SLIM: Style-Linguistics Mismatch Model for Generalized Audio Deepfake Detection

Yi Zhu Surya Koppisetti Trang Tran Gaurav Bharaj
Reality Defender Inc.
{yi,surya,trang,gaurav}@realitydefender.ai

Abstract

Audio deepfake detection (ADD) is crucial to combat the misuse of speech synthesized by generative AI models. Existing ADD models suffer from generalization issues to unseen attacks, with a large performance discrepancy between in-domain and out-of-domain data. Moreover, the black-box nature of existing models limits their use in real-world scenarios, where explanations are required for model decisions. To alleviate these issues, we introduce a new ADD model that explicitly uses the **S**tyle-**LI**nguistics **M**ismatch (SLIM) in fake speech to separate them from real speech. SLIM first employs self-supervised pretraining on only real samples to learn the style-linguistics dependency in the real class. The learned features are then used in complement with standard pretrained acoustic features (e.g., Wav2vec) to learn a classifier on the real and fake classes. When the feature encoders are frozen, SLIM outperforms benchmark methods on out-of-domain datasets while achieving competitive results on in-domain data. The features learned by SLIM allow us to quantify the (mis)match between style and linguistic content in a sample, hence facilitating an explanation of the model decision.

1 Introduction

The growing interest in generative models has led to an expansion of publicly available tools that can closely mimic the voice of a real person [73]. Text-to-speech (TTS) or voice conversion (VC) systems can now be used to synthesize a fake voice from only a few seconds of real speech recordings [72]. When these generation tools are used by bad actors, the generated fake human voices (hereinafter referred to as *audio deepfakes*) can pose serious dangers. Examples include impersonation of celebrities/family members for robocalls [29], illegal access to voice-guarded bank accounts [14], or forgery of evidence in court [56]. Reliable audio deepfake detection (ADD) tools are therefore urgently needed.

State-of-the-art (SOTA) detection systems [81, 93] employ self-supervised learning (SSL) encoders as the frontend feature extractors, and append classification backends to map the high-dimensional feature representations to a binary real/fake decision [35, 93, 81]. Common SSL encoders for this task are the Wav2vec [7], WavLM [11], and HuBert [25], among others. These models are usually trained in a fully-supervised manner, with fake samples generated using off-the-shelf TTS/VC tools [34, 93, 12, 81, 52, 92, 28]. However, current ADD systems are known to underperform on deepfakes crafted by unseen generative models (i.e., *unseen attacks*) [40, 51, 67, 93]. To tackle this issue, some works have focused on extracting more robust features from the input representation [26, 71, 92]. Additional improvements have been reported by finetuning the SSL frontend during downstream supervised training [71, 81, 44] and by increasing the diversity of labeled samples via data augmentation or continual training on vocoded data [70, 87, 82, 83]. While shown to be effective for in-domain deepfakes, frontend finetuning increases the cost of training drastically.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2168

Additionally, outputs from existing ADD systems are hard to explain, i.e., it is unclear to a typical user why an ADD makes a certain prediction, which leads to lack of trust [93, 34]. For practical applications, it is crucial to understand what information the model is relying on to make decisions, and under which circumstances the model would fail to successfully detect deepfakes. A group of works uses explainable AI (xAI) methods [4] to interpret model decisions [21, 32, 36], but they mainly rely on *post-hoc* visualizations such as saliency maps, which are known to be sensitive to training set-ups [90] and can therefore be inconsistent. Other models focus on specific vocal attributes, such as breath [33], or vocal tract [10] to derive explanations. However, most of these interpretable attributes only account for a subset of deepfake-related characteristics, hence resulting in a large gap in detection performance compared to SOTA methods [93, 34]. We note that while "interpretability/explainability" is often ambiguous [39], in this work we mean the model's ability to provide reasons for a certain prediction, e.g., a sample is likely fake because its style and linguistics representations are more misaligned than those of real samples (as will be shown in Section 4.2).

In this study, we propose a generalizable ADD model that explicitly explores the style-linguistics mismatch in deepfakes to separate them from real audios, and thereby facilitates an explanation on the model decision. We hypothesize that in real speech, a certain dependency exists between the *linguistics* information embedded in the verbal content and the *style* information embedded in the vocal attributes, such as speaker identity and emotion. To synthesize a deepfake audio, both TTS and VC systems artificially combine the verbal content with the vocal attributes of a target speaker, and thereby introduce an artificial style-linguistics dependency that would differ from the real speech. Our two-stage framework explicitly studies the Style-LInguistics Mismatch (SLIM) in the fake class to separate it from the real class. During Stage 1, the style-linguistics dependency in the real class is learned by contrasting the style and linguistic subspace representations and generating a set of dependency features from each subspace. The learned pairs of style and linguistics features are expected to be more correlated for real speech than for deepfake speech. In Stage 2, we employ supervised training, where we fuse the learned dependency features from Stage 1 with the original style and linguistic representations and train a light-weight projection head to classify the input representations as real or fake.

Our main contributions are summarized as follows:

1. We propose SLIM model, which leverages Style and LInguistics Mismatch in deepfakes to achieve better generalization to unseen attacks than existing models, and has interpretation capability.
2. SLIM outperforms existing SOTA methods on out-of-domain datasets (`In-the-wild`, `MLAAD`) while being competitive on in-domain datasets (`ASVspoof2019`, `2021`). This is achieved without increasing the amount of labeled data or the added cost from end-to-end finetuning.
3. Unlike black-box ADD models, the style-linguistics features learned by SLIM can be used to interpret model decisions. We present analyses to show how the interpretation can be performed on a group level as well as on individual speech samples.

2 Related works

2.1 Audio deepfake detection

State-of-the-art ADD systems mainly rely on fully-supervised training, where the model architectures comprise of one or more speech SSL frontends and a backend classifier [93, 2, 45]. Guo et al. [23] developed a multi-fusion attentive classifier to process the output from a WavLM frontend; Yang et al. [92] fused outputs from multiple SSL frontends and reported improvements over using a single frontend. However, existing ADD systems experience severe degradation in performance when tested on unseen data [51, 67], which questions their applicability and trustworthiness for real-world scenarios. To address this issue, multiple works have explored methods to improve model generalizability. With added training cost, improvements have been reported when frontends are finetuned alongside the backend classifiers during downstream training [71, 81]. Further improvements were achieved with data augmentations such as RawBoost [70, 71] and neural vocoding [82]. More recent works also show that distilled student models can generalize better than large teacher models [43, 83]. Still, large discrepancies between in-domain and out-of-domain performance are common [93, 34].

In addition to generalization, existing ADD models also fall short on interpretability. Several studies have shown that current SOTA models may be focusing on artifacts introduced in the frequency

domain during voice synthesis and/or the artifacts in non-speech segments [67, 50, 40, 96]. While a line of work proposed to extract speech-related features, such as breath [33], vocal tract, and articulatory movement [10], the overall detection performance was inferior to SSL-based methods. Other works resort to xAI methods [4] for model interpretation, such as SHAP [21], GradCAM [32], and Deep Taylor [36]. However, these post-hoc analysis approaches are known to be sensitive to training set-ups [90] and therefore not viable for practical use. Both generalization and interpretability remain challenging issues for current ADD systems.

2.2 Style-linguistics modelling

One standard approach for modelling speech is to decompose it into two subspaces, style and linguistics. The former refers to short and long-term paralinguistic attributes, such as speaker identity, emotion, accent, and health state [65]. The latter corresponds to the verbal content of speech [31]. For representing style information, early works relied on handcrafted features, such as GeMAPS [19, 18]. Later studies showed improved performance by representations learned end-to-end by deep neural networks (DNN), such as the x-vector [68] and ECAPA-TDNN embeddings [16]. Similarly, the linguistic representations follow a similar trend where DNN-based embeddings, such as Whisper [60], outperform handcrafted features for content-related tasks [17]. More recent studies have shown that style and linguistics information can be efficiently encoded together in the SSL representations [7, 11, 25]. To investigate how speech information is encoded in DNNs, a group of works conducted layer-wise analysis and showed that early to middle layers carry more style related attributes, such as speaker identity [5], emotion [63], and articulatory movement [13]; while later layers encode linguistics attributes, such as phonetic information and semantics [55, 66].

Despite these approaches, it is unclear if completely disentangling style and linguistics information in speech is possible. Studies have shown that a certain dependency exists between these two subspaces: the link between emotional states and word choices [38], the relation between prosody and language understanding [15], and the impact of age on sentence coherence [58]. Effectively modeling both the independent and dependent aspects of style and linguistics in speech still remains a challenge.

