SELF-ATTENTIVE AUTOMATIC SPEAKER IDENTIFICATION WITH DEFENDING ADVERSARIAL PERTURBATIONS

Ruirui Li[†] *Jyun-Yu Jiang*[§] *Xian Wu*[¶] *Chu-Cheng Hsieh*[†]

[†] Amazon [§]University of California Los Angeles [¶]University of Notre Dame
[†]ruirul@amazon.com, [§]jyunyu@cs.ucla.edu, [¶]xwu9@nd.edu, [†]chucheng@ucla.edu

ABSTRACT

Speaker identification that recognizes people from voices is one of the most fundamental problems in the field of speech processing. The identification technology enables versatile downstream applications, such as personalization and authentication. With the advent of deep learning, most of the state-of-the-art methods apply machine learning techniques and derive acoustic features from utterances with convolutional neural networks (CNNs) and recurrent neural networks (RNNs). However, there are two inherent limitations. First, complementary formant information, which expresses speaker identities, might not be fully captured by CNNs and RNNs, as they focus on local feature extraction and adjacent dependencies modeling, respectively. Second, complicated deep learning models can be extremely fragile and highly sensitive to subtle but intentional changes in the model parameters and inputs, also known as adversarial perturbations. To distill informative global acoustic information from utterances and defend potential adversarial perturbations, we propose a Self-Attentive Adversarial Speaker-Identification method (*SAASI*). We conduct experiments on the VCTK dataset and compare *SAASI* with four state-of-the-art baseline methods. The experimental results conclude that *SAASI* significantly outperforms all competitive baseline methods in recognizing both existing and new speakers.

Index Terms— Self-attention, adversarial perturbation, speaker identification

1. INTRODUCTION

Automatic speaker identification paves the way towards effective human-machine voice interactions and benefits versatile downstream applications. Smart speakers equipped with speaker identification, such as Amazon Echo and Google Home, own a tremendous user base of around 50 million users in US [1]. More invigoratingly, the user base is bound to skyrocket as more and more new users are ready to enjoy the convenience from flexible voice interactions. To encourage the user engagement and retention, how to accurately identify voiceprints of the users, which empowers them to jump in smoothly and deliver personalized services, becomes a daunting task.

Deep learning-based speaker identification methods have been gaining notable attraction as they outperformed the prevalent i-vector and GMM-UBM solutions [2, 3]. For example, Deep Speaker [4] and VGGVox [5] adopt CNN-based residual networks to learn voice acoustic representations based on utterance spectrograms while SincNet [6] applies CNNs to perform speaker identification from raw voice waveforms. GE2E [7] utilizes RNNs to model utterances so that the similarity between utterance representations for the same speaker can be maximized. SNL [8] further extends GE2E by introducing the dot-product attention mechanism, thereby obtaining more informative acoustic features for speaker identification. However, these conventional methods can suffer from capturing the complementary dependencies among frames in an utterance, leading to inferior formant and suboptimal acoustic representations.

Adversarial training, which minimizes the maximal risk for label-preserving input perturbations, has been proved to be effective to enhance the security and generalization of deep learning models [9–11]. [12, 13] apply domain adversarial training, but focuses on adapting a well trained speaker model to a new domain/language other than boosting the robustness of the model. [14] strives to enhance the robustness of speaker identification through multi-task learning, without defending adversarial perturbations explicitly. Therefore, it remains a challenge to defend adversarial attacks and enhance security and robustness in speaker identification.

To address the above challenges, we first leverage the self-attention mechanism [15, 16] to extract satisfactory acoustic features from utterances. Precisely, the self-attention mechanism allows us to fully utilize the complementary dependencies among frames and formants, resulting in informative global acoustic representations of utterances. Moreover, we craft dynamic perturbations at the embedding level to form effective adversarial examples. These adversarial examples are formed by applying small but deliberate perturbations to training utterances. The model is then trained in an adversarial manner, which not only learns from the original training data but also improves based on the dynamically constructed perturbed examples. As a result, adversarial training boosts the robustness of the speaker identification model, which is crucial for security-sensitive tasks.

