbb{R}^{T' \times d_v}$) clusters visually similar phonemes into perceptually indistinguishable visual speech units, such as /p/ and /b/ merging into a single viseme class. It is derived from $\mathbf{Z}^p$ using a second transformer decoder:

$$\mathbf{Z}^v = \text{TransformerDecoder}_v(\mathbf{Z}^p, \mathbf{E}^v; \theta_v) \tag{24}$$

where $\mathbf{E}^v \in \mathbb{R}^{T' \times d_v}$ is the learnable viseme query embedding, and T' is a reduced temporal resolution via average pooling.

The word level ($\mathbf{Z}^w \in \mathbb{R}^{d_w}$) is a high-level semantic embedding that compresses temporal information into a fixed-dimensional vector for word classification:

$$\mathbf{Z}^w = \text{TransformerDecoder}_w(\mathbf{Z}^v, \mathbf{E}^w; \theta_w) \tag{25}$$

where $\mathbf{E}^w \in \mathbb{R}^{1 \times d_w}$ is a singleton query embedding that aggregates all viseme-level information into a single word embedding vector via global attention and MLP projection.

3.5.2 Bilevel Optimization Formulation

Following bilevel optimization principles [19], we formulate the nested learning process as a multi-stage hierarchical optimization problem. The inner loop (phoneme optimization) updates the phoneme decoder parameters θ_p by minimizing the phoneme-level cross-entropy loss:

$$\theta_p^* = \arg\min_{\theta_p} \mathcal{L}_p(f_p(\mathbf{Z}; \theta_p), \mathbf{y}^p) \tag{26}$$

where $f_p(\cdot)$ is the phoneme decoder model, $\mathbf{y}^p$ is the ground-truth phoneme sequence, and $\mathcal{L}_p$ is the classification loss.

The middle loop (viseme optimization) updates the viseme decoder θ_v using the optimized phoneme features $\mathbf{Z}^p(\theta_p^*)$ as input:

$$\theta_v^*(\theta_p^*) = \arg\min_{\theta_v} \mathcal{L}_v(f_v(\mathbf{Z}^p(\theta_p^*); \theta_v), \mathbf{y}^v) \tag{27}$$

where $\mathbf{y}^v$ is the corresponding viseme label sequence.

The outer loop (word optimization) updates the word decoder θ_w conditioned on the optimized viseme representations:

$$\theta_w^*(\theta_p^*, \theta_v^*) = \arg\min_{\theta_w} \mathcal{L}_w(f_w(\mathbf{Z}^v(\theta_p^*, \theta_v^*); \theta_w), \mathbf{y}^w) \tag{28}$$

where $\mathbf{y}^w$ is the word label.

3.5.3 Practical Implementation via Joint Optimization

Solving the hierarchical optimization problem exactly is computationally expensive, so we instead employ a tractable approximation based on a jointly optimized weighted loss:

$$\mathcal{L}_{\text{nested}} = \lambda_p \mathcal{L}_p + \lambda_v \mathcal{L}_v + \lambda_w \mathcal{L}_w + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}} \tag{29}$$

where $\lambda_p, \lambda_v, \lambda_w, \lambda_{\text{cons}} \in \mathbb{R}^+$ are scalar weights balancing contributions from each level.

We clarify the relationship between the theoretical bilevel formulation and our practical implementation. The bilevel optimization formulation (Eqs. (27)–(29)) provides a principled framework for understanding how hierarchical supervision should flow from lower to higher linguistic levels. However, true bilevel optimization requires expensive nested gradient computations and multiple inner-loop iterations per outer update, which is computationally prohibitive for deep networks. Our practical implementation approximates this structure through the weighted joint loss (Eq. (30)), which trains all levels simultaneously but preserves the hierarchical dependency structure through the consistency loss $\mathcal{L}_{\text{cons}}$ and the architectural design where higher levels receive input from lower levels. Empirically, we found that this approximation retains the key benefits of hierarchical supervision (multi-scale learning signals, regularization from auxiliary tasks) while being tractable for training. We conducted experiments comparing alternating updates (K_p = 5, K_v = 3 inner iterations) versus joint optimization and show that alternating updates yield only modest improvements (+0.3% accuracy) at significantly higher computational cost (2.4× training time), justifying our use of the joint approximation.

Each level-specific objective is a cross-entropy classification loss:

$$\mathcal{L}_x = -\sum_{c=1}^{C_x} y_c^x \log(\hat{y}_c^x), \quad x \in \{p, v, w\} \tag{30}$$

where $y_c^x \in \{0,1\}$ is the binary ground-truth indicator for class c at level x, $\hat{y}_c^x \in [0,1]$ is the predicted probability for class c, and C_x is the number of classes at hierarchy level x.

We introduce a hierarchical consistency loss to promote alignment between successive levels:

$$\mathcal{L}_{\text{cons}} = \left\| \mathbf{Z}^v - g_{p\to v}(\mathbf{Z}^p) \right\|_2^2 + \left\| \mathbf{Z}^w - g_{v\to w}(\mathbf{Z}^v) \right\|_2^2 \tag{31}$$

where $g_{p\to v}(\cdot)$ and $g_{v\to w}(\cdot)$ are learnable projection functions implemented as multilayer perceptrons (MLPs).

