INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Speaker Tracking using Graph Attention Networks with Varying Duration Utterances across Multi-Channel Naturalistic Data: Fearless Steps Apollo-11 Audio Corpus

Meena M. C. Shekar[1], *John H.L. Hansen*[1]

[1]Center of Robust Speech Systems,
The University of Texas at Dallas, US

meena.chandrashekar@utdallas.edu, john.hansen@utdallas.edu

Abstract

Speaker tracking in spontaneous naturalistic data continues to be a major research challenge, especially for short turn-taking communications. The NASA Apollo-11 space mission brought astronauts to the moon and back, where team based voice communications were captured. Building robust speaker classification models for this corpus has significant challenges due to variability of speaker turns, imbalanced speaker classes, and time-varying background noise/distortions. This study proposes a novel approach for speaker classification and tracking, utilizing a graph attention network framework that builds upon pre-trained speaker embeddings. The model's robustness is evaluated on a number of speakers (10-140), achieving classification accuracy of 90.78% for 10 speakers, and 79.86% for 140 speakers. Furthermore, a secondary investigation focused on tracking speakers-of-interest(SoI) during mission critical phases, essentially serves as a lasting tribute to the 'Heroes Behind the Heroes'

Index Terms: speaker classification, graph attention networks, speaker tracking, graph neural networks.

1. Introduction

Speaker tracking is the process of assigning an unknown speech utterance to one of known speakers in a set of target speakers, including following the speaker's voice over time in the course of an audio stream. The first step in speaker tracking involves classifying speakers and identifying the target speaker. The goal is to identify all speech segments uttered by the same speaker in an audio recording and assign unique labels. The second step involves tracking speaker of interest throughout the audio segment. For our work, we also identify roles of our speakers using their speech duration. To perform speaker tracking, good representations of data and features that reflect the semantic meaning are required to produce a robust model.

In our experiments, we employ CRSS-UTDallas Fearless Steps Apollo 11 audio corpus[1] consisting of +9k audio data (100 hr hand labelled) involving more than +400 personnel serving as mission specialists who communicate across 30 audio loops[1, 2]. Each channel reflects a single communications loop (channel) that can contain anywhere from 3-65 speakers over extended time periods. Due to strict NASA communication protocols in such time-critical missions, most personnel employed a compact speaking style, with information turn-taking over 3-5sec windows [3]. Furthermore, some speakers had less than 3 seconds of utterance duration. This poses a unique and challenging research problem of finding 'needles in a haystack' from a speaker tracking perspective [1, 4, 5].

[1]This work was supported by National Science Foundation under Grant Award number 2016725

Performing speaker classification on short duration utterances is challenging as such utterances do not contain enough contextual information to accurately classify the speaker. Previous work [6, 7] have explored building robust models for short duration utterance, by extracting speaker specific features. However, this requires a substantial amount of training data to perform well and they do not consider varying duration utterances. [8] proposes an approach that works on varying duration speech data by aggregating information across multiple utterances, although this system can handle varying duration speech data, it may not work well with short duration utteranced.

Graph Neural Networks (GNNs) have rapidly developed with powerful variants such as Graph Convolutional Network (GCN) [9], Graph Attention Network (GAT) [10], and GraphSAGE [11]. Despite their success, GNNs have not been used or studied often in the context of speaker classification or speaker tracking. [12] proposes a graph convolution network for speaker verification and uses attention mechanism to obtain speaker representations. However, this model requires large amounts of training data and does not consider short duration utterance. Hence, in our study, we propose using a variation of the Graph Attention Network (GAT) framework with a dynamic attention, which can handle varying duration utterance and works with a small training dataset, while remaining robust to noise.

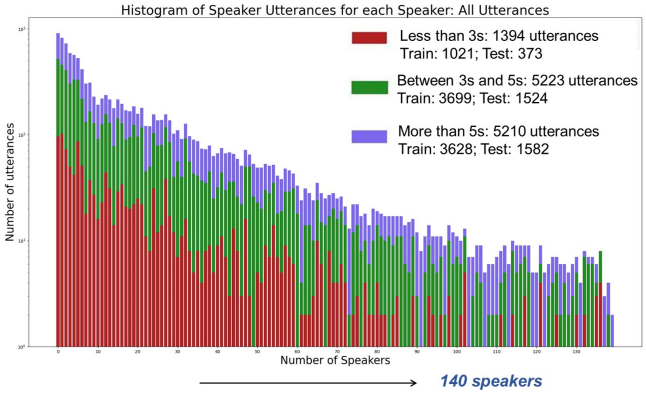

Figure 1: *Histogram of varying speaker utterances for 140 speakers*

The main contributions of this study are: [i] proposing a Graph Attention Network framework that utilizes dynamic attention and edge softmax to produce stronger representations for varying duration utterances. This formulation produces stronger representations that outperform baseline and other popular GNN models;[ii] comparing our models on a range of speakers, starting with 10 speakers expanding to 140 speak-