3 Method

3.1 Motivation

For the majority of generative speech models, the style and linguistic subspaces are assumed to be independent of each other [73, 27, 78, 48]. For example, VC systems change the voice of an utterance by replacing the source speaker's embeddings with those of the target speaker [48, 78], assuming that these embeddings contain no linguistics information. Similarly, modern TTS systems rely on independently learned representations to model different speech aspects (e.g., text, speaker, emotion) to synthesize expressive speech [8, 16, 77]. Because of this disentanglement assumption, a mismatch likely exists between the style and linguistics information in TTS/VC speech that differentiates it from real speech. To study this hypothesis, we conduct a proof-of-concept experiment on a sample subset of `ASVSpoof2019` [75]. Following previous research [61, 30, 54], we use canonical correlation analysis (CCA) to derive a subspace where the linear projections of the style and linguistics embeddings are maximally correlated for the real class. We choose the last layer output of pretrained `wav2vec2-large-xlsr-53-english` [22] for linguistics representation, and the pretrained `ECAPA-TDNN` embeddings [16] for style representation.

Table 1: Mean and standard deviation of Pearson correlation coefficients (r) calculated between style and linguistics embeddings for real and TTS/VC samples across 5 unseen speakers. Significant difference (as per Welch's t-test) is seen between real speech and all types of generated speech.

Class	Real	A01 (TTS)	A02 (TTS)	A03 (TTS)	A04 (TTS)	A05 (VC)	A06 (VC)
r	**.308±.025**	.202±.033	.217±.020	.243±.024	.253±.021	.214±.026	.252±.020

We randomly select 100 real speech samples from `ASVspoof2019` [75] training set to fit 20-dim CCA features for both linguistics and style representations. We then apply the CCA projection to

https://doi.org/10.52202/079017-2168

200 audios from 5 unseen speakers and 6 TTS/VC systems, and compute the correlation values between these projected style and linguistics vectors to quantify the subpace similarities. Details of the tested TTS/VC systems can be found in Appendix A.1. Table 1 shows these results: A higher r is seen for the real samples, whereas significantly lower correlations are observed for both TTS and VC generated samples. Moreover, TTS-samples on average show lower r (0.228) than VC-samples (0.236), indicating that VC-samples are closer to real speech in terms of style-linguistics dependency. This could explain why VC samples were found to be more challenging to detect than TTS samples in the ASVspoof2019 challenge [40]. While our findings demonstrate the usefulness of CCA for validating the subspace mismatch, its limitations, such as that it only explores the linear composites of the variables [85], might make it sub-optimal to be used independently for deepfake detection. We therefore develop a detection framework that *explicitly* studies the style-linguistic mismatch and scales to larger amount of data.

3.2 Formulation of SLIM

Our two-stage Style-LInguistics Mismatch (SLIM) learning framework is outlined in Figure 1. The first stage operates on the real class only and employs self-supervised learning to build style and linguistic representations and their dependencies for real speech. In the second stage, a classifier is fit onto the learned representations via supervised training over deepfake datasets with binary (real/fake) labels.

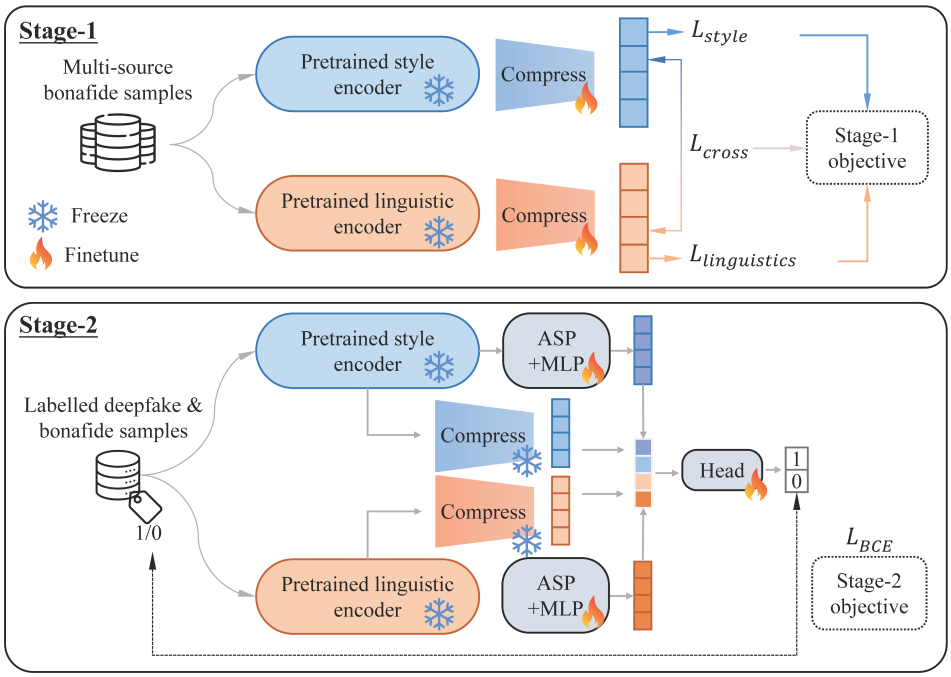

Figure 1: SLIM – A two-stage training framework for ADD. Stage 1 extracts style and linguistics representations from frozen SSL encoders, compresses them, and aims to minimize the distance between the compressed representations ($\mathcal{L}_{cross}$), as well as the intra-subspace redundancy ($\mathcal{L}_{style}$ and $\mathcal{L}_{linguistics}$). The Stage 1 features and the original subspace representations (pretrained SSL embeddings) are combined in Stage 2 to learn a classifier via supervised training.

3.2.1 Stage 1: One-class self-supervised contrastive training

The goal of the first stage is to learn pairs of dependency features from style and linguistics subspaces, which are expected to be highly correlated for real samples and minimally correlated for deepfakes. Since only real samples are needed, we incorporate other open-source speech datasets to diversify the style variations. Given a speech sample, we first extract the style and linguistics representations separately using pretrained networks. Since recent SSL models achieve superior performance on

multiple speech downstream tasks compared to conventional speech representations (e.g., ECAPA-TDNN) [7, 11, 25, 91, 57] we select a group of SSL models finetuned for paralinguistics and linguistics tasks as candidate encoders [20, 79, 57, 84, 6]. In addition, it has been shown that early to middle model layers carry paralinguistics information, while later layers encode linguistics content [54, 5, 37, 66]; we conducted thorough analyses to examine the cross-correlation between pretrained SSL model layers (Appendix A.2) and chose layer 0-10's output from `Wav2vec-XLSR` finetuned for speech emotion recognition to represent style, and layer 14-21's output from `Wav2vec-XLSR` finetuned for automatic speech recognition, to represent linguistics information.

Both style ($\mathbf{X}_S$) and linguistics ($\mathbf{X}_L$) embeddings are three-dimensional tensors $\in \mathbb{R}^{K\times F\times T}$ where K denotes the transformer layer index, F denotes the feature size, and T denotes the number of time steps. These subspace embeddings are sent into compression modules $\mathcal{C}(\cdot)$, which average the transformer layer outputs and reduce the feature size from 1024 to 256 (more details in Appendix A.4). We refer to the output from the compression modules as dependency features: $\mathbf{S}_{f,t} = \mathcal{C}(\mathbf{X}_S)$ for style and $\mathbf{L}_{f,t} = \mathcal{C}(\mathbf{X}_L)$ for linguistics, and their temporally averaged versions are denoted $\bar{\mathbf{S}}_f$ and $\bar{\mathbf{L}}_f$. These dependency features are learned by minimizing the self-contrastive loss $\mathcal{L}_{con}$, defined as:

$$\mathcal{L}_{con} = \mathcal{L}_{cross} + \lambda\mathcal{L}_{intra}, \qquad \mathcal{L}_{intra} = \mathcal{L}_{style} + \mathcal{L}_{linguistics} \tag{1}$$

$$\mathcal{L}_{cross} = \frac{1}{T}\sum_{t=0}^{T} \|\mathbf{S}_{f,t} - \mathbf{L}_{f,t}\|_{\mathbf{F}}^2, \qquad \mathcal{L}_{intra} = \|\bar{\mathbf{S}}_f\bar{\mathbf{S}}_f^\intercal - \mathbb{I}\|_{\mathbf{F}}^2 + \|\bar{\mathbf{L}}_f\bar{\mathbf{L}}_f^\intercal - \mathbb{I}\|_{\mathbf{F}}^2 \tag{2}$$

$\mathcal{L}_{cross}$ denotes the cross-subspace loss; $\mathcal{L}_{intra}$ is the intra-subspace loss, defined in terms of $\mathcal{L}_{style}$ and $\mathcal{L}_{linguistics}$ (Figure 1); $\lambda \in [0, 1]$ is a hyperparameter that weighs the two loss terms, T is the number of time steps; and $\|(.)\|_{\mathbf{F}}^2$ is the Frobenius norm. The $\mathcal{L}_{cross}$ term reduces distance between the compressed style and linguistic embeddings, while the $\mathcal{L}_{intra}$ term reduces redundancy within the (temporally averaged) style and linguistic features by pushing off-diagonal elements to zero. The learned dependency features from Stage 1 can be used to quantify whether a mismatch exists between the style and linguistics of an audio. We further demonstrate this in Section 4.2.

