INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Self-supervised Spoofing Audio Detection Scheme

Ziyue Jiang[1], Hongcheng Zhu[1], Li Peng[1], Wenbing Ding[1], Yanzhen Ren[2,1,]*

[1]School of Cyber Science and Engineering, Wuhan University, China
[2]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, China

ZiyueJiang341@gmail.com, hongcheng Z@whu.edu.cn, pl19990329@gmail.com, 2017301500258@whu.edu.cn, renyz@whu.edu.cn

Abstract

With the development of deep generation technology, spoofing audio technology based on speech synthesis and speech conversion is closer to reality, which challenges the credibility of the media in social networks. This paper proposes a self-supervised spoofing audio detection scheme(SSAD). In SSAD, eight convolutional blocks are used to capture the local feature of the audio signal. The temporal convolutional network (TCN) is used to capture the context features and realize the operation in parallel. Three regression workers and one binary worker are designed to achieve better performance in fake and spoofing audio detection. The experimental results on ASVspoof 2019 dataset show that the detection accuracy of SSAD outperforms the state-of-art. It shows that the self-supervised method is effective for the task of spoofing audio detection.

Index Terms: self-supervised learning, ASVspoofing detection, anti-spoofing, deepfake

1. Introduction

The rapid advancement of speech synthesis technology contributes to the crimes of synthesizing spoofing audio to fool others, which poses a significant threat to the global political economy and social stability. Speech carries a lot of language information. Therefore, audio spoofing technology is usually used to control public opinion. What is more, generative adversarial networks' progress makes spoofing audio more realistic, dramatically challenging spoofing audio detection.

The traditional spoofing audio detection schemes mainly depend on the differential characteristics of biological information such as speech rate, voiceprint, and spectrum distribution. Nowadays, deep neural networks have become the mainstream method. Wu [1] proposed a light neural network (Light CNN) using a max-feature-map (MFM) activation function to learn a light model that is efficient in computational costs and storage spaces. Gomez-Alanis [2] proposed a light convolutional gated recurrent neural network (LC-GRNN) by merging Light CNN [1] and RNN based on gated recurrent units and used LC-GRNN as a deep feature extractor. LC-GRNN combines advantages of Light CNN and RNN, which is capable of extracting discriminative features at the frame level as well as learning contextual features. Li [3] proposed a spoofing audio detection architecture based on multiple features integration and multi-task learning (MFMT), which uses the Mel frequency cepstrum coefficient (MFCC), constant Q cepstral coefficient(CQCC), and power spectrogram as its features. What is more, it completes multi-task learning based on the butterfly unit (BU), improving the scheme's generalization capacity. Based on the differential characteristics between the real and fake audio spectrum, Dessa [4] proposed a deep forged speech detection model based on the spectrogram, which uses a temporal convolutional network (TCN) [5]. On the competition dataset [6] in ASVspoof 2019, the model's accuracy can reach 90%. Besides, many professional teams participated in the ASVspoof 2019 challenge [7], in which they presented their excellent results. As shown in the examples above, their schemes are mainly based on supervised learning.

Recently, self-supervised learning has been widely used in natural language processing [8] and computer vision [9, 10]. BERT [8] is designed to pre-train deep bidirectional representations from an unlabeled text by jointly conditioning on both left and right contexts in all layers. As a result, the pre-trained BERT model can provide state-of-the-art results on many natural language processing tasks. Some models like MoCo [9] and SimCLR [10] witnesses competitive results in self-supervised visual representation learning. The learned feature representation can be well applied to downstream tasks. Pascual [11] proposed a multi-task self-supervised approach called PASE to learn problem-agnostic high-level speech representations in speech signal processing. An improved version of PASE, called PASE+ [12], has an excellent performance in speech recognition, such as speaker identification, emotion classification, and automatic speech recognition.

Inspired by PASE+ [12], in this paper, we propose SSAD, a self-supervised spoofing audio detection scheme consisting of an encoder and two kinds of workers. SSAD encodes the raw speech waveform into a representation by which the regression workers and the binary worker are fed. Regression workers aim to predict the target features computed from the input waveform. The goal features we choose, such as LPS, LFCC, and CQCC, are specially designed to suit our task. After training, the workers can minimize the mean squared error (MSE) between the target features and the network predictions. Binary task worker deals with either positive or negative samples and separates them by training. SSAD introduced a binary task named congener info max (CIM). CIM aims to minimize the distance between two similar kinds of audio and maximizing the distance between two different kinds of audio. Both regression and binary task workers contribute to helping the encoder discover the higher-level representations, which proves to be crucial to derive both meaningful and robust representations.