10.21437/Interspeech.2023-1258

ers that are selected based on the presence of atleast 3 speaker utterances;[iii] Using the concept of 'Finding Waldo' to identify and track key speakers of interest (SOI) :Flight Director (FD), Capsule Communicator (CAPCOM), Guidance, Navigation and, Control (GNC), Electrical, Environmental, and Consumables Manager (EECOM), and Network (NTWK) on three mission critical phases and compare their speech duration on different phases;[iv] Contributing to archiving and serving as a lasting tribute to the 'Heroes Behind the Heroes of Apollo', thus preserving the "words spoken in space".

2. Dataset Description

The UTDallas Fearless-Steps Apollo corpus comprises of 19,000 hours of audio, which presents unique and multiple challenges due to severe noise and degradation, as well as overlapping instances over 30 channels. For our work, we selected a subset of 100 hours [13, 14, 15], which were manually transcribed by professional annotators for speaker labels. The 100 hrs were obtained from three mission-critical events: Lift-Off (25 hours), Lunar-Landing (50 hours), and Lunar-Walking (25 hours).

Out of 30 channels, we selected five channels with the most speech activity over the selected events: Flight Director (FD), Mission Operations Control Room (MOCR), Guidance Navigation and Control (GNC), Network Controller (NTWK), and Electrical, Environmental, and Consumables Manager (EECOM).

Although the corpus contains 100 hours of audio data, the total amount of actual speech content is approximately 17 hours. As shown in Fig 1, speaker duration can range from 10 to 3000 seconds per speaker and each utterance duration from 1s-15s. Shorter duration utterances are particularly difficult to classify, recognize, and track as they do not provide sufficient contextual information to make accurate predictions about speaker labels. Notably, the dataset includes a substantial proportion of short test duration utterances (less than 5 seconds), as highlighted in Fig 1. For this work, we divided the 100 hours into training (70% of data) and test (30% of data) sets. The Fearless Steps dataset consists of 183 speakers; however, we considered a total of 140 speakers who have at least 3 utterances, with each utterance being at least 1 second long [16].

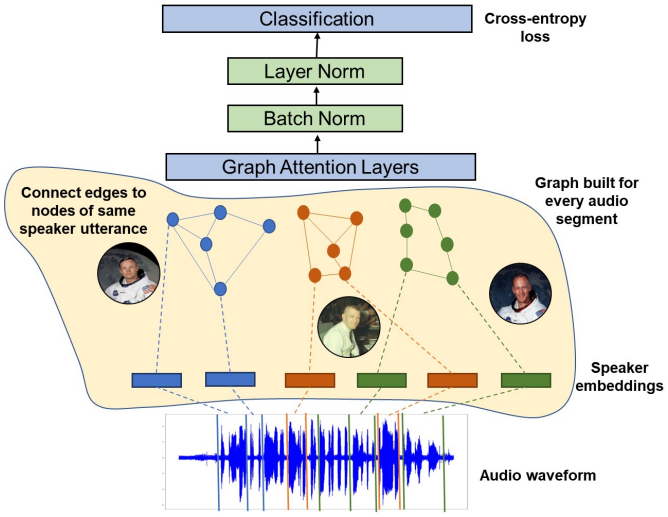

Figure 2: *Overview of the proposed GAT method*

3. Baseline Systems

3.1. i-Vector

This system is based on a Gausian Mixture Model-Universal background Model (GMM-UBM) system [17, 18], which serves as the acoustic-feature system. Here, the UBM model is trained on the NIST SRE 16 corpus to create a 2048 component full-covariance GMM. A 600 dim i-Vector speaker embedding is developed and extracted.

3.2. x-Vector

To extract x-Vectors, a feed-forward Deep Neural Network (DNN) computes the speaker embeddings from variable-length acoustic segments [19, 20]. The DNN embeddings are trained on the SRE16 dataset and extracted x-Vectors are 512 dim vectors. The Kaldi speech recognition toolkit [21] was used to train both i-Vectors and x-Vectors.