3.6 Graph Contrastive Learning for Homophene Disambiguation

For each mini-batch, we construct positive and negative pairs based on their linguistic identity. Positive pairs are obtained from distinct video instances of the same word, encouraging invariance to speaker, lighting, and pose variations. Hard negatives are chosen from lexically distinct words that are visually similar in articulation, known as homophenes.

We employ the InfoNCE loss, augmented with a graph-based regularization component:

$$\mathscr{L}_{\text{contrast}} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j^+)/\tau_c)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau_c)} \tag{32}$$

where $\mathbf{z}_i \in \mathbb{R}^{d_w}$ is the encoded word embedding of the i-th sample in the batch, $\mathbf{z}_j^+$ is its positive counterpart (same word), $\tau_c > 0$ is a learnable temperature parameter, $\text{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v}/(\|\mathbf{u}\|_2 \|\mathbf{v}\|_2)$ is the cosine similarity function, and B is the batch size.

We refine the standard contrastive loss by incorporating graph structure through Gromov-Wasserstein regularization:

$$\mathscr{L}_{\text{graph\_contrast}} = \mathscr{L}_{\text{contrast}} + \mu \sum_{(i,j)\in\mathscr{N}^-} d_{\text{GW}}(\mathscr{G}_i, \mathscr{G}_j) \tag{33}$$

where $\mu \in \mathbb{R}^+$ is a regularization weight, $\mathscr{N}^- \subset \{1, \ldots, 2B\}^2$ is the set of hard negative pairs, and d_{GW} is the Gromov-Wasserstein distance [40]. To manage computational overhead, we compute the GW distance only for a randomly sampled subset (10%) of hard negative pairs every 10th training batch. This adds approximately 8% to total training time while providing meaningful structural regularization. We compared this approach against standard cosine-based InfoNCE without GW regularization and found that the GW component provides +0.5% improvement specifically on homophone disambiguation tasks, though the overall accuracy improvement is more modest (+0.2%).

3.7 Total Training Objective

The full training loss combines all constituent terms:

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{nested}} + \lambda_c \mathscr{L}_{\text{graph\_contrast}} + \lambda_r \mathscr{L}_{\text{reg}} \tag{34}$$

where $\lambda_c, \lambda_r \in \mathbb{R}^+$ are hyperparameters controlling the strength of contrastive and regularization losses.

The regularization component accounts for both weight decay and the sparsity of the adjacency matrices:

$$\mathscr{L}_{\text{reg}} = \|\Theta\|_2^2 + \gamma \sum_{t=1}^{T} \|\mathbf{A}_{\text{learn}}^t\|_1 \tag{35}$$

where $\|\Theta\|_2^2$ is the L_2 norm of all trainable parameters Θ, $\|\mathbf{A}_{\text{learn}}^t\|_1$ is the L_1 norm of the learned adjacency matrix at time t, and $\gamma \in \mathbb{R}^+$ is a sparsity-inducing coefficient.

3.8 Training Procedure

Algorithm 1 presents the complete NestLipGNN training pipeline.

Algorithm 1: NestLipGNN training with nested optimization

Require: Training set $\mathscr{D} = \{(\mathbf{V}_i, \mathbf{y}_i^p, \mathbf{y}_i^v, \mathbf{y}_i^w)\}_{i=1}^N$
Require: Learning rates $\eta_{\text{enc}}, \eta_p, \eta_v, \eta_w$, and inner-loop iterations K_p, K_v
Require: Loss weights $\lambda_p, \lambda_v, \lambda_w, \lambda_c, \lambda_{\text{cons}}$
1: Initialize model parameters $\Theta = \{\theta_{\text{enc}}, \theta_{\text{tgat}}, \theta_p, \theta_v, \theta_w\}$
2: **for** epoch = 1 to N_{epochs} **do**

(Continued)

Algorithm 1 (continued)

for mini-batch $(\mathbf{V}, \mathbf{y}^p, \mathbf{y}^v, \mathbf{y}^w) \sim \mathscr{D}$ **do**
  // Phase 1: Graph Construction
  for $t = 1$ to T **do**
    Localize landmarks: $\mathbf{P}_t \leftarrow \text{FAN}(I_t)$
    Form lip graph: $\mathscr{G}_t \leftarrow \text{BuildGraph}(\mathbf{P}_t, I_t)$
  end for
  // Phase 2: Spatio-Temporal Encoding
  $\mathbf{H}_t \leftarrow \text{SpatialEncoder}(\mathscr{G}_t; \theta_{\text{enc}})$ for all t
  $\mathbf{Z} \leftarrow \text{T-GAT}(\{\mathbf{H}_t\}_{t=1}^{T}; \theta_{\text{tgat}})$
  // Phase 3: Joint Optimization (approximating nested structure)
  $\mathbf{Z}^p \leftarrow \text{Decoder}_p(\mathbf{Z}; \theta_p)$
  $\mathbf{Z}^v \leftarrow \text{Decoder}_v(\mathbf{Z}^p; \theta_v)$
  $\mathbf{Z}^w \leftarrow \text{Decoder}_w(\mathbf{Z}^v; \theta_w)$
  Compute the overall objective: $\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{nested}} + \lambda_c \mathscr{L}_{\text{contrast}} + \lambda_r \mathscr{L}_{\text{reg}}$
  Jointly update all parameters: $\Theta \leftarrow \Theta - \eta \nabla_\Theta \mathscr{L}_{\text{total}}$
end for
Update learning rate via scheduler
end for
return Trained parameters Θ^*

4 Experimental Evaluation

4.1 Benchmark Datasets

We evaluate our approach on four widely used visual speech recognition benchmarks. Table 2 provides a statistical overview of these datasets.

