Interspeech 2024
1-5 September 2024, Kos, Greece

Exploring Self-supervised Embeddings and Synthetic Data Augmentation for Robust Audio Deepfake Detection

Juan M. Martín-Doñas[1], *Aitor Álvarez*[1], *Eros Rosello*[2], *Angel M. Gomez*[2], *Antonio M. Peinado*[2]

[1]Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), San Sebastián, Spain
[2]Dept. Signal Theory, Telematics and Communications and CITIC, Universidad de Granada, Spain

{jmmartin,aalvarez}@vicomtech.org, {erosrosello,amgg,amp}@ugr.es

Abstract

This work explores the performance of large speech self-supervised models as robust audio deepfake detectors. Despite the current trend of fine-tuning the upstream network, in this paper, we revisit the use of pre-trained models as feature extractors to adapt specialized downstream audio deepfake classifiers. The goal is to keep the general knowledge of the audio foundation model to extract discriminative features to feed up a simplified deepfake classifier. In addition, the generalization capabilities of the system are improved by augmenting the training corpora using additional synthetic data from different vocoder algorithms. This strategy is also complemented by various data augmentations covering challenging acoustic conditions. Our proposal is evaluated under different benchmark datasets for audio deepfake and anti-spoofing tasks, showing state-of-the-art performance. Furthermore, we analyze the relevant parts of the downstream classifier to achieve a robust system.

Index Terms: audio deepfake detection, anti-spoofing, self-supervised models, data augmentation, vocoders

1. Introduction

The research on audio deepfake detection algorithms has become critical in recent years due to the astounding speech quality achieved by novel speech synthesis and voice conversion algorithms [1]. These detection systems can help protect against malicious use of deepfakes for misinformation and identity theft, and even as countermeasures for automatic speaker verification systems [2, 3]. Thus, several initiatives such as the ASVspoof series [4, 5] or the recent ADD challenges [6, 7] have been promoted to develop more robust models able to cope with different attacks and acoustic conditions, but obtaining a generalizable model to unknown deepfakes is still a challenging task.

In recent years, the proliferation of large self-supervised learning (SSL) speech foundation models [8] has impacted the speech community, including the anti-spoofing and audio deepfake research areas. Preliminary works showed that SSL models such as wav2vec2 [9] were able to outperform supervised deep learning classifiers when used as feature extractors [10] or fine-tuned for the downstream task [11, 12]. As a result, research interest moved towards the development of deepfake detectors built upon these SSL architectures, focusing on the following aspects: improving the downstream network [13, 14], using different upstreams such as WavLM [15] or Whisper [16], exploring multi-corpus with domain-invariant training [17], or parameter-efficient fine-tuning of the SSL network [18]. In parallel, other works have focused on techniques to extend the training data used for these models. For example, [19] proposed an active learning approach to select representative audio samples from a data pool. Wang et al. [20] explores spoofing data augmentation by generating new deepfake samples directly using vocoder systems, while [21] analyzed the system performance under known attack variations for synthetic data training. Recently, [22] showed state-of-the-art (SOTA) performance when using large-scale vocoded spoofed data for continuous SSL training. Indeed, the combination of large SSL models and synthetic data augmentation techniques is a promising research line for achieving robust audio deepfake detection.

In this work, we revisit the use of pre-trained SSL models as deep embedding extractor networks, where only the downstream classifier is fine-tuned for the anti-spoofing task. Following the SUPERB setup [23], we consider the hidden embeddings from the SSL encoder layers to obtain a more discriminative representation for the target task, as proposed on [10] and other speech-related tasks [24, 25]. Thus, we propose a simple yet effective downstream classifier with few processing blocks, the impact of which in the final performance are also investigated. The downstream is fine-tuned on the well-known ASVspoof 2019 dataset [4] considering data augmentation techniques to improve the performance on varying acoustic conditions. Moreover, the training data is extended by using spoofing attack augmentation with several vocoders applied over the in-domain clean data [20], avoiding the need to create large-scale spoofed data from additional corpora to fine-tune the SSL upstream. We hypothesize that the general audio representations from the SSL model can be used as highly discriminative features when applied along with a properly fine-tuned classifier with heterogeneous and representative training data covering different acoustic conditions and deepfake generation algorithms. Our straightforward approach is evaluated under different anti-spoofing and audio deepfake benchmarks, showing SOTA results and outperforming previous methods regarding generalization capabilities across diverse unseen test sets using a single detection system.