3.3. ECAPA TDNN

Emphasized Channel Attention, Propagation and Aggregation (ECAPA) Time Delay Neural Networks is a deep learning architecture which combines time-delay neural networks (TDNNs) and convolutional neural networks (CNNs). The speaker embeddings are extracted from the output of the bottleneck layer resulting in a 192 dim vector. The system is pretrained on Voxceleb1+Voxceleb2 training data. The embeddings are extracted using attentive statistical pooling [22, 23].

4. Graph structure

A graph is defined by its node set $V = \{v_1, ...v_n\}$ and edge set $E \subseteq \{(v_i, v_j) \mid v_i, v_j \in V\}$ where $(i, j) \in E$ denotes an edge from node j to node i. Each node has a node feature vector and associated with other nodes by it's edges. A message exchange is performed at each round where a node sends messages to its neighbors, and aggregates incoming messages from its neighbors through a message function f(.). Each graph has a unique message passing function and aggregation function AGG(.) [24].

For our experiments, we consider popular GNN variants such as the Graph Convolutional Network (GCN) [9], Graph Attention Network (GAT) [10], Graph Neural Network with convolutional auto-regressive moving average filters (ARMA) [25], and GraphSAGE [11]. These four GNNs are all considered.

4.1. GAT

Graph Attention Networks (GAT) use attention mechanisms to model the interaction between nodes in a graph. To learn the node representation for each node, attention coefficients between pairs of nodes are computed to weigh the contribution of each neighbor representation of the target node. The output of the final layer are a set of new features for each node [10]. The propagation function can be defined as:

$$\mathbf{h}_i' = \sum_{j \in \mathcal{N}i} \alpha_{ij}.\mathbf{W}\mathbf{h}_j \tag{1}$$

where $\mathcal{N}_i$ is the set of neighboring nodes of node i, and $\mathbf{h}_i'$ is the updated feature representation of node i. Let $\mathbf{h}_i \in \mathbb{R}^F$ be the input features of node i and $\mathbf{h}_j \in \mathbb{R}^F$ be the input features of a neighboring node j. The normalized attention coefficients are used to compute a linear combination of the features corresponding to them. To serve as the final output features for every

		Number of Speakers				
Speaker Embedding	**Classifier**	***10***	***20***	***50***	***100***	***140***
i-Vector	cosine distance	69.19%	62.28%	55.40%	50.17%	47.84%
i-Vector	GCN	79.40%	78.71%	75.02%	73.81%	69.05%
i-Vector	GAT	80.56%	80.44%	75.94%	74.72%	70.98%
i-Vector	ARMA	82.18%	79.66%	75.19%	72.98%	68.48%
i-Vector	GraphSAGE	73.89%	67.04%	64.38%	60.83%	59.05%
i-Vector	***Ours***	***87.14%***	***83.43%***	***79.08%***	***76.36%***	***74.20%***
x-Vector	cosine distance	72.98%	65.06%	60.52%	54.50%	52.35%
x-Vector	GCN	83.58%	81.31%	80.11%	78.90%	74.22%
x-Vector	GAT	82.72%	80.15%	79.69%	77.44%	75.87%
x-Vector	ARMA	81.55%	79.54%	75.37%	73.83%	73.61%
x-Vector	GraphSAGE	77.13%	71.71%	67.88%	64.38%	63.13%
x-Vector	***Ours***	***86.10%***	***85.34%***	***82.98%***	***80.44%***	***79.12%***
ECAPA	cosine distance	66.25%	62.28%	58.48%	51.78%	48.38%
ECAPA	GCN	87.36%	85.62%	80.41%	75.21%	72.59%
ECAPA	GAT	88.60%	86.86%	81.94%	79.01%	77.38%
ECAPA	ARMA	87.01%	85.10%	84.16%	79.70%	75.76%
ECAPA	GraphSAGE	81.30%	72.82%	71.10%	66.50%	64.67%
ECAPA	**Ours**	**90.74%**	**88.62%**	**86.00%**	**81.64%**	**79.86%**

Table 1: *Speaker classification accuracy on a range of speakers using several frameworks*

node, the attention mechanism is therefore defined as:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\overrightarrow{\mathbf{a}}^T\left[\mathbf{W}\boldsymbol{h}_i||\mathbf{W}\boldsymbol{h}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\overrightarrow{\mathbf{a}}^T\left[\mathbf{W}\boldsymbol{h}_i||\mathbf{W}\boldsymbol{h}_k\right]\right)\right)} \tag{2}$$

where $\mathbf{a} \in \mathbb{R}^{2F}$ is a trainable attention parameter vector, $\mathbf{W} \in \mathbb{R}^{F\times F'}$ is a trainable weight matrix, $||$ denotes concatenation, and LeakyReLU is a non-linearity.

