Learning Long-Term Spatial-Temporal Graphs for Active Speaker Detection

Kyle Min[1*], Sourya Roy[2*†], Subarna Tripathi[1], Tanaya Guha[3], and Somdeb Majumdar[1]

[1]Intel Labs, [2]UC Riverside, [3]University of Glasgow
{kyle.min,subarna.tripathi,somdeb.majumdar}@intel.com

Abstract. Active speaker detection (ASD) in videos with multiple speakers is a challenging task as it requires learning effective audiovisual features and spatial-temporal correlations over long temporal windows. In this paper, we present SPELL, a novel spatial-temporal graph learning framework that can solve complex tasks such as ASD. To this end, each person in a video frame is first encoded in a unique node for that frame. Nodes corresponding to a single person across frames are connected to encode their temporal dynamics. Nodes within a frame are also connected to encode inter-person relationships. Thus, SPELL reduces ASD to a node classification task. Importantly, SPELL is able to reason over long temporal contexts for all nodes without relying on computationally expensive fully connected graph neural networks. Through extensive experiments on the AVA-ActiveSpeaker dataset, we demonstrate that learning graph-based representations can significantly improve the active speaker detection performance owing to its explicit spatial and temporal structure. SPELL outperforms all previous state-of-the-art approaches while requiring significantly lower memory and computational resources. Our code is publicly available: https://github.com/SRA2/SPELL

1 Introduction

Holistic scene understanding in the wild is still a challenge in computer vision despite recent breakthroughs in several other areas. A *scene* represents real-life events spanning complex visual and auditory information, which are often intertwined. Active speaker detection (ASD) is a key component in scene understanding and is an inherently multimodal (audio-visual) task. The objective here is, given a video input, to identify which persons are speaking in each frame. This has numerous practical applications ranging from speech enhancement systems [1] to human-robot interaction [31,30].

Earlier efforts on ASD had limited success due to the unavailability of large datasets, powerful learning models, or computing resources [7,8,9]. With the release of AVA-ActiveSpeaker [26], a large and diverse ASD dataset, a number of promising approaches have been developed including both visual-only

* Authors contributed equally
† Work partially done during an internship at Intel Labs

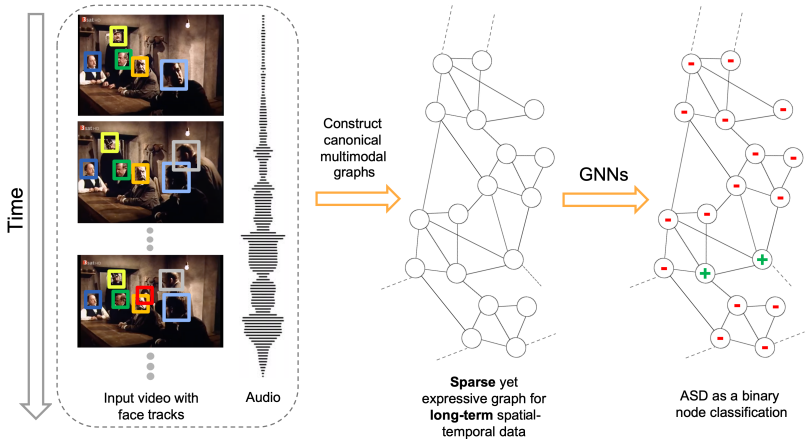

Fig. 1. SPELL converts a video into a canonical graph from the audio-visual input data, where each node corresponds to a person in a frame, and an edge represents a spatial or temporal interaction between the nodes. The constructed graph is dense enough for modeling long-term dependencies through message passing across the temporally-distant but relevant nodes, yet sparse enough to be processed within low memory and computation budget. The Active Speaker Detection (ASD) task is posed as a binary node classification in this long-range spatial-temporal graph.

and audio-visual methods. As visual-only methods [8] are unable to distinguish between verbal and non-verbal lip movements, more recent approaches have focused on joint modeling of the audio-visual information. Audio-visual approaches [2,37,33,18,17] address the task by first encoding visual (primarily facial) and audio features from videos, and then by classifying the fused multimodal features. Such models generally have multi-stage frameworks [2,18,37,17] and show good detection performance. However, state-of-the-art methods have relied on complex architectures for processing the audio-visual features with high computation and memory overheads. For example, TalkNet [33] suggests using a transformer-style architecture [34] to model the cross-modal information from the audio-visual input. ASDNet [17], which is the leading state-of-the-art method, uses a complex 3D convolutional neural network (CNN) to extract more powerful features. These approaches are not scalable and may not be suitable for real-world situations with limited memory and computation budgets.

In this paper, we propose an efficient graph-based framework, which we call SPELL (**Sp**atial-T**e**mpora**l** Graph **L**earning). Figure 1 illustrates an overview of our framework. We construct a multimodal graph from the audio-visual data and cast the active speaker detection as a graph node classification task. First, we create a graph where each node corresponds to each person at each frame and the edges represent spatial or temporal relationships among them. The initial node features are constructed using simple and lightweight 2D CNNs instead of a complex 3D CNN or a transformer. Next, we perform binary node classi-

fication – active or inactive speaker – on this graph by learning a three-layer graph neural network (GNN) model each with a small number of parameters. In our framework, graphs are constructed specifically for encoding the spatial and temporal dependencies among the different facial identities. Therefore, the GNN can leverage this graph structure and model the temporal continuity in speech as well as the long-term spatial-temporal context, while requiring low memory and computation.