The rest of the paper is organized as follows. Section 2 describes the proposed architecture, including the upstream SSL and downstream classifiers, as well as the acoustic and spoofing data augmentation techniques used during training. The experimental framework and results are presented and analyzed in Section 3. Finally, the final conclusions and future work are summarized in Section 4.

2. Audio deepfake detection models and data augmentation

The architecture of our proposed audio deepfake detection network is depicted in Figure 1. In the next subsections, we describe the different processing blocks involved in both the upstream and downstream models. In addition, the strategies for synthetic data augmentation are also explained.

10.21437/Interspeech.2024-942

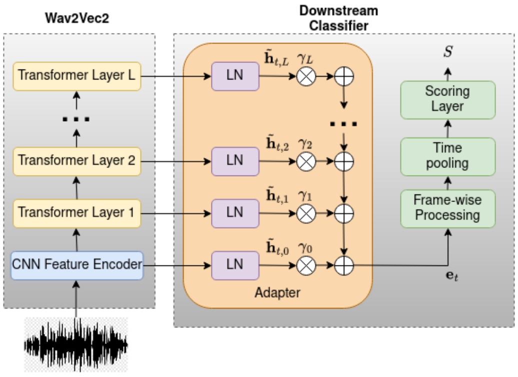

Figure 1: *Architecture of the proposed audio deepfake detection system based on pre-trained Wav2Vec2.*

Table 1: *Number of parameters for the different blocks and their variants. For the scoring layer, it depends on the dimension of the output vector from the time pooling.*

Block	**Params.**
Wav2Vec2	315M
Adapter	25
Frame-wise proc.	
Proj	262K
NN	328K
Temp. pool.	
SP	0
ASP/ACP	66.8K
Scoring layer	
*SP/ASP	65.8K
*ACP	4.2M

2.1. Self-supervised speech embedding extractor

In this work, we considered the Wav2Vec2 XLS-R-128 model proposed in [26] as the SSL embeddings extractor. This 315M based SSL model consists of a feature encoder of 7 convolutional layers and a Transformer encoder of L layers ($L = 24$). It is trained in a cross-language setting covering 128 different languages. We chose this SSL as it has shown good performance in previous related works, but other models like WavLM or Whisper could effortlessly be integrated into our approach. This upstream model extracts 1024-dimension embeddings each 20 ms. However, we consider the representations from all the different encoder layers, including the convolutional one, which gives a representation per sequence of dimension (T, 1024, 25), where T is the embedding sequence length. It is worth mentioning that we use variable length sequences, using zero-padding during training to fit batches. Thus, the sequence padding mask is given as input to the upstream model to avoid convolutional and attention computations on the padding parts, and the resulting embedding padding mask is used in the following downstream model.

2.2. Audio deepfake downstream classifier

The SSL sequence representation is processed by the downstream classifier, which is fine-tuned to discriminate between genuine and fake audio. Our proposed classifier consists of several blocks related to different processing tasks. In the following, we describe each block, including the different neural network architectures we considered for each of them:

- *Adapter*: This block combines the sequence representations from the different SSL layers into a single embedding per frame, giving a sequence of (T, 1024). First, a layer normalization (LN) is applied to the embeddings (without element-wise affine transformation). Then, the output representation is computed as $\mathbf{e}_t = \sum_{l=0}^{L} \gamma_l \tilde{\mathbf{h}}_{t,l}$, where $\tilde{\mathbf{h}}_{t,l}$ are the normalized hidden representations, and t and l are the time and layer indexes, respectively. The weights γ_l are obtained through a learnable vector α_l fed into a softmax layer.
- *Frame-wise processing*: The next block performs dimensionality reduction on each time embedding. A straightforward technique for achieving this is through an affine projection (Proj), which, in our implementation, reduces the embedding dimensionality to 256. Alternatively, a more sophisticated approach that we have also explored involves incorporating a ReLU activation, followed by dropout, and a subsequent affine transformation with the same dimensionality. We will denote this variant as NN, which has the potential to facilitate the learning of more complex representations.
- *Time pooling*: We need to summarize the time embeddings into a single representation for each sequence. A common practice is statistical pooling (SP), which computes the time mean and standard deviation (factoring in the padding mask) per feature dimension and concatenates them into a single 512-dimensional vector. However, as some time embeddings could be more informative than others, a multi-head attention mechanism is also considered. Following [25], the attention weights for each head are computed by feeding the embeddings into two affine transformation layers with an intermediate ReLU activation. The output dimensions of these transformations are 256 and H, respectively, with H denoting the number of heads (we used $H = 4$ heads). The time outputs of each head are combined through *logsumexp* operation (soft maximum) and normalized via a softmax layer to obtain the final attention weights (considering also the padding mask). Apart from this attentive SP version (ASP), we also evaluated the attentive correlation pooling (ACP) proposed in [25], which considers the normalized cross-correlation terms (the upper-diagonal terms of the correlation matrix) vectorized into a single 32640-dimensional vector. In this case, channel dropout is applied to the time embedding sequence before the attention computation and pooling for regularization purposes.
- *Scoring layer*: Finally, the last block of the downstream computes a single score S for the audio sample, which is eventually used to classify the audio as genuine or fake. The dimension of the output vector for the time pooling block is first reduced to 128 via an affine transformation. Then, the score is computed as the cosine similarity between this vector and a learnable vector network parameter $\mathbf{w}$ as in [10], where $\mathbf{w}$ represents the direction of genuine samples in the corresponding vector space.

Table 1 shows the number of parameters for each processing block, including the different variants evaluated. As it can be observed, the fine-tuned parameters represent a small fraction of the whole architecture, where the pre-trained SSL model, frozen during training, contributes to nearly 99% of the total parameters of the audio deepfake detection system.

2.3. Acoustic and spoofed data augmentation

The robustness and generalization capabilities of an audio deepfake detection system mainly depend on the variety of training data used regarding speakers, acoustic conditions, spoofed attacks, among others. Therefore, we explored the use of acoustic and synthetic data augmentation on the training database to improve the performance of the resulting systems.

First, the acoustic data augmentation is achieved using the RawBoost technique proposed in [27], which applies several perturbations over the speech signals, including convolutive (linear and non-linear), impulsive and stationary noises. This technique has shown good performance in improving robustness on different conditions, such as telephonic channels or codified audio, but it also contributes to better generalization capabilities with out-domain data. On the other hand, previous works [28] have pointed out that the ASVspoof 2019 has artifacts related to the silence duration on genuine and fake samples that can be exploited during training, preventing learning more relevant features that could better generalize. Thus, we also explored an additional pre-processing for the training samples, which involves trimming the leading and trailing silences.

Furthermore, we considered [20] for spoofing data augmentation. This approach creates new fake samples from the genuine speech in the training data using only vocoder techniques, disregarding the more complex acoustic models frequently used in speech synthesis or voice conversion methods. This paper considers the following vocoders: HiFiGAN, WaveGlow, Harmonic-plus-noise neural source filter (Hn-NSF), and the fusion of Hn-NSF and HiFiGAN. In contrast to [20], we extended the original training data by including these new spoofed attacks to increase variability. Moreover, we do not rely on external large-corpus to create spoofed data for continuous SSL training of the upstream model as in [22], but the in-domain extended training corpus is only used for fine-tuning the downstream classifier. The idea is to cover a broader range of conditions so that the classifier learns better to detect deepfake samples while exploiting the general audio representations from the pre-trained upstream. As the SSL model is frozen, we avoid over-fitting or catastrophic forgetting issues during the continuous SSL pre-training and fine-tuning.