To the best of our knowledge, SSAD is the first application of multi-task self-supervised learning in spoofing audio detection. Moreover, our experiments suggest that SSAD performs well in spoofing audio detection. The result, tested on ASVspoof 2019 datasets, shows that SSAD significantly outperforms the state-of-the-art schemes and has high transferability when combined with different spoofing audio detection schemes.

The paper is organized as follows. Section 2 describes the details of SSAD's architecture. In section 3, we introduce the dataset used in our experiments and how we conduct the experi-

* corresponding author

http://dx.doi.org/10.21437/Interspeech.2020-1760

ments. In section 4, we compare our results with previous work and show SSAD's excellent transferability and generalization capacity. Finally, we conclude this paper in section 5.

2. Self-Supervised Learning with SSAD

The SSAD architecture, shown in Figure 1, is equipped with an encoder and two kinds of workers (three regression workers and one binary worker). The encoder includes eight convolutional blocks, a temporal convolutional network (TCN) layer, and a nonlinear projection layer. All of these workers are based on small feed-forward neural networks. In this section, we describe these modules in detail. Particular attention should be paid to the parts where we make specific modifications in our architecture for better performance.

2.1. SSAD Encoder

Firstly, we use eight convolutional blocks to perform the convolution of the raw input waveform chunks. Each block is composed of a one-dimensional convolution (Conv1d), followed by batch normalization (BN) [13] and a multi-parametric rectified linear unit (PReLU) activation [14]. We also introduce the skip-conn of the intermediate convolution layers to transfer different levels of abstractions to the final representations for improving gradient flows, and also set convolutions' sliding window with a shift of 10 ms. SSAD modifies the encoder's architecture in [12] as follows:

1) Temporal convolutional network(TCN): with a TCN placed on the top of the convolutional layers, SSAD can learn long-term dependencies more efficiently. Like the time-delay neural network (TDNN), TCN is a unique convolution neural network that combines the causal convolutions, dilated convolutions, and the residual connections structure. For a 1-D sequence input $x \in \mathbb{R}^n$ and a filter $f : \{0, \cdots, k-1\} \rightarrow \mathbb{R}$, the dilated convolution operation F on element s of the sequence is defined as

$$F(s) = (x \cdot_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \qquad (1)$$

where d is the dilation factor, k is the filter size, and $s - d \cdot i$ accounts for the direction of the past. Dilation is thus equivalent to introducing a fixed step between every two adjacent filter taps. The TCN can be computed in parallel and has a flexible receptive field size to adapt to our task. The TCN architecture appears more accurate than recurrent canonical networks such as LSTMs and GRUs [5], but also simpler and more explicit.

2) Nonlinear projection: in contrastive learning tasks, a nonlinear projection performs better than a linear projection in the encoder's output layer. As shown in [10], the default nonlinear projection with one additional hidden layer (and ReLU activation), similar to [15], improves the representations' quality of the layer before it.

2.2. Workers

Workers are fed by the encoded representation produced by the frontend model. Moreover, regression or discrimination tasks are solved by workers as self-supervised tasks (Figure 1). Then the average error of the workers propagates back to help the encoder discover better high-level representations. The workers we choose are based on small feed-forward networks (one hidden layer with 256 hidden units with PReLU activation). It is noteworthy that using simple workers can significantly encourage the encoder, for the encoder can discover high-level features that can be well decoded even by workers with limited capacity.

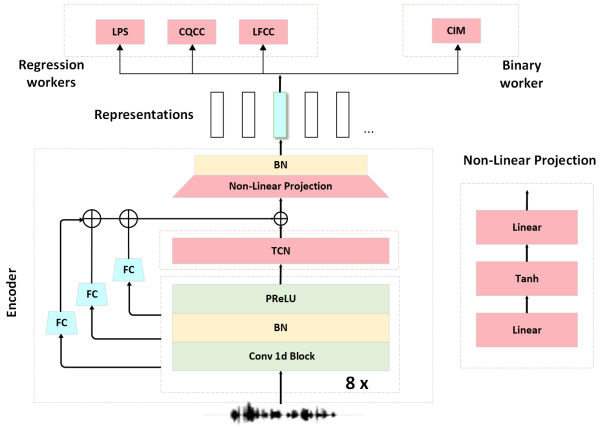

Figure 1: *The proposed SSAD architecture. In pink are the main differences with PASE+ [12].*

For our task of spoof audio detection, we here propose self-supervised tasks as follows:

2.2.1. Regression Tasks

The regression workers aim to predict the target features extracted by standard ways using librosa [16] or script designed by us. Additionally, we specially choose and design the goal features to suit spoofing audio detection using the parameters stated below. These workers are trained to minimize the mean squared error (MSE) between the target features and the network predictions. We choose the following three regression workers to conduct these tasks:

1) Log power spectrum(LPS): we compute the LPS from the original speech signals without voice activity detection (VAD), pre-emphasis, or dereverberation. We set the NFFT to 1724 and hop length to 130 to capture more local information.