Although the proposed graph structure can model the long-term spatial-temporal information from the audio-visual features, it is likely that some of the short-term information may be lost in the process of feature encoding. This is because we use 2D CNNs that are not well-suited for processing the spatial-temporal information when compared to the transformer or the 3D CNNs. To encode the short-term information, we adopt TSM [19] - a generic module for 2D CNNs that is capable of modeling temporal information without introducing any additional parameters or computation. We empirically verify that SPELL can benefit both from the supplementary short-term information provided by TSM and the long-term information modeled by our graph structure.

We show the effectiveness of SPELL by performing extensive experiments on the AVA-ActiveSpeaker dataset [26]. Using our spatial-temporal graph framework on top of the TSM-inspired feature encoders, SPELL outperforms all previous state-of-the-art approaches. Critically, SPELL requires significantly less hardware resources for the visual feature encoding (0.7 GFLOPs, 11.2M #Params) compared to ASDNet [17] (13.2 GFLOPs, 48.6M params), which is the leading state-of-the-art method. In addition, SPELL achieved 2nd place in the AVA-ActiveSpeaker challenge at ActivityNet 2022[1], which also demonstrates the effectiveness of our method (please refer to the technical report [22]).

There are three main contributions in this paper:

- We present a graph-based approach for solving the task of active speaker detection over long time supports by casting it as a node classification problem.
- Our model, SPELL, learns from videos to model the short-term and long-term spatial-temporal information. Specifically, we propose to construct graphs on the TSM-inspired audio-visual features. The graphs are dense enough for message passing across temporally-distant nodes, yet sparse enough to model their interactions within tight memory and compute constraints.
- SPELL notably outperforms existing methods with lower memory and computation complexity on the active speaker detection benchmark dataset, AVA-ActiveSpeaker.

2 Related Work

We discuss related works in two relevant areas: application of GNNs in video scene understanding and active speaker detection.

[1] https://research.google.com/ava/challenge.html

GNNs for scene understanding. CNNs, Long Short Term Memory (LSTM), and their variants have long dominated the field of video understanding. In recent times, two new types of models are gaining traction in many areas of visual information processing: Transformers [34] and GNNs. They are not necessarily in competition with the former models, but it has been shown that they can augment the performance of CNN/LSTM based models. Applications of specialized GNN models in video understanding include visual relationship forecasting [21], dialog modeling [11], video retrieval [32], emotion recognition [28], and action detection [38]. GNN-based generalized video representation frameworks have also been proposed [3,23,24] that can be used for multiple downstream tasks. For example, in Arnab *et al.* [3], a fully connected graph is constructed over the foreground nodes from video frames in a sliding window fashion, and a foreground node is connected to other context nodes from its neighboring frames. The message passing over the fully connected spatial-temporal graph is expensive in terms of the computational time and memory. Thus in practice such models end up using a small sliding window, making them unable to process longer-term sequences. SPELL also operates on foreground nodes - particularly, faces. However, the graph structure is not fully connected. We construct the graph such that it enables interactions only between relevant nodes over space and time. The graph remains sparse enough such that the longer-term context can be accommodated within a comparatively smaller memory and compute budget.

Active speaker detection (ASD). Earlier work on active speaker detection by Cutler *et al.* [7] detects correlated audio-visual signals using a time-delayed neural network. Subsequent works depend only on visual information and considers a simpler set-up focusing on lip and facial gestures [8]. More recently, high-performing ASD models rely on large networks - developed for capturing the spatial-temporal variations in audio-visual signals, often relying on ensemble networks or complex 3D CNN features [2,33]. Sharma *et al.* [27] and Zhang *et al.* [36] both used large 3D CNN architectures for audio-visual learning. The Active Speaker in Context (ASC) model [2] uses non-local attention modules with an LSTM to model the temporal interactions between audio and visual features encoded by two-stream ResNet-18 networks [13]. TalkNet [33] achieves superior performance through the use of a 3D CNN and a couple of Transformers [34] resulting in an effectively large model. Another recent work, the ASDNet [17], uses 3D-ResNet101 for encoding visual data and SincNet [25] for audio. The Unified Context Network (UniCon) [37] proposes relational context modules to capture visual (spatial) and audio-visual context based on convolutional layers. Much of these advances are due to the availability of the AVA-ActiveSpeaker dataset [26]. Previously available multimodal datasets (e.g. [4]) were either smaller or constrained or lacked variability in data. The work by Roth *et al.* [26] also introduced a competitive baseline along with the large dataset. Their baseline involves jointly learning an audio-visual model that is end-to-end trainable. The audio and visual branches in this model are CNN-based which uses a depth-wise separable technique.

MAAS [18] presents a different multimodal graph approach. Our work differs from MAAS in several ways, where the main difference is in the handling of temporal context. While MAAS focuses on short-term temporal windows to construct their graphs, we focus on constructing longer-term audio-visual graphs. More specifically, in MAAS, different faces are connected only between consecutive frames. In contrast, SPELL directly connects faces in a longer-term neighborhood controlled by the time threshold hyperparameter, τ (defined in Sec 3.2). In addition, SPELL exploits the temporal ordering patterns of the face tracks by using all the forward/backward/undirected edges in the time domain. In SPELL, each graph can span from 13 to 55 seconds (refer to Sec 4.2) of a video depending on the number of nodes. This is significantly larger than MAAS where the time window size is fixed at 1.59 seconds. During inference, SPELL performs single forward pass, whereas MAAS performs multiple forward passes.