4.2. Proposed framework

The motivation for GAT is to compute a representation for every node as a weighted average of its neighbors. However, GAT is severely constrained, because it can only calculate the static attention. This means that the attention function always weighs one key at least as much as any other key, regardless of the query. This limitation can be problematic when attempting to fit to the available training data, as the model may not be able to focus on the most relevant inputs. Furthermore, our corpus consists of varying duration utterance where it may be required to assign alternate weights to different duration utterances. To address varying duration utterances, we make use of an edge softmax function; a normalization function that converts edge weights into a probability distribution, allowing all nodes to contribute to the representation, albeit with different weights. The attention score e_{ij} on the nodes can then be computed as:

$$e_{ij} = a(\mathbf{W}\boldsymbol{h}_i||\mathbf{W}\boldsymbol{h}_j) \tag{3}$$

This equation indicates the importance of node j's features to node i. e_{ij} is only computed for nodes $j \in N_i$, where N_i is some neighborhood of node i in the graph. Given a set of edge weights, the softmax function normalizes the weights such that they add up to one and is given as:

$$\text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}i}\exp(e_{ik})} \tag{4}$$

where $\text{softmax}(e_{ij})$ is the normalized weight for the edge connecting node i and node j, and e_{ij} is the unnormalized weight for that edge. The edge softmax function can make use of contextual information from both shorter and longer duration utterances by effectively learning the underlying graph structure, and thereby capturing important patterns in the data. This helps to prevent over-emphasizing longer utterance duration, by giving appropriate weights to all nodes in the graph. To resolve the static attention problem, our framework will be using dynamic attention in GAT [26]. To create a dynamic graph attention network, the order of the internal operators in the attention coefficient function is modified and given as:

$$\alpha_{ij} = \frac{\exp\left(\overrightarrow{\mathbf{a}}^T\,\text{LeakyReLU}\left(\left[\mathbf{W}\boldsymbol{h}_i||\mathbf{W}\boldsymbol{h}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\overrightarrow{\mathbf{a}}^T\,\text{LeakyReLU}\left(\left[\mathbf{W}\boldsymbol{h}_i||\mathbf{W}\boldsymbol{h}_k\right]\right)\right)} \tag{5}$$

The proposed modification in this study significantly enhances the robustness of the GAT function by allowing it to decay noisy edges, which leads to better performance in the presence of edge noise. For our study, this modification is particularly useful since the nodes have varying duration as it can prioritize nodes with longer duration, while assigning lower attention scores to those with shorter duration. Using the edge softmax prevents assigning extremely low attention scores for the shorter duration nodes, in turn learning a better generalized node representation. In addition, we propose using normalization techniques such as batch normalization and layer normalization. Batch normalization [27] improves the stability of the attention weights by normalizing the nodes to each layer of the network, allowing the network to handle channel noise in the input data. Layer normalization [28] can also be used to normalize the outputs of the attention layer to help regularize the network and prevent overfitting. Overall, the above techniques are expected to help in improving the effectiveness, robustness, and stability of the model.