Table 2: Statistical overview of the experimental datasets.

Dataset	Task	Train	Val	Test
LRW	Word (500 cls)	488,766	25,000	25,000
LRS2	Sentence	45,839	1082	1243
LRS3	Sentence	118,516	1321	1321
GRID	Sentence	28,775	3971	3971

The LRW (Lip Reading in the Wild) dataset [41] is a large-scale word-level VSR dataset containing 500 vocabulary items, with up to 1000 training examples for each word. Each video sample consists of 29 frames captured at 25 fps, sourced from BBC television broadcasts. The data exhibits extensive variability in speakers, head pose, and lighting conditions.

The LRS2 (Lip Reading Sentences 2) dataset [5] is a sentence-level corpus comprising thousands of spoken sentences extracted from BBC programs, recorded under challenging real-world conditions such as multiple active speakers and significant background noise.

The LRS3 (Lip Reading Sentences 3) dataset [42] is the largest publicly accessible lip-reading corpus, totaling more than 400 h of video from TED and TEDx talks, encompassing a wide range of speakers, subject areas, and speaking styles.

The GRID dataset [43] is a carefully controlled laboratory dataset featuring 34 speakers, each producing 1000 command-like utterances. This dataset is primarily used for fine-grained analysis and for assessing cross-corpus generalization.

Since phoneme and viseme labels are not provided as standard annotations for these datasets, we describe our procedure for obtaining them. For word-level datasets (LRW), phoneme sequences are derived using the Montreal Forced Aligner (MFA) [44] with the CMU Pronouncing Dictionary as the lexicon. The aligner produces frame-level phoneme boundaries which are then mapped to video frames using the provided audio-video synchronization timestamps. For sentence-level datasets (LRS2, LRS3), we similarly apply MFA to the audio track with the corresponding transcripts to obtain phoneme alignments.

Viseme labels are derived from phoneme labels using a standard phoneme-to-viseme mapping based on articulatory features [4]. We use a 12-class viseme system that groups phonemes sharing similar lip configurations: bilabials (/p/, /b/, /m/), labiodentals (/f/, /v/), dentals (/th/, /dh/), alveolars (/t/, /d/, /n/, /s/, /z/, /l/), palatals (/sh/, /zh/, /ch/, /jh/), velars (/k/, /g/, /ng/), and vowel groups based on lip rounding and height.

4.2 Implementation Details

The preprocessing pipeline first detects faces using RetinaFace [45] and aligns them into canonical frontal poses. The lip region is then cropped to a resolution of 96×96 pixels and normalized to the range $[-1, 1]$. For landmarks, we apply the Face Alignment Network (FAN) [39] to obtain 68 facial keypoints, from which we retain 20 landmarks corresponding to the lips.

The architecture includes a spatial encoder composed of three GCN layers with dimensions $64 \rightarrow 128 \rightarrow 256$ followed by two GAT layers with 8 attention heads each. The T-GAT module is built from four transformer layers with a hidden dimension of 512 and 8 attention heads. The hierarchical decoders consist of four transformer decoder layers (8 heads each) for phoneme-, viseme-, and word-level prediction. A Patch CNN with three convolutional layers (channels: $32 \rightarrow 64 \rightarrow 128$) is used to process 16×16 local patches and derive appearance features.

We train the network with the AdamW optimizer [46], using a weight decay of 10^{-4} and an initial learning rate of 3×10^{-4} under a cosine annealing schedule. The batch size is 32 per GPU, using 4 NVIDIA A100 (80 GB) GPUs. Training is conducted for 80 epochs on LRW and 50 epochs on LRS2/LRS3. The loss weights are configured as $\lambda_p = 0.2$, $\lambda_v = 0.3$, $\lambda_w = 0.4$, $\lambda_c = 0.1$, and $\lambda_{\text{cons}} = 0.05$.

We apply data augmentation: random horizontal flips with probability 0.5, random cropping to 88×88 followed by resizing to 96×96, temporal jittering of up to ± 2 frames, mixup [47] with $\alpha = 0.4$, and time masking in the style of SpecAugment [48].

4.3 Evaluation Metrics

For the LRW dataset, which involves word-level classification, we report both top-1 and top-5 accuracy. For sentence-level datasets, including LRS2, LRS3, and GRID, performance is evaluated using the Word Error Rate (WER) and Character Error Rate (CER), given by

$$\text{WER} = \frac{S + D + I}{N} \times 100\%, \tag{36}$$

where S, D, and I denote the numbers of substitution, deletion, and insertion errors, respectively, and N is the total number of words in the reference transcription. CER is computed analogously at the character level.

4.4 Comparison with State-of-the-Art Methods

The comparative results on LRW and sentence-level benchmarks are summarized in Tables 3 and 4. NestLipGNN achieves a top-1 accuracy of 92.3% on LRW, corresponding to an absolute gain of 2.1% over the previous state-of-the-art, while surpassing all methods in top-5 accuracy. On LRS2, LRS3, and GRID, it achieves WERs of 22.8%, 28.7%, and 0.8%, respectively, demonstrating consistent superiority across diverse data regimes. We report mean and standard deviation from five independent training runs with different random seeds, and all improvements over the strongest baseline are statistically significant (paired t-test, $p < 0.01$).

