Speaker Identification for Household Scenarios with Self-attention and Adversarial Training

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

[†] Amazon [§]University of California Los Angeles [¶]University of Notre Dame

[†]{ruirul, stolcke}@amazon.com, [§]jyunyu@cs.ucla.edu, [¶]xwu9@nd.edu, [†]chucheng@ucla.edu

Abstract

Speaker identification that recognizes people from voices is one fundamental problem 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 embedding representations from utterances with convolutional neural networks (CNNs) and recurrent neural networks (RNNs). However, there are two inherent limitations. First, complementary long-duration voice characteristics, which express speaker identities, might not be fully captured by CNNs and RNNs, as they are designed to 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 model inputs, also known as adversarial perturbations. To distill informative global acoustic embedding representations from utterances and defend against 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 baselines. The experimental results conclude that *SAASI* significantly outperforms all competitive baselines in recognizing both known and new speakers.

1. Introduction

Smart speakers like Amazon Echo and Google Home allow convenient voice-enabled access to a wide variety of services and experiences, and have gained widespread use. As these devices are typically used by multiple speakers in a household, speaker identification is key to enable many important functionalities such as authentication and user-based customization. In this paper, we develop two novel techniques for speaker identification geared toward household-like scenarios with a small number of competing speaker identities.

Deep learning-based speaker identification methods have gained notable attraction in virtue of significant improvements over the prevalent i-vector and GMM-UBM solutions [1,2]. For example, Deep Speaker [3] and VGGVox [4] adopt CNN-based residual networks to learn voice acoustic representations based on utterance spectrograms while SincNet [5] applies CNNs to perform speaker identification from raw voice waveforms. Generalized end-to-end speaker identification (GE2E) [6] utilizes RNNs to model utterances and develops a similarity-based loss function so that the similarity between utterance representations from the same/different speaker is maximized/minimized, respectively. GE2E with shared-parameter non-linear attention (SNL) [7] further extends GE2E to obtain more informative acoustic features by weighting contributions of RNN outputs differently. However, these conventional methods suffer from capturing the complementary dependencies among all frames across an utterance, leading to inferior voice characteristic 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 [8–13]. Although previous studies [14, 15] apply domain adversarial training, they only focus on adapting a well trained speaker model to a new domain or a fresh language instead of boosting the model robustness. Li et al. [13] investigates the vulnerability of Gaussian Mixture Model i-vector based speaker verification systems to adversarial attacks. Meng et al. [16] strive to enhance the robustness of speaker identification through multi-task learning. Suthokumar et al. [17] utilize adversarial multi-task learning with the focus of distinguishing genuine and replayed speech. However, it remains a daunting task to defend against adversarial attacks and enhance security and robustness in speaker identification.

To address the above challenges, we first leverage the self-attention mechanism [18, 19] to extract the acoustic features from utterances. More precisely, the self-attention mechanism allows us to fully utilize the complementary dependencies among all frames across an utterance, resulting in informative global acoustic embedding representations of an utterance. Moreover, we craft dynamic perturbations at spectrogram level to form effective adversarial examples. These adversarial examples are formed by applying small but deliberate perturbations to spectrograms of 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 examples, 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 when the utterances are only 1.5 seconds.

2. Problem Statement & Methodology

We formulate the objective of our work as followings. Suppose the system has a set of known users with a few voice utterances for each user as training data. Given a closed test set of known or new users, with a few short registered voice utterances for each user as enrollment, and another short test utterance from a test user, the goal of this study is to recognize the speaker identity behind the test utterance. In this work, we focus on text-independent speaker identification and presume that each utterance is very short [6, 20–23], 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 an 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 voice characteristics across an 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 query, key, and value matrices in the self-attention mechanism, 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. To incorporate the frame location information, we follow [18] and add sinusoidal positional embedding $\boldsymbol{E}_p$ into $\boldsymbol{SP}_u$ before fusion. Formally,

$$E_p(t, pos) = \begin{cases} \sin \frac{pos}{10000^{t/d}}, \text{if } t \text{ is even,} \\ \cos \frac{pos}{10000^{t/d}}, \text{if } t \text{ is odd,} \end{cases} \tag{2}$$

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. Formally,

$$\boldsymbol{SP}_u^{'} = \boldsymbol{SP}_u + \boldsymbol{E}_p. \tag{3}$$

$$\tilde{\boldsymbol{E}}_u = \text{Self-Att}(\boldsymbol{SP}_u^{'} \cdot \boldsymbol{W}^Q, \boldsymbol{SP}_u^{'} \cdot \boldsymbol{W}^K, \boldsymbol{SP}_u^{'} \cdot \boldsymbol{W}^V), \tag{4}$$

where $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$, and $\boldsymbol{W}^V$ are query, key, and value projection matrices, respectively.

