ASVspoof Workshop 2024

Exploring WavLM Back-ends for Speech Spoofing and Deepfake Detection

Theophile Stourbe, Victor Miara, Theo Lepage, Reda Dehak

EPITA Research Laboratory (LRE), France

Introduction

- With advancements in deep learning, audio spoofing techniques (speech synthesis and voice conversion) are making significant progress, highlighting the importance of robust speech spoofing and deepfake detection systems in the context of speaker verification.

➔ Submission to ASVspoof 5 challenge (Track 1) [1]

- ◆ Speech Deepfake Detection: bonafide/spoof speech classification
- ◆ Open Condition: pretrained models and external training datasets are allowed

[1] Xin Wang et al., “ASVspoof 5: Crowdsourced data, deepfakes and adversarial attacks at scale,” in ASVspoof 2024 workshop (submitted), 2024.

Method – Overview

- Other works have successfully applied large self-supervised models for speech processing tasks [1, 2]

- We adopt a pre-trained WavLM [4] as a front-end feature extractor
  - Transformer-based model designed for Automatic Speech Recognition (ASR)
  - Pre-trained in a self-supervised way on a masked speech denoising and prediction task that also captures non-ASR information

- We experiment with different back-ends to aggregate information (WA and MHFA)

- We implement two regularization components to limit overfitting and explore different training strategies (hyper-params and data-augmentation)

[1] Xin Wang and Junichi Yamagishi, "Investigating Self-Supervised Front Ends for Speech Spoofing Countermeasures," in Odyssey, 2022.
[2] Hemlata Tak et al., "Automatic Speaker Verification Spoofing and Deepfake Detection Using Wav2vec 2.0 and Data Augmentation," in Odyssey, 2022.
[3] Sanyuan Chen et al., "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," IEEE JSTSP, 2022.

Method – Weighted Average (WA) back-end

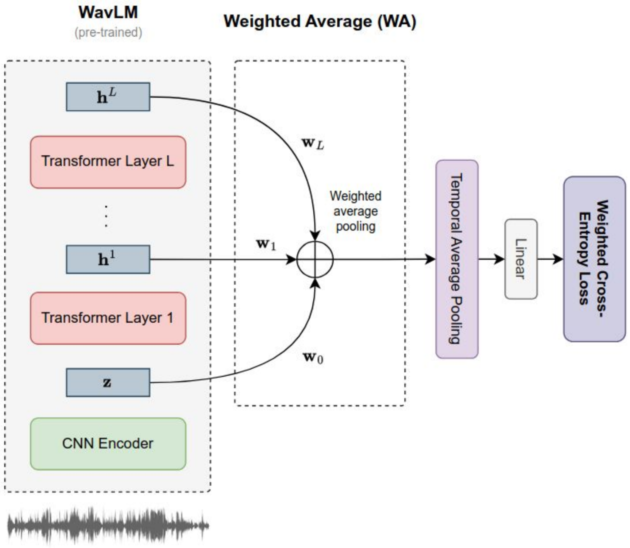

Figure 1. Diagram of our framework for fine-tuning WavLM with Weighted Average (WA) back-end.

- Intermediate representations of self-supervised models contains essential features

- Progressive abstraction of information across layers

  Top layers tend to be the most helpful for ASR while speech and speaker features are mainly represented in the low- and mid-level features

- Weighted Average (WA) back-end: weighted average of the Transformer outputs with learnable weights

Method – Multi-Head Factorized Attention (MHFA) back-end

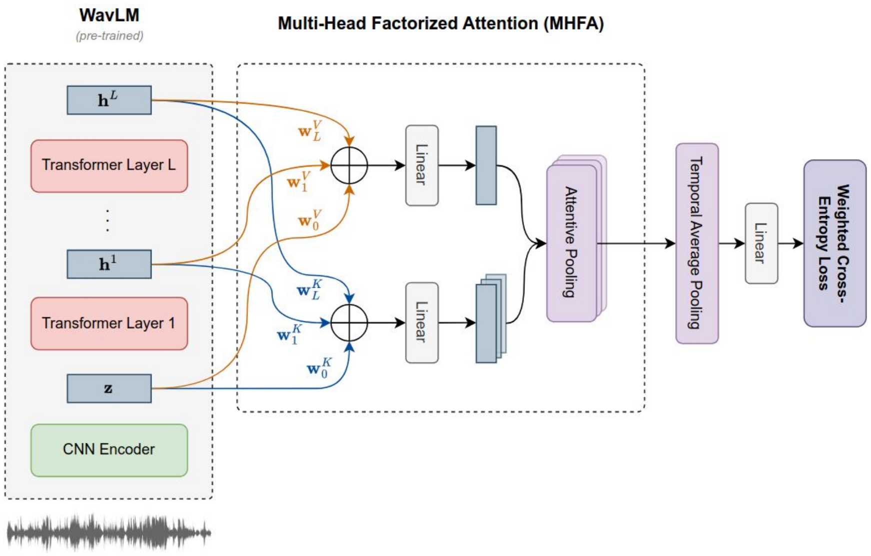

Figure 2. Diagram of our framework for fine-tuning WavLM with Multi-Head Factorized Attention (MHFA) back-end.