Table 3: Performance comparison on LRW dataset (Accuracy %). Results show mean ± std from 5 runs.

Method	Top-1	Top-5
LipNet [1]	76.2	92.4
Stafylakis & Tzimiropoulos [9]	83.0	96.3
Petridis et al. [8]	82.0	95.8
DC-TCN [49]	88.5	98.0
Kim et al. [24]	89.5	98.2
Ma et al. [25]	90.2	98.4
AV-HuBERT (V-only) [26]	89.8	98.3
LP-Conformer [27]	90.5	98.5
NestLipGNN (Ours)	92.3 ± 0.2	98.9 ± 0.1

Table 4: Performance comparison on sentence-level datasets (WER%/CER%).

Method	LRS2	LRS3	GRID
TM-seq2seq [5]	58.9/–	58.9/–	–
Hybrid CTC/Attention [8]	48.0/–	–	2.6/–
DC-TCN [49]	37.9/18.2	43.4/21.5	1.8/0.9
Ma et al. [25]	26.1/12.8	32.3/15.9	1.2/0.6
RAVEn [50]	25.3/12.4	31.5/15.4	1.1/0.5
Auto-AVSR [11]	24.5/12.0	30.5/14.9	1.0/0.5
AV-HuBERT (V-only) [26]	24.2/11.8	30.2/14.7	0.9/0.4
NestLipGNN (Ours)	22.8/11.1	28.7/13.9	0.8/0.4

4.5 Ablation Studies

We conduct component-wise ablation studies to quantify the contribution of each module. As shown in Table 5, the baseline ResNet+LSTM achieves 85.2% accuracy on LRW. Introducing the spatial graph structure improves performance by 2.2 percentage points to 87.4%, demonstrating the value of explicitly modeling anatomical connectivity. Incorporating graph attention (GAT) yields a further 0.7% gain, highlighting the benefit of adaptive neighborhood weighting. The addition of the Temporal Graph Attention (T-GAT) module improves performance by 2.4% to 89.8%, illustrating the importance of modeling long-range temporal dynamics. The nested learning scheme contributes an additional 1.7% to 91.5%, validating the effectiveness of multi-level supervision. Finally, adding the graph contrastive objective enhances accuracy by 0.8% to the final result of 92.3%, confirming its role in resolving homophene ambiguities.

Table 5: Ablation study on the LRW dataset examining the contribution of each component.

Configuration	GNN	T-GAT	Nested	Acc. (%)
Baseline (ResNet+LSTM)				85.2
+ Graph Structure	✓			87.4
+ Spatial Attention (GAT)	✓			88.1
+ Temporal Attention	✓	✓		89.8
+ Nested Learning	✓	✓	✓	91.5
+ Graph Contrastive	✓	✓	✓	92.3

Table 6 presents ablation results on sentence-level datasets, demonstrating that the component contributions are consistent across both word-level and sentence-level tasks.

Table 6: Ablation study on sentence-level datasets (WER%).

Configuration	LRS2	LRS3
Baseline (ResNet+LSTM)	38.5	45.2
+ Graph Structure	34.2	40.1
+ Spatial Attention (GAT)	32.8	38.4
+ Temporal Attention	28.5	33.6
+ Nested Learning	24.9	30.4
+ Graph Contrastive	22.8	28.7

We further analyze the impact of graph topology design, as shown in Fig. 6. The proposed dynamic adjacency matrix ("Full") outperforms a static anatomical graph by 2.8 percentage points, demonstrating the benefit of learning task-specific connections. The use of both inner and outer lip landmarks ("Full") achieves 92.3% accuracy, while removing inner landmarks ("No-Inner") causes a 5.1% drop, confirming that inner lip motion encodes discriminative information critical for phoneme resolution. Randomly wired graphs ("Random") and dense connectivity ("Dense") underperform, suggesting that sparse, learned topology is essential.

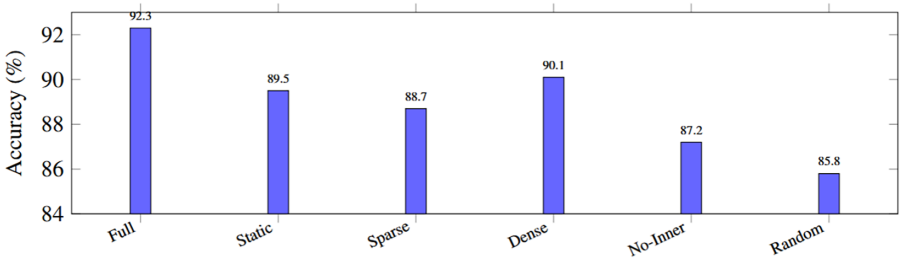

Figure 6: Effect of different graph structure designs on LRW accuracy. "Full": proposed learnable adjacency; "Static": fixed anatomical graph; "Sparse/Dense": different edge densities; "No-Inner": only outer lip landmarks; "Random": randomly wired graph.