2) Log-frequency cepstral coefficients(LFCC): 20 coefficients from 20 log filter banks (FBANKs) are extracted from the speech signal chunk. And then, the first and second order deltas of LFCC are computed and concatenated with the original LFCC.

3) Constant Q cepstral coefficients(CQCC): in the ways shown at [17], 29 CQCC coefficients are extracted with first and second order deltas computed.

2.2.2. Binary Task

We design the binary task for one worker to capture higher-level abstraction from speech signals. The encoder and the binary task worker cooperate to derive good representations. This task works by the defined sampling strategy that draws an anchor s_a, a positive s_r, and a negative s_f from the set of SSAD-encoded representations available in the training set. We draw the anchor from the encoded feature extracted from a random audio sentence while sampling the positive from encoded features extracted from a random congeneric speech and sampling the negative from a random speech of different types. Thus we introduce a binary task named congener info max (CIM). CIM is dedicated to minimizing the distance between two similar kinds of speeches (defined as $L1$), but maximizing the distance

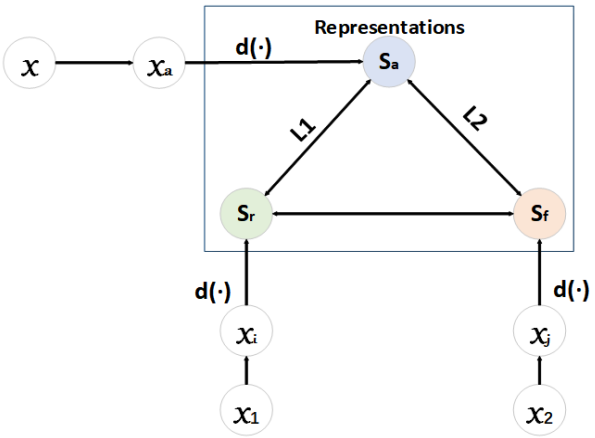

Figure 2: *A binary task framework for contrastive learning of high-level representations. S_a and S_r are representations extracted by the encoder $d(\cdot)$ from chunks sampled from random congeneric speech, while S_f is from a random speech of the other type. Then cross-entropy loss is computed between S_r, S_f, and S_a to minimize the distance between two similar kinds of speeches and maximize the distance between two different kinds of speeches.*

between two different kinds of speeches (defined as $L2$). As shown in Figure 2, we use a formulation of the cross-entropy to quantify this distance:

$$L1 = E_{S_r}[log(d(s_a, s_r))] \tag{2}$$

$$L2 = E_{S_f}[log(1 - d(s_a, s_f))] \tag{3}$$

$$L = L1 + L2 \tag{4}$$

where d is the discriminator function, and E_{S_r} and E_{S_p} denote the expectation over positive and negative samples. Obviously, by minimizing L, our model is trained to link the anchor and positive example more closely than negative. As we know, L, equal to mutual information, is a significant measure of divergence that can capture complex nonlinear relationships between random variables [18, 19]. In our work, we mainly rely on CIM to help the binary worker learn how to discriminate between bonafide and spoof speeches.

2.3. Details of Training SSAD

The workers' learning rate is initially set to 0.5×10^{-3} and then will decline continuously, which depends on a polynomial scheduler [20]. Adam [21] serves as an optimizer. The average loss of all workers propagates back to the encoder to optimize its parameters. Furthermore, we use mini-batches of 16 waveform chunks, each of which is 2 seconds long. All the SSAD systems in our experiments are trained for 100 epochs in an Nvidia Tesla V100 GPU. The training time for each experiment is about 5 days.

3. Corpora and Task

3.1. Dataset

We conduct our experiment using the ASVspoof 2019 dataset [22], which encompasses two partitions for the assessment of logical access (LA) and physical access (PA) scenarios. In this paper, we focus on the LA partition originating from the VCTK base corpus, which includes speech data recorded from 107 speakers, containing 46 males and 61 females. LA partition owns three datasets, named by training, development (Dev), and evaluation (Eval) [23]. Importantly, the spoofing methods of the Dev set and training set are the same, while the Eval set has 11 unknown attacks. As known to us, some models perform exceedingly well in the Dev set but perform not so good in the Eval set for overfitting. Thus in Sec.4, we use the equal error rate (EER) computed in the Eval set to measure the generalization capacity of the scheme we propose and the schemes for comparison.

