A METHOD TO REVEAL SPEAKER IDENTITY IN DISTRIBUTED ASR TRAINING, AND HOW TO COUNTER IT

Trung Dang[⋆,‡]*, Om Thakkar*[†]*, Swaroop Ramaswamy*[†]*, Rajiv Mathews*[†]*, Peter Chin*[⋆]*, Françoise Beaufays*[†]

[⋆]Boston University [†]Google Inc.

ABSTRACT

End-to-end Automatic Speech Recognition (ASR) models are commonly trained over spoken utterances using optimization methods like Stochastic Gradient Descent (SGD). In distributed settings like Federated Learning, model training requires transmission of gradients over a network. In this work, we design the first method for revealing the identity of the speaker of a training utterance with access only to a gradient. We propose Hessian-Free Gradients Matching, an input reconstruction technique that operates without second derivatives of the loss function (required in prior works), which can be expensive to compute. We show the effectiveness of our method using the DeepSpeech model architecture, demonstrating that it is possible to reveal the speaker's identity with 34% top-1 accuracy (51% top-5 accuracy) on the LibriSpeech dataset. Further, we study the effect of Dropout on the success of our method. We show that a dropout rate of 0.2 can reduce the speaker identity accuracy to 0% top-1 (0.5% top-5).

***Index Terms*—** ASR, distributed training, privacy

1. INTRODUCTION

End-to-end automatic speech recognition (ASR) has increasingly been gaining popularity over conventional pipeline frameworks [1, 2, 3, 4]. State-of-the-art ASR models have achieved human parity in conversational speech recognition [4]. Training such models often requires a large amount of user-spoken utterances comprising of audios and their transcripts, access to which can expose sensitive information.

In distributed frameworks such as Federated Learning (FL) [5], model training is done via mobile devices with transmission of gradients over the network, allowing training over large populations [6] while ensuring such data remains on-device. Many works have shown the competitive performance of FL-trained models on sequential modeling tasks [7], as well as in speech recognition [8].

A recent line of work [9, 10, 11, 12] has focused on demonstrating leakages of information about training data, from the gradients used in model training. At a high level, these works aim to reconstruct training samples by designing optimization methods for constructing objects that have a gradient *matching* to the observed gradient. For instance, existing methods have been shown to successfully reconstruct images used for training image classification models [10, 13]. In the speech context, there are fundamental challenges, e.g., variable-sized inputs/outputs, rendering such methods inapplicable.

In this work, we study information leakage from gradients in ASR model training. We design a method to reveal the speaker id of a training utterance from a model gradient computed using the utterance. Given that ASR models can have training utterances and transcripts of arbitrary lengths, for computational efficiency and to avoid potential false positives, we assume that the transcript, and the length of the training utterance are known. We start by designing Hessian-Free Gradients Matching (HFGM), a technique to reconstruct speech features used in computing gradients for training ASR models. HFGM eliminates the need of second-order derivatives (i.e., Hessian) of the loss function, which were required in prior works [9, 10], and can be expensive to compute. Next, our method uses the reconstructed features and a speaker identification (SI) model to uniquely identify the speaker from a list of speakers by comparing speaker embeddings.

To our knowledge, this is the first method in the speech domain that can be used for revealing information about training samples from gradients. We demonstrate the efficacy of our method by conducting experiments using the LibriSpeech data set [14] on the DeepSpeech [2] model architecture. We find that our method is successful in revealing the speaker id with 34% top-1 accuracy (51% top-5 accuracy) among ≈2.5k speakers. We also show that using the standard training technique of Dropout [15] can reduce the speaker id accuracy of our method to 0% top-1 (0.5% top-5), without compromising utility of the trained model. To spur further research, we provide an open-source implementation[1] of our framework.

We conclude by exploring the effectiveness of our method in two complex regimes, where instead of access to individual gradients, the method can only access 1) the average gradient from a mini-batch of samples, or 2) the update comprising multiple gradient descent steps using a training sample. We demonstrate that in both of the above settings, our method reveals speaker id with non-trivial accuracy, whereas training

[‡]Work performed while at Google.

[1]https://github.com/googleinterns/deepspeech-reconstruction

with dropout is effective in reducing its success.

