MSc Artificial Intelligence
Master Thesis

Impact of content and channel on automatic speaker verification

by
David Bikker
14667045

June 26, 2025

36 EC
January - June 2025

Supervisor:
Eleni Sergidou

Examiner:
Prof Dr Ing Zeno Geradts

Second reader:
Charlotte Pouw

This research was carried out during an internship at the Netherlands Forensic Institute.

Universiteit van Amsterdam

Acknowledgements

Developing and completing this thesis was a great experience. It could not have been so without the support of others. I am very grateful to my external supervisor Elina, who proposed the original idea for the thesis project and has always been keen to be involved. Our weekly meetings were something to look forward to, as I could share my progress and discuss relevant issues, to get new ideas. Elina's writing feedback was also very helpful: I know myself to have an eye for detail, but was impressed with the granularity at which I received feedback, when appropriate.

Likewise I am very grateful to my internal supervisor, Charlotte. She was excited to supervise the project from the beginning, which further increased my enthusiasm. Meeting up with Charlotte or both my supervisors at important milestones was very constructive, in receiving feedback on my approach or writing. Furthermore, thanks go to Charlotte for inviting me to meetings of the Cognition, Language and Computation lab, where I was able to present my ideas and gain new ones. Lastly, thanks to Charlotte for setting up a meeting with Hosein Mohebbi, main author of the paper of which I adopt the disentanglement approach, and thanks to Hosein for being interested in my research and giving his ideas on my work.

I am quite thankful for my time at the Netherlands Forensic Institute (NFI), where Elina helped me to get involved. *Bewijswaardering en statistiek*'s *dagstart* was a great regularity whether working from NFI or from home, and I also enjoyed having lunch with the group. Other events that were open to me, in particular the 'data science guild' demos, were a great way to spice up my week and get renewed inspiration. I also want to thank my fellow interns at *Digitale en biometrische sporen*. It was fun to share ideas with peers. Lastly, I want to thank the organisation of NFI's Science Fair, where I was able to present a poster on the first part of my work. This was a great experience in itself, but also lead to several improvements to my thesis.

With regard to support from people in my private sphere, I refrain from enumerating names. However, besides my friends and family, I do want to thank my student's association, C.S.V. Ichthus Utrecht, and my study association, U.S.C.K.I. Incognito, for offering diversion from my studies. My focus has been on this thesis for the past six months, but it was good to put time into other activities as well.

Contents

Abstract

In automatic speaker verification, the goal is to verify whether two utterances originate from the same speaker. Given an audio signal, a system based on deep neural networks can be used to derive an embedding that represents its speaker. This embedding can then be compared against a speaker embedding of another audio signal, computing a similarity score. Ideally, these speaker embeddings should capture inter-speaker variability and be robust to variability unrelated to the speaker's identity. However, previous research shows that speaker embeddings can also capture other attributes of utterances, such as content and channel information. First, this work investigates the degree to which content and channel are represented in speaker embeddings. We use probing, an interpretability technique from natural language processing, and find that all investigated speaker verification models, with the exception of x-vector, suppress content and channel information. This suppression happens mainly at the final model layer. Next, we explore whether we can further disentangle speaker, content and channel in ECAPA-TDNN, a speaker verification model that showed good disentanglement in our experiments, and is considered state-of-the-art. We apply a disentanglement approach based on the Variational Information Bottleneck. Comparing to a general pretrained speech model and x-vector, results reinforce our finding that the level of disentanglement present in ECAPA-TDNN is almost optimal, i.e., hard to improve. Finally, we contribute a novel synthesized dataset that includes controlled variation of speaker, content and channel.

Chapter 1

Introduction

When we listen to speech, we not only process what is said, but also by whom the utterance is made, and familiar voices are easily recognised. If a voice is unfamiliar, however, it can be hard to verify whether two utterances were spoken by the same person (Bimbot et al., 2004). In automatic speaker recognition, a task within the field of speech processing, human speaker recognition ability is sought to be replicated and exceeded.

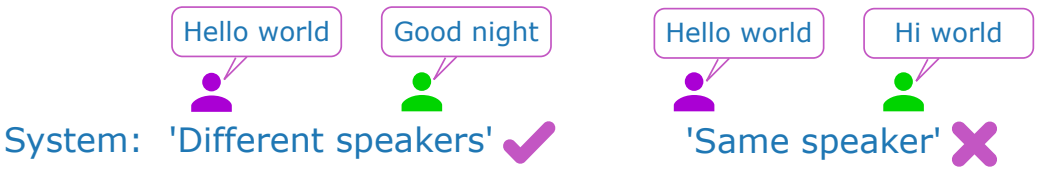
Speaker recognition can be divided into speaker verification, speaker identification and speaker diarization (Bai and Zhang, 2021). In speaker verification, the goal is to verify whether two utterances[1] originate from the same speaker. The aim of speaker identification is detecting which of the speakers in a database produced a given utterance. Finally, speaker diarization aims to partition a multi-participant conversation into utterances from individual speakers. All three branches make use of speaker embeddings that aim to represent the characteristics of individual speakers (Wang et al., 2024b).