In a nutshell, our proposed solution focuses on effective global acoustic feature extractions and adversarial perturbation defense in speaker identification. To achieve this goal,

we leverage the self-attention mechanism to extract acoustic features from utterances (Section 2.1). We generate dynamic adversarial examples, which serve as additional out-of-distribution training instances, and train a model with strong robustness to unseen data (Section 2.3). Our experiments demonstrate that our proposed solution, *SAASI* outperforms all baseline methods by a large margin (Section 3.2) even the utterances are only 1.5 seconds.

2. PROBLEM STATEMENT & METHODOLOGY

We formulate the objective of our work as the following. Suppose the system has a set of existing users with a few voice utterances for each user as training data. Given a test set of existing or new users, with a few short registered voice utterances for each user as enrollment, and another short testing voice utterance of a user within the test user set, the goal of this study is to recognize the speaker identity behind the testing voice utterance. In this work, we focus on text-independent speaker identification and presume that each utterance is very short [7, 17–20], for example, one to two seconds.

2.1. Self-Attentive Utterance Representation Learning

In this section, we discuss how we extract the acoustic features from an utterance and represent it into a fix-length vector. Each utterance u is first represented by a sequence of frames and each frame gives the frequency distribution during a particular short period of time. In this work, we use the spectrogram $\boldsymbol{SP}_u$ of a utterance u as the input and further learn the acoustic features of u. First, we aim at mining inter-relationships among all frames in an utterance, which allows us to comprehensively utilize all frames and fuse the all-inclusive formant information in the utterance. Second, we aggregate the fused frame embeddings of an utterance and summarize them into a fix-length embedding vector that expresses the acoustic information of the utterance.

As we mentioned above, each utterance u is expected to be first represented by a set of fused frame embeddings. A fused frame embedding encodes the acoustic information with attention to itself and the other frames in u. To achieve this, we develop a fusion module based on the self-attention mechanism. Formally,

$$\text{Self-Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q} \cdot \boldsymbol{K}^T}{\sqrt{d_Q}})\boldsymbol{V}, \tag{1}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent the attention query, key, and value matrices, respectively. The scale factor $\sqrt{d_Q}$ is used to avoid overly large values of the inner product, where d_Q is the feature dimension of $\boldsymbol{Q}$.

In our case, the self-attention operation takes the utterance spectrograms $\boldsymbol{SP}_u \in \mathbb{R}^{c \times d}$, where c is number of frames and d gives the dimension of a frame, as the inputs and feeds them into the self-attention layer to learn fused frame representations. Formally,

$$\tilde{\boldsymbol{E}}_u = \text{Self-Att}(\boldsymbol{SP}_u, \boldsymbol{SP}_u, \boldsymbol{SP}_u) \tag{2}$$

The self-attention result $\tilde{\boldsymbol{E}}_u$ learns the fused embeddings of frames by comparing the pairwise closeness between frames. Each fused frame embedding is a weighted sum of frame embedding of itself and other related frames, where each weight gauges the similarity between one frame and another one in u. In this way, $\tilde{\boldsymbol{E}}_u$ encodes the fused frame information, with each one frame explained by itself and others. In particular, $\tilde{\boldsymbol{E}}_u$ is good at modeling distant frame relationships, as no matter how distant two frames are, the longest possible path between them is one in the self-attention mechanism. The shorter the path between any combination of frames in an utterance, the easier to learn long-range dependencies. This allows the acoustic information, especially the formant information, in an utterance get fused and complement each other.

To incorporate the frame location information, we follow [15] and add sinusoidal positional embedding $\boldsymbol{E}_p$ into $\boldsymbol{SP}_u$ before fusion. Formally,

$$E_p(t, pos) = \begin{cases} \sin \frac{pos}{10000^{t/d}}, \text{if } t \text{ is even,} \\ \cos \frac{pos}{10000^{t/d}}, \text{if } t \text{ is odd,} \end{cases} \tag{3}$$

where pos is the position of a frame, d is the dimension of a frame, and $E_p(t, pos)$ gives the t-th element in the positional embedding of a frame, which is at position pos.