2. BACKGROUND

Gradients Matching (GM) & Deep Leakage from Gradients (DLG) Algorithm DLG was introduced by [9] as a method to reconstruct an input $\hat{x}$ and output $\hat{y}$ given a model gradient $\nabla_\theta \mathcal{L}(\hat{x}, \hat{y})$, where $\mathcal{L}$ denotes the loss function and θ denotes the model parameters (when it is clear that the gradient is w.r.t. model parameters θ, we just denote it by $\nabla\mathcal{L}(\cdot,\cdot)$). The algorithm attempts to find an input-output pair (x, y) that matches $\nabla\mathcal{L}(x, y)$ with $\nabla\mathcal{L}(\hat{x}, \hat{y})$. The general idea is also referred to as Gradients Matching (GM). A dummy input x and a dummy label y are fed into the model to get dummy gradients $\nabla\mathcal{L}(x, y)$. Reconstructed objects are obtained by minimizing the Euclidean distance between the dummy gradients and the client update: $x^*, y^* = \text{argmin}_{x,y} \|\nabla\mathcal{L}(x, y) - \nabla\mathcal{L}(\hat{x}, \hat{y})\|^2$.

Zeroth-order Optimization Zeroth-order optimization is the process of minimizing an objective, given access to the objective values at chosen inputs. A direct search algorithm (also known as pattern search, derivative-free search, or black-box search), which samples a vector u and moves x to $x + u$, has been shown to perform well in several settings (e.g. [16])

Speaker Identification (SI) with Triplet Loss Recent works [17, 18] formulate SI as learning speaker discriminative embeddings for an utterance. In this work, we adopt the triplet loss, which operates on pairs of embeddings, trying to minimize the distance of embeddings from the same speaker, and maximize the distance with other negative samples.

3. A METHOD TO REVEAL SPEAKER IDENTITY

Now, we describe our method to reveal the speaker id of a training sample $\hat{x}$ given its model gradient $\nabla\mathcal{L}(\hat{x}, \hat{y})$. Here, $\hat{x} \in \mathbb{R}^{T\times d}$ denotes the input speech features created from the training utterance, T is the length of the input, d is the dimension of the input speech features, $\hat{y}$ is the output label sequence, and $\mathcal{L}$ denotes the training loss function. We split our method into two phases: (1) Using Hessian-Free Gradients Matching to reconstruct the input speech features, and (2) Identify the speaker from the reconstructed speech features. Figure 1 provides an illustration of our method.

Reconstruction Phase: Hessian-Free GM (HFGM) Given access to a gradient $\nabla\mathcal{L}(\hat{x}, \hat{y})$, generally one would like to find a pair of speech features and transcript (x, y) such that $\nabla\mathcal{L}(x, y) = \nabla\mathcal{L}(\hat{x}, \hat{y})$. However, ASR models are typically sequence-to-sequence models that can map arbitrary length speech features ($\hat{x}$) to arbitrary length transcripts ($\hat{y}$). With no additional information, the possible values y can take is exponential in the label set size, searching through which can incur a prohibitive computational cost. To circumvent this issue and make our problem simpler, we assume $\hat{y}$, the transcript of the training utterance, is given.[2] Even though $\hat{y}$ is given, there could exist multiple length speech features x', where $|x'| \neq T$, such that $\nabla\mathcal{L}(x', \hat{y}) = \nabla\mathcal{L}(\hat{x}, \hat{y})$. Thus, we also assume T is given. Note that even if the transcript and the length of the input speech features are known, revealing the identity of the speaker can still result in a significant breach of privacy. We leave designing efficient reconstruction methods that operate without these assumptions for future work.

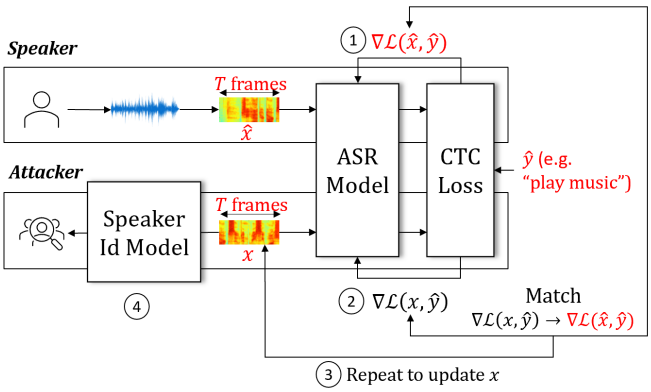

Fig. 1. An illustration of our method. (1) A gradient is accessible to an attacker. (2) The attacker computes dummy gradients. (3) The attacker compares the gradient received with dummy gradients and repeats to optimize x. (4) The attacker reveals the speaker id. Notations in red are known to the attacker.