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 in an utterance to get fused and to complement each other.

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:

$$\tilde{\boldsymbol{E}}_u^f = \boldsymbol{W}_2^f \cdot \text{ReLU}(\boldsymbol{W}_1^f \cdot \tilde{\boldsymbol{E}}_u + \boldsymbol{b}_1^f) + \boldsymbol{b}_2^f, \tag{5}$$

where $\boldsymbol{W}_1^f$, $\boldsymbol{W}_2^f$, and $\boldsymbol{b}_2^f$, $\boldsymbol{b}_2^f$ 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 [24].

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

$$\boldsymbol{E}_u = \frac{\bar{\boldsymbol{E}}_u^f}{||\bar{\boldsymbol{E}}_u^f||_2}. \tag{6}$$

2.2. End-To-End Training

We follow [6, 7] 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,

$$\boldsymbol{C}_j = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{E}_{jm} \tag{7}$$

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

$$\boldsymbol{S}_{ji,k} = \boldsymbol{W}^s \cdot cos(\boldsymbol{E}_{ji}, \boldsymbol{C}_k) + \boldsymbol{b}^s, \tag{8}$$

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 of 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:

$$L(\boldsymbol{E}_{ji}|\Theta) = -\boldsymbol{S}_{ji,j}^{\Theta} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}_{ji,k}^{\Theta}), \tag{9}$$

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.

$$L(\boldsymbol{S}|\Theta) = \sum_{j,i} L(\boldsymbol{E}_{ji}|\Theta) \tag{10}$$

2.3. Defending Against Adversarial Attacks

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

$$\begin{gathered} L_{adv}(\boldsymbol{S}|\Theta) = L(\boldsymbol{S}|\Theta) + \lambda L(\boldsymbol{S}_{\Delta_{adv}}|\Theta), \\ \text{where } \Delta_{adv} = \arg \max_{\Delta, \|\Delta\| \le \epsilon} L(\boldsymbol{S}_{\Delta}|\hat{\Theta}), \end{gathered} \tag{11}$$

where Δ denotes the perturbations on input utterances, $\boldsymbol{S}_{\Delta_{adv}}$ is corresponding similarity matrix regarding the perturbed utterances, $\epsilon \ge 0$ ensures that the perturbations are perceptible but not too large, and $\hat{\Theta}$ denotes the current model parameters. In this formulation, the adversarial term $L(\boldsymbol{S}_{\Delta_{adv}}|\Theta)$ 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:

$$\Theta_{opt}, \Delta_{opt} = \arg \min_{\Theta} \max_{\Delta, \|\Delta\| \le \epsilon} L(\boldsymbol{S}|\Theta) + \lambda L(\boldsymbol{S}_{\Delta}|\Theta), \tag{12}$$

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 adversarial perturbations Δ_{adv} to be constructed are expected to best fool the current model. Therefore, the problem of constructing Δ_{adv} is formulated as maximizing

$$\ell_{adv}(\boldsymbol{E}_{ji}|\hat{\Theta}) = -\boldsymbol{S}^{\hat{\Theta}}_{\Delta\, ji,j} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\hat{\Theta}}_{\Delta\, ji,k}), \tag{13}$$

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 [9] to estimate Δ_{adv}, where we approximate the objective function around Δ as a linear function. To maximize the approximated linear function, we move toward the gradient direction of the objective function with respect the Δ. With the max-norm constraint $\|\Delta\| \le \epsilon$, we approximate Δ_{adv} as:

$$\Delta_{adv} = \epsilon \frac{\tau}{\|\tau\|}, \text{ where } \tau = \frac{\partial \ell_{adv}(\boldsymbol{E}_{ji}|\hat{\Theta})}{\partial \boldsymbol{SP}_{ji}}. \tag{14}$$

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

$$\begin{gathered} \ell_{adv}(\boldsymbol{E}_{ji}|\Theta) = -\boldsymbol{S}^{\Theta}_{ji,j} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\Theta}_{ji,k}) \\ -\lambda\{\boldsymbol{S}^{\Theta}_{\Delta_{adv}\, ji,j} - \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\Theta}_{\Delta_{adv}\, ji,k})\}, \end{gathered} \tag{15}$$

where Δ_{adv} is obtained from Equation 14.