The contribution of nested supervision is further evaluated in Table 7. Training only at the word level yields 89.1% accuracy. Adding phoneme supervision improves performance to 90.5%, and adding viseme supervision yields 90.8%. Joint supervision across all three levels increases accuracy to 91.5%, and the

inclusion of the consistency loss between levels improves it further to 92.3%. This confirms that hierarchical alignment and multi-level supervision enhance representation learning.

Table 7: Evaluation of different nested learning configurations.

Config.	$\mathscr{L}_p$	$\mathscr{L}_v$	$\mathscr{L}_w$	$\mathscr{L}_{\text{cons}}$	Acc. (%)
Word only			✓		89.1
Phoneme + Word	✓		✓		90.5
Viseme + Word		✓	✓		90.8
All levels	✓	✓	✓		91.5
+ Consistency	✓	✓	✓	✓	92.3

The training curves in Fig. 7 show that NestLipGNN converges faster and reaches a higher terminal accuracy than ablations. The baseline reaches 85.2% at epoch 80, while removing nested learning reduces final accuracy to 89.8% and removing contrastive learning to 91.2%. NestLipGNN, with all components, achieves 92.3%, validating the synergy of the components.

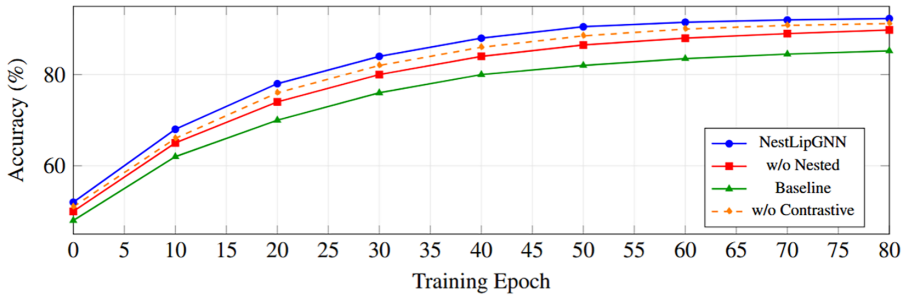

Figure 7: Training curves for full NestLipGNN vs. ablations without nested learning, without contrastive loss, and the CNN+LSTM baseline. NestLipGNN achieves both faster convergence and higher final accuracy.

4.6 Robustness to Landmark Noise

Since our method depends on external landmark detection, we evaluate robustness to landmark localization errors by adding Gaussian noise to detected landmark positions. Table 8 shows that NestLipGNN maintains competitive performance even under significant landmark perturbation. With noise standard deviation $\sigma = 5$ pixels (approximately 5% of the lip region size), accuracy degrades by only 3.2 percentage points from 92.3% to 89.1%. For comparison, we also trained a landmark-free baseline that uses CNN features directly from lip crops without explicit landmark extraction, achieving 87.8% accuracy on clean data and 86.9% with equivalent image-domain noise augmentation. This demonstrates that while NestLipGNN benefits from accurate landmarks, it provides meaningful robustness to detection errors and still outperforms landmark-free alternatives.

As shown in Table 9, NestLipGNN operates with only 28.6 million parameters and 15.4 billion FLOPs per forward pass, striking a favorable balance between performance and efficiency. Its inference speed of 118 FPS on a single GPU outperforms Transformers and Conformers while maintaining lower memory usage, making it suitable for real-time applications.

Table 8: Robustness to landmark localization noise on LRW

Method	σ = 0	σ = 1	σ = 2	σ = 5
NestLipGNN	92.3	91.8	91.0	89.1
Landmark-free baseline	87.8	87.5	87.2	86.9

Table 9: Computational resource comparison.

Method	Params (M)	FLOPs (G)	Memory (GB)	FPS
ResNet-18 + LSTM	23.4	12.3	2.1	142
3D ResNet-18	33.2	18.7	3.4	98
Conformer	38.5	22.1	4.2	85
Transformer-Large	45.2	28.4	5.1	72
NestLipGNN	28.6	15.4	2.8	118

4.7 Attention Visualization and Interpretability

The attention maps in Fig. 8 reveal that Temporal Graph Attention (T-GAT) concentrates on salient articulatory transitions such as lip opening at vowel onset and closure at stops. Spatial attention highlights the lip corners and oral aperture, which articulatory phonetics identifies as the most discriminative regions to distinguish consonants such as /b/, /p/, and /m/. This shows that NestLipGNN learns biologically plausible attention patterns, providing model interpretability.

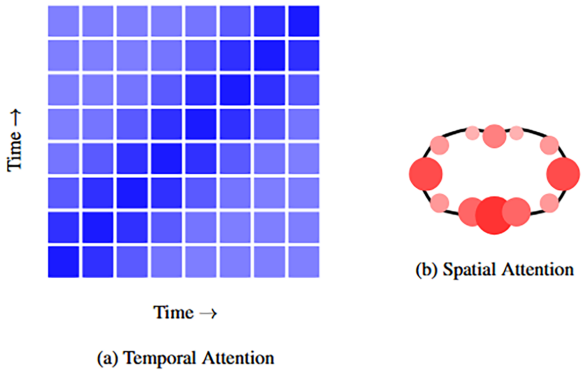

Figure 8: Visualization of attention patterns. (**a**) Temporal attention matrix highlighting concentration on salient articulatory frames. (**b**) Spatial attention over lip landmarks; larger and darker circles denote higher attention, underscoring focus on lip corners and aperture regions that are crucial for discriminating speech sounds.