In self-supervised training, we split the training set into three parts. The training part contains 20,558 audio, the validation part contains 2,284 audio, and the test part contains 2,538 audio. All of them are sampled with a length of 32,000 frames when fed into SSAD. This split corresponds to approximately 19 hours for training, 2 hours for validation, and 2 hours for testing.

3.2. Classifiers

Our work adopts LCNN-big, LCNN-small, and SENet12 in [24] as classifiers to evaluate SSAD's transferability when combined with different model structures. The architecture of LCNN-big is the same as in [25], and LCNN-small is similar to LCNN-big with fewer parameters. As for SENet12 architecture, it is similar to that in [26], while it has fewer parameters like the one in [24]. The number of trainable parameters of these three models is shown in Table 1. LCNN-big owns the largest model capacity among these three models, while LCNN-small and SENet12 are smaller than it.

Table 1: *Number of trainable parameters of LCNN-big, LCNN-small and SENet12 model*

	SENet12	**LCNN-big**	**LCNN-small**
Parameters	478546	3189536	158608

In the training process, LCNN-big and LCNN-small model exploit the angular softmax (A-Softmax) loss function to enhance the anti-spoofing performance as [24] has done. Meanwhile, the SENet12 model uses original softmax and cross-entropy loss. A-Softmax loss function is represented as:

$$L_{ang} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|X_i\|\cos(m\theta_{y_i,i})}}{e^{\|X_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|X_i\|\cos(\theta_{y_j,i})}}\right) \tag{5}$$

where N is the number of training samples, $\{x_i\}_{i=1}^N$ and their labels $\{y_i\}_{i=1}^N$ are training pairs, $\theta_{y_i,i}$ is the angle between x_i and the corresponding column y_i of weights W in the fully connected classification layer, and m is an integer that controls the size of an angular margin between classes.

3.3. Our Task Outline

To evaluate the performance and generalization capacity of the learned representations, we conduct our experiments by inputting different features (containing the ones extracted by SSAD) into different classifiers. We use the equal error rate (EER) as evaluation criteria for results. Results shown as followed suggest that we do make much progress.

4. Results

4.1. Comparison with PASE+ [12]

Experiment results have proved our modifications to be valid and effective. The first three rows in Table 2 shows the EER(%) obtained on the ASVspoof 2019 dataset with the PASE+ [12], SSAD(GRU), and SSAD(TCN) while SENet12 serves as a classifier. Then the next four rows show the comparison between SSAD and some results in the ASVspoof 2019.

The third row shows the results of SSAD(TCN), while SENet12 serves as a classifier. Significantly, the results outperform the original PASE+ [12] with a relative improvement of 2.49% in the Eval set when we attempt to substitute the workers and use TCN. Besides, the best result in the Dev set is performed by SSAD(GRU), but SSAD(TCN) performs best in the Eval set. It means TCN helps the encoder learn representations with more generalization capacity. The results prove that our work enables the encoder to learn better representations from contrastive samples, which is of great importance in self-supervised training tasks.

The 4th and 5th rows are the results achieved by the baseline in the ASVspoof 2019 [7]. The 6th row shows the best result of the teams using a single classifier. It is worth noting that our approach outperforms team T04's by 0.43% and significantly outperforms the baseline.

Table 2: *The first three rows in Table 2 shows the EER(%) obtained on the ASVspoof 2019 dataset with the PASE+ [12], SSAD(GRU), and SSAD(TCN), while SENet12 serves as a classifier. Then the next four rows show the comparison between SSAD and some results in the ASVspoof 2019.*

		Dev	Eval
PASE+	+ SENet12	1.51	9.04
SSAD(GRU)	+ SENet12	0.15	7.24
SSAD(TCN)	+ SENet12	0.47	**6.55**
baseline(LFCC)		-	8.09
baseline(CQCC)		-	9.57
team T04		-	5.74
SSAD	+ LCNN-big	0.78	**5.31**

4.2. Comparison with common speech features

In this experiment, we compare SSAD representations with characteristic features such as LPS, LFCC, and CQCC [17]. These are the most common speech features used in ASV schemes, and it is relatively uneasy to find alternatives that outperform them. To provide a fair comparison, we extract these features from speech chunks sized 2s. CQCCs and LFCCs are also concatenated with their first and second derivatives. The difference is that standard acoustic features are pre-extracted, while the SSAD's encoder is combined with the classifier when training. It is worth noting that SSAD is an end-to-end model. Thus we can directly train the classifier with the original speech signals.