Automatic speaker verification is used in forensic speaker comparison, where the goal is to produce a *likelihood ratio* of a recording coming from a suspected speaker, versus it coming from an arbitrary speaker from the relevant population (Champod and Meuwly, 2000; Hansen and Hasan, 2015; Morrison et al., 2020). In this context, and in our study, the task is *text-independent* speaker verification, meaning that there is no constraint on the content of the utterance to be compared (Bimbot et al., 2004; Kabir et al., 2021).

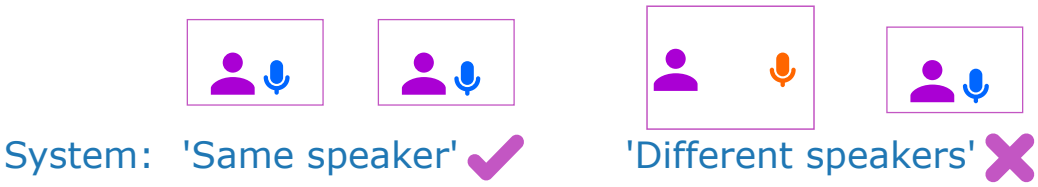
Recently, the field of automatic speaker verification has seen a shift from statistical models to deep learning (Jakubec et al., 2024; Sharma et al., 2024). In the early 2010s, the state-of-the-art in automatic speaker verification was based on Gaussian Mixture Models (Reynolds, 2009). Several deep learning alternatives were proposed, of which *x-vector* (Snyder et al., 2017, 2018) became the dominant approach (Sharma et al., 2024). x-vector represents an utterance by a low-dimensional speaker embedding of constant length. Since its release, state-of-the-art speaker verification models have used deep learning to learn a speaker embedding (Jakubec et al., 2024).

Ideally, speaker embeddings should capture inter-speaker variability and be robust to variability unrelated to the speaker's identity. However, variability in a speech signal can come from many different sources, and it has been shown in earlier research that deep speaker embeddings capture other attributes of utterances too (Peri et al., 2020a; Raj et al., 2019; Wang et al., 2017). In this work, we focus on representation of content (i.e., *what* is said) and channel (persistent acoustic information in a recording, such as background noise and room acoustics). These attributes are commonly targeted in speaker verification research, as they are the main sources of variation in a recording, besides those resulting from the speaker. If content and channel are represented by speaker embeddings, this can affect the outputs of automatic speaker

[1] As common in the field of speaker recognition, we use 'utterance' to refer to a complete audio signal representing a (short) recording of human speech, including background noise, silence, etc.



(a) Content.



(b) Channel.

Figure 1.1: **Illustration of potential impact of content and channel.** Speaker identity and microphone type are indicated by different colours.

verification systems, as illustrated in Figure 1.1. This is undesirable for any application, and especially for forensic speaker comparison, where a reported likelihood ratio should not be affected by unintended factors.

We investigate to what degree content and channel information are increased or suppressed in embeddings from deep neural models trained for speaker verification. We use probing, an interpretability technique from natural language processing. Improving over earlier research, we implement minimum-description length probing (Voita and Titov, 2020), which has been shown to be more robust against hyperparameter choices, and compare against multiple baselines.

Next, we attempt to further disentangle content and channel from speaker embeddings, without decreasing speaker verification performance. We use a recently proposed disentanglement technique by Mohebbi et al. (2024).

The remainder of this thesis is structured as follows. In Chapter 2, we introduce related work regarding interpretability[2] of automatic speaker verification and disentangled speaker representation learning. Chapter 3 gives a technical introduction to automatic speaker verification, focusing on the common pipeline and what has been done to improve robustness to irrelevant variability. In Chapter 4, we introduce the interpretability and disentanglement techniques used in this work and go over the models investigated and the datasets used. In Chapter 5, we present our experimental results and draw conclusions. Finally, in Chapter 6, we reflect on our contributions, touch on limitations and give directions for future research.