3.2.2 Stage 2: Supervised training

The second stage of SLIM follows a standard supervised training scheme, where the dependency features and subspace representations are concatenated and fed into a classification head to predict a binary real/fake outcome. As shown in Figure 1, the subspace SSL encoders and compression modules are obtained from Stage 1 and are all frozen during Stage 2. Since the dependency features are specifically designed to capture the style-linguistics mismatch alone, we complement them with the original embeddings in order to capture other artifacts that can help separate real samples from the fake class. The original embedding's dimensions are reduced from 1024 to 256 through an attentive statistics pooling (ASP) layer and a multi-layer perceptron (MLP) network. The projected subspace embeddings when concatenated with dependency features result in 1024-dim vectors. The classification head consists of two fully-connected layers and a dropout layer. Binary cross-entropy loss is used to jointly train the ASP and MLP modules alongside the classification head.

4 Experiments

Based on the preliminary results from Section 3.1, we systematically assess the in-domain and cross-domain detection performance of SLIM using multiple datasets, and demonstrate how such framework would benefit the interpretation of model decisions.

4.1 Experimental set-up

Stage 1 training. Unlike benchmark models which are trained end-to-end in a supervised manner, our model relies on two-stage training where each stage requires different training data to avoid information leakage. Since only real samples are needed in Stage 1, we take advantage of open-source speech datasets by aggregating subsets from the `Common Voice` [3] and `RAVDESS` [41] as training data and use a small portion of real samples from the `ASVspoof2019 LA train` for validation. Both `Common Voice` and `RAVDESS` cover a variety of speaker traits. The former is a crowdsourced dataset collected online from numerous speakers with uncontrolled acoustic environments, while the latter is

 https://doi.org/10.52202/079017-2168

an emotional speech corpus with large variations in prosodic patterns. Such data variety enables our model to learn a wider range of style-linguistics combinations.

Stage 2 training and evaluation. For a fair comparison with existing works, we adopt the standard train-test partition, where only the `ASVspoof2019` logical access (LA) training and development sets are used for training and validation. For evaluation, we use the test split from `ASVspoof2019 LA` [75] and `ASVspoof2021 DF` [40]. `ASVspoof2019 LA` and `ASVspoof2021 DF` have been used as standard datasets for evaluating deepfake detection models, where real speech recordings originate from the `VCTK` and `VCC` datasets [89, 42, 94] and the spoofed ones are generated with a variety of TTS and VC systems. Compared to `ASVspoof2019 LA`, `ASVspoof2021 DF` contains more than 100 different types of generated speech in the evaluation set with a variety of different compression codecs. This provides a more challenging setting for testing generalization to unseen attacks and robustness to compression. In addition, we assess our model's generalizability on two more recent datasets: `In-the-wild` [51], and the English subset from `MLAAD v3` [53]. `In-the-wild` consists of on audio clips collected from English-speaking celebrities and politicians, featuring more realistic and spontaneous speech samples. The English subset of `MLAAD` (hereinafter referred to as `MLAAD-EN`) is a recent dataset with spoofed samples generated using state-of-the-art open-source TTS and VC systems (more details in Appendix A.3).

Metrics. Equal error rate (EER) is a standard metric for evaluating deepfake detection systems [75, 40]. It refers to the point in the detection error tradeoff curve where the false acceptance rate equals the false rejection rate; lower EER suggests better performance. We also report F1-beta scores (β=1) to account for the class imbalance; higher F1 scores suggest better performance.

Benchmarks. We consider several SOTA models to benchmark against and broadly categorize them as follows, based on the training cost: (i) methods which freeze feature extraction frontends and finetune only the backend classifiers, and (ii) methods which finetune frontends together with the classifiers during supervised training. As benchmarks representing the former case, we consider Wav2vec-XLSR+LLGF (W2V-LLGF) [87], Wav2vec-XLSR+LCNN (W2V-LCNN) [87], six different models that share a similar backend classifier as SLIM (W2V/WLM/HUB-ASP), a model that fuses different SSL representations (SSL-fusion) [92], as well as three methods that do not rely on large SSL encoders, namely, LCNN [12], RawNet2 [69], and PS3DT [88]. For the end-to-end fine-tuning benchmarks, we consider the model in [44] with a backend classifier similar to SLIM's (W2V-ASP-ft), and the model in [70] with RawBoost augmentation and AASIST backend (W2V-AASIST). Using frozen frontends, five variants of SLIM are considered, where the input at Stage 2 is: (i) only the style embedding, (ii) only the linguistics embedding, (iii) the combination of style and linguistics embeddings, (iv) only the style-linguistics dependency features, and (v) the fusion of style and linguistic embeddings and their dependency features. We emphasize that the original SLIM framework does not involve any finetuning of frontends, since the finetuning may change the disentanglement of style and linguistics embeddings and thus hamper model explainability. However, to compare with finetuned benchmarks, we include a variant of SLIM that finetunes all modules during Stage 2, noting that this would compromise the feature interpretation.

Implementation details. We implement our models using the SpeechBrain toolkit [62] v1.0.0. The hyperparameters used for Stage 1 and Stage 2 training are provided in Appendix A.7. When setting up our customized benchmark models, we followed consistent training recipes where only the model architectures were changed and the same data augmentation method was used. Each round of evaluation was repeated three times with different random seeds, and the mean values are reported. Unlike some previous works which limit the duration of test speech, we only applied amplitude normalization during evaluation and kept the full duration of all samples.

4.2 Experiment results

Detection performance. Table 2 summarizes the detection performance of all models and compares the number of trainable parameters. We discuss the models with *frozen frontend* here, and compare the models with *finetuned frontend* in Section. 4.3. `ASVspoof2019 eval` set contains 19 types of attacks, out of which 6 are seen during training. This makes it the simplest of the four test datasets. We see that a majority of the models achieve near-perfect performance, with several including SLIM reporting EER below 1%. As expected, degradation is seen when models are tested

on ASVspoof2021, where the majority of attacks are unseen. Both W2V-LCNN and SLIM are top-performers, with no significant difference between the two. With the out-of-domain datasets (In-the-wild and MLAAD-EN), more severe degradation is observed, where the majority report EERs over 20%. SLIM, however, outperforms the others with EER of 12.9% and 13.5% on In-the-wild and MLAAD-EN, respectively. It should be noted that although ASVspoof2021 is often used as a standard dataset to evaluate model generalizability to unseen attacks [40], part of the real samples in ASVspoof2021 originate from the same dataset (the VCTK corpus [89]) as the ASVspoof2019 training data [75, 12, 69, 87, 81]. As a result, the real samples from ASVspoof2019 and ASVspoof2021 share a similar distribution, whereas the In-the-wild and MLAAD-EN samples share nearly no overlap with ASVspoof (further discussion in Appendix A.3). Generalization to In-the-wild and MLAAD-EN is therefore more challenging than to ASVspoof2021. The large gains reported by SLIM demonstrates how the style-linguistics mismatch helps with generalization to unseen data.

In Table 2, we also demonstrate the benefits of introducing Stage 1 by considering features from SLIM variants as inputs to Stage 2: dependency features, the style and linguistics embeddings (Enc_{sty} and Enc_{ling}), as well as their combination. The architecture of the classification head is kept the same, except for the number of neurons in the input layer. The dependency features outperform the rest on the two out-of-domain datasets, while the subspace embeddings perform better on ASVspoof2021. Simply concatenating the style and linguistics embeddings does not yield significant improvements when compared to benchmark models. This suggests that the style-linguistics dependency may not be fully captured by supervised training methods without explicit guidance.

Table 2: Detection performance on different deepfake datasets. Experiments were repeated three times with different random seeds, and average metric values are reported. #Param refers to the number of trainable parameters (in millions). For SLIM, we sum up parameters trained at both stages. A few models do not make their code open-source, we therefore include the metrics reported in their papers and skip parameter calculation (N/A). Lowest EERs are bolded per category.