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

$$L_{adv}(\boldsymbol{S}|\Theta) = \sum_{j,i} \ell_{adv}(\boldsymbol{E}_{ji}|\Theta) \tag{16}$$

We can obtain the SGD update rule for Θ:

$$\Theta = \Theta - \eta \frac{\partial L_{adv}(\boldsymbol{S}|\Theta)}{\partial \Theta}, \tag{17}$$

where η denotes the learning rate.

Algorithm 1: Adversarial parameter optimizations

```
Input: Training utterances $U$, max iteration $iter_{\max}$;
Output: Model parameters $\Theta$
Initialization: initialize $\Theta$ with Normal distribution
  $N(0,0.01)$, $iter = 0$, $\Theta_{opt} = \Theta$, $EER_{opt} = EER_{vali}$;
repeat
    foreach batch of training utterances do
        // Updating model parameters;
        $\Theta \leftarrow$ Equation 10;
        // Constructing adversarial perturbations;
        $\Delta_{adv} \leftarrow$ Equation 14;
        // Updating model parameters with adversarial
          training;
        $\Theta \leftarrow$ Equation 17;
    if $EER_{vali} < EER_{opt}$ then
        $EER_{opt}$ = $EER_{vali}$;
        $\Theta_{opt} = \Theta$;
    $iter++$;
until $iter > iter_{max}$;
Return $\Theta_{opt}$;
```

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 follow Equation 10 to calculate the loss based on utterances from this mini-batch and optimize the model. After that, we construct a corresponding mini-batch of contaminated utterances with adversarial perturbations, feed them into the model, and update model parameters so that the resulting model learns to resist such adversarial perturbations. The training involves multiple training steps and stops until reaching a certain number of training iterations. The parameters achieving the best equal error rate (EER) on the validation dataset are utilized for evaluations.

3. Experiments

We conduct experiments on the VCTK dataset 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 publicly available VCTK dataset[1]. Table 1 shows the statistics of the dataset. For the dataset, 80% of speakers are treated as known users and the remaining 20% of speakers are treated as new users. Utterances

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

from the known users are used for training and unseen utterances from both known and new users are used for evaluation. We follow the previous work [23] 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		Age		
	Female	Male	[10, 20)	[20, 30)	[30, 40)
# of speakers	61	47	14	91	3

	Major Accent					
	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
# 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** [6] adopts LSTM to construct utterance embeddings and optimizes the end-to-end speaker identification system by maximizing the similarity among utterances coming from the same speaker.
- **SNL** [7] extends GE2E by adding a shared-parameter nonlinear 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. As most smart speakers, such Google Home, serve customers in household scenarios, we propose to use household-level equal error rate (H-EER) as the evaluation metric. To form a household, we randomly shuffle the test speakers and then sample 1000 different households with replacement, with each household composed of 4 speakers. For each household, 20 utterances are tested. Then, H-EER calculates the equal error rate for all speakers in each household and averages the equal error rate in each household by treating the importance of each household equally.

Table 3: *H-EER performance on known users.*

Utt Length	Embed Size	GE2E	GE2E$_{adv}$	SNL	SNL$_{adv}$	*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%

To imitate the scenarios and differences when serving known users and new users, we test the performances of different methods on unseen utterances from both known users and new users. The known users are the ones who are involved when training the model while the new users are the ones who are not involved when training the model. For each speaker, 5 utterances are used for enrollments and each utterance is about 1.5 seconds long. Tables 3 and 4 show the performance of different methods on known users and new users, respectively.

Table 4: *H-EER performance on new users.*

Utt Length	Embed Size	GE2E	GE2E$_{adv}$	SNL	SNL$_{adv}$	*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%

We have three observations from the results on the VCTK dataset. First, GE2E$_{adv}$ and SNL$_{adv}$ outperform GE2E and SNL in all settings, respectively. This demonstrates the effectiveness of training with adversarial examples in speaker identification. Adversarial examples serve as out-of-distribution augmented data and help generalize the model. Second, SNL and SNL$_{adv}$ achieve lower H-EER than GE2E and GE2E$_{adv}$ in all settings, respectively. The improvement gain stems from the involvement of the shared-parameter non-linear attention mechanism in SNL and SNL$_{adv}$, as it summarizes the RNN outputs differently with the consideration of their contributions to identification performance. In this way, more informative global acoustic features are extracted from voice utterances. Third, we observe that *SAASI* consistently achieves the best H-EER 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 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 self-attention mechanism applied in this work can easily and comprehensively fuse the acoustic information among all frames across an utterance, yielding more informative acoustic embedding representations of utterances. 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 test utterances contain perturbations, especially for new speakers.