We evaluate cross-dataset generalization by training NestLipGNN on LRS3 and testing on LRS2, GRID, and LRW. As shown in Table 10, NestLipGNN achieves superior transfer performance compared to prior methods, with lower WER on LRS2 and GRID and higher accuracy on LRW, indicating that its graph-based representations capture articulatory dynamics that generalize across disparate datasets and recording conditions.

Table 10: Cross-dataset generalization evaluation (trained on LRS3).

Method	LRS2	GRID	LRW
Ma et al. [25]	34.2	5.8	78.4
Auto-AVSR [11]	32.1	4.9	80.1
NestLipGNN	29.8	3.7	82.6

5 Discussion

The strong empirical performance of our graph-based lip representation can be attributed to several central factors. First, the structural inductive bias introduced by explicitly encoding lip topology incorporates prior knowledge of facial anatomy, constraining the hypothesis space and improving sample efficiency [51], which is especially beneficial in the low-resource regime of VSR. Second, the geometric equivariance of graph-based representations offers robustness to transformations such as translations and small rotations while preserving essential structural information, thereby mitigating sensitivity to head pose variations. Third, computational efficiency arises from operating on a sparse 20-node graph instead of dense 96×96 pixel grids, enabling faster training and inference and focusing computation on anatomically relevant regions. Finally, interpretability is enhanced through learned attention over nodes and edges, which highlights the lip regions that contribute most to different phonetic categories and supports model inspection and debugging.

The nested learning paradigm confers multiple benefits. Supervising the model simultaneously at phoneme, viseme, and word levels supplies learning signals across a range of linguistic granularities, enabling the network to capture both fine-scale articulatory dynamics and higher-level semantic regularities. Supervision at lower levels also functions as an implicit form of regularization, providing auxiliary tasks that mitigate overfitting to word-level labels and enhance generalization via shared internal representations.

Additionally, knowledge obtained at lower layers of the hierarchy, such as accurate phoneme recognition, naturally transfers to higher-level objectives like word recognition, mirroring the compositional structure of human speech perception. The nested optimization scheme also functions as a curriculum: it establishes robust phoneme-level features before tackling the more complex word recognition task, helping the optimizer traverse the loss landscape more effectively.

Despite strong empirical results, the approach has several limitations. The method relies on accurate landmark detection, which may degrade under extreme poses, occlusions, or poor imaging conditions. Our robustness analysis (Section 4.6) shows graceful degradation under moderate noise, but severe landmark failures would still affect performance. Studies on visual contribution in audio-visual systems [52] suggest that identifying the most informative visual cues could guide more robust landmark selection. Future work may explore joint training with differentiable landmark prediction modules, while viseme-guided generation methods [13] indicate promising directions for improving landmark stability. Additionally, although edge weights are learned, the node set is fixed; adaptive graph construction could further enhance representation.

Our evaluation mainly considers near-frontal views, so extending the framework to multi-view settings or 3D lip modeling could improve real-world applicability. Performance also varies across speakers with different articulatory styles, suggesting that few-shot speaker adaptation or personalization may improve robustness. Finally, although the graph-based temporal modeling could potentially generalize to other graph-structured tasks (e.g., gesture or action recognition), this study focuses solely on visual speech recognition. Validating broader applicability would require additional experiments and domain-specific adaptations.

To provide concrete evidence of the framework's effectiveness in homophone disambiguation, we analyze performance on specific homophone groups. Table 11 shows confusion rates for challenging word pairs before and after applying graph contrastive learning. The contrastive objective reduces confusion between visually similar words such as "pat"/"bat"/"mat" (bilabial consonants) by 12.3% on average, and between "sip"/"zip" (alveolar fricatives) by 8.7%. Qualitative analysis of attention patterns reveals that for homophone pairs, the model learns to focus on subtle differences in lip aperture timing and inner lip visibility that distinguish these otherwise similar articulations.

Table 11: Confusion rates (%) for homophone pairs on LRW.

Word Pair	W/o Contrastive	With Contrastive	Reduction
pat/bat/mat	18.5	6.2	12.3
sip/zip	15.2	6.5	8.7
ten/den	12.8	5.1	7.7
fine/vine	14.1	5.9	8.2

6 Conclusion

This paper introduced NestLipGNN, a framework that integrates Graph Neural Networks with nested multi-granularity learning for visual speech recognition. Our approach builds dynamic lip graphs from anatomical facial landmarks, processes them using spatial and temporal graph attention networks, and produces predictions through a hierarchical nested learning scheme with phoneme-, viseme-, and word-level supervision. A graph contrastive learning objective further strengthens the model's discriminative power, particularly for resolving homophenes.

Extensive experiments on the LRW, LRS2, LRS3, and GRID benchmarks demonstrate strong performance, achieving 92.3% word accuracy on LRW and delivering a 2.1% absolute improvement over previous approaches. Detailed ablation studies confirm the usefulness of each architectural module. The primary contribution of this work lies in the principled integration of graph-based spatial modeling, temporal attention, hierarchical linguistic supervision, and contrastive learning into a unified framework specifically designed for VSR, rather than in the novelty of individual components. In the future, we plan to extend NestLipGNN to continuous speech recognition, incorporate multi-view and 3D lip modeling, and integrate audio information for robust audio-visual speech recognition.