3 Method

In this section, we describe our approach in detail. Figure 2 illustrates how SPELL constructs a graph from an input video where each node corresponds to a face within a temporal window of the video. SPELL is unique in terms of its canonical way of constructing the graph from a video. The graph is able to reason over long temporal contexts for all nodes without being fully-connected. This is an important design choice to reduce memory and computation overheads. The edges in the graph are only between *relevant* nodes needed for message passing, leading to a sparse graph that can be accommodated within a small memory and computation budget. After converting the video into a graph, we train a lightweight GNN to perform binary node classification on this graph. The model architecture is illustrated in Figure 3. The model utilizes three separate GNN modules for the forward, backward, and undirected graph, respectively. Each module has three layers where the weight of the second layer is shared across all the above three modules. More details and the intuition behind the design choice are described in Section 3.4.

3.1 Notations

Let $G = (V, E)$ be a graph with the node set V and edge set E. For any $v \in V$, we define N_v to be the set of neighbors of v in G. We will assume the graph has self-loops, i.e., $v \in N_v$. In addition, let X denote the set of given node features $\{\mathbf{x}_v\}_{v \in V}$ where $\mathbf{x}_v \in \mathbb{R}^d$ is the feature vector associated with the node v. Given this setup, we can define a k-layer GNN as a set of functions $\mathcal{F} = \{f_i\}_{i \in [k]}$ for $i \geq 1$ where each $f_i : V \to \mathbb{R}^m$ (m will depend on layer index i). All f_i is parameterized by a set of learnable parameters. Furthermore, $X_V^i = \{\mathbf{x}_v\}_{v \in V}$ is the set of features at layer i where $\mathbf{x}_v = f_i(v)$. Here, we assume that f_i has access to the graph G and the feature set from the last layer X_V^{i-1}.

- SAGE-CONV aggregation: This aggregation was proposed by [12] and has a computationally efficient form. Given a d-dimensional feature set X_V^{i-1}, the

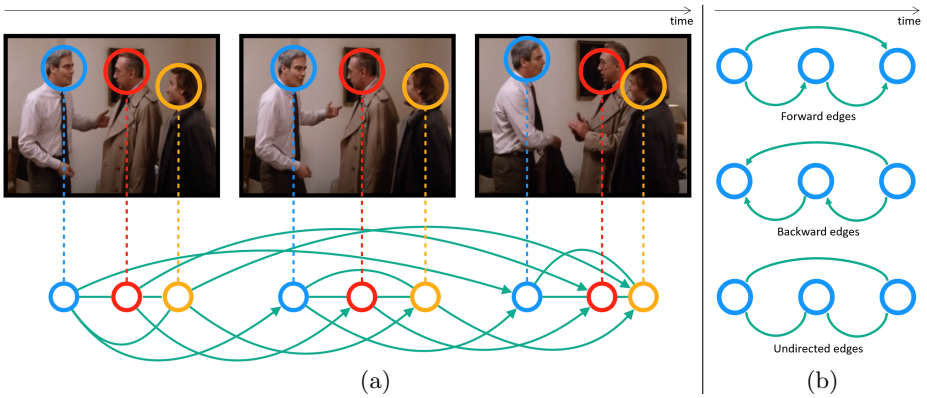

Fig. 2. (a): An illustration of our graph construction process. The frames above are temporally ordered from left to right. The three colors of blue, red, and yellow denote three identities that are present in the frames. Each node in the graph corresponds to each face in the frames. SPELL connects all the inter-identity faces from the same frame with the undirected edges. SPELL also connects the same identities by the forward/backward/undirected edges across the frames. In this example, the same identities are connected across the frames by the forward edges, which are directed and only go in the temporally forward direction. (b): The process for creating the backward and undirected graph is identical, except in the former case the edges for the same identities go in the opposite direction and the latter has no directed edge. Each node also contains the audio information which is not shown here.

function $f_i : V \to \mathbb{R}^m$ is defined for $i \geq 1$ as follows:

$$f(v) = \sigma\Big(\sum_{w \in N_v} \mathsf{M}_i \mathbf{x}_w \Big)$$

where $\mathbf{x}_w \in X_V^{i-1}$, $\mathsf{M}_i \in \mathbb{R}^{m \times d}$ is a learnable linear transformation, and $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-linear activation function applied point-wise.

- EDGE-CONV aggregation: EDGE-CONV [35] models global and local-structures by applying channel-wise symmetric aggregation operation on the edge features associated with all the edges emanating from each node. The aggregation function $f_i : V \to \mathbb{R}^m$ can be defined as:

$$f_i(v) = \sigma\Big(\sum_{w \in N_v} \mathsf{g}_i\big(\mathbf{x}_v \circ \mathbf{x}_w\big) \Big)$$

where $\circ$ denotes concatenation and $\mathsf{g}_i : \mathbb{R}^{2d} \to \mathbb{R}^m$ is a learnable transformation. Often g_i is implemented by MLPs. The number of parameters for EDGE-CONV is larger than SAGE-CONV. This gives the EDGE-CONV layer more expressive power at a cost of higher complexity and possible risk of overfitting. For our model, we set g_i to be an MLP with two layers of linear transformation and a non-linearity. We describe the details in section 4.