4. Conclusion

In this work, we investigate a household-based speaker identification task that reflects speaker recognition as typically used on smart speaker devices. 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 end-to-end manner so that the identification system is equipped with the capability of defending against adversarial perturbations. We propose to use household-level equal error rate to measure the speaker identification for household scenarios. Our experiments on the publicly available VCTK benchmark dataset show that the proposed solution outperforms the four baseline methods proposed recently: GE2E, SNL, GE2E$_{adv}$, and SNL$_{adv}$. This is because: (1) The self-attention mechanism precisely captures the acoustic similarity among different frames across an utterance so that the learned representations can obtain more contextual and global knowledge; (2) the constructed adversarial perturbations provide strong examples as hints to significantly boost the robustness and generalization of the model.

5. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Trans. Audio, Speech & Language Processing*, pp. 788–798, 2011.

[2] 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.

[3] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep Speaker: an End-to-End Neural Speaker Embedding System," *CoRR*, vol. abs/1705.02304, 2017.

[4] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," *CoRR*, vol. abs/1806.05622, 2018. [Online]. Available: http://arxiv.org/abs/1806.05622

[5] M. Ravanelli and Y. 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.

[6] 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.

[7] F. A. R. R. Chowdhury, Q. Wang, I. Lopez-Moreno, and L. Wan, "Attention-Based Models for Text-Dependent Speaker Verification," in *Proceedings of ICASSP, Calgary, AB, Canada, April 15-20, 2018*, pp. 5359–5363.

[8] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, "Freelb: Enhanced adversarial training for language understanding," in *Proceedings of ICLR, 26-30 April 2020, Addis Ababa, Ethiopia*, pp. 770–778.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *ICLR*, 2015.

[10] N. Carlini and D. 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.

[11] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proceedings of ICML, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, J. G. Dy and A. Krause, Eds., pp. 284–293.

[12] H. Wu, S. Liu, H. Meng, and H. Lee, "Defense against adversarial attacks on spoofing countermeasures of ASV," in *Proceedings of ICASSP, Barcelona, Spain, May 4-8, 2020.*

[13] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial attacks on GMM i-vector based speaker verification systems," in *Proceedings of ICASSP, Barcelona, Spain, May 4-8, 2020.*

[14] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proceedings of ICASSP, Calgary, AB, Canada, April 15-20, 2018*, pp. 4889–4893.

[15] G. Bhattacharya, M. J. Alam, and P. 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*, pp. 6041–6045.

[16] Z. Meng, Y. Zhao, J. Li, and Y. Gong, "Adversarial speaker verification," in *Proceedings of ICASSP, Brighton, United Kingdom, May 12-17, 2019*, pp. 6216–6220.

[17] G. Suthokumar, V. Sethu, K. Sriskandaraja, and E. Ambikairajah, "Adversarial multi-task learning for speaker normalization in replay detection," in *Proceedings of ICASSP, Barcelona, Spain, May 4-8, 2020.*

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Proceedings of NIPS, 4-9 December 2017, Long Beach, CA, USA*, pp. 5998–6008.

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

[20] 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.

[21] R. Li, J.-Y. Jiang, X. Wu, H. Mao, C.-C. Hsieh, and W. Wang, "Bridging Mixture Density Networks with Meta-learning for Automatic Speaker Identification," in *Proceedings of ICASSP, Barcelona, Spain, May 04-08, 2020*, pp. 5359–5363.

[22] R. Li, J. Jiang, J. Liu, C. Hsieh, and W. Wang, "Automatic Speaker Recognition with Limited Data," in *Proceedings of WSDM, Houston, Texas, USA, February 3-7*, 2020.

[23] P. Anand, A. K. Singh, S. Srivastava, and B. Lall, "Few Shot Speaker Recognition using Deep Neural Networks," *CoRR*, vol. abs/1904.08775, 2019.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778.