To increase the non-linearity of the self-attention mechanism, we further feed the fused frame embeddings $\tilde{\boldsymbol{E}}_u$ into a feed-forward neural network:

$$\tilde{\boldsymbol{E}}^f_u = \boldsymbol{W}^f_2 \cdot \text{ReLU}(\boldsymbol{W}^f_1 \cdot \tilde{\boldsymbol{E}}_u + \boldsymbol{b}^f_1) + \boldsymbol{b}^f_2, \tag{4}$$

where $\boldsymbol{W}^f_1$, $\boldsymbol{W}^f_2$, and $\boldsymbol{b}^f_2$, $\boldsymbol{b}^f_2$ are the weight matrices and bias in the feed-forward layer. To comprehensively fuse the frame information in an utterance, we perform the self-attention operations twice via residual shortcut connection [21].

To derive a summarized global acoustic representation of an utterance, we average $\tilde{\boldsymbol{E}}^f_u$ over the time dimension into one embedding vector, denoted as $\bar{\boldsymbol{E}}^f_u$. In addition, the summarized embedding vector is further L2 normalized. Formally, an utterance u is represented by a fix-length vector $\boldsymbol{E}_u$:

$$\boldsymbol{E}_u = \frac{\bar{\boldsymbol{E}}^f_u}{||\bar{\boldsymbol{E}}^f_u||_2}. \tag{5}$$

2.2. End-To-End Training

We follow [7, 8] and train the speaker identification model in an end-to-end manner. We construct a batch by $N \times M$ utterances, where N is the number of speakers and M is the number of utterances from each speaker. We use u_{ji} to represent the i-th utterance from speaker j. Moreover, we use $\boldsymbol{E}_{ji}$ to represent the embedding vector of the j-th speaker's i-th utterance. The acoustic biometry of speaker j is further represented by the embedding centroid $\boldsymbol{C}_j$ of his/her M utterances. Formally,

$$\boldsymbol{C}_j = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{E}_{jm} \tag{6}$$

The similarity matrix $\boldsymbol{S}_{ji,k}$ is defined as the scaled cosine similarities between each embedding vector $\boldsymbol{E}_{ji}$ to all centroids $\boldsymbol{C}_k$:

$$\boldsymbol{S}_{ji,k} = \boldsymbol{W}^s \cdot cos(\boldsymbol{E}_{ji}, \boldsymbol{C}_k) + \boldsymbol{b}^s, \tag{7}$$

where $\boldsymbol{W}^s$ and $\boldsymbol{b}^s$ are learnable parameters.

During training, the embedding of each utterance is expected to be similar to the centroid of all that speaker's embeddings, while at the same time, far from other speakers' centroids. The loss on each embedding vector $\boldsymbol{E}_{ji}$ is defined as:

$$L(\boldsymbol{E}_{ji}|\Theta) = -\boldsymbol{S}^{\Theta}_{ji,j} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\Theta}_{ji,k}), \tag{8}$$

where Θ is the model parameters. The loss function allows us to push each embedding vector close to its centroid and pull it away from all other centroids. The final end-to-end loss is the sum of all losses over all utterances involved in the similarity matrix.

$$L(\boldsymbol{S}|\Theta) = \sum_{j,i} L(\boldsymbol{E}_{ji}|\Theta) \tag{9}$$

2.3. Defending Adversarial Attacks

Adversarial attacks refer to techniques that fool models through malicious input with perturbations. To defend adversarial attacks and enhance the robustness, we enforce the model to perform consistently well even when the adversarial perturbations are presented. To achieve this goal, we additionally optimize the model to minimize the objective function with the perturbed parameters. Formally, we define the objective function with adversarial examples incorporated as:

$$\begin{aligned} L_{adv}(\boldsymbol{S}|\Theta) &= L(\boldsymbol{S}|\Theta) + \lambda L(\boldsymbol{S}|\Theta + \Delta_{adv}), \\ \text{where } \Delta_{adv} &= \arg \max_{\Delta, \|\Delta\| \leq \epsilon} L(\boldsymbol{S}|\hat{\Theta} + \Delta), \end{aligned} \tag{10}$$

where Δ denotes the perturbations on model parameters, $\epsilon \geq 0$ controls the magnitude of the perturbations, and $\hat{\Theta}$ denotes the current model parameters. In this formulation, the adversarial term $L(\boldsymbol{S}|\Theta + \Delta_{adv})$ can be treated as a model regularizer, which stabilizes the identification performance. λ is introduced to control the strength of the adversarial regularizer, where the intermediate variable Δ maximizes the objective function to be minimized by Θ. The training process can be summarized as playing a minimax game:

$$\Theta_{opt}, \Delta_{opt} = \arg \min_{\Theta} \max_{\Delta, \|\Delta\| \leq \epsilon} L(\boldsymbol{S}|\Theta) + \lambda L(\boldsymbol{S}|\Theta + \Delta), \tag{11}$$

where the optimizer for the model parameters Θ acts as the minimizing player while the procedure to derive dynamic perturbations Δ acts as the maximizing player. The maximizing player strives to construct the worst-case perturbations against the current model. The two players alternately play the minmax game until convergence.

Constructing Adversarial Perturbations. Given a training utterance u_{ji}, the problem of constructing adversarial perturbations Δ_{adv} is formulated as maximizing

$$\ell_{adv}(\boldsymbol{E}_{ji}|\Delta) = -\boldsymbol{S}^{\hat{\Theta}+\Delta}_{ji,j} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\hat{\Theta}+\Delta}_{ji,k}), \tag{12}$$

where $\hat{\Theta}$ denotes a set of current model parameters. As it is difficult to derive the exact optimal solutions of Δ_{adv}, we apply the fast gradient method proposed in [10] to estimate Δ_{adv}, where we approximate the objective function around Δ as a linear function. To maximize the approximated linear function, we move towards the gradient direction of the objective function with respect the Δ. With the max-norm constraint $\|\Delta\| \leq \epsilon$, we approximate Δ_{adv} as:

$$\Delta_{adv} = \epsilon \frac{\tau}{\|\tau\|}, \text{ where } \tau = \frac{\partial \ell_{adv}(\boldsymbol{E}_{ji}|\Delta)}{\partial \Delta}. \tag{13}$$

Learning Model Parameters. We now explain how to learn model parameters Θ. The local objective function to minimize for a training instance u is as follows:

$$\begin{aligned} \ell_{adv}(\boldsymbol{E}_{ji}|\Theta) = -\boldsymbol{S}^{\Theta}_{ji,j} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\Theta}_{ji,k}) \\ -\lambda\{\boldsymbol{S}^{\Theta+\Delta_{adv}}_{ji,j} - \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\Theta+\Delta_{adv}}_{ji,k})\}, \end{aligned} \tag{14}$$

where Δ_{adv} is obtained from Equation 13.

The final adversarial end-to-end loss is the sum of all adversarial losses over all utterances.

$$L_{adv}(\boldsymbol{S}|\Theta) = \sum_{j,i} \ell_{adv}(\boldsymbol{E}_{ji}|\Theta) \tag{15}$$

We can obtain the SGD update rule for Θ:

$$\Theta = \Theta - \eta \frac{\partial L_{adv}(\boldsymbol{S}|\Theta)}{\partial \Theta}, \tag{16}$$

where η denotes the learning rate.

Algorithm 1: Adversarial parameter optimizations

Input: Training utterances U, max iteration $iter_{\text{max}}$;
Output: Model parameters Θ

Initialization: initialize Θ with Normal distribution $N(0,0.01)$, $iter = 0$, $\Theta_{opt} = \Theta$, $L_{opt} = L_{vali}$;
repeat
  foreach *batch of training utterances* **do**
    // Constructing adversarial perturbations;
    $\Delta_{adv} \leftarrow$ Equation 13;
    // Updating model parameters;
    $\Theta \leftarrow$ Equation 16;
  if $L_{vali} < L_{opt}$ **then**
    $L_{opt} = L_{vali}$;
    $\Theta_{opt} = \Theta$;
  $iter$ + +;
until $iter > iter_{max}$;
Return Θ_{opt};

Algorithm 1 summarizes the training process. In each training step, we first randomly generate a mini-batch of utterances from N speakers, with each speaker M utterances. We then construct a corresponding mini-batch of contaminated utterances with adversarial perturbations, and update network

parameters so that the resulting model learns to resist such such adversarial perturbations. The training involves multiple training steps and stops until reaching a certain number of training iterations.

3. EXPERIMENTS

We conduct experiments on VCTK data set to evaluate the performance of *SAASI* against four state-of-the-art algorithms.