3.2 Video as a multimodal graph

We represent a video as a multimodal graph that is suitable for the task of active speaker detection. We assume that the bounding-box information of every face region in each frame is given as per the problem set up. For simplicity, we assume that the entire video is represented by a single graph - if the video has n faces in it, the graph will have n nodes. In our actual implementation, we temporally order the set of all faces in a video, divide them in contiguous sets, and then construct one graph for each such set.

Let B be the set of all face images cropped from an input video (i.e. face-crops). Then, each element $b \in B$ can be represented by a tuple $(\mathsf{Box}, \mathsf{Time}, \mathsf{Id})$, where Box is the normalized bounding-box coordinates of a face-crop in its frame, Time is the time-stamp of its frame, and Id is a unique string that is common to all the face-crops that shares the same identity.

In other words, B can be represented by a set of nodes $[n]$ where $n = |B|$ is the total number of faces that appear in the video. Box is treated as a map such that $\mathsf{Box}(i)$ is defined by the bounding-box coordinates of the i-th face for any $i \in [n]$. Similarly, $\mathsf{Time}(i)$ and $\mathsf{Id}(i)$ correspond to the time and identity of the i-th face, respectively. With this setup, the node set of $G = (V, E)$ is $V = [n] \cong B$, and for any $(i, j) \in [n] \times [n]$, we have $(i, j) \in E$ if either of the following two conditions are satisfied:

- $\mathsf{Id}(i) = \mathsf{Id}(j)$ and $|\mathsf{Time}(i)\text{-}\mathsf{Time}(j)| \leq \tau$
- $\mathsf{Time}(i) = \mathsf{Time}(j)$

where τ is a hyperparameter for the maximum time difference between the nodes having the same identities. In essence, we connect two nodes (faces) if they share the same identity and are temporally close or if they belong to the same frame. Thus, the interactions between different speakers and the temporal variations of the same speaker can jointly be modeled.

To pose the active speaker detection task as a node classification problem, we also need to specify the feature vectors for each node $v \in V$. We use a two-stream 2D ResNet [13] architecture as in [26,2] for extracting the visual features of each face-crop and the audio features of each frame. Then, a feature vector of node v is defined to be $x_v = [v_{\mathsf{visual}} \circ v_{\mathsf{audio}}]$ where v_{visual} is the visual feature of face-crop v and v_{audio} is the audio feature of v's frame where $\circ$ denotes the concatenation. Finally, we can write $G = (V, E, X)$ where X is the set of the node features.

3.3 ASD as a node classification task

In the previous section, we described our graph construction procedure that converts a video into a graph $G = (V, E, X)$ where each node has its own audio-visual feature vector. During the training process, we have access to the ground-truth labels of all face-crops indicating if each of the face-crop is active speaker or not. Therefore, the task of active speaker detection can be naturally posed

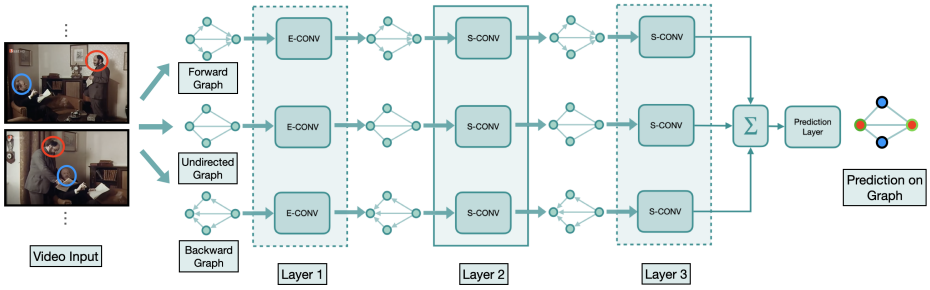

Fig. 3. An illustration of our proposed *Bi-directional* (a.k.a. *Bi-dir*) GNN model for active speaker detection. Here, we have three separate GNN modules for the forward, backward, and undirected graph, respectively. Each module has three layers where the weight of the second layer is shared by all three graph modules. The second layer is placed inside a solid-lined box to indicate the weight sharing while for the first and the third layer we use dotted-lines. E-CONV and S-CONV are shorthand for EDGE-CONV and SAGE-CONV, respectively. We use the color coding: blue and red to denote different identities in input frames. The output of the third layers are added together and then passed to the prediction layer. It applies the sigmoid function to the summed features of every node and produces node classification probabilities.

as a binary node classification problem in the constructed graph G, whether a node is speaking or not speaking. Specifically, we train a three-layer GNN for this classification task. The first layer in the network uses EDGE-CONV aggregation to learn pair-wise interactions between the nodes. For the last two layers, we observe that using SAGE-CONV aggregation provides better performance than EDGE-CONV, possibly due to EDGE-CONV's tendency to overfit.

3.4 SPELL

We now describe how our graph construction and embedding strategy takes temporal ordering into consideration. Specifically, as we use the criterion: $|\mathsf{Time}(i) - \mathsf{Time}(j)| \leq \tau$ for connecting the nodes having the same identities across the frames, the resultant graph becomes undirected. In this process, we lose the information of the temporal ordering of the nodes. To address this issue, we explicitly incorporate temporal direction as shown in Figure 2(b). The undirected GNN is augmented with two other parallel networks; one for going forward in time and another for going backward in time.