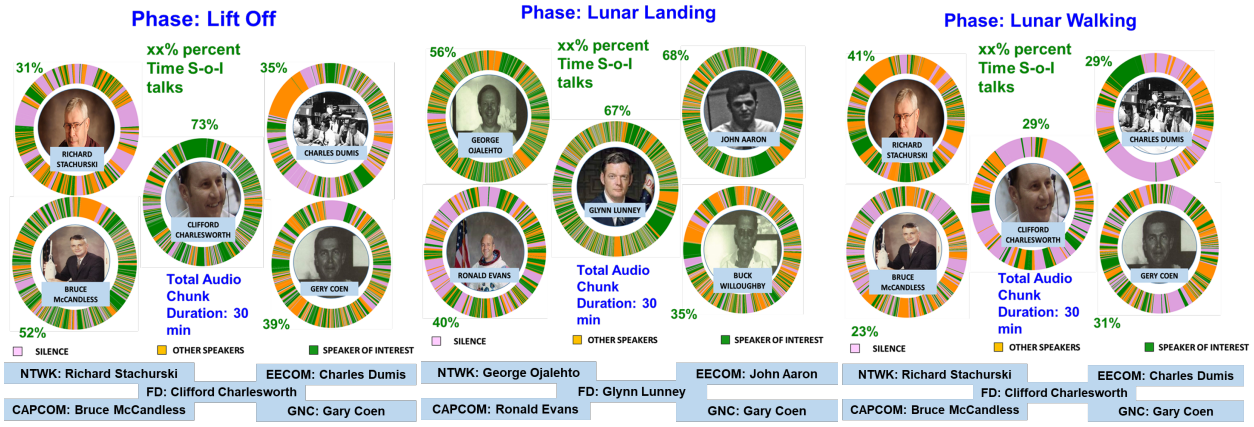

Figure 3: *Tracking Speaker of Interests on three critical phases of the mission: Apollo 11*

5. Implementation Details

For this study, a graph is built for every 30 minute audio segment, where each node represents a speaker utterance which can vary from 1s to 15s, meaning that every graph can have a variable number of nodes. The features of each node v_i is represented by extracted speaker embeddings denoted by F dimensions depending on the type of speaker embedding $x_i, i \in 1, 2, ..., N$ where N represents the number of nodes/speaker utterances. For our proposed framework, we construct a 2 layer GATv2 network using PyTorch Geometric [29]. After every GATv2 layer, we use a batch normalization and layer normalization. Each GATv2 layer has an edge softmax function. Finally, we use a cross entropy loss to perform classification on the speakers. Each network mentioned in Sec 4, including our proposed framework, is a 2 layer network that uses a cross entropy loss trained with an Adam optimizer with a learning rate of 0.0001 with a batch size of 13. Note that every graph will have a different number of nodes depending on the number of speaker utterances in that 30 minute audio segment.

5.1. Discussion

The performance of our proposed framework was evaluated with three different pretrained speaker embeddings: i-Vector, x-Vector, and ECAPA-TDNN. We compare our network with the baseline systems and other popular Graph Networks. Additionally, we test the networks against different number of speakers to determine its efficacy on small vs large amounts of data. Starting with 10 speakers who had the highest amount of speech data, we progressively increase the number of speakers to 140 speakers. For each case, our network framework outperforms the baseline models and the popular Graph models ranging from 90.74% speaker classification accuracy for 10 speakers to 79.86% speaker classification accuracy for 140 speakers. As the number of speakers increase, the classification accuracy drops. As seen in Fig 1, this corpus duration labels are highly imbalanced among speakers, with speech duration per speaker decreasing as we increase the number of speakers. This explains the drop in classification accuracy as we increase the number of speakers, since the training model does not have sufficient number of examples to accurately classify every test sample.

6. Tracking Speaker-of-Interest on Mission Phases

As noted, there are three mission critical phases: Lift Off(52 speakers, 25hrs), Lunar Landing(92 speakers, 50 hrs), and Lunar Walking(37 speakers, 25 hrs). The Green team was responsible for Lift Off phase and the Lunar Walking phase. Hence, we see the same speakers operating these phases in Fig 3. The donut plots are a new visualization tool to analyze primary speaker duration (speaker in the center, highlighted by green) vs other speakers (speakers that interact with primary speakers, highlighted by orange). For all three phases, we observe that Flight Director has a longer duration (73%, 67%, and 29% of primary speaker duration for Lift Off, Lunar Landing, and Lunar Walking) compared to his secondary speakers (includes multiple speakers) and other S-o-I. The CAPCOM was the only flight controller authorized to communicate directly with the spacecraft's crew. Therefore, the secondary speakers (i.e., astronauts) are more active in this channel than the CAPCOM (52%, 40%, and 23% of primary speaker duration for Lift Off, Lunar Landing, and Lunar Walking). With our speaker tracking, we assigned names to each speaker instead of using their speaker handle since each speaker handle represents 3 to 4 individuals, who worked in a day. This approach allows us to honor the "Heroes Behind the Heroes", and archive the speech of these individuals for posterity.