[2]In the field of explainable AI, the terms *explainability* and *interpretability* are often used interchangeably (Linardatos et al., 2021). In this work, we use *interpretability* to refer to interpreting the information represented in embeddings from speaker verification models.

Chapter 2

Related work

This thesis concerns interpretability and disentanglement of embeddings for automatic speaker verification. This chapter introduces related work: in Section 2.1, we introduce four lines of research into speaker verification interpretability, and in Section 2.2 we discuss variables in existing work into disentanglement of speaker verification.

2.1 Interpretability of automatic speaker verification

There has been a growing interest in interpretability of speaker verification (Wang et al., 2024b). This section introduces four lines of research. We consider the strengths and weaknesses of techniques within each, to be able to decide how to investigate representation of content and channel in speaker embeddings.

Probing speaker embeddings. In probing techniques, which originate in natural language processing, a new model is trained to investigate the representations of the model of interest (Belinkov and Glass, 2019). The weights of the investigated model are frozen, and representations on an annotated dataset are generated. A different classifier is then trained on the activations from the frozen model, to predict an attribute of interest (e.g., speaker identity). Performance of this classifier is taken as a proxy for how well the original model represents the attribute. In speaker verification research, probing was first used in Wang et al. (2017). Later studies include Raj et al. (2019), Peri et al. (2020a) and Zhao et al. (2022). Each study differs in what properties are probed for, but a common finding is that speaker embeddings include information that does not seem relevant for speaker verification, including representation of content and channel. It should be noted that probing tasks have some shortcomings, relating to whether performance on the probing task is a reliable proxy for the information represented and used by the original model (Belinkov, 2022). Important design decisions include what baselines and controls to use, and what architecture and complexity to use for the probe. These issues are commonly overlooked in research using probing, including previous work into interpretability of speaker verification.

Visualizing signal salience. Another line of research uses visualization techniques to show what parts of the signal are important to the model's decision. This technique stems from computer vision, where Class Activation Mapping (CAM) (Zhou et al., 2016) and variants have been popular techniques to explain model decisions. CAM techniques overlay a heatmap on an image to indicate what regions are most important for the model's decision. They have been applied to speaker verification to show what parts of the signal's spectrogram a model focuses on (Li et al., 2023b; Yao et al., 2024; Zhou et al., 2021). However, it was demonstrated by Li et al. (2022b) that different CAM variants can yield different and potentially unreliable

visualizations in the context of speaker verification. Moreover, CAM-based methods can only be used on models based on convolutional neural networks (Krizhevsky et al., 2012). Zhang et al. (2023, 2024) propose and use a new visualization framework that can be used with other model architectures, comparing different attribution algorithms used in explainable AI. However, the reported observations are not apparently grounded in linguistic theory.

Similarity analysis. In similarity analysis, layers from different models, or different layers within a model, are compared. This technique has been used in natural language processing for several years (Wu et al., 2020; Chung et al., 2021), but has not yet seen much application in the field of speaker verification. Ashihara et al. (2024) compare layers within and across models, looking into self-supervised speech models, a self-supervised speaker model and a supervised speaker model.

Intrinsic interpretability. The methods hereto presented focus on generating post-hoc explanations from a black-box model. It has been argued that inherently interpretable model architectures should be developed instead (Rudin, 2019). Ben-Amor and Bonastre (2022) compute binary attributes representing voice characteristics present or absent in utterances, which Ben-Amor et al. (2023) match to phonetic variables. In Ma et al. (2025), the speaker verification model uses representations of phonetic traits to aggregate a speaker embedding, and an explanation can be generated that shows which phones are or are not similar between the two utterances. In contrast to these phonetic explanations, Wu et al. (2024) use a two-stage architecture where in the first stage classifiers are trained to predict nationality, age, gender and profession, and in the second stage the softmax label distributions of these classifiers are used to train a speaker verification model. While this approach gives more insight into the attributes used by the model, the decrease in speaker verification performance in the experiments in Wu et al. (2024) is quite significant.

2.2 Disentangled speaker representation learning

Deep learning models are often described as *black boxes*, learning representations without (explicitly) distinguishing different attributes of the data. Disentangled representation learning aims to learn representations that disentangle distinct features underlying the data, which can improve interpretability, controllability and generalizability (Bengio et al., 2013; Wang et al., 2024c).