More precisely, in addition to the undirected graph, we create a forward graph where we connect (i, j) if and only if $0 \geq \mathsf{Time}(i) - \mathsf{Time}(j) \geq -\tau$. Similarly, (i, j) is connected in a backward graph if and only if $0 \leq \mathsf{Time}(i) - \mathsf{Time}(j) \leq \tau$. This gives us three separate graphs where each of the graphs can model different spatial-temporal relationships between the nodes. Furthermore, the weights of the second layer of each graph is shared across the three graphs. This weight sharing technique can enforce the temporal consistencies among the

different information modeled by the three graphs as well as reduce the number of parameters. For the remaining parts of this paper, we will refer to this network that is augmented with the foward/backward graphs as *Bi-directional* or *Bi-dir* for short.

The proposed three-layer *Bi-dir* is illustrated in Figure 3. We note that right before the *Bi-dir* is applied, the audio and the visual features are further encoded by two learnable MLP layers (linear transformation with ReLU activation) separately and then added to form the fused features for the graph nodes. After the fused features are processed by the first and the second layers, the third layer aggregates all the information and reduce the feature dimension to 1. These 1D features coming from the three separate graphs are added and applied to the sigmoid function to get the final prediction score for each node.

3.5 Feature learning

Similar to ASC [2], we use a two-stream 2D ResNet [13] architecture for the audio-visual feature encoding. The networks take as visual input k consecutive face-crops and take as audio input the Mel-spectrogram of the audio wave sliced along the time duration of the face-crops for the visual stream. Although the 2D ResNet requires significantly lower hardware resources than 3D CNN counterparts or a transformer-style architecture [34], it is not specifically designed for processing spatial-temporal information that is crucial in understanding video contents. To better encode the spatial-temporal information, we augment the visual feature encoder with TSM [19], which provides 2D CNNs with a capability to model the short-term temporal information without introducing any additional parameters or computation. This additional use of TSM can greatly improve the quality of the visual features, and we empirically establish that SPELL benefits from the supplementary short-term information. The audio-visual features from the two stream are concatenated to be node features $\{\mathbf{x}_v\}$.

Data augmentation. Reliable ASD models should be able to detect speaking signals even if there is a noise in the audio. To make our method robust to noise, we make use of data augmentation methods while training the feature extractor. Inspired by TalkNet [33], we augment the audio data by negative sampling. For each audio signal in a batch, we randomly select another audio sample from the whole training dataset and add it after decreasing its volume by a random factor. This technique can effectively increase the amount of training samples for the feature extractor by selecting negative samples from the whole training dataset.

Spatial feature. The visual features encoded by the 2D ResNet do not have any information about where each face is spatially located in each frame because we only use the cropped face regions in the visual feature encoding. Here, we argue that the spatial locations of speakers can be another type of inductive bias. In order to exploit the spatial information of each face-crop, we incorporate the spatial features corresponding to each face as additional input to the node feature as follows: We project the 4-D spatial feature of each face region parameterized by the normalized center location, height and width (x, y,

Table 1. Performance comparisons with other state-of-the-art methods on the validation set of AVA-ActiveSpeaker datset [26]. We report mAP (mean average precision). SPELL outperforms all the previous approaches. 3D Conv denotes an additional use of one or more 3D convolutional layers. Note that TSM [19] does not increase memory usage nor the computation cost.

Method	**Feature encoding network**	**mAP(%)**
Roth *et al.* [26]	MobileNet [14]	79.2
Zhang *et al.* [36]	3D ResNet-18 [29] + VGG-M [5]	84.0
MAAS-LAN [18]	2D ResNet-18 [13]	85.1
Chung *et al.* [6]	VGG-M [5] + 3D Conv	85.5
ASC [2]	2D ResNet-18 [13]	87.1
MAAS-TAN [18]	2D ResNet-18 [13]	88.8
UniCon [37]	2D ResNet-18 [13]	92.0
TalkNet [33]	2D ResNet-18/34 [13] + 3D Conv	92.3
ASDNet [17]	3D ResNeXt-18 [16] + SincDSNet [25]	93.5
SPELL (Ours)	2D ResNet-18-TSM [13,19]	**94.2**
SPELL+ (Ours)	2D ResNet-50-TSM [13,19]	**94.9**

h, w) to a 64-D feature vector using a single fully-connected layer. The resulting spatial feature vector is then concatenated to the visual feature at each node.

4 Experiments

We perform experiments on the large-scale AVA-ActiveSpeaker dataset [26]. Derived from Hollywood movies, this dataset comes with a number of face tracks for active and inactive speakers and their audio signals. Its extensive annotations of the face tracks is a key feature that was missing in its predecessors.