- Multi-Head Factorized Attention (MHFA) [1] back-end: aggregate layer-wise outputs from WavLM's Transformer layers into a light-weight attentive pooling mechanism

- MHFA clusters frame-level representations into acoustic units discovered by the transformer model

[1] Junyi Peng et al., "An Attention-Based Backend Allowing Efficient Fine-Tuning of Transformer Models for Speaker Verification," in IEEE SLT, 2022.

Method – Training stability improvements

To mitigate the effect of overfitting from the WavLM font-end, we rely on two components:

1. L2 regularization between the updated weights and the initial weights from the pre-trained WavLM model → reduces overfitting caused by the large number of parameters;

2. Layer-wise learning rate decay → allows more flexible weight updates in higher layers to adapt ASR capabilities, while ensuring lower layers preserve speech signals-related information.

Experimental setup – Data-augmentation

- Background noises: add noise randomly selected from the MUSAN corpus [1]. SNR is uniformly sampled between 0 and 15 dB.

- Reverberation: convolve the input audio segment with an impulse response randomly sampled from the Simulated Room Impulse Response Database [2].

- Codecs: use torchaudio library to apply low and high-quality mp3 and ogg encoder. We also tested four trans-codecs configuration:
  - high mp3 → high ogg
  - low mp3 → low ogg
  - high mp3 → low ogg
  - high ogg → low mp3

- RawBoost: we also experiment with RawBoost similar to [3].

[1] D. Snyder, G. Chen, and D. Povey, “MUSAN: A Music, Speech, and Noise Corpus,” arXiv preprint arXiv:1510.08484, 2015.
[2] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, “A study on data augmentation of reverberant speech for robust speech recognition,” in ICASSP, 2017.
[3] Hemlata Tak et al., “Rawboost: A Raw Data Boosting and Augmentation Method Applied to Automatic Speaker Verification Anti-Spoofing,” in ICASSP, 2022.

Experimental setup – Implementation details

- Models
  - Front-end: WavLM Base (~94M params)
    - CNN encoder
    - 12 Transformer layers (768-d)
  - WA back-end: 1,5K params
  - MHFA back-end: ~1M params

- Evaluation
  - Test score is computed on the full speech utterance
  - Results are reported in terms of EER and minDCF following the evaluation plan

- Training
  - Train data: ASVspoof 5
  - Input data: 4s frames of raw audio
  - Loss: weighted CE loss
  - Epochs: 100 (early stopping)
  - Batch size: 120 or 32
  - Optimizer: Adam
  - Learning rate:
    - Front-end: 2×10^{-5}
    - Back-end: 5×10^{-3}
  - LR scheduler: reduced by 5% every epoch
  - Hardware: NVIDIA A100 80 GB GPU

Results – Experiments (1/3)

	Model		Training	Data-augmentation			Scoring Dataset		Progress Dataset	
#	Back-end	Fine-tune WavLM	Batch size	Noise and RIR	Rawboost	Codec	**EER (%)**	**minDCF**	**EER (%)**	**minDCF**
	Baseline (ResNet)		120	✓		✓	15.60	0.3469	16.19	0.3915
1	MHFA		120				6.78	0.1581		
2	MHFA		120	✓			8.78	0.2155		
3	MHFA	✓	120				6.41	0.1628		
4	MHFA	✓	120	✓			3.37	0.0872	1.42	0.0380

Table 1. Spoof detection results of the different models trained during the ASVspoof 5 challenge on our scoring and progress datasets. The best performances are represented in bold text.