Now, we define the reconstruction task as constructing an $x \in \mathbb{R}^{T\times d}$ such that $\nabla\mathcal{L}(x, \hat{y})$ is close to the observed gradient $\nabla\mathcal{L}(\hat{x}, \hat{y})$. Following [10], we choose cosine distance as our measure of closeness, and formulate our optimization problem to find x^* s.t. $x^* = \text{argmin}_x \mathcal{D}\left(x, \nabla\mathcal{L}(\hat{x}, \hat{y})\right)$, where $\mathcal{D}(x, \nabla\mathcal{L}(\hat{x}, \hat{y})) = 1 - \frac{\langle\nabla\mathcal{L}(x,\hat{y}), \nabla\mathcal{L}(\hat{x},\hat{y})\rangle}{\|\nabla\mathcal{L}(x,\hat{y})\|\|\nabla\mathcal{L}(\hat{x},\hat{y})\|}$.

To solve this non-convex optimization problem using gradient-based methods as in prior work [9, 10, 11], we need to compute the second-order derivatives of the loss function $\mathcal{L}$. However, computing the second derivative of Connectionist Temporal Classification (CTC) loss involves backpropagating twice through a dynamic programming algorithm which we found to be intractable.[3] To tackle this challenge, and also address a broader family of loss functions, we adopt a zeroth-order optimization algorithm.

We use a direct search approach (Algorithm 1). We initialize x with uniformly random values. At each iteration, we sample K random unit vectors and apply them to x. The value $\mathcal{D}$ is evaluated at each of these K points. We choose only the vectors that lower $\mathcal{D}$, sum these up and apply the sum with a learning rate α. We repeat the process until we reach the

[2] Note that the transcript could be a common phrase, e.g., "play music". Our objective is to identify the speaker of an utterance regardless of the contents of its transcript.

[3] Additionally, the second derivatives of CTC loss are not implemented in common deep learning frameworks like TensorFlow and PyTorch.

Algorithm 1 Hessian-Free Gradients Matching

Input: $\nabla\mathcal{L}(\hat{x}, \hat{y})$, $\mathcal{D}(x, \nabla\mathcal{L}(\hat{x}, \hat{y}))$, α, $\hat{y}$, and T. Parameters: number of samplings K, number of iterations N
Initialize $x \in \mathbb{R}^{T\times d}$.
for $n = 1$ **to** N **do**
  $V \leftarrow \varnothing$
  Sample K unit vectors $v_1, ..., v_K \in \mathbb{R}^{T\times d}$
  for $k = 1$ **to** K **do**
    if $\mathcal{D}(x + \alpha v_k, \nabla\mathcal{L}(\hat{x}, \hat{y})) < \mathcal{D}(x, \nabla\mathcal{L}(\hat{x}, \hat{y}))$ **then**
      Add v_k to V
  $x \leftarrow x + \alpha \sum_{v\in V} v$

convergence criteria.

Inference Phase: Revealing Speaker Identity In the second part of our method, we use the reconstructed speech features (x^*) to identify the speaker of the utterance from a list of possible speakers. We assume that we have access to some public utterances for each possible speaker to identify them. We use the SI model to create embeddings for each speaker from the public utterances. We take the reconstructed speech features (x^*), create an embedding using the SI model, and compare it using consine similarity with embeddings for each speaker.

Comparison with Related Prior Works Our work differs from the related prior works [9, 10, 11] in a few ways. 1) The input to ASR models is typically speech features which are computed from the raw audio using a series of lossy transformations. While prior works on image recognition models demonstrate breach of privacy by directly reconstructing the input to the model, we incorporate an additional inference phase where we use the reconstructed input to reveal the identity of the speaker. 2) The models we focus on use CTC loss instead of cross-entropy loss.

4. EXPERIMENTS

Model Architectures Following prior work [19], we choose the DeepSpeech [2] model architecture for our experiments. We conduct our experiments using randomly initialized weights for the model. For the inference phase, we follow [20] to train a text-independent SI model on 26-dim normalized Mel-frequency cepstral coefficients (MFCCs), similar to the speech features for DeepSpeech.

Dataset We choose the LibriSpeech ASR corpus [14] for our experiments. For training the SI model, we first combine all sets to obtain 300k utterances from 2,484 speakers, and use only the first 5 utterances of each speaker for training.