Category	Model	ASVspoof19		ASVspoof21		In-the-wild		MLAAD-EN		#Param (million)
		EER↓	F1↑	EER↓	F1↑	EER↓	F1↑	EER↓	F1↑	
Frozen frontend (Section. 4.2)	LCNN [12]	3.7	.834	25.5	.197	65.6	.373	37.2	.654	4
	RawNet2 [69]	3.0	.875	22.3	.213	37.8	.602	33.9	.676	4
	PS3DT [88]	4.5	–	–	–	29.7	–	–	–	N/A
	W2V-ASP	3.3	.858	19.6	.233	30.2	.705	29.1	.715	9
	WLM-ASP	**0.3**	.983	9.0	.426	25.4	.751	30.3	.709	9
	HUB-ASP	0.5	.975	15.4	.289	29.9	.718	31.0	.702	9
	W2V-LLGF [81]	2.3	.936	9.4	.402	25.1	.756	27.8	.731	10
	W2V-LCNN [87]	0.6	–	**8.1**	–	24.5	–	–	–	N/A
	W2V+WLM	1.8	.916	22.5	.203	30.3	.704	27.0	.739	9
	W2V+HUB	0.9	.956	14.2	.310	27.9	.737	27.6	.732	9
	WLM+HUB	0.8	.963	16.7	.269	29.2	.724	28.5	.720	9
	SSL-Fusion [92]	**0.3**	.981	8.9	.419	24.2	.765	26.5	.739	10
	SLIM variants (ours)									
	Enc_{sty}	6.7	.740	8.6	.438	29.2	.724	25.4	.756	9
	Enc_{ling}	5.9	.764	9.3	.407	30.4	.708	25.0	.760	9
	$\text{Enc}_{style+ling}$	3.5	.834	9.0	.429	25.1	.757	23.9	.772	10
	Dependency	2.8	.897	20.5	.234	25.8	.750	19.8	.811	9
	Full	0.6	.969	**8.3**	.451	**12.9**	.895	**13.5**	.865	11
Finetuned frontend (Section. 4.3)	W2V-ASP [44]	0.3	.984	4.5	.646	18.6	.836	19.2	.817	317
	W2V-AASIST [71]	**0.2**	.991	**3.6**	.707	17.5	.847	14.5	.856	317
	SLIM (ours)	**0.2**	.989	4.4	.651	**12.5**	.898	**10.7**	.892	253

Style-linguistics mismatch of deepfakes. Figure 2 shows the distribution of cosine distances between the style and linguistics dependency features for the real and fake classes; larger distances indicate a higher mismatch. Since the distance values approximately follow a Gaussian distribution with unequal variances, we further conduct a Welch's t-test [1] to examine the statistical significance

https://doi.org/10.52202/079017-2168

of the difference between real and fake samples. For all three datasets, the average cosine distance is found to be significantly lower for real speech than for deepfake samples ($p < 1e^{-5}$). This further corroborates our hypothesis that a higher style-linguistics mismatch exists for fakes. On the other hand, the distance distributions of real and fake samples still share a large overlap, indicating that dependency features alone are not sufficient for perfectly discriminating between the two classes.

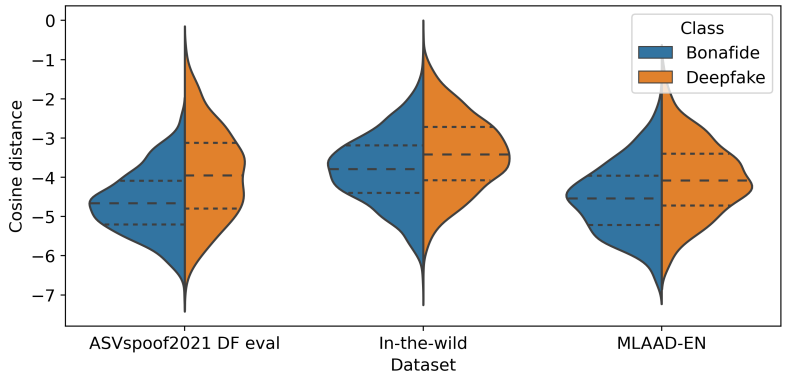

Figure 2: Cosine distance (log scale) calculated between the style and linguistics dependency features for `ASVspoof2021 DF eval`, `In-the-wild`, and `MLAAD-EN`. Whiskers from top to bottom represent the 75% quartile, median, and 25% quartile of the distribution.

Analysis of style-linguistics dependency features. Table 2 demonstrates that style-linguistics dependency features can provide better generalizability than the subspace embeddings (Table 2 SLIM variants, rows 1–4). To examine these results, we first aggregate `ASVspoof2021`, `In-the-wild`, and `MLAAD-EN`, and project the dependency features as well as the concatenated subspace embeddings to a 2-dim space using t-SNE for visualization (Figure 3). Since we use frozen frontends, the embeddings input to Stage 2 training are not affected by backpropagation. Ideal embeddings would exhibit maximal separation between the real and fake classes, while showing minimal shift within each class for different dataset distributions. In Figure 3, we see that the dependency features show larger discrimination between real and fakes (3(c) and 3(d)) than the concatenated subspace embeddings (3(a) and 3(b)), and also a smaller shift between datasets: fake and real samples from the same dataset (color) clusters have less overlap in distribution in the plots.

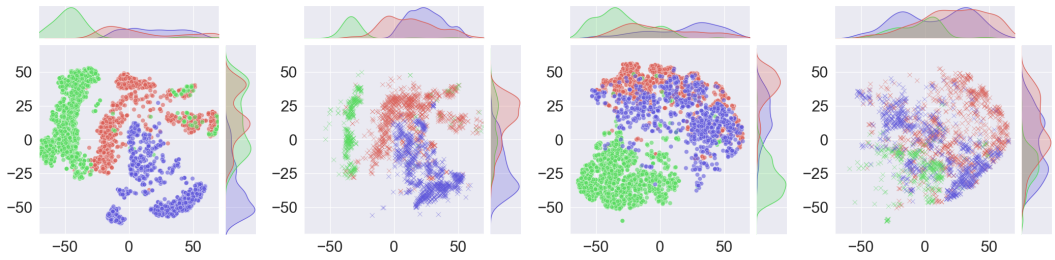

Figure 3: Projected embeddings using t-SNE for style-linguistic representations: (a) subspace embeddings - real class, (b) subspace embeddings - fake class, (c) dependency features - real class, (d) dependency features - fake class. Data distributions are visualized on the upper and right side of the embedding plots. Red: `ASVspoof2021`; Green: `In-the-wild`; Blue: `MLAAD-EN`.

Interpretation of model decisions. Next, we perform a qualitative evaluation of the model decisions. Figure 4 shows the mel-spectrograms of four samples selected from `In-the-wild`. These four demonstrate typical acoustic characteristics that represent a larger group of recordings: (1) top-left is a *fake* sample with audible artifacts at high-frequency region; (2) top-right is a *fake* sample with unnaturally long pauses heard before and after the phrase "but not"; (3) bottom left is a *real* sample

with an atypical speech style where the word pronunciations are elongated; (4) bottom right is a *real* speech recorded in a noisy condition. We find that among the top-performing systems shown in Table 2, only SLIM classified all four samples correctly (both frozen and fine-tuned versions; with all features), while others mostly failed on (2) and (4). Findings here suggest that SLIM provides guidance when abnormalities in style and linguistics occur. Such guidance can be complemented via *post-hoc* analysis tools such as human evaluations or saliency maps [4] for further interpretation.

Additionally, we note that the decisions made by dependency features and the original subspace representations are complementary to each other. Samples in the right column are correctly identified as fake by the dependency features but missed by the original subspace representations, and vice versa (left column missed by dependency features). These results corroborate with the nature of the two feature types. The dependency features are learned by modelling the general style-linguistics relationship seen in real speech, therefore samples with mismatched style-linguistics pattern are likely to be flagged as "not real." The original style and linguistics embeddings, on the other hand, are sensitive to signal artifacts, which could be the deepfake imperfections generated during speech synthesis [67], or the amount of background noise and device artifacts. By combining the two features, SLIM captures a variety of abnormalities and achieves improved classification.