3. Experimental results

3.1. Experimental framework

Datasets: As it was previously mentioned, in this work, we used the ASVspoof 2019 LA train and development sets [4] for the training and validation of the models. These sets contain about 25K audio samples from 20 and 10 speakers, covering clean English genuine audio data and 6 different spoofing attacks. In addition, we extended this dataset using the spoofed data of the Voc.v4 partition[1] released in [20] including 4 different vocoders fine-tuned in the ASVspoof 2019 real data. This process incorporates an additional 10K spoof samples into both the training and development sets.

For the evaluation, multiple test datasets are considered to measure the performance in both in- and out-domain conditions. The test set of ASVspoof 2019 LA [4] is included (similar conditions but different spoofing attacks), as well as the evaluation sets of ASVspoof 2021 LA (telephonic channels) and DF (audio codecs and novel data from voice conversion challenges) [5]. To evaluate the impact of silence duration artifacts, we also considered a version of the 2019 LA test in which initial and end silences were removed, as well as the hidden sets of 2021 LA and DF. Finally, to further evaluate the performance with out-domain datasets, we also included WaveFake [29] (two female speakers, English and Japanese) and In-the-Wild [30] (over 50 English politicians and celebrities) datasets.

Training setup: The different models were trained using the one-class softmax loss function [31], dropout rates of 0.2 when applicable, and the ADAM optimizer [32] with a learning rate of $3 \cdot 10^{-4}$. Only the downstream parameters are trained, while the upstream is frozen. During training, a batch size of 8 audio samples was used with gradient accumulation across 8 consecutive batches (i.e., a total effective batch of 64 samples). The audio duration was set to a maximum of 8 seconds, removing the remaining samples if longer. We validated the models at each epoch on the development set and stopped the training after 10 epochs without improvements. Moreover, we trained 3 model instances with different seeds in order to provide averaged performance metrics. The experiments were done using the Pytorch-lightning library [33], and the wav2vec2 model was obtained from its HuggingFace repository[2]. Finally, the trainings were done in a Nvidia A40 48 GB GPU[3].

Evaluation metrics: We compared the different approaches in terms of the equal error rate (EER) on both individual and pooled test sets. To this end, we considered the averaged EER resulting from the 3 different model instances previously trained. To ensure the results are statistically significant, we followed the same methodology as in [34] for the pair-wise comparison of the models at a 95% significance level with Holm-Bonferroni correction.

3.2. Results and analysis

Table 2 shows the test results for the different configurations of our proposed approach in terms of downstream classifier architecture, data augmentations, and a comparison with SOTA models. In the following, we analyze these results.

Data augmentations: We first considered NN-ASP/ACP downstreams and evaluated using trimming and training with extended vocoded data. Trimming the silences impacts on ASVspoof 2019 and 2021 LA, where trained models can exploit these undesired artifacts. Nevertheless, it improves results both for in-domain corpus with trimmed silences and out-domain sets in the NN-ASP classifier, while the impact is less significant or even harmful when using ACP. On the other hand, training with additional vocoded data has potential benefits for the model performance, improving the results in ASVspoof 2021 DF and out-domain datasets, and yielding astounding pooled test results. Adding new types of fake data allows to encompass a wide specturm of deepfake attacks, particularly in out-domain datasets where this vocoded data align with similar attacks included in those sets. Simultaneously, this preserves high performance levels in the attacks featured in ASVspoof 2019, with potential improvements observed even in its trimmed version. These improvements are not evident in ASVspoof 2021 LA, likely due to the main degradation influenced by the effect of telephonic channels, which are not directly impacted by the introduction of new vocoders. Nevertheless, with the exception of LA eval 2019/2021, using both data augmentation yields the most optimal performance across individual and pooled sets.

[1] https://zenodo.org/records/7315515

[2] https://huggingface.co/facebook/wav2vec2-xls-r-300m

[3] We would also like to point out that the training models fit in a Nvidia TITAN X 12 GB, but this yields higher training times.