For the reconstruction phase, we trim the leading and ending silences, based on the intensity of every 10ms chunk, from each utterance in the remaining combined test set. Next, we randomly sample a total of 600 utterances, 100 for each interval of audio length in $\{[1, 1.5s), [1.5, 2s), \ldots, [3.5, 4s)\}$. The average audio length in our sampled set is 2.5 seconds, and average transcript length is 40.6 characters.

Implementation Details In this section, all the experiments consider the scenario of revealing speaker id from a single gradient computed using a single utterance. For computational efficiency, we match gradients only for the last layer ($\sim$60k parameters). Each dummy input in our reconstruction is initialized with uniformly random values in $[-1, 1]$. When performing direct search, we sample 128 unit vectors per iteration, each of which only updates a single frame. We set the step size to 1, and reduce by half after every 2.5k iterations s.t. the loss does not decrease by more than 5%. We stop the reconstruction when the step size reaches 0.125.

Evaluation Metrics To evaluate our reconstruction, we use the Mean Absolute Error (MAE) to measure the distance of normalized MFCCs to those of the original utterance. During inference, the similarity scores of a reconstructed object's embedding with each of 5 available utterances' embeddings in the training data are averaged and ranked to identify the speaker. We use Top-1 Accuracy, Top-5 Accuracy and Mean Reciprocal Rank (MRR) to evaluate the speaker id leakage.

4.1. Empirical Results

We present the results of using our method from Section 3 to reveal speaker id from 600 individual gradients, each gradient computed using a unique utterance from our sampled set. Table 1 shows the overall values of the average MAE, Top-1 accuracy, Top-5 accuracy, and MRR of SI results from the original and reconstructed speech features. We see that while SI from original utterances results in 42% top-1 (57% top-5) accuracy, the same from the reconstructed features is 34% top-1 (51% top-5), providing 81% (89.5%) relative performance.

Table 1. MAE, Top-1 Accuracy (%), Top-5 Accuracy (%), and MRR of SI on 600 utterances.

	MAE	TOP-1	TOP-5	MRR
ORIGINAL	0.00	42.0	57.0	0.554
RECONSTRUCTED	0.25	34.0	51.0	0.419

Training with Dropout Dropout [15] has been adopted in training deep neural networks as an efficient way to prevent overfitting to the training data. While prior work [11] has mentioned dropout in the context of information leakage from gradients, it does not provide any empirical evidence of the effect of training with dropout on such leakages. When parameters are shared in the network, for e.g., a fully-connected layer operating frame-wise on a sequence of speech features, each part of the output typically uses an i.i.d. random dropout

Table 2. MAE, Top-1 (%), Top-5 (%), and MRR of SI when reconstructed from dropped-out gradients.

d	MAE	TOP-1	TOP-5	MRR	WER (CLEAN)	WER (OTHER)
0	0.25	34.0	51.0	0.419	10.5	28.4
0.1	0.59	0.8	2.0	0.019	11.9	28.2
0.2	0.72	0.0	0.5	0.006	9.2	25.6
0.3	0.81	0.1	0.3	0.005	9.5	27.1

mask, making it difficult to infer dropout masks from a gradient. Table 2 shows reconstruction quality and training error rates for different dropout rates. Even for the lowest dropout rate of 0.1, we see that the top-1 accuracy of SI is $\sim$1%. At the same time, we observe that for models trained with dropout, the WER is comparable (or sometimes even lower) than the baseline training.

Visualizing Reconstructed Features In Figure 2, we provide two examples of spectrograms from the reconstruction of a short and a long utterance. For comparison, we also provide spectrograms from reconstructions of the same utterances from training with a dropout rate of 0.1.

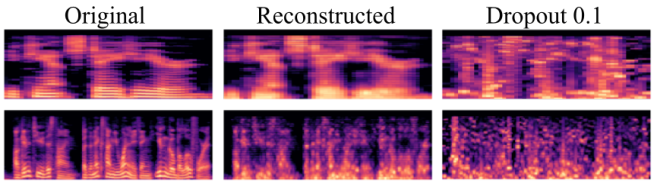

Fig. 2. Spectrograms obtained from the original and reconstructed MFCCs, as well as training with dropout rate 0.1. For reconstructions on training using Dropout, we see that reconstruction quality deteriorates.