3.1. Dataset and Experimental Settings

The experiments are conducted on the VCTK dataset, which is publicly available[1]. Table 1 shows the statistics of the dataset. For the dataset, 80% of speakers are treated as existing users and the remaining 20% of speakers are treated as new users for the purpose of evaluation. We follow the previous work [20] to extract acoustic features from the raw audios. The 40-dimensional spectrograms are extracted from each utterance after an energy-based voice activity detection. Table 2 shows the main parameters and their default values to tune in the experiments.

Table 1: The statistics of the experimental dataset.

Gender	Females	Males	Age	[10, 20)	[20, 30)	[30, 40)
# of speakers	61	47		14	91	3
Major accents	English	American	Scottish	Irish	Canadian	South African
# of speakers	33	22	25	9	8	4

Table 2: Main parameters of SAASI in the experiments.

Parameters	Value	Parameters	Value
Learning rate η	0.01	Max number of iterations	5000
Regularizer weight λ	1	Perturbation bound ϵ	0.1
Number of speakers N in a batch	4	Utterances per speaker M	5

Baseline Methods. To evaluate the performance of *SAASI*, the following four methods are adopted as baselines.

- **GE2E** [7] adopts LSTM as the building block and optimizes the end-to-end speaker identification system by maximizing the similarity among utterances coming from the same speaker.
- **SNL** [8] extends GE2E by adding a dot-product attention layer on top of LSTM to extract more informative acoustic features in utterances to conduct speaker identification.
- **GE2E**$_{adv}$ extends GE2E by conducting training in an adversarial manner similarly as described in Section 2.3.
- **SNL**$_{adv}$ conducts adversarial training on SNL.

3.2. Identification Performance

In this section, we evaluate the performances of *SAASI* against different baseline methods on the VCTK dataset. We adopt household-level equal error rate (EER) as the evaluation metric. To form a household, we randomly shuffle the test speakers and then group them into 1000 different households with replacements. The household-level EER first calculates the EER for all speakers in each household and then averages the EER in each household by treating the importance of them equally.

Table 3: Speaker identification EER on existing users.

Utt length	Embed size	GE2E	GE2E$_{adv}$	PNL	PNL$_{adv}$	*SAASI*
1.5s	64	6.95%	5.76%	4.22%	4.13%	3.67%
1.5s	128	6.49%	5.66%	4.03%	3.85%	3.39%

Table 4: Speaker identification EER on new users.

Utt length	Embed size	GE2E	GE2E$_{adv}$	PNL	PNL$_{adv}$	*SAASI*
1.5s	64	13.84%	13.58%	10.86%	9.31%	6.56%
1.5s	128	13.11%	12.73%	10.30%	9.11%	6.39%

To imitate the scenarios of serving existing users and new users, we test the performances of different methods on unseen utterances from both existing users and new users. Moreover, we also set the duration of each utterance to small values, varying from 2 to 4 seconds. Tables 3 and 4 show the performance of different methods on the VCTK dataset.

We have three observations from the results on the VCTK dataset. First, GE2E$_{adv}$ and PNL$_{adv}$ outperform GE2E and PNL in all settings, respectively. This demonstrates the effectiveness of adversarial training in speaker identification. Second, PNL and PNL$_{adv}$ achieve lower EERs than GE2E and GE2E$_{adv}$ in all settings, respectively. The improvement gain stems from the involvement of the dot-product attention mechanism in PNL and PNL$_{adv}$, as more informative global acoustic features are extracted from voice utterances. Third, we observe that *SAASI* consistently achieves the best EERs comparing with the four baselines in all settings. The superior performance of SAASI comes from the adoption of the self-attention mechanism and adversarial training. As the conventional dot-product attention mechanism on RNNs would inadvertently give a higher weight to frames in an utterance closer to a position and therefore it is hard to find relations between frames far apart in the utterance. The adversarial training helps generalize the model and makes it more robust against noise that an utterance might include. It therefore prompts the model less likely to make wrong predictions when testing utterances contain perturbations, especially for new speakers.

