Intel Labs at ActivityNet Challenge 2022: SPELL for Long-Term Active Speaker Detection

Kyle Min[1], Sourya Roy[2†], Subarna Tripathi[1], Tanaya Guha[3], Somdeb Majumdar[1]
[1]Intel Labs, [2]UC Riverside, [3]University of Glasgow
{kyle.min,subarna.tripathi,somdeb.majumdar}@intel.com

Abstract

In this report, we describe SPELL, a novel spatial-temporal graph learning framework for active speaker detection (ASD). First, each person in a video frame is encoded in a unique node for that frame. The nodes corresponding to each 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 with low computation cost.

1. Introduction

Active speaker detection (ASD) is a multimodal (audio-visual) task where the goal is to identify which persons are speaking in each frame given a video. It has numerous practical applications ranging from speech enhancement systems [1] to human-robot interaction [14, 13].

Most of the previous state-of-the-art approaches [2, 15, 9, 8] address the task by first encoding visual and audio features from videos, and then by classifying the fused multimodal features. However, recent methods have relied on complex architectures for processing the audio-visual features with high computation and memory overheads. For example, TalkNet [15] suggests using a transformer-style architecture [16] to model the cross-modal information from the audio-visual input. ASDNet [8] 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 report, we propose an efficient graph-based framework, which we call SPELL (**Sp**atial-**T**empora**l** Graph **L**earning). Figure 1 illustrates an overview of our framework. 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. Next, we perform binary node classification – active or inactive speaker – on this graph by learning

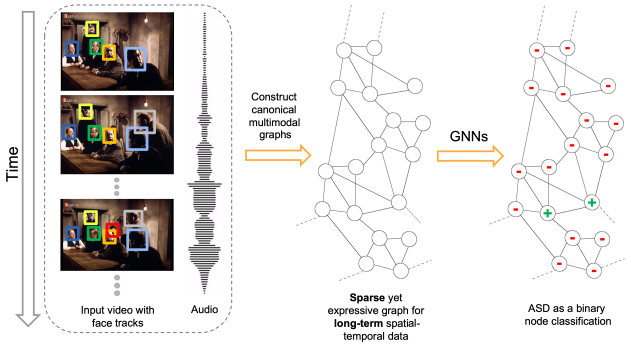

Figure 1: SPELL converts a video into a canonical graph from the audio-visual input data, where each node corresponds to a person in a frame, and an edge represents a spatial or temporal interaction between the nodes. The constructed graph is dense enough for modeling long-term dependencies through message passing across the temporally-distant nodes, yet sparse enough to be processed within low memory and computation budget.

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 [10] - 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.

[†]Work partially done during an internship at Intel Labs

2. Method

2.1. 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 $f_i : V \rightarrow \mathbb{R}^m$ (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}. We use two different types of aggregation methods, which are SAGE-CONV [5] and EDGE-CONV [17].

2.2. 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. 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 [6] architecture as in [12, 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.

2.3. ASD as a node classification task

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 posed 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.

2.4. 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. Specifically, 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. For the remaining parts of this report, we will refer to this network that is augmented with the foward/backward graphs as *Bi-directional* or *Bi-dir* for short.

2.5. Feature learning

Similar to ASC [2], we use a two-stream 2D ResNet [6] architecture for the audio-visual feature encoding. The networks take as visual input 11 consecutive face-crops (144×144) 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 [16], 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 [10], 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. To make our method robust to noise, we make use of data augmentation methods while training the feature extractor. Inspired by TalkNet [15], 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.

Spatial feature. 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, 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.

3. Experiments

Implementation details. Following ASC [12], we utilize a two-stream network with a ResNet-18 [6] 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 2.5. We extract the encoded audio, visual, and spatial features for each face-crop to make the node feature. For SPELL, we implement it using PyTorch Geometric library [4]. 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 [7]. The learning rate starts at 5×10^{-4} and decays following the cosine annealing schedule [11]. The whole training process of 120 epochs takes less than two hours using a single GPU (TITAN V).