5. ADDITIONAL EXPERIMENTS

The experiments in Section 4 focused on revealing speaker id using our method on a single gradient from a single utterance. In distributed settings like FL, model training is performed under more complex settings. In this section, we conduct experiments to evaluate the success of our method on two natural extensions of the setting in Section 4. We demonstrate that in both of the settings, our method can reveal speaker id with non-trivial accuracy. All the experiments in this section are conducted using the 200 utterances of audio length 1-2s.

5.1. Averaged Gradients from Batches

Now, we study the performance of our method for revealing speaker ids from an averaged gradient computed using a batch of utterances. In the reconstruction phrase, our objective function (Section 3) does not change; however, we instead try to reconstruct $(x_1 : x_N)$, where $x_i \in \mathbb{R}^{T_i \times d}$ for $i \in [B]$. Here, B is the number of samples in the batch, and T_i is the length of input $i \in [B]$. We conduct our experiments for batch sizes in $\{2, 4, 8\}$. For each batch size, the 200 utterances are sorted by audio length, and grouped into batches. We provide the results in Table 3, comparing them with the results (batch size 1) on same 200 utterances in Section 4.1. We see that while SI accuracy decreases with increasing batch sizes, the top-1 accuracy is still as high as 19% for batch size 4. Training with a dropout rate of 0.1 shows that reconstruction for batch size 2 reduces the accuracy to 1% top-1 (4% top-5), compared to 2% top-1 (4% top-5) for batch size 1.

Table 3. Reconstruction MAE, Top-1 (%) and Top-5 (%) SI accuracy from averaged gradients of a batch.

	MAE	TOP-1	TOP-5	MRR
ORIGINAL	0.00	42.0	57.0	0.490
BATCH SIZE 1	0.14	40.0	55.0	0.470
BATCH SIZE 2	0.21	37.0	54.0	0.451
BATCH SIZE 4	0.37	19.0	31.0	0.249
BATCH SIZE 8	0.48	5.0	11.0	0.084

Table 4. Reconstruction MAE, Top-1 (%) and Top-5 (%) SI accuracy from multi-step updates from a sample.

	MAE	TOP-1	TOP-5	MRR
ORIGINAL	0.00	42.0	57.0	0.490
1-STEP	0.14	40.0	55.0	0.470
2-STEP	0.33	26.5	39.5	0.333
8-STEP	0.33	24.5	39.0	0.321

5.2. Multi-Step Updates from a Sample

Next, we study the success of our method in revealing speaker identities from an update comprising of multiple update steps using a single utterance. For computational efficiency, we reduce the number of unit vectors sampled to 8 (as opposed to 128, in the experiments in Sections 4 and 5.1) in each iteration of our zeroth-order optimization. Table 4 shows the results of our experiment, comparing them with the same (1-step) from Section 4.1. Since the optimization for multi-step reconstruction is different, the results are not directly comparable with those of single-step setting. We see that the success of our method in revealing speaker id is still as high as 24% top-1 accuracy for 8-step updates. Using dropout in training is still effective: a dropout rate of 0.1 reduces the accuracy to 2% top-1 (3.5% top-5) in the 2-step case.

6. REFERENCES

[1] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[2] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[3] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[4] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Toward human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.

[5] Brendan McMahan and Daniel Ramage, "Federated learning: Collaborative machine learning without centralized training data," *Google Research Blog*, vol. 3, 2017.

[6] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al., "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[7] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[8] Dimitrios Dimitriadis, Kenichi Kumatani, Robert Gmyr, Yashesh Gaur, and Sefik Emre Eskimez, "A federated approach in training acoustic models," in *Proc. Interspeech*, 2020.

[9] Ligeng Zhu, Zhijian Liu, and Song Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, 2019, pp. 14774–14784.

[10] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?," *arXiv preprint arXiv:2003.14053*, 2020.

[11] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu, "A framework for evaluating gradient leakage attacks in federated learning," *arXiv preprint arXiv:2004.10397*, 2020.

[12] Trung Dang, Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, Peter Chin, and Françoise Beaufays, "Revealing and protecting labels in distributed training," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[13] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16337–16346.

[14] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] Horia Mania, Aurelia Guy, and Benjamin Recht, "Simple random search of static linear policies is competitive for reinforcement learning," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. 2018, vol. 31, pp. 1800–1809, Curran Associates, Inc.

[17] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[18] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[19] Nicholas Carlini and David Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[20] 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," *arXiv preprint arXiv:1705.02304*, vol. 650, 2017.