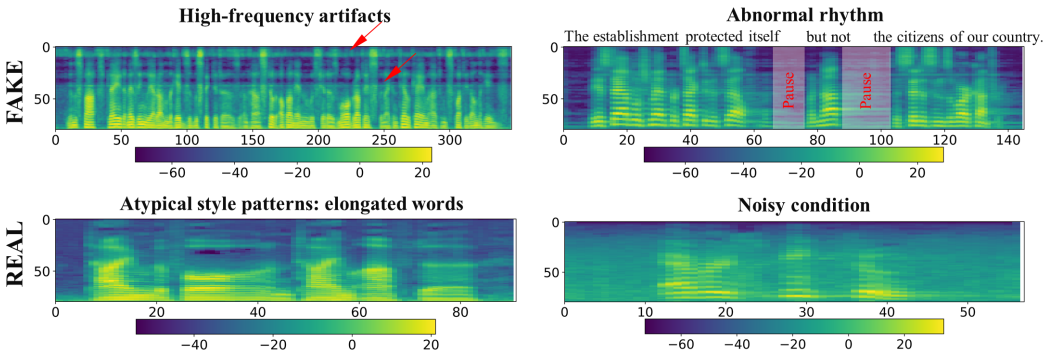

Figure 4: Mel-spectrograms of select samples from `In-the-wild`. SLIM classifies all four correctly, and when reporting fakes, provides guidance on abnormalities in style and/or linguistics. Also, the dependency and subspace features in SLIM are complementary to each other. Left: samples missed by dependency features but correctly identified by the style and linguistic features; right: vice versa.

4.3 Ablation studies

Effects of finetuning SSL frontend. From Table 2, we see that the frontend finetuning helps to further decrease the EER for SLIM. The finetuned version of SLIM performs better than the rest on `In-the-wild` and `MLAAD-EN`, while providing comparable performance on `ASVspoof2019` and `ASVspoof2021`. However, it should be noted that the interpretation of style-linguistics mismatch becomes difficult after finetuning, since the two subspace representations may no longer be maximally disentangled.

Effects of classification backend. In Stage 2, subspace representations are sent into ASP+MLP layers, which output 256-dim embeddings to fuse with the dependency features. Previous works have shown that different backend architectures may lead to a significant difference in the detection performance [81]. With the input fixed (dependency features and subspace embeddings), we find that removing the ASP and MLP layers degrades EER across the four datasets (Table 4, Appendix A.6), while using the LCNN [87] or LLGF [81] backends improves EER on `ASVspoof2019` and `ASVspoof2021`, but not on `In-the-wild` and `MLAAD-EN`.

Misclassifications made by SLIM. We studied the misclassifications made by SLIM through a combination of qualitative analysis methods, such as listening tests, visualizations of mel-spectrograms, noise-level estimation, and prediction of number of speakers. In general, we found that shorter and low-quality samples, e.g., those with estimated Mean Opinion Score (MOS) lower than 2 (using NISQA [47]), as well as recordings with multiple speakers were more likely to be misclassified. While the former is a general challenge faced by all detection methods, the latter can be explained by

https://doi.org/10.52202/079017-2168

the design of SLIM, where the pretraining stage does not involve any multi-speaker samples. In a multi-speaker scenario (e.g., conversational speech), speech patterns from more than one speaker could lead to higher misalignment between linguistic content and speaker attributes, resulting in false alarms.

5 Limitations

Since our framework explicitly focuses on style-linguistics mismatch, it is possible that real speech samples with atypical style-linguistics dependency (e.g., dysarthric speech [59] or children speech) may be misclassified. With the rapidly growing number of speech datasets, one countermeasure is to conduct self-supervised pretraining on datasets with more diverse speech styles (e.g., [95]). Also, while we focused on English samples due to the scarcity of multilingual deepfake datasets at the time of writing, the proposed framework can be scaled to more languages with recent deepfake datasets, such as MLAAD-v4. Finally, although SLIM can benefit from frontend finetuning and more advanced backends, doing so would affect the feature interpretation and will require modifications to the training approach. We plan to explore these directions in the future.

6 Conclusion

We present SLIM, a new audio deepfake detection framework that explicitly models the style-linguistics mismatch in speech to detect deepfakes. Without requiring a large amount of labeled data or the added cost of end-to-end finetuning on pretrained encoders, SLIM outperforms existing benchmarks on out-of-domain datasets, while being competitive on in-domain datasets. The learned style-linguistics dependency features are complementary to the individual pretrained style and linguistics subspace representations, and facilitate result interpretation by guiding our attention to where style-linguistics misalignment occurs, which can be further analyzed by a variety of qualitative studies.

References

[1] Nor Aishah Ahad and Sharipah Soaad Syed Yahaya. Sensitivity analysis of welch'st-test. In *AIP Conference proceedings*, volume 1605, pages 888–893. American Institute of Physics, 2014.

[2] Zaynab Almutairi and Hebah Elgibreen. A review of modern audio deepfake detection methods: Challenges and future directions. *Algorithms*, 15(5):155, 2022.

[3] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, 2020.

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[5] Takanori Ashihara, Marc Delcroix, Takafumi Moriya, Kohei Matsuura, Taichi Asami, and Yusuke Ijima. What do self-supervised speech and speaker models learn? new findings from a cross model layer-wise analysis. *arXiv preprint arXiv:2401.17632*, 2024.

[6] Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, et al. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*, 2021.

[7] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33: 12449–12460, 2020.

[8] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR, 2022.

[9] Susan Baldwin. Compute canada: advancing computational research. In *Journal of Physics: Conference Series*, volume 341, page 012001. IOP Publishing, 2012.

[10] Logan Blue, Kevin Warren, Hadi Abdullah, Cassidy Gibson, Luis Vargas, Jessica O'Dell, Kevin Butler, and Patrick Traynor. Who are you (i really wanna know)? detecting audio {DeepFakes} through vocal tract reconstruction. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2691–2708, 2022.

[11] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[12] Akash Chintha, Bao Thai, Saniat Javid Sohrawardi, Kartavya Bhatt, Andrea Hickerson, Matthew Wright, and Raymond Ptucha. Recurrent convolutional structures for audio spoof and video deepfake detection. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1024–1037, 2020.

[13] Cheol Jun Cho, Peter Wu, Abdelrahman Mohamed, and Gopala K Anumanchipalli. Evidence of vocal tract articulation in self-supervised learning of speech. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[14] Joseph Cox. How i broke into a bank account with an ai-generated voice. `https://www.vice.com/en/article/dy7axa/how-i-broke-into-a-bank-account-with-an-ai-generated-voice`, 2023. Accessed: 2024-04-30.

[15] Anne Cutler, Delphine Dahan, and Wilma Van Donselaar. Prosody in the comprehension of spoken language: A literature review. *Language and speech*, 40(2):141–201, 1997.

[16] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[17] Amandeep Singh Dhanjal and Williamjeet Singh. A comprehensive survey on automatic speech recognition using neural networks. *Multimedia Tools and Applications*, 83(8):23367–23412, 2024.

[18] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462, 2010.

 https://doi.org/10.52202/079017-2168

[19] Florian Eyben, Klaus R Scherer, Björn W Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y Devillers, Julien Epps, Petri Laukka, Shrikanth S Narayanan, et al. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE transactions on affective computing*, 7(2):190–202, 2015.

[20] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu. Exploring wav2vec 2.0 on speaker verification and language identification. *Interspeech 2021*, 2021.

[21] Wanying Ge, Jose Patino, Massimiliano Todisco, and Nicholas Evans. Explaining deep learning models for spoofing and deepfake detection with shapley additive explanations. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6387–6391. IEEE, 2022.

[22] Jonatas Grosman. Fine-tuned XLSR-53 large model for speech recognition in English. `https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english`, 2021.

[23] Yinlin Guo, Haofan Huang, Xi Chen, He Zhao, and Yuehai Wang. Audio deepfake detection with self-supervised wavlm and multi-fusion attentive classifier. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12702–12706. IEEE, 2024.

[24] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. Voice conversion from non-parallel corpora using variational auto-encoder. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–6. IEEE, 2016.

[25] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[26] Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6367–6371. IEEE, 2022.

[27] Navdeep Kaur and Parminder Singh. Conventional and contemporary approaches used in text to speech synthesis: A review. *Artificial Intelligence Review*, 56(7):5837–5880, 2023.

[28] Awais Khan, Khalid Mahmood Malik, and Shah Nawaz. Frame-to-utterance convergence: A spectra-temporal approach for unified spoofing detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10761–10765. IEEE, 2024.

[29] Kate Knibbs. Researchers say the deepfake biden robocall was likely made with tools from ai startup elevenlabs. `https://www.wired.com/story/biden-robocall-deepfake-elevenlabs/`, 2024. Accessed: 2024-04-30.

[30] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[31] William A Kretzschmar. *The linguistics of speech*. Cambridge University Press, 2009.