Implementation details. Following ASC [26], we utilize a two-stream network with a ResNet [13] backbone for the audio-visual feature encoder. In the training process, we perform visual augmentation including horizontal flipping, color jittering, and scaling and audio augmentation as described in Section 3.5. We extract the encoded audio, visual, and spatial features for each face-crop to make the node feature. For the visual features, we use a stack of 11 consecutive face-crops (resolution: 144×144). We implement SPELL using PyTorch Geometric library [10]. Our model consists of three GCN layers, each with 64 dimensional filters. The first layer is implemented by an EDGE-CONV layer that uses a two-layer MLP for feature projection. The second and third GCN layers are of type SAGE-CONV and each of them uses a single MLP layer. We set the number of nodes n to 2000 and τ parameter to 0.9, which ensures that each graph fully spans each of the face tracks. We train SPELL with a batch size of 16 using the Adam optimizer [15]. The learning rate starts at 5×10^{-3} and decays following the cosine annealing schedule [20]. The whole training process of 70 epochs takes less than an hour using a single GPU (TITAN V).

Table 2. Performance comparison of context-reasoning with state-of-the-art methods. SPELL without TSM [19] demonstrates the higher context-reasoning capacity of our method when compared to the other 2D CNN-based approaches.

Method	**Stage-1 mAP**	**Final mAP**	**ΔmAP**
MAAS-LAN [18]	79.5	85.1	5.6
ASC [2]	79.5	87.1	7.6
MAAS-TAN [18]	80.2	88.8	8.6
Unicon [37]	84.0	92.0	8.0
ASDNet [17]	**88.9**	93.5	4.6
SPELL (Ours)	88.0	**94.2**	6.2
SPELL (Ours) w/o TSM	82.6	92.0	**9.4**

4.1 Comparison with the state-of-the-art

We summarize the performance comparisons of SPELL with other state-of-the-art approaches on the validation set of the AVA-ActiveSpeaker dataset [26] in Table 1. We want to point out that SPELL significantly outperforms all the previous approaches using the two-stream 2D ResNet-18 [13]. Critically, SPELL's visual feature encoding has significantly lower computational and memory overhead (0.7 GFLOPs and 11.2M parameters) compared to ASDNet [17] (13.2 GFLOPs, 48.6M #Params), the leading state-of-the-art method. A concurrent and closely related work MAAS [18] also uses a GNN-based framework. MAAS-LAN uses a graph that is generated on a short video clip. To improve the detection performance, MAAS-TAN extends MAAS-LAN by connecting the graphs over time, which makes 13 temporally-linked graph spanning about 1.59 seconds. This time span is relatively shorter than SPELL since the SPELL graph spans around 13-55 seconds, as explained in the next subsection. In addition, SPELL requires a single forward pass when MAAS performs multiple forward passes for each inference process.

4.2 Context-reasoning capacity

Most of the previous approaches have multi-stage frameworks, which includes a feature-encoding stage for audio-visual feature extraction that is followed by one or more context-reasoning stages for modeling long-term interactions and the context information. For example, SPELL has a single context-reasoning stage that uses a three-layer *Bi-dir* GNN for modeling long-term spatial-temporal information. In Table 2, we compare the performance of context-reasoning stages with previous methods. Specifically, we analyze the detection performance when using only the feature-encoding stage (Stage-1 mAP) and the final performance. The difference between the two scores can provide a good insight on the capacity of the context-reasoning modules. Because ASDNet [17] uses 3D CNNs, it is likely that some degree of the temporal context is already incorporated in the feature-encoding stage, which leads to a low context-reasoning performance. Sim-

Table 3. Complexity comparisons of the context-reasoning stage. SPELL achieves the best performance while requiring the lowest memory and computation consumption.

Method	#Params(M)	Size(MB)	mAP(%)
ASC [2]	1.13	4.32	87.1
MAAS-TAN [18]	0.16	0.63	88.8
ASDNet [17]	2.56	9.77	93.5
SPELL (Ours)	0.11	0.45	94.2

ilarly, using TSM [19] provides the short-term context information in the feature-encoding stage, which leads to a smaller score difference between the Stage-1 and Final mAP and thus underestimates the context-reasoning capacity. Therefore, we also estimate the performance of SPELL without TSM. In this case, the context-reasoning performance of SPELL outperforms all the other methods, which shows the higher context-reasoning capacity of our method, thanks to the longer-term context modeling. Note that although ASC [2], MAAS [18], Unicon [37], and SPELL use the same 2D ResNet-18 [13], their Stage-1 mAP can be different due to the inconsistency of input resolution, number of face-crops, and training scheme.

Long-term temporal context. Note that τ (= 0.9 second in our experiments) in SPELL imposes *additional* constraint on direct connectivity across temporally distant nodes. The face identities across consecutive time-stamps are always connected. Below is the estimate of the effective temporal context size of SPELL. AVA-ActiveSpeaker dataset contains 3.65 million frames and 5.3 million annotated faces, resulting into 1.45 faces per frame. With an average of 1.45 faces per frame, a graph with 500 to 2000 faces in sorted temporal order spans over 345 to 1379 frames which correspond to 13 to 55 seconds for a 25-fps video. In other words, the nodes in the graph might have a time-difference of about 1 minute, and SPELL is able to reason over that long-term temporal window within a limited memory and compute budget, thanks to the effectiveness of the proposed graph structure. It is note worthy that the temporal window size in MAAS [18] is 1.9 seconds and TalkNet [33] uses up to 4 seconds as long-term sequence-level temporal context.