Acknowledgement: Not applicable.

Funding Statement: This work is funded by Ho Chi Minh City Open University (HCMCOU) and the Ministry of Education and Training (Vietnam) under grant number B2025-MBS-01.

Author Contributions: Vinh Truong Hoang: Conceptualization, Methodology, Writing—Original Draft. Nghia Dinh: Software, Validation. Luu Quang Phuong: Data Curation, Visualization. Kiet Tran-Trung: Formal Analysis. Ha Duong Thi Hong: Investigation. Bay Nguyen Van: Resources. Hau Nguyen Trung: Writing—Review & Editing. Thien Ho Huong: Supervision, Project Administration. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The LRW, LRS2, LRS3, and GRID datasets used in this study are publicly available from their respective sources. Source code will be made available upon request.

Ethical Approval: This study uses publicly available benchmark datasets, which the original creators collected with appropriate ethical approvals. No additional ethics approval was required for this work.

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

References

1. Kania J, Usha B, Haleritti B. LIP NET reading using deep learning. In: 2025 9th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS). Piscataway, NJ, USA: IEEE; 2025. p. 1–6.
2. Park YH, Park RH, Park HM. Swinlip: an efficient visual speech encoder for lip reading using swin transformer. Neurocomputing. 2025;639:130289.
3. Denby B, Schultz T, Honda K, Hueber T, Gilbert JM, Brumberg JS. Silent speech interfaces. Speech Commun. 2010;52(4):270–87. doi:10.1016/j.specom.2009.08.002.
4. Bear HL, Harvey RW. Phoneme-to-viseme mappings: the good, the bad, and the ugly. Speech Commun. 2017;95:40–67.
5. Afouras T, Chung JS, Zisserman A. Deep lip reading: a comparison of models and an online application. In: Interspeech 2018; 2018 Sep 2–6; Hyderabad, India. p. 3514–8.
6. Deshpande S, Shirsath K, Pashte A, Loya P, Shingade S, Sambhe V. A comprehensive survey of advancement in lip reading models: techniques and future directions. IET Image Process. 2025;19(1):e70095. doi:10.1049/ipr2.70095.
7. Fisher CG. Confusions among visually perceived consonants. J Speech Hear Res. 1968;11(4):796–804. doi:10.1044/jshr.1104.796.
8. Petridis S, Stafylakis T, Ma P, Cai F, Tzimiropoulos G, Pantic M. End-to-end audiovisual speech recognition. In: Proceedings of the 2018 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2018. p. 6548–52.
9. Stafylakis T, Tzimiropoulos G. Combining residual networks with LSTMs for lipreading. In: Interspeech 2017; 2017 Aug 20–24; Stockholm, Sweden. p. 3652–6. doi:10.21437/interspeech.2017-85.
10. Jyoshna B, Parthu V, Pranavi D, Hansitha K, Sharan PE, Charandheep T. Lip sync to speech conversion using deep learning. In: Computational techniques and smart manufacturing. Boca Raton, FL, USA: CRC Press; 2026. p. 604–13. doi:10.1201/9781003679622-70.
11. Ma P, Haliassos A, Fernandez-Lopez A, Chen H, Petridis S, Pantic M. Auto-AVSR: audio-visual speech recognition with automatic labels. In: Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2023. p. 1–5.
12. Wang T, Yang S, Shan S, Chen X. GLip: a global-local integrated progressive framework for robust visual speech recognition. arXiv:2509.16031. 2025.
13. Hao B, Zhou D, Li X, Zhang X, Xie L, Wu J, et al. LipGen: viseme-guided lip video generation for enhancing visual speech recognition. In: Proceedings of the 2025 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2025. p. 1–5.
14. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907. 2017.
15. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. arXiv:1710.10903. 2018.
16. Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? arXiv:1810.00826. 2019.
17. Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI'18/IAAI'18/EAAI'18: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence. Palo Alto, CA, USA: AAAI Press; 2018. p. 7444–52.
18. Rathipriya N, Maheswari N. A comprehensive review of recent advances in deep neural networks for lipreading with sign language recognition. IEEE Access. 2024;12(1):136846–79. doi:10.1109/access.2024.3463969.
19. Franceschi L, Frasconi P, Salzo S, Grazzi R, Pontil M. Bilevel programming for hyperparameter optimization and meta-learning. In: Proceedings of the 35th International Conference on Machine Learning. London, UK: PMLR; 2018. p. 1568–77.