[32] Il-Youp Kwak, Sungsu Kwag, Junhee Lee, Youngbae Jeon, Jeonghwan Hwang, Hyo-Jung Choi, Jong-Hoon Yang, So-Yul Han, Jun Ho Huh, Choong-Hoon Lee, et al. Voice spoofing detection through residual network, max feature map, and depthwise separable convolution. *IEEE Access*, 2023.

[33] Seth Layton, Thiago De Andrade, Daniel Olszewski, Kevin Warren, Carrie Gates, Kevin Butler, and Patrick Traynor. Every breath you don't take: Deepfake speech detection using breath. *arXiv preprint arXiv:2404.15143*, 2024.

[34] Menglu Li, Yasaman Ahmadiadli, and Xiao-Ping Zhang. Audio anti-spoofing detection: A survey. *arXiv preprint arXiv:2404.13914*, 2024.

[35] Yuang Li, Min Zhang, Mengxin Ren, Miaomiao Ma, Daimeng Wei, and Hao Yang. Cross-domain audio deepfake detection: Dataset and analysis. *arXiv preprint arXiv:2404.04904*, 2024.

[36] Suk-Young Lim, Dong-Kyu Chae, and Sang-Chul Lee. Detecting deepfake voice using explainable deep learning techniques. *Applied Sciences*, 12(8):3926, 2022.

[37] Guan-Ting Lin, Chi-Luen Feng, Wei-Ping Huang, Yuan Tseng, Tzu-Han Lin, Chen-An Li, Hung-yi Lee, and Nigel G Ward. On the utility of self-supervised models for prosody-related tasks. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 1104–1111. IEEE, 2023.

[38] Kristen A Lindquist, Jennifer K MacCormack, and Holly Shablack. The role of language in emotion: Predictions from psychological constructionism. *Frontiers in psychology*, 6:121301, 2015.

[39] Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, jun 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340. URL https://doi.org/10.1145/3236386.3241340.

[40] Xuechen Liu, Xin Wang, Md Sahidullah, Jose Patino, Héctor Delgado, Tomi Kinnunen, Massimiliano Todisco, Junichi Yamagishi, Nicholas Evans, Andreas Nautsch, et al. Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[41] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[42] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling. The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. In *The Speaker and Language Recognition Workshop*, pages 195–202. ISCA, 2018.

[43] Jingze Lu, Yuxiang Zhang, Wenchao Wang, Zengqiang Shang, and Pengyuan Zhang. One-class knowledge distillation for spoofing speech detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11251–11255. IEEE, 2024.

[44] Juan M. Martín-Doñas and Aitor Álvarez. The vicomtech audio deepfake detection system based on wav2vec2 for the 2022 add challenge. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9241–9245, 2022. doi: 10.1109/ICASSP43922.2022.9747768.

[45] Momina Masood, Mariam Nawaz, Khalid Mahmood Malik, Ali Javed, Aun Irtaza, and Hafiz Malik. Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. *Applied intelligence*, 53(4):3974–4026, 2023.

[46] Driss Matrouf, J-F Bonastre, and Corinne Fredouille. Effect of speech transformation on impostor acceptance. In *2006 IEEE international conference on acoustics speech and signal processing proceedings*, volume 1, pages I–I. IEEE, 2006.

[47] Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Moller. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *Interspeech 2021*, 2021.

[48] Seyed Hamidreza Mohammadi and Alexander Kain. An overview of voice conversion systems. *Speech Communication*, 88:65–82, 2017.

[49] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7): 1877–1884, 2016.

[50] Nicolas Müller, Franziska Dieckmann, Pavel Czempin, Roman Canals, Konstantin Böttinger, and Jennifer Williams. Speech is silver, silence is golden: What do asvspoof-trained models really learn? *2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021.

[51] Nicolas Müller, Pavel Czempin, Franziska Diekmann, Adam Froghyar, and Konstantin Böttinger. Does audio deepfake detection generalize? *Interspeech 2022*, 2022.

[52] Nicolas M Müller, Philip Sperl, and Konstantin Böttinger. Complex-valued neural networks for voice anti-spoofing. *arXiv preprint arXiv:2308.11800*, 2023.

[53] Nicolas M Müller, Piotr Kawa, Wei Herng Choong, Edresson Casanova, Eren Gölge, Thorsten Müller, Piotr Syga, Philip Sperl, and Konstantin Böttinger. Mlaad: The multi-language audio anti-spoofing dataset. *arXiv preprint arXiv:2401.09512*, 2024.

[54] Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. Layer-wise analysis of a self-supervised speech representation model. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 914–921. IEEE, 2021.

[55] Ankita Pasad, Bowen Shi, and Karen Livescu. Comparative layer-wise analysis of self-supervised speech models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

 https://doi.org/10.52202/079017-2168

[56] Terry Pender. Ai threatens courts with fake evidence, uw prof says. https://www.jdsupra.com/legalnews/ai-threatens-courts-with-fake-evidence-7371356/, 2023. Accessed: 2024-05-05.

[57] Leonardo Pepino, Pablo Riera, and Luciana Ferrer. Emotion recognition from speech using wav2vec 2.0 embeddings. *Interspeech 2021*, 2021.

[58] Natalie Pereira, Ana Paula Bresolin Gonçalves, Mariana Goulart, Marina Amarante Tarrasconi, Renata Kochhann, and Rochele Paz Fonseca. Age-related differences in conversational discourse abilities a comparative study. *Dementia & Neuropsychologia*, 13:53–71, 2019.

[59] Zhaopeng Qian, Kejing Xiao, and Chongchong Yu. A survey of technologies for automatic dysarthric speech recognition. *EURASIP Journal on Audio, Speech, and Music Processing*, 2023(1):48, 2023.

[60] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL https://arxiv.org/abs/2212.04356.

[61] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[62] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. Speechbrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*, 2021.

[63] Alexandra Saliba, Yuanchao Li, Ramon Sanabria, and Catherine Lai. Layer-wise analysis of self-supervised acoustic word embeddings: A study on speech emotion recognition. *arXiv preprint arXiv:2402.02617*, 2024.

[64] Marc Schröder, Marcela Charfuelan, Sathish Pammi, and Ingmar Steiner. Open source voice creation toolkit for the mary tts platform. In *12th Annual Conference of the International Speech Communication Association-Interspeech 2011*, pages 3253–3256. ISCA, 2011.

[65] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian MüLler, and Shrikanth Narayanan. Paralinguistics in speech and language—state-of-the-art and the challenge. *Computer Speech & Language*, 27(1):4–39, 2013.

[66] Jui Shah, Yaman Kumar Singla, Changyou Chen, and Rajiv Ratn Shah. What all do audio transformer models hear? probing acoustic representations for language delivery and its structure. *arXiv preprint arXiv:2101.00387*, 2021.

[67] Tsu-Hsien Shih, Chin-Yuan Yeh, and Ming-Syan Chen. Does audio deepfake detection rely on artifacts? In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12446–12450. IEEE, 2024.

[68] 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)*, pages 5329–5333. IEEE, 2018.

[69] Hemlata Tak, Jee-Weon Jung, Jose Patino, Madhu Kamble, Massimiliano Todisco, and Nicholas Evans. End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. In *ASVSPOOF 2021, Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 1–8. ISCA, 2021.

[70] Hemlata Tak, Madhu Kamble, Jose Patino, Massimiliano Todisco, and Nicholas Evans. Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6382–6386. IEEE, 2022.

[71] Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas Evans. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. In *The Speaker and Language Recognition Workshop (Odyssey 2022)*. ISCA, 2022.

[72] Xu Tan. *Neural text-to-speech synthesis*. Springer Nature, 2023.

[73] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*, 2021.

[74] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[75] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. Asvspoof 2019: Future horizons in spoofed and fake audio detection. *arXiv preprint arXiv:1904.05441*, 2019.

[76] Markus Toman and Michael Pucher. An open source speech synthesis frontend for hts. In *Text, Speech, and Dialogue: 18th International Conference, TSD 2015, Pilsen, Czech Republic, September 14-17, 2015, Proceedings 18*, pages 291–298. Springer, 2015.

[77] Andreas Triantafyllopoulos and Björn W Schuller. Expressivity and speech synthesis. *arXiv preprint arXiv:2404.19363*, 2024.

[78] Andreas Triantafyllopoulos, Björn W Schuller, Gökçe İymen, Metin Sezgin, Xiangheng He, Zijiang Yang, Panagiotis Tzirakis, Shuo Liu, Silvan Mertes, Elisabeth André, et al. An overview of affective speech synthesis and conversion in the deep learning era. *Proceedings of the IEEE*, 2023.

[79] Nik Vaessen and David A Van Leeuwen. Fine-tuning wav2vec2 for speaker recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7967–7971. IEEE, 2022.