7. Conclusion

In this study, we proposed a solution to the problem of varying utterance duration in the Fearless Steps Apollo audio corpus by using a Graph Attention Network framework with dynamic attention. Furthermore, we assessed the performance of our proposed network on a range of speakers, from small to large and highly imbalanced duration and speaker count dataset, and demonstrate that it outperforms all baseline models and other popular Graph Networks. Additionally, we use the concept of 'Finding Waldo' to track and tag speaker-of-interest during three critical phases of the mission, thus providing recognition to individuals who contributed to the success of the mission. We analyze the speaker duration of primary and secondary speakers using donut plots, which reveal an intriguing global perspective of speaker interactions between NASA mission specialists. Ultimately, our analysis of key speakers-of-interest serves as a lasting tribute to the "Heroes Behind the Heroes of Apollo".

8. References

[1] A. Sangwan, L. Kaushik, C. Yu, J. H. L. Hansen, and D. W. Oard, "'houston, we have a solution': using nasa apollo program to advance speech and language processing technology." *ISCA INTERSPEECH*, pp. 1135–1139, 2013.

[2] A. Joglekar and J. H. Hansen, "Fearless steps challenge phase-1 evaluation plan," *arXiv preprint arXiv:2211.02051*, 2022.

[3] M. C. Shekar, *Knowledge Based Speaker Analysis Using a Massive Naturalistic Corpus: Fearless Steps Apollo-11*. The University of Texas at Dallas, 2020.

[4] A. Joglekar, S. O. Sadjadi, M. Chandra-Shekar, C. Cieri, and J. H. L. Hansen, "Fearless Steps Challenge Phase-3 (FSC P3): Advancing SLT for Unseen Channel and Mission Data Across NASA Apollo Audio," pp. 986–990, 2021.

[5] M. M. C. Shekar and J. H. Hansen, "Historical audio search and preservation: Finding waldo within the fearless steps apollo 11 naturalistic audio corpus," *IEEE Signal Processing Magazine*, vol. 40, no. 3, pp. 30–38, 2023.

[6] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," *IEEE ICASSP*, pp. 4052–4056, 2014.

[7] J. S. Chung and S. Lee, "Deep speaker embeddings for short-duration speaker verification," *IEEE Spoken Language Technology Workshop (SLT)*, pp. 31–36, 2018.

[8] L. Li, L. Wan, and J. McAuley, "Utterance-level aggregation for speaker recognition in the wild," *ACM on Multimedia Conference*, pp. 640–648, 2017.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *International Conference on Learning Representations*, 2017.

[10] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[11] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017.

[12] M. Hu, H. Wang, H. Yu, and B. Yang, "Graph convolutional networks with attention mechanism for speaker verification," *IEEE ICASSP*, pp. 6944–6948, 2021.

[13] J. Hansen, A. Joglekar, M. C. Shekhar, V. Kothapally, C. Yu, L. Kaushik, and A. Sangwan, "The 2019 inaugural fearless steps challenge: A giant leap for naturalistic audio," *ISCA INTERSPEECH*, pp. 1851–1855, 2019.

[14] A. Joglekar, J. H. Hansen, M. C. Shekar, and A. Sangwan, "FEARLESS STEPS Challenge (FS-2): Supervised Learning with Massive Naturalistic Apollo Data," *ISCA INTERSPEECH 2020*, pp. 2617–2621, 2020.

[15] J. H. Hansen, A. Joglekar, S.-J. Chen, M. C. Shekar, and C. Belitz, "Fearless steps apollo: Advanced naturalistic corpora development," in *LREC 2022*, 2022, pp. 14–19.

[16] L. Kaushik, A. Sangwan, and J. H. Hansen, "Multi-channel apollo mission speech transcripts calibration." *ISCA INTERSPEECH*, pp. 2799–2803, 2017.

[17] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[18] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans on Audio, Speech, and Lang Proc*, vol. 19, no. 4, pp. 788–798, 2010.

[19] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." *ISCA INTERSPEECH*, pp. 999–1003, 2017.

[20] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *IEEE ICASSP*, pp. 5329–5333, 2018.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," *IEEE ASRU*, Dec. 2011.

[22] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," *ISCA INTERSPEECH*, pp. 3830–3834, 2020.

[23] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[24] J. You, J. Leskovec, K. He, and S. Xie, "Graph structure of neural networks," *International Conference on Machine Learning*, pp. 10 881–10 891, 2020.

[25] F. M. Bianchi, D. Grattarola, and C. Alippi, "Graph neural networks with convolutional arma filters," *International Conference on Learning Representations*, 2019.

[26] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" in *International Conference on Learning Representations*.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International conference on machine learning*, pp. 448–456, 2015.

[28] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[29] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019.