Table 3 reports the EER (%) obtained with the baseline in [7], SENet12, LCNN-small, and LCNN-big utilizing different features. Since that LCNN's architecture is designed for LPS, we do not conduct experiments by feeding LFCC and CQCC into LCNN. Besides, the results of the baseline system (base on GMM) originate from [7]. All systems are trained for 10 epochs, and the recorded results are the lowest EER (%) of each system. The results shown in the table suggest the excellent transferability of self-supervised SSAD features when fed into different classifiers. The Eval set's best result is achieved by LCNN-big, a vast neural network architecture when fed by SSAD's representations. Meanwhile, small architectures, SENet12 and LCNN-small, also show outstanding performances in the Eval set when SSAD serves as the feature extractor. Our results turn out that SSAD's high-level representations perform good transferability when fed into different types of classifiers.

Table 3: *EER(%) obtained on the ASVspoof 2019 dataset with different spoofing audio detection schemes. Rows report EER(%) when using different types of common features or SSAD.*

	baseline	**SENet12**		**LCNN small**		**LCNN big**	
	Eval	Dev	Eval	Dev	Eval	Dev	Eval
LPS	-	0.04	9.27	0.09	9.76	0.12	6.82
LFCC	8.09	1.73	8.45	-	-	-	-
CQCC	9.57	8.01	15.14	-	-	-	-
SSAD	-	0.47	**6.55**	0.86	**7.16**	0.78	**5.31**

5. Conclusion

This paper proposes a multi-task self-supervised learning scheme for spoofing audio detection, which is composed of an encoder and two kinds of workers (three regression workers and one binary worker). The encoder includes eight convolutional blocks, a TCN layer, and a nonlinear projection layer. These workers are based on small feed-forward neural networks to help the encoder discover high-level features with higher capacity. As is shown in our experiment results, SSAD turns out to significantly outperform those features extracted by supervised learning and usually used in ASVspoof's task. Our experiments conducted in the ASVspoof 2019 dataset indicate that SSAD performs outstanding generalization capacity. Besides, SSAD offers amazing transferability when combined with different types of classifiers. Our work shows the great potential of self-supervised representations in the spoofing audio detection field. In our future work, we will improve our network architecture to better use the advantages of self-supervised learning and increase SSAD accuracy.

6. Acknowledgements

This work is supported by the Natural Science Foundation of China (NSFC) under the grant NO. 61872275, U1836112, 61876134, and the China Scholarship Council.

7. References

[1] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[2] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A light convolutional gru-rnn deep feature extractor for asv spoofing detection," *Proc. Interspeech 2019*, pp. 1068–1072, 2019.

[3] R. Li, M. Zhao, Z. Li, L. Li, and Q. Hong, "Anti-spoofing speaker verification system with multi-feature integration and multi-task learning." in *Interspeech*, 2019, pp. 1048–1052.

[4] (2019). Detecting Audio Deepfakes With AI [On-Line]. Available: https://medium.com/dessanews/detecting-audio-deepfakes-f2edfd8e2b35.

[5] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[6] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, "The asvspoof 2019 database," *arXiv preprint arXiv:1911.01601*, 2019.

[7] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," *arXiv preprint arXiv:1911.05722*, 2019.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[11] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *arXiv preprint arXiv:1904.03416*, 2019.

[12] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," *arXiv preprint arXiv:2001.09239*, 2020.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[15] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Advances in Neural Information Processing Systems*, 2019, pp. 15 509–15 519.

[16] B. M. et al., "librosa/librosa: 0.7.2," Jan. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3606573

[17] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, 2017.

[18] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[19] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018.

[20] R. Ge, S. M. Kakade, R. Kidambi, and P. Netrapalli, "Rethinking learning rate schedules for stochastic optimization," 2018.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 019–10 029.

[23] "Asvspoof 2019: The 3rd automatic speaker verification spoofing and countermeasures challenge database," https://datashare.is.ed.ac.uk/handle/10283/3336.

[24] S. Liu, H. Wu, H.-y. Lee, and H. Meng, "Adversarial attacks on spoofing countermeasures of automatic speaker verification," *arXiv preprint arXiv:1910.08716*, 2019.

[25] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "Stc antispoofing systems for the asvspoof2019 challenge," *arXiv preprint arXiv:1904.05576*, 2019.

[26] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "Assert: Anti-spoofing with squeeze-excitation and residual networks," *arXiv preprint arXiv:1904.01120*, 2019.