[80] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12, 2016.

[81] Xin Wang and Junichi Yamagishi. Investigating self-supervised front ends for speech spoofing countermeasures. *arXiv preprint arXiv:2111.07725*, 2021.

[82] Xin Wang and Junichi Yamagishi. Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[83] Xin Wang and Junichi Yamagishi. Can large-scale vocoded spoofed data improve speech spoofing countermeasure with a self-supervised front end? In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10311–10315. IEEE, 2024.

[84] Yingzhi Wang, Abdelmoumene Boumadane, and Abdelwahab Heba. A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding. *arXiv preprint arXiv:2111.02735*, 2021.

[85] David Weenink. Canonical correlation analysis. In *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, volume 25, pages 81–99. University of Amsterdam Amsterdam, 2003.

[86] Zhizheng Wu, Oliver Watts, and Simon King. Merlin: An open source neural network speech synthesis system. In *9th ISCA Speech Synthesis Workshop*, pages 202–207, 2016.

[87] Yuankun Xie, Haonan Cheng, Yutian Wang, and Long Ye. Learning a self-supervised domain-invariant feature representation for generalized audio deepfake detection. In *Proc. INTERSPEECH*, volume 2023, pages 2808–2812, 2023.

[88] Amit Kumar Singh Yadav, Ziyue Xiang, Kratika Bhagtani, Paolo Bestagini, Stefano Tubaro, and Edward J Delp. Ps3dt: Synthetic speech detection using patched spectrogram transformer. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 496–503. IEEE, 2023.

[89] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit, 2019.

[90] Masahiro Yanagawa and Junya Sato. Seeing is not always believing: Discrepancies in saliency maps. *Radiology: Artificial Intelligence*, 6(1):e230488, 2023.

[91] Shu-wen Yang, Heng-Jui Chang, Zili Huang, Andy T Liu, Cheng-I Lai, Haibin Wu, Jiatong Shi, Xuankai Chang, Hsiang-Sheng Tsai, Wen-Chin Huang, et al. A large-scale evaluation of speech foundation models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[92] Yujie Yang, Haochen Qin, Hang Zhou, Chengcheng Wang, Tianyu Guo, Kai Han, and Yunhe Wang. A robust audio deepfake detection system via multi-view feature. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13131–13135. IEEE, 2024.

 https://doi.org/10.52202/079017-2168

[93] Jiangyan Yi, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao. Audio deepfake detection: A survey. *arXiv preprint arXiv:2308.14970*, 2023.

[94] Zhao Yi, Wen-Chin Huang, Xiaohai Tian, Junichi Yamagishi, Rohan Kumar Das, Tomi Kinnunen, Zhen-Hua Ling, and Tomoki Toda. Voice conversion challenge 2020—intra-lingual semi-parallel and cross-lingual voice conversion–. In *Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*. ISCA, 2020.

[95] Chang Zeng, Xin Wang, Xiaoxiao Miao, Erica Cooper, and Junichi Yamagishi. Improving generalization ability of countermeasures for new mismatch scenario by combining multiple advanced regularization terms. *arXiv preprint arXiv:2305.10940*, 2023.

[96] Yuxiang Zhang, Zhuo Li, Jingze Lu, Hua Hua, Wenchao Wang, and Pengyuan Zhang. The impact of silence on speech anti-spoofing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

A Appendix

A.1 Details of generative models tested in the CCA analysis

Details on the systems tested in Table. 1 are as follows. A01 is a TTS model that uses Flite [76] that converts text into a sequence of linguistic features. A Hidden Markov Model (HMM) is then used to estimate the duration of phones. The acoustic features are predicted by an NN-based acoustic model, including MFCCs, F0, and voicing flags, which are input to the WaveNet vocoder [80] for waveform synthesis. A02 and A03 follows a similar linguistic-acoustic-vocoder pipeline as A01, where A02 uses the WORLD vocoder [49] and A03 relies on recipes from the Merlin toolbox [86]. A04 is a waveform concatenation TTS model based on the MaryTTS platform [64], a major difference between A04 and the previous three models is that A04 preserves the short-term acoustic features of the natural speech, which may make it more difficult to separate from real speech. A05 and 06 are VC models. A05 encodes the spectral features into speaker-independent embeddings using a VAE and decode them with a desired target speaker representation [24]. A06 relies on the source-filter modelling of speech where the conversion can be achieved by replacing the filters of the input signals by that of the target speaker [46]. The non-speech segments are kept intact during conversion.

A.2 Layer-wise analysis of pretrained SSL models

As mentioned in Section. 3.2.1, we use the `Wav2vec-XLSR` model finetuned for emotion recognition (Wav2vec-SER) and speech recognition (Wav2vec-ASR) tasks to extract the style and linguistics representations, respectively.

The style representation is based on the pretrained model obtained from `https://huggingface.co/ehcalabres/wav2vec2-lg-xlsr-en-speech-emotion-recognition` and the linguistics representation is based on the pretrained model obtained from `https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english`. To obtain a maximal disentanglement between the two subspace representations, we calculate Spearman's rank correlation coefficient values between different layers from the two models to examine the layer-wise similarity. These correlation values and our final layer selection are demonstrated in Figure 5. Based on existing works, which showed how linguistics and paralinguistics information propagate through layers, we choose layer 0-10 from Wav2vec-SER backbone to represent style information, and layer 14-21 from Wav2vec-ASR backbone to represent linguistics information. The correlation values between these two groups are shown close to 0, indicating a better disentanglement.

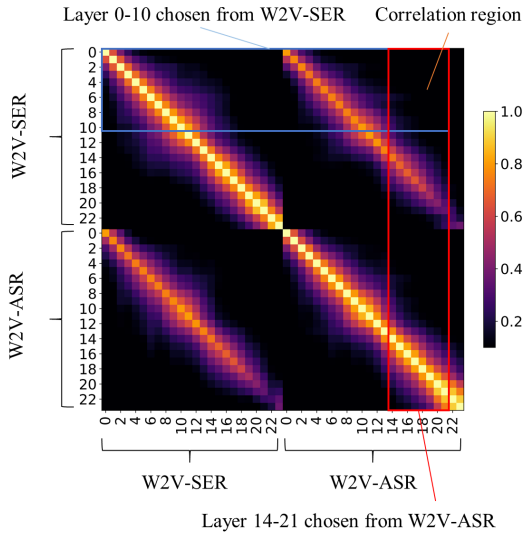

Figure 5: Spearman correlation coefficients calculated across all layers from two pretrained Wav2vec-XLSR backbones. Blue highlights layers 0-10 from Wav2vec-SER to represent style information. Red highlights layers 14-21 from Wav2vec-ASR to represent linguistics information. The correlation values between the selected layers can be read from the overlapping region.

https://doi.org/10.52202/079017-2168

A.3 Dataset details

Table 3 describes the details of datasets used for Stage 1 and Stage 2 training and evaluation. Figure 6 shows the projected WavLM embeddings for real and fake samples from the four employed datasets using t-SNE. We choose WavLM since it is the top-performing model in the single-encoder category (Table 2). For both classes, an overlap can be seen between `ASVspoof2019` and `ASVspoof2021` samples, while samples from `In-the-wild` and `MLAAD-EN` can be separated nearly perfectly. This corroborates with the results reported in Table 2 where all employed ADD systems trained on `ASVspoof2019` perform better on `ASVspoof2021` than `In-the-wild` and `MLAAD-EN`.

Table 3: Summary of datasets used for Stage 1 and Stage 2 training and evaluation.