Table 2: *Average EERs (%) results on the evaluated audio deepfake detection test set. We indicate if trimming (Trim.) or additional vocoding data (Voc.) are used in our models. Results in **bold** are not statistically significant compared to the best model (**underlined**).*

Classifier	Trim.	Voc.	asv19LA		asv21LA		asv21DF		IntheWild	WaveFake	Pooled
			Eval	Trim	Eval	Hidden	Eval	Hidden			
NN-ASP	×	×	**0.22**	6.71	**1.90**	44.03	7.90	36.95	11.10	9.33	9.90
	✓	×	5.04	5.56	8.15	12.50	3.31	11.45	9.49	8.36	8.53
	×	✓	**0.14**	**2.81**	8.26	14.39	1.46	**5.64**	3.76	2.82	**5.31**
	✓	✓	2.42	**2.52**	9.32	**10.10**	**1.28**	**5.76**	**3.55**	2.24	**5.13**
NN-ACP	×	×	**0.19**	5.63	**1.83**	13.87	4.34	11.84	11.09	5.79	8.51
	✓	×	7.47	8.09	9.58	13.77	4.26	13.08	10.27	8.44	9.06
	×	✓	**0.22**	3.27	11.08	13.19	1.59	**6.51**	**3.59**	4.04	5.79
	✓	✓	2.30	**2.54**	9.78	**9.91**	**1.36**	**6.06**	**3.10**	1.90	5.35
Proj-SP	✓	✓	2.44	**2.36**	10.04	**9.84**	**1.30**	**5.64**	**3.39**	2.27	5.71
NN-SP			2.42	**2.69**	9.96	**10.56**	**1.36**	**5.55**	3.96	1.83	5.65
Proj-ASP			3.18	3.14	8.66	**10.43**	**1.16**	**5.99**	**3.27**	2.12	**5.23**
Proj-ACP			2.46	**2.70**	7.79	**9.47**	1.55	**6.69**	**3.55**	2.23	**5.25**
NII-B1 [22]			3.45	**2.69**	17.59	13.93	6.53	8.89	6.78	7.33	11.13
NII-B1-b [22]			**0.22**	7.37	2.69	15.56	4.27	9.16	13.52	23.75	12.76
NII-P1 [22]			2.09	3.33	16.88	16.02	4.34	7.71	5.84	1.94	10.54
NII-P3 [22]			1.91	3.28	15.92	14.97	5.67	8.84	6.10	**1.30**	9.98

Downstream architecture: We also evaluated the different combinations for the downstream classifiers in terms of frame-wise processing and time pooling. Regarding the former, using Proj or NN variants does not yield significant differences in performance, indicating that SSL embeddings are a discriminant feature requiring few transformations to be exploited by the classifier. On the other hand, when comparing time poolings, the attentive mechanism does not particularly affect individual datasets. Still, the effect is significant when evaluating the pooled test, suggesting that the operating point for the EER is more similar when using these models. Therefore, we conclude that attention mechanisms contribute to the development of more generalizable models across different sets. There are no significant performance differences between ASP and ACP, but the former requires less trainable parameters. Thus, Proj-ASP is a preferred solution for downstream implementation, with NN-ASP as alternative candidate in some particular conditions.