4. CONCLUSION

We present *SAASI*, a framework that utilizes self-attention to learn global acoustic features from voice utterances. Moreover, the model is trained in an adversarial manner so that the identification system is equipped with the capability of defending adversarial perturbations. Our experiments on VCTK

[1]VCTK: http://homepages.inf.ed.ac.uk/jyamagis

dataset show that the proposed solution outperforms the four baseline methods: GE2E, PNL, GE2E$_{adv}$, and PNL$_{adv}$.

5. REFERENCES

[1] John Koetsier, "Amazon echo, google home installed base hits 50 million; apple has 6% market share, report says," https://www.forbes.com/sites/johnkoetsier/2018/08/02/amazon-echo-google-home-installed-base-hits-50-million-apple-has-6-market-share-report-says, 2018.

[2] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[3] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[4] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep Speaker: an End-to-End Neural Speaker Embedding System," *CoRR*, vol. abs/1705.02304, 2017.

[5] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "VoxCeleb2: Deep Speaker Recognition," *CoRR*, vol. abs/1806.05622, 2018.

[6] Mirco Ravanelli and Yoshua Bengio, "Speaker Recognition from Raw Waveform with SincNet," in *2018 IEEE Spoken Language Technology Workshop, SLT, Athens, Greece, December 18-21*, 2018, pp. 1021–1028.

[7] L. Wan, Q. Wang, A. Papir, and I. Moreno, "Generalized End-to-end Loss For Speaker Verification," in *Proceedings of ICASSP, Calgary, AB, Canada, April 15-20*, 2018, pp. 4879–4883.

[8] F. A. Rezaur Rahman Chowdhury, Quan Wang, Ignacio Lopez-Moreno, and Li Wan, "Attention-Based Models for Text-Dependent Speaker Verification," in *Proceedings of ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. 2018, pp. 5359–5363, IEEE.

[9] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu, "Freelb: Enhanced adversarial training for language understanding," in *Proceedings of ICLR 2020, 26-30 April 2020, Addis Ababa, Ethiopia*, 2020, pp. 770–778.

[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and Harnessing Adversarial Examples," in *ICLR*, 2015.

[11] Nicholas Carlini and David A. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*. 2018, pp. 1–7, IEEE Computer Society.

[12] Qing Wang, Wei Rao, Sining Sun, Lei Xie, Eng Siong Chng, and Haizhou Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proceedings of ICASSP, Calgary, AB, Canada, April 15-20, 2018*. 2018, pp. 4889–4893, IEEE.

[13] Gautam Bhattacharya, Md. Jahangir Alam, and Patrick Kenny, "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training," in *Proceedings of ICASSP, Brighton, United Kingdom, May 12-17, 2019*. 2019, pp. 6041–6045, IEEE.

[14] Zhong Meng, Yong Zhao, Jinyu Li, and Yifan Gong, "Adversarial speaker verification," in *Proceedings of ICASSP, Brighton, United Kingdom, May 12-17, 2019*. 2019, pp. 6216–6220, IEEE.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is All you Need," in *Proceedings of NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.

[16] Shigeki Karita, Xiaofei Wang, and et al, "A comparative study on transformer vs RNN in speech applications," in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*. 2019, pp. 449–456, IEEE.

[17] J. Wang, K. Wang, M. Law, F. Rudzicz, and M. Brudno, "Centroid-based Deep Metric Learning For Speaker Recognition," in *Proceedings of ICASSP, Brighton, United Kingdom, May 12-17*, 2019, pp. 3652–3656.

[18] Ruirui Li, Jyun-Yu Jiang, Xian Wu, Hongda Mao, Chu-Cheng Hsieh, and Wei Wang, "Bridging Mixture Density Networks with Meta-learning for Automatic Speaker Identification," in *Proceedings of ICASSP 2020, Barcelona, Spain, May 04-08, 2020*. 2020, pp. 5359–5363, IEEE.

[19] Ruirui Li, Jyun-Yu Jiang, Jiahao Liu, Chu-Cheng Hsieh, and Wei Wang, "Automatic Speaker Recognition with Limited Data," in *Proceedings of WSDM, Houston, Texas, USA, February 3-7*, 2020.

[20] Prashant Anand, Ajeet Kumar Singh, Siddharth Srivastava, and Brejesh Lall, "Few Shot Speaker Recognition using Deep Neural Networks," *CoRR*, vol. abs/1904.08775, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778.