Stage 1 datasets							
Name	Split	#Sample	#Real	#Fake	#Attacks	Speech type	Environment
`Common Voice`	Train	3k	3k	–	–	Scripted	Crowdsourced
`RAVDESS`	Train	3k	3k	–	–	Scripted	Studio
`19 LA train`	Valid	500	500	–	–	Scripted	Studio
Stage 2 datasets							
Name	Split	#Sample	#Real	#Fake	#Attacks	Speech type	Environment
`19 LA train`	Train	25380	2580	22800	6	Scripted	Studio
`19 LA dev`	Valid	24884	2548	22336	6	Scripted	Studio
`19 LA eval`	Test	71237	7355	63882	17	Scripted	Studio
`21 DF eval`	Test	611829	22617	589212	100+	Scripted	Studio
`In-the-wild`	Test	31779	11816	19963	N/A	Spontaneous	In-the-wild
`MLAAD-EN`	Test	37998	18999	18999	25	Scripted	Studio

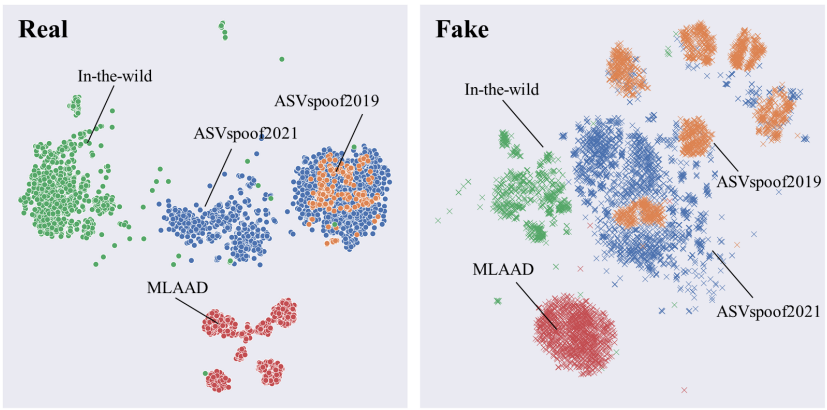

Figure 6: Projected WavLM embeddings for real and fake classes from the four employed datasets. Left: real class embeddings. Right: fake class embeddings.

A.4 Details of the compression module

Figure 7 shows the architecture of the compression module. The input is first passed through a pooling layer to obtain an average of different SSL layer outputs. Since the goal of compression modules is to project the original style/linguistics embeddings to a subspace where the compressed embeddings can be maximally correlated, we use bottleneck layers to remove the redundant information that is not shared across the two subspaces. Similar to the design of an autoencoder [74], the bottleneck layer first compresses the feature dimension from 1024-dim to 256-dim, then recovers it back to 1024-dim. In practice, we found using only one bottleneck layer is enough to obtain meaningful compressed representations. A projection head is applied at the end to reduce the final output dimension to 256.

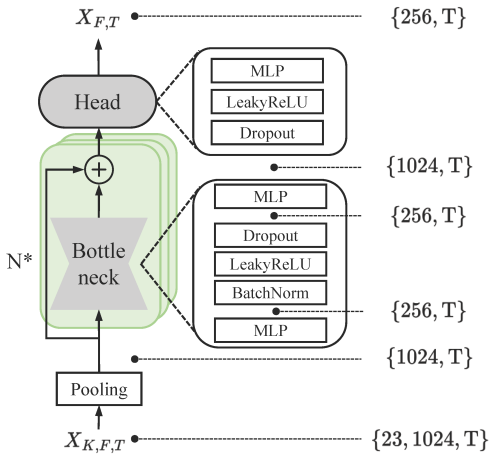

Figure 7: Architecture of the compression module with input and output dimensions. Input $\mathbf{X_{K,F,T}}$ represents the original subspace representation encoded by the SSL frontend, where K denotes the transformer layer index, F denotes the feature size, and T denotes the number of time steps.

A.5 PyTorch implementation of the Stage 1 training objective

Algorithm 1 shows a PyTorch-style implementation of the Stage 1 training objective, which minimizes the cross-subspace distance loss and an intra-subspace redundancy loss. The subspace embeddings are first normalized across the whole batch before passing into the loss calculations. We experimented with two types of distance for the cross-subspace loss: Euclidean and Cosine distance. While no significant difference is found when comparing the performance achieved by the two, the former provides slightly better results, hence is adopted as the final distance measure.

Algorithm 1: PyTorch-style code for the Stage 1 loss function

Input: x_{style}, $x_{linguistic}$
Output: L_{stage1}

```
// Normalize embeddings from both subspaces
batch_size = x_style.shape[0]
x_style_norm = torch.nn.BatchNorm1d(x_style, affine=False) / batch_size
x_linguistic_norm = torch.nn.BatchNorm1d(x_linguistic, affine=False) / batch_size
// Computation of cross-subspace loss
D = torch.linalg.norm(x_style_norm - x_linguistic_norm, ord='fro')
D = torch.pow(D, 2)
// Computation of intra-subspace redundancy loss
v_linguistic = torch.mm(x_linguistic_norm.T, x_linguistic_norm)
C_linguistic = torch.linalg.norm(v_linguistic-torch.eye(v_linguistic.shape[-1]))
C_linguistic = torch.pow(C_linguistic, 2)
v_style = torch.mm(x_style_norm.T, x_style_norm)
C_style = torch.linalg.norm(v_style-torch.eye(v_style.shape[-1]))
C_style = torch.pow(C_style, 2)
// Final loss term
L_stage1 = D + λ (C_style+C_linguistic)
```

A.6 Performance comparison of different backend classifiers

Table 4 shows the performance obtained when the ASP+MLP layers are swapped with other layer choices.

https://doi.org/10.52202/079017-2168

Table 4: Performance comparison of different backend classifiers used in SLIM. Frontend encoders are frozen.

SLIM backend	EER			
	ASVspoof2019	ASVspoof2021	In-the-wild	MLAAD-EN
Original (ASP+MLP)	0.6	8.3	12.9	13.5
None	0.9	9.1	13.1	13.7
LLGF	0.4	7.5	13.5	13.0
LCNN	0.3	7.9	12.8	13.9

A.7 Hyperparameters and computation resources

Table 5 describes the optimal hyperparameters and architecture details of SLIM used for Stage 1 and Stage 2 training. The hyperparameter names of the data augmentation modules can be found in SpeechBrain *v1.0.0*. Experiments were conducted on the Compute Canada cluster [9] with a total of four NVIDIA V100 GPUs (32GB RAM).

Table 5: Hyperparameters and architecture details of SLIM.

Parameter	SLIM
Stage 1 Optimization	
Batch size	16
Epochs	50
GPUs	4
Audio length	5s
Optimizer	AdamW
LRscheduler	Linear
Starting LR	.005
End LR	.0001
Early-stop patience	3 epochs
λ	.007
Training time	3h
SSL frontend	
Style encoder	Wav2vec-XLSR-SER
Style layers	0-10
Linguistic encoder	Wav2vec-XLSR-ASR
Linguistic layers	14-21
Compression module	
Bottleneck layers	1
BN dropout	0.1
FC dropout	0.1
Compression output dim	256
Stage 2 Optimization	
Batch size	2
Epochs	10
GPUs	4
Audio length	5s
Optimizer	AdamW
LRscheduler	Linear
Starting LR	.0001
End LR	.00001
Early-stop patience	3 epochs
Training time	10h
Classifier	
FC dropout	0.25
Stage 2 data augmentation	
Num augmentations	1
Concat with original	True
Augment prob	1
Augment choices	Noise, Reverb, SpecAug
SNR_high	15dB
SNR_low	0dB
Reverb	RIR noise
Drop_freq_low	0
Drop_Freq_high	1
Drop_freq_count_low	1
Drop_freq_count_high	3
Drop_freq_width	.05
Drop_chunk_count_low	1
Drop_chunk_count_high	5
Drop_chunk_length_low	1000
Drop_chunk_length_high	2000

https://doi.org/10.52202/079017-2168

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions and scope of our study are listed at the end of Section 1, and summarized in the abstract. The experimental results in Section 4 support our claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 5 and in parts of Section 4.3. In Section 2, we also present gaps in the literature and the resulting limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We take an experimental approach to validate our hypothesis on style-linguistics mismatch in deepfake audios. Details on the motivation and the experiments are presented in Section 3.1. The training objectives of our framework are detailed in Section 3.2.1, Equations 1 and 2.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our proposed framework is described in detail in Section 3. We also provide details regarding model architecture and its implementation (Appendix A.4, A.5), pseudocode of the training objective (Appendix A.5), and hyperparameters used for obtaining the reported performance (Appendix A.7).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

https://doi.org/10.52202/079017-2168

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets employed in this study are publicly available (Appendix A.3). While the training script and model weights are not explicitly released, we provided sufficient details in Section 3 and Appendix A to faithfully reproduce and verify our results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and test procedures are detailed in Section 4.1. The data splits are provided in Appendix A.3. The hyperparameters are listed in Appendix A.7.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We conducted statistical significance tests for our experiments. The details can be found in Section 3.1 and Section 4.2.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computational resources are described in Appendix A.7.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Authors have reviewed and respect NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

https://doi.org/10.52202/079017-2168

Justification: As described in Section 1, our proposed framework is designed to detect deepfake audios, hence our work can be used for positive societal impacts. We have also presented limitations in Sections 4.3 and 6 to scope-limit the scenarios in which our framework can be used.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The employed datasets, code, and model backbones are all properly cited (Sections 3, 4.1, 4.1). The URLs for obtaining pretrained models are also provided (Appendix A.2).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

https://doi.org/10.52202/079017-2168

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.