Comparison with SOTA models: Finally, we compare our approach with several systems presented by NII labs in [22], including SOTA performance methods when evaluating in this multi-dataset setup. Both B1 and B1-b systems are based on the Wav2Vec2 XLS-R-53 model, and the complete system is fine-tuned either using the vocoded Voc.v4 data (and corresponding genuine data) or the ASVspoof 2019 train set. The P1 system is like B1 but also considers vocoded data from a large external corpus, VoxCeleb2, to first perform continuous SSL training on the upstream model. Their proposed P3 system uses the embedding differences between the original and SSL-trained model on spoofed data, distilling the information to a different upstream through a teacher-student framework. This avoids using two different upstream networks. Our proposed approach outperforms the continuous SSL training strategy proposed in [22] by extending the original training set with the vocoded data without requiring an additional large corpus for pre-training the upstream. The P3 system still yields the best results on WaveFake, but our models achieve better results on the remaining test sets and the best overall performance (5.13% vs 9.98% EER). Moreover, our approach directly exploits the hidden embeddings from a pre-trained SSL model while fine-tuning the downstream classifier with the task domain datasets. Moreover, even when only the original training data for ASVspoof 2019 is used (i.e., without additional vocoded data), our corresponding best system still achieves a competitive overall performance (NN-ACP with 8.51% EER) compared with the systems proposed in [22]. These results show that the hidden layers of the original SSL model contain enough discriminative information to be exploited for the classification task. Furthermore, a simple classifier with adequate processing blocks can efficiently make use of the information contained in the embeddings for deepfake detection. Finally, we can boost the system detection capabilities by creating new fake data using vocoder systems adapted to the in-domain real training data. This can allow us to quickly adapt our detection systems to new generative algorithms without re-training the upstream model.

4. Conclusions

This work evaluates the combination of pre-trained SSL embeddings and a fine-tuned downstream classifier using spoofing data augmentation for audio deepfake detection. By means of the latter, we extended the original training dataset with additional fake samples generated with different vocoder systems to adapt the final classifier while the upstream is frozen. Moreover, we analyzed the performance of small-sized downstream models to exploit the information provided by these SSL embeddings. Our proposed approach exhibits overall SOTA performance across a different range of in- and out-of-domain benchmarks. This demonstrates that the original SSL embeddings serve as a sufficiently discriminative source for the classification task, particularly when considering various deepfake attacks for data augmentation. As a result, this strategy requires only the adaptation of a simple yet efficient downstream model with appropriate processing blocks. In future work, we will evaluate the performance of ensembled downstream networks adapted to different domain conditions and use active learning techniques to select the most relevant data for fine-tuning these classifiers.

5. Acknowledgements

This work has received funding from the European Union's Horizon Europe research and innovation programme in the context of project EITHOS under Grant Agreement No. 101073928. It has also been supported in part by the Project PID2022-138711OB-I00 funded by the MICIU/AEI/10.13039/501100011033 and by ERDF/EU, and the FPI grant PRE2022-000363.

6. References

[1] M. Masood, M. Nawaz, K. M. Malik, A. Javed, A. Irtaza, and H. Malik, "Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward," *Applied intelligence*, vol. 53, no. 4, pp. 3974–4026, 2023.

[2] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[3] C. B. Tan *et al.*, "A survey on presentation attack detection for automatic speaker verification systems: State-of-the-art, taxonomy, issues and future direction," *Multimedia Tools and Applications*, vol. 80, no. 21-23, pp. 32 725–32 762, 2021.

[4] A. Nautsch *et al.*, "ASVspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, 2021.

[5] X. Liu *et al.*, "ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 2507–2522, 2023.

[6] J. Yi *et al.*, "ADD 2022: The first audio deep synthesis detection challenge," in *Proc. ICASSP 2022*, 2022, pp. 9216–9220.

[7] ——, "ADD 2023: The second audio deepfake detection challenge," in *Proc. IJCAI 2023 DADA Workshop*, 2023, pp. 125–130.

[8] A. Mohamed *et al.*, "Self-supervised speech representation learning: A Review," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1179–1210, 2022.

[9] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[10] J. M. Martín-Doñas and A. Álvarez, "The vicomtech audio deepfake detection system based on wav2vec2 for the 2022 ADD challenge," in *Proc. ICASSP 2022*, 2022, pp. 9241–9245.

[11] X. Wang and J. Yamagishi, "Investigating self-supervised front ends for speech spoofing countermeasures," in *Proc. Speaker Odyssey 2022*, 2022, pp. 100–106.

[12] H. Tak, M. Todisco, X. Wang, J. weon Jung, J. Yamagishi, and N. Evans, "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation," in *Proc. Speaker Odyssey 2022*, 2022, pp. 112–119.