4.3 Efficiency of the context-reasoning stage

In Table 3, we compare the complexity of the context-reasoning stage of SPELL with ASC [2], MAAS-TAN [18], and ASDNet [17]. These methods release the source code for their models, so we use the official code to compute the number of parameters and the model size of the context-reasoning stage. ASC has about 10 times more parameters and model size than ours. Nevertheless, SPELL achieves 7.1% higher mAP than ASC. SPELL has fewer number of parameters than MAAS-TAN even while achieving 5.4% higher mAP. When compared to the leading state-of-the-art method, ASDNet, SPELL is one order of magnitude more computationally efficient.

Table 4. Performance comparisons of different ablative settings: TSM [19], Cx-reason (context-reasoning only with an undirected graph), *Bi-dir* (augmenting with forward/backward graphs), Audio-aug (audio data augmentation), Sp-feat (spatial features).

TSM	Cx-reason	*Bi-dir*	Audio-aug	Sp-feat	mAP(%)
-	-	-	-	-	80.2
-	-	-	✓	-	82.6
✓	-	-	✓	-	88.0
✓	✓	-	✓	-	92.4
✓	✓	✓	✓	-	93.9
✓	✓	✓	✓	✓	94.2

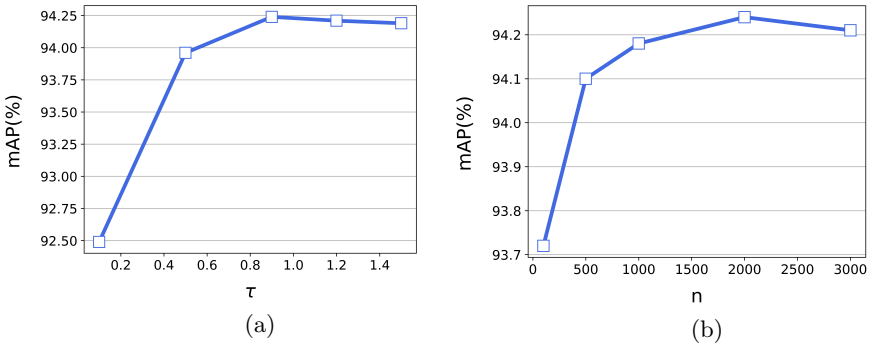

Fig. 4. Study on the impact of two hyperparameters, which are τ (when n is set to 2000) and n (when τ is fixed at 0.9).

4.4 Ablation study

We perform an ablative study to validate the contributions of individual components, namely TSM, Cx-reason (context-reasoning only with an undirected graph), *Bi-dir* (augmenting context-reasoning with the forward/backward graphs), Audio-aug (audio data augmentation), and Sp-feat (spatial features). We summarize the main contributions in Table 4. We can observe that TSM, *Bi-dir* graph structure, and audio data augmentation play significant roles in boosting the detection performance. This implies that 1) retaining short-term information in the feature-encoding stage is important, 2) processing the spatial-temporal information using our graph structure is effective, and 3) the negatively sampled audio makes our model more robust to the noise. Additionally, the spatial features also bring meaningful performance gain.

In addition, we analyze the impact of two hyperparameters: τ (Section 3.2) and the number of nodes in a graph embedding process. τ controls the connectivity or the edge density in the graph construction. Specifically, larger values for τ allow us to model longer temporal correlations but increases the average degree of the nodes, thus making the system more computationally expensive.

Table 5. Comparisons of the detection performance and the model size with different filter dimensions.

Filter Dim	#Params(M)	Size(MB)	mAP(%)
16	0.02	0.10	93.5
32	0.05	0.21	93.9
64	0.11	0.45	94.2
128	0.29	1.14	94.1
256	0.88	3.38	94.1

In Figure 4, we can observe that connecting too distant face-crops deteriorates the detection performance. One potential reason behind this could be that the aggregation procedure becomes too smooth due to the high degree of connectivity. Interestingly, we also found that a larger number of nodes does not always lead to higher performance. This might be because after a certain point, larger number of nodes leads to over-fitting.

We perform additional experiments with different filter dimensions of the EDGE-CONV and SAGE-CONV. In Table 5, we show how the detection performance and the model size change depending on the filter dimension. We can observe that increasing the filter dimension above 64 does not bring any performance gain when the model size increases significantly.

We also perform an ablation study of the input modalities. When using only the visual features, the detection performance drops significantly from 94.2% to 84.9% mAP (when using only the audio: 55.6%), which shows that both the audio and video modalities are important for this application.

4.5 Qualitative analysis

In the supplementary material, we show several detection examples to provide a qualitative analysis. The selected frames have multiple faces and have a long time-span about 5-10 seconds. In all of the provided examples in the supplementary material, SPELL correctly classifies all the speakers when the counterpart fails to do. The qualitative analysis demonstrates that SPELL is effective and that it is good at modeling spatial-temporal long-term information.

5 Conclusion

We proposed SPELL - an effective graph-based approach to active speaker detection in videos. The main idea is to capture the long-term spatial and temporal relationships among the cropped faces through a graph structure that is aware of the temporal order of the faces. SPELL outperforms all the previous approaches and requires significantly less hardware resources when compared to the leading state-of-the-art method. The model we propose is also generic - it can be used to address other video understanding tasks such as action localization and audio source localization.