➔ WavLM outperformed the baseline (System 1)

➔ Fine-tuning the front-end was necessary to reach better performance (Systems 2 and 4)

➔ Data-augmentation with noise and reverberation is fundamental (Systems 3 and 4)

Results – Experiments (2/3)

	Model		Training	Data-augmentation			Scoring Dataset		Progress Dataset	
#	Back-end	Fine-tune WavLM	Batch size	Noise and RIR	Rawboost	Codec	EER (%)	minDCF	EER (%)	minDCF
	Baseline (ResNet)		120	✓		✓	15.60	0.3469	16.19	0.3915
1	MHFA		120				6.78	0.1581		
2	MHFA		120	✓			8.78	0.2155		
3	MHFA	✓	120				6.41	0.1628		
4	MHFA	✓	120	✓			3.37	0.0872	1.42	0.0380
5	MHFA	✓	120		✓		28.91	0.7160		
6	MHFA	✓	120	✓		✓	2.18	0.0552	1.22	0.0320
7	MHFA	✓	32	✓		✓	**1.82**	**0.0498**	1.13	0.0279

Table 1. Spoof detection results of the different models trained during the ASVspoof 5 challenge on our scoring and progress datasets. The best performances are represented in bold text.

- ➔ Applying RawBoost augmentation did not perform well (System 5)
- ➔ Applying codec augmentations improved downstream results (Systems 4 and 6)
- ➔ Reducing batch size from 120 to 32 provided better generalization (Systems 6 and 7)

Results – Experiments (3/3)

	Model		Training	Data-augmentation			Scoring Dataset		Progress Dataset	
#	Back-end	Fine-tune WavLM	Batch size	Noise and RIR	Rawboost	Codec	**EER (%)**	**minDCF**	**EER (%)**	**minDCF**
	Baseline (ResNet)		120	✓		✓	15.60	0.3469	16.19	0.3915
1	MHFA		120				6.78	0.1581		
2	MHFA		120	✓			8.78	0.2155		
3	MHFA	✓	120				6.41	0.1628		
4	MHFA	✓	120	✓			3.37	0.0872	1.42	0.0380
5	MHFA	✓	120		✓		28.91	0.7160		
6	MHFA	✓	120	✓		✓	2.18	0.0552	1.22	0.0320
7	MHFA	✓	32	✓		✓	**1.82**	**0.0498**	1.13	0.0279
8	WA	✓	32	✓		✓	1.89	0.0503	**1.01**	**0.0251**

Table 1. Spoof detection results of the different models trained during the ASVspoof 5 challenge on our scoring and progress datasets. The best performances are represented in bold text.

- ➔ WA performs a little bit worse than MHFA but obtained the best result on the progress dataset as it is less subject to overfitting (Systems 7 and 8)
- ➔ We would need more training samples or data-augmentations for the MHFA back-end

Results – Final fused system

	Model		Training	Data-augmentation			Scoring Dataset		Progress Dataset	
#	Back-end	Fine-tune WavLM	Batch size	Noise and RIR	Rawboost	Codec	EER (%)	minDCF	EER (%)	minDCF
	Baseline (ResNet)		120	✓		✓	15.60	0.3469	16.19	0.3915
1	MHFA		120				6.78	0.1581		
2	MHFA		120	✓			8.78	0.2155		
3	MHFA	✓	120				6.41	0.1628		
4	MHFA	✓	120	✓			3.37	0.0872	1.42	0.0380
5	MHFA	✓	120		✓		28.91	0.7160		
6	MHFA	✓	120	✓		✓	2.18	0.0552	1.22	0.0320
7	MHFA	✓	32	✓		✓	**1.82**	**0.0498**	1.13	0.0279
8	WA	✓	32	✓		✓	1.89	0.0503	**1.01**	**0.0251**
9	**Fusion of model 6, 7 and 8**						**1.10**	**0.0272**	**0.88**	**0.0226**

Table 1. Spoof detection results of the different models trained during the ASVspoof 5 challenge on our scoring and progress datasets. The best performances are represented in bold text.

- ➔ Fusing systems achieved the best result → complementarity between WA and MHFA
- ➔ For the challenge, with unseen acoustic conditions, we achieve 0.0937 minDCF, 3.42% EER, 0.1927 Cllr, and 0.1375 actDCF

Conclusions

- We showed that WavLM representations are effective for speech spoofing and deepfake detection.
- Our final system outperforms the baseline and achieves 0.0937 minDCF and 3.42% EER on ASVspoof 5 Track 1: Speech Deepfake Detection - Open Condition.
- MHFA back-end was more subject to overfitting than WA but their fusion achieved the best performance showing the complementarity between the two techniques.
- Perspective: combine SV and speech spoofing detection with a back-end for each downstream task as WavLM also contain valuable speaker identity information.

Please do not hesitate to contact us if you have any questions.

theophile.stourbe@epita.fr

victor.miara@epita.fr

theo.lepage@epita.fr

reda.dehak@epita.fr