3.1. Comparison with the state-of-the-art

We summarize the performance comparisons of SPELL with other state-of-the-art approaches on the AVA-ActiveSpeaker dataset [12] in Table 1. We want to point out that SPELL significantly outperforms all the previous approaches using the two-stream 2D ResNet-18 [6]. Critically, SPELL's visual feature encoding has significantly lower computational and memory overhead (0.7 GFLOPs and 11.2M parameters) compared to ASDNet [8] (13.2 GFLOPs, 48.6M #Params),

Method	val mAP(%)	test mAP(%)
Roth *et al.* [12]	79.2	82.1
Zhang *et al.* [18]	84.0	83.5
Chung *et al.* [3]	87.8	87.8
ASC [2]	87.1	86.7
MAAS-TAN [9]	88.8	-
TalkNet [15]	92.3	90.8
ASDNet [8]	93.5	91.7
SPELL (Ours)	**94.2**	-
SPELL+ (Ours)	**95.3**	**93.2**

Table 1: Performance comparisons with other state-of-the-art methods on the AVA-ActiveSpeaker dataset [12]. We report mAP (mean average precision).

the leading state-of-the-art method. A concurrent and closely related work MAAS [9] 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 discussion section. In addition, SPELL requires a single forward pass when MAAS performs multiple forward passes for each inference process. For the challenge, we use the features encoded by a two-stream ResNet-50 to boost the performance (SPELL+). Here, the visual encoder is augmented with TSM and takes as input 23 consecutive face-crops.

4. Discussion

Long-term temporal context. Here, we estimate the effective temporal context span of SPELL. AVA-ActiveSpeaker dataset contains 5.3 million frames and 3.65 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 [9] is 1.9 seconds and TalkNet [15] uses up to 4 seconds as long-term sequence-level temporal context.

Conclusion. We have presented an effective graph-based approach for active speaker detection in videos. The main idea is to capture the long-term spatial and temporal relationships among the face-crops through a graph structure that is aware of temporal orders of them. SPELL is generic; it can be used to address other video understanding tasks such as action localization.

References

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. *arXiv preprint arXiv:1804.04121*, 2018.

[2] Juan Leon Alcazar, Fabian Caba Heilbron, Long Mai, Federico Perazzi, Joon-Young Lee, Pablo Arbelaez, and Bernard Ghanem. Active Speakers in Context. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, page 12465–12474, 2020.

[3] Joon Son Chung. Naver at activitynet challenge 2019 - task B active speaker detection (AVA). *CoRR*, abs/1906.10555, 2019.

[4] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[5] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Okan Köpüklü, Maja Taseska, and Gerhard Rigoll. How to Design a Three-Stage Architecture for Audio-Visual Active Speaker Detection in the Wild. In *Proc. Internal Conference on Computer Vision*, June 2021.

[9] Juan León-Alcázar, Fabian Caba Heilbron, Ali Thabet, and Bernard Ghanem. MAAS: Multi-modal Assignation for Active Speaker Detection. In *Internal Conference on Computer Vision*, 2021.

[10] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[11] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

[12] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, 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)*, pages 4492–4496. IEEE, 2020.

[13] Kalin Stefanov, Jonas Beskow, and Giampiero Salvi. Vision-based active speaker detection in multiparty interaction. In *Grounding Language Understanding GLU2017 August 25, 2017, KTH Royal Institute of Technology, Stockholm, Sweden*, 2017.

[14] Kalin Stefanov, Akihiro Sugimoto, and Jonas Beskow. 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*, pages 22–27, 2016.

[15] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3927–3935, 2021.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[17] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[18] Yuan-Hang Zhang, Jingyun Xiao, Shuang Yang, and Shiguang Shan. Multi-task learning for audio-visual active speaker detection. *The ActivityNet Large-Scale Activity Recognition Challenge*, pages 1–4, 2019.