[13] E. Rosello, A. Gomez-Alanis, A. M. Gomez, and A. Peinado, "A conformer-based classifier for variable-length utterance processing in anti-spoofing," in *Proc. InterSpeech 2023*, 2023, pp. 5281–5285.

[14] Y. Zhang, J. Lu, Z. Shang, W. Wang, and P. Zhang, "Improving short utterance anti-spoofing with AASIST2," in *Proc. ICASSP 2024*, 2024.

[15] Y. Guo, H. Huang, X. Chen, H. Zhao, and Y. Wang, "Audio deepfake detection with self-supervised WavLM and multi-fusion attentive classifier," in *Proc. ICASSP 2024*, 2024.

[16] P. Kawa, M. Plata, M. Czuba, P. Szymański, and P. Syga, "Improved deepfake detection using Whisper features," in *Proc. InterSpeech 2023*, 2023, pp. 4009–4013.

[17] Y. Xie, H. Cheng, Y. Wang, and L. Ye, "Learning a self-supervised domain-invariant feature representation for generalized audio deepfake detection," in *Proc. InterSpeech 2023*, 2023, pp. 2808–2812.

[18] C. Wang, J. Yi, X. Zhang, J. Tao, L. Xu, and R. Fu, "Low-rank adaptation method for wav2vec2-based fake audio detection," in *Proc. IJCAI 2023 DADA Workshop*, 2023, pp. 101–106.

[19] X. Wang and J. Yamagishi, "Investigating active-learning-based training data selection for speech spoofing countermeasure," in *Proc. 2022 IEEE SLT Workshop*, 2023, pp. 585–592.

[20] ——, "Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders," in *Proc. ICASSP 2023*, 2023.

[21] W. Ge, X. Wang, J. Yamagishi, M. Todisco, and N. Evans, "Spoofing attack augmentation: Can differently-trained attack models improve generalisation?" in *Proc. ICASSP 2024*, 2024.

[22] X. Wang and J. Yamagishi, "Can large-scale vocoded spoofed data improve speech spoofing countermeasure with a self-supervised front end?" in *Proc. ICASSP 2024*, 2024.

[23] S. Yang *et al.*, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. InterSpeech 2021*, 2021, pp. 1194–1198.

[24] T. Stafylakis, L. Mošner, S. Kakouros, O. Plchot, L. Burget, and J. Ćernockỳ, "Extracting speaker and emotion information from self-supervised speech models via channel-wise correlations," in *Proc. 2022 IEEE SLT Workshop*, 2023, pp. 1136–1143.

[25] S. Kakouros, T. Stafylakis, L. Mošner, and L. Burget, "Speech-based emotion recognition with self-supervised models using attentive channel-wise correlations and label smoothing," in *Proc. ICASSP 2023*, 2023.

[26] A. Babu *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. InterSpeech 2022*, 2022, pp. 2278–2282.

[27] H. Tak, M. Kamble, J. Patino, M. Todisco, and N. Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *Proc. ICASSP 2022*, 2022, pp. 6382–6386.

[28] N. Müller, F. Dieckmann, P. Czempin, R. Canals, K. Böttinger, and J. Williams, "Speech is silver, silence is golden: What do ASVspoof-trained models really learn?" in *Proc. 2021 ASVspoof Challenge Workshop*, 2021, pp. 55–60.

[29] J. Frank and L. Schönherr, "WaveFake: A data set to facilitate audio deepfake detection," in *Proc. NeurIPS Datasets and Benchmarks Track (Round 2)*, 2021.

[30] N. Müller, P. Czempin, F. Diekmann, A. Froghyar, and K. Böttinger, "Does audio deepfake detection generalize?" in *Proc. InterSpeech 2022*, 2022, pp. 2783–2787.

[31] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[33] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," 2019. [Online]. Available: https://github.com/Lightning-AI/lightning

[34] X. Wang and J. Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," in *Proc. InterSpeech 2021*, 2021, pp. 4259–4263.