In the context of speech representations, disentangled representations can benefit several applications. In speech synthesis, disentangled representations can help control the style of synthesized examples while keeping content the same (Kumar et al., 2021), which is also useful in voice conversion (Chou et al., 2018) and speaker anonymization (Matassoni et al., 2024). In automatic speech recognition, disentangled representations can help reduce the effects of inter-speaker variability on transcription performance (Meng et al., 2018). In speaker verification, the focus of this thesis, disentangled representations improve robustness against different sources of signal variability, which might stem from the speaker (e.g., emotion, age) but can also be external (e.g., background noise, room acoustics) (Hansen and Hasan, 2015; Peri et al., 2020b). This section discusses some variables in disentangled speaker representation learning.

Attributes to be disentangled. When disentangling representations, it is possible to let the latent attributes be determined by the algorithm. Hsu et al. (2017), Nagrani et al. (2020a) and Li and Mandt (2018) apply such a general approach to speech data and evaluate on speaker verification. Tai et al. (2020) and Kwon et al. (2020) focus on speaker verification and aim to disentangle all speaker-unrelated information generally.

However, much research has a goal of disentangling specific attributes of a speech signal. One common objective is removing channel information from speech signals. In this context, *channel* refers to the persistent acoustic information in a recording, which includes factors such as background noise, room response and distance from the microphone (Hansen and Bořil, 2018; Kang et al., 2020; Morrison et al., 2020). In this area, domain adaptation is a common objective, where a system trained on a domain with sufficient data is adapted to a domain for which less resources are available (Bai and Zhang, 2021). Domain adaptation approaches use either labeled or unlabeled data from the domain that is evaluated on (Li et al., 2023a; Yi and Mak, 2022; Zhou et al., 2019). There are also approaches that generalise to several target domains, but do require (unlabeled) data from the domains that are evaluated on in training (Wang et al., 2021c; Wei et al., 2022). If the goal is to disentangle channel information generally, irrespective of the target domain, the model has to be trained and evaluated on different domains (Kang et al., 2022a; Meng et al., 2019; Tu et al., 2019).

Another common goal is reducing the effect of the linguistic content of utterances. Some approaches aim to disentangle on the phoneme-level (Chen and Bao, 2021; Hong et al., 2023; Tawara et al., 2020; Wang et al., 2022a,b), while others target larger lexical structures (Krishna and Ganapathy, 2024; Mohebbi et al., 2024).

Besides content and channel, various other attributes have been targeted. Considered attributes include language identity (Kang et al., 2022b; Nam et al., 2023), style/genre (Kang et al., 2022b; Williams and King, 2019), emotion (Kang et al., 2020; Li et al., 2020; Williams and King, 2019), speaking rate (Tong et al., 2022), gender (Noé et al., 2021, 2022) and age (Qin et al., 2022). Instead of disentangling speaker identity from other attributes, Luu et al. (2022) aim to disentangle speaker identity into nationality, gender and age.

Supervision signal. An important distinction in disentangled representation learning is whether a method is supervised or unsupervised (Wang et al., 2024c). Many disentanglement methods, also in the context of speaker embeddings, are unsupervised. A common justification for this is the cost of obtaining large labeled datasets (Nagrani et al., 2020a; Peri et al., 2020b; Qu et al., 2024). If a training signal is obtained from the data by the framework, an approach might also be called self-supervised: in either case, no dataset labels are required. Some research into disentanglement of speaker embeddings uses no dataset labels at all, only exploiting the information present in the audio representation. Krishna and Ganapathy (2024) and Tjandra et al. (2021) construct frame-based and utterance-based representations and use a mutual information loss between them. Lin et al. (2023) use neural factor analysis to learn utterance-level representations of attributes such as speaker and language. Other work into unsupervised disentanglement uses speaker labels, but does not require labels for the other attributes to be disentangled (Li et al., 2024; Liu et al., 2023; Peri et al., 2020b).

While unsupervised or self-supervised learning alleviates the need for large labeled datasets, it has been shown that, in general, disentangled representations cannot be learnt reliably without supervision, and assuming no access to labeled data, successfully disentangled models cannot be identified (Locatello et al., 2019). While, for content and channel disentanglement, there is a large body of work applying unsupervised techniques, there is also a lot of research using datasets labeled for content (Liu et al., 2022; Mohebbi et al., 2024; Wang et al., 2022b) and channel (Chen et al., 2020; Fang et al., 2019; Kang et al., 2020). Alternatively, some approaches construct a dataset for content by using automatic speech recognition (Chen and Bao, 2021; Hong et al., 2023; Wang et al., 2019), or a dataset for channel using noise augmentation (Kang et al., 2022a; Meng et al., 2019; Yi and Mak, 2022; Zhou et al., 2019).

Disentanglement position. Speaker representation disentanglement can be specific to a model architecture or work on embeddings generally. Most approaches incorporate disentanglement techniques in pretraining (Li et