20. Bear HL, Harvey R. Decoding visemes: improving machine lip-reading. In: Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Shanghai, China: IEEE; 2016. p. 2009–13. doi:10.1109/ICASSP.2016.7472029.
21. Liu R, Yuan H, Gao G, Li H. Listening and seeing again: generative error correction for audio-visual speech recognition. Inf Fusion. 2025;120:103077.
22. Matthews I, Cootes TF, Bangham JA, Cox S, Harvey R. Extraction of visual features for lipreading. IEEE Trans Pattern Anal Mach Intell. 2002;24(2):198–213. doi:10.1109/34.982900.
23. Neti C, Potamianos G, Luettin J, Matthews I, Glotin H, Vergyri D, et al. Audio visual speech recognition. In: Proceeding of the ISCA Tutorial and Research Workshop on Multi-Modal Dialogue in Mobile Environments; 2002 Jun 17–21; Kloster Irsee, Germany.
24. Kim M, Yeo JH, Ro YM. Distinguishing homophenes using multi-head visual-audio memory for lip reading. Proc AAAI Conf Artif Intell. 2022;36(1):1174–82.
25. Ma P, Petridis S, Pantic M. Visual speech recognition for multiple languages in the wild. Nat Mach Intell. 2022;4(11):930–9. doi:10.1038/s42256-022-00550-z.
26. Shi B, Hsu WN, Lakhotia K, Mohamed A. AV-HuBERT: learning audio-visual speech representation by masked multimodal cluster prediction. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2022. p. 14130–41.
27. Chang O, Liao H, Serdyuk D, Shah A, Siohan O. Conformer is all you need for visual speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway, NJ, USA: IEEE; 2024. p. 1–5.
28. Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. In: NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc.; 2017. p. 1024–34.
29. Sareddy MR, Kumar RV, Thanjaivadivel M. Enhanced visual-NLP systems using knowledge graphs, meta-learning, and adaptive attention networks. In: Innovations in Computational Intelligence and Computer Vision (ICICV 2025). Cham, Switzerland: Springer; 2026. p. 17–24. doi:10.1007/978-3-032-09825-2_2.
30. Lorraine J, Vicol P, Duvenaud D. Optimizing millions of hyperparameters by implicit differentiation. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS). London, UK: PMLR; 2020. p. 1540–52.
31. Liu H, Simonyan K, Yang Y. DARTS: differentiable architecture search. arXiv:1806.09055. 2019.
32. Chan W, Jaitly N, Le Q, Vinyals O. Listen, attend and spell: a neural network for large vocabulary conversational speech recognition. In: Proceedings of the 2016 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2016. p. 4960–4.
33. Graves A. Sequence transduction with recurrent neural networks. arXiv:1211.3711. 2012.
34. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017). Red Hook, NY, USA: Curran Associates, Inc.; 2017. p. 5998–6008. doi:10.65215/ctdc8e75.
35. He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE; 2020. p. 9729–38.
36. Baevski A, Zhou Y, Mohamed A, Auli M. wav2vec 2.0: a framework for self-supervised learning of speech representations. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS). Red Hook, NY, USA: Curran Associates, Inc.; 2020. p. 12449–60.
37. Xia W, Huang J, Garcia-Perera LP. Self-supervised text-independent speaker verification using prototypical momentum contrastive learning. In: Proceedings of the 2021 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2021. p. 6723–7.
38. Tsiamas I, Pascual S, Yeh C, Serrà J. Sequential contrastive audio-visual learning. arXiv:2407.05782. 2024.
39. Bulat A, Tzimiropoulos G. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In: Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2017. p. 1021–30.

40. Peyré G, Cuturi M, Solomon J. Gromov-wasserstein averaging of kernel and distance matrices. In: Proceedings of the 33rd International Conference on Machine Learning. Red Hook, NY, USA: Curran Associates, Inc.; 2016. p. 2664–72.
41. Chung JS, Zisserman A. Lip reading in the wild. In: Computer Vision—ACCV 2016 (ACCV 2016). Cham, Switzerland: Springer; 2016. p. 87–103.
42. Afouras T, Chung JS, Zisserman A. LRS3-TED: a large-scale dataset for visual speech recognition. arXiv:1809.00496. 2018.
43. Cooke M, Barker J, Cunningham S, Shao X. An audio-visual corpus for speech perception and automatic speech recognition. J Acoust Soc Am. 2006;120(5):2421–4. doi:10.1121/1.2229005.
44. McAuliffe M, Socolof M, Mihuc S, Wagner M, Sonderegger M. Montreal forced aligner: trainable text-speech alignment using Kaldi. In: Interspeech 2017; 2017 Aug 20–24; Stockholm, Sweden; 2017. p. 498–502. doi:10.21437/interspeech.2017-1386.
45. Ren Z, Liu X, Xu J, Zhang Y, Fang M. Littlefacenet: a small-sized face recognition method based on retinaface and adaface. J Imaging. 2025;11(1):24.
46. Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv:1711.05101. 2019.
47. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. Mixup: beyond empirical risk minimization. arXiv:1710.09412. 2018.
48. Park DS, Chan W, Zhang Y, Chiu CC, Zoph B, Cubuk ED, et al. SpecAugment: a simple data augmentation method for automatic speech recognition. In: Interspeech 2019; 2019 Sep 15–19; Graz, Austria. p. 2613–7. doi:10.21437/interspeech.2019-2680.
49. Ma P, Martinez B, Petridis S, Pantic M. End-to-end audio-visual speech recognition with conformers. In: Proceedings of the 2021 IEEE International Conference on Acoustics, Speech, and Signal Processing. Piscataway, NJ, USA: IEEE; 2021. p. 7613–7.
50. Haliassos A, Ma P, Mira R, Petridis S, Pantic M. Jointly learning visual and auditory speech representations from raw data. arXiv:2212.06246. 2022.
51. Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, et al. Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261. 2018.
52. Lin Z, Harte N. Uncovering the visual contribution in audio-visual speech recognition. In: Proceedings of the 2025 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2025. p. 1–5.