References

1. Afouras, T., Chung, J.S., Zisserman, A.: The conversation: Deep audio-visual speech enhancement. arXiv preprint arXiv:1804.04121 (2018)
2. Alcazar, J.L., Heilbron, F.C., Mai, L., Perazzi, F., Lee, J.Y., Arbelaez, P., Ghanem, B.: Active Speakers in Context. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. p. 12465–12474 (2020)
3. Arnab, A., Sun, C., Schmid, C.: Unified graph structured models for video understanding. arXiv preprint arXiv:2103.15662 (2021)
4. Chakravarty, P., Tuytelaars, T.: Cross-modal supervision for learning active speaker detection in video. ArXiv **abs/1603.08907** (2016)
5. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: Proceedings of the British Machine Vision Conference. BMVA Press (2014)
6. Chung, J.S.: Naver at activitynet challenge 2019 - task B active speaker detection (AVA). CoRR **abs/1906.10555** (2019), http://arxiv.org/abs/1906.10555
7. Cutler, R., Davis, L.: Look who's talking: Speaker detection using video and audio correlation. In: 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532). vol. 3, pp. 1589–1592. IEEE (2000)
8. Everingham, M., Sivic, J., Zisserman, A.: Hello! my name is... buffy"–automatic naming of characters in tv video. In: BMVC. vol. 2, p. 6 (2006)
9. Everingham, M., Sivic, J., Zisserman, A.: Taking the bite out of automated naming of characters in tv video. Image and Vision Computing **27**, 545–559 (2009)
10. Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019), http://arxiv.org/abs/1903.02428
11. Geng, S., Gao, P., Hori, C., Le Roux, J., Cherian, A.: Spatio-temporal scene graphs for video dialog. arXiv e-prints pp. arXiv–2007 (2020)
12. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
16. Kopuklu, O., Kose, N., Gunduz, A., Rigoll, G.: Resource efficient 3d convolutional neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
17. Köpüklü, O., Taseska, M., Rigoll, G.: How to Design a Three-Stage Architecture for Audio-Visual Active Speaker Detection in the Wild. In: Proc. Internal Conference on Computer Vision (Jun 2021)
18. León-Alcázar, J., Heilbron, F.C., Thabet, A., Ghanem, B.: MAAS: Multi-modal Assignation for Active Speaker Detection. In: Internal Conference on Computer Vision (2021)
19. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7083–7093 (2019)

20. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (ICLR) (2017)
21. Mi, L., Ou, Y., Chen, Z.: Visual relationship forecasting in videos. arXiv preprint arXiv:2107.01181 (2021)
22. Min, K., Roy, S., Tripathi, S., Guha, T., Majumdar, S.: Intel labs at activitynet challenge 2022: Spell for long-term active speaker detection (2022)
23. Nagarajan, T., Li, Y., Feichtenhofer, C., Grauman, K.: Ego-topo: Environment affordances from egocentric video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 163–172 (2020)
24. Patrick, M., Asano, Y.M., Huang, B., Misra, I., Metze, F., Henriques, J., Vedaldi, A.: Space-time crop & attend: Improving cross-modal video representation learning. arXiv preprint arXiv:2103.10211 (2021)
25. Ravanelli, M., Bengio, Y.: Speaker recognition from raw waveform with sincnet. In: 2018 IEEE Spoken Language Technology Workshop (SLT). pp. 1021–1028. IEEE (2018)
26. Roth, J., Chaudhuri, S., Klejch, O., Marvin, R., Gallagher, A., Kaver, L., Ramaswamy, S., Stopczynski, A., Schmid, C., Xi, Z., et al.: Ava active speaker: An audio-visual dataset for active speaker detection. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4492–4496. IEEE (2020)
27. Sharma, R., Somandepalli, K., Narayanan, S.: Crossmodal learning for audio-visual speech event localization. arXiv preprint arXiv:2003.04358 (2020)
28. Shirian, A., Tripathi, S., Guha, T.: Learnable graph inception network for emotion recognition. IEEE Transactions on Multimedia (2020)
29. Stafylakis, T., Tzimiropoulos, G.: Combining residual networks with lstms for lipreading. arXiv preprint arXiv:1703.04105 (2017)
30. Stefanov, K., Beskow, J., Salvi, G.: Vision-based active speaker detection in multiparty interaction. In: Grounding Language Understanding GLU2017 August 25, 2017, KTH Royal Institute of Technology, Stockholm, Sweden (2017)
31. Stefanov, K., Sugimoto, A., Beskow, J.: Look who's talking: visual identification of the active speaker in multi-party human-robot interaction. In: Proceedings of the 2nd Workshop on Advancements in Social Signal Processing for Multimodal Interaction. pp. 22–27 (2016)
32. Tan, R., Xu, H., Saenko, K., Plummer, B.A.: Logan: Latent graph co-attention network for weakly-supervised video moment retrieval. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2083–2092 (2021)
33. Tao, R., Pan, Z., Das, R.K., Qian, X., Shou, M.Z., Li, H.: Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3927–3935 (2021)
34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
35. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
36. Zhang, Y.H., Xiao, J., Yang, S., Shan, S.: Multi-task learning for audio-visual active speaker detection. The ActivityNet Large-Scale Activity Recognition Challenge pp. 1–4 (2019)

37. Zhang, Y., Liang, S., Yang, S., Liu, X., Wu, Z., Shan, S., Chen, X.: UniCon: Unified Context Network for Robust Active Speaker Detection, p. 3964–3972. Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3474085.3475275
38. Zhang, Y., Tokmakov, P., Hebert, M., Schmid, C.: A structured model for action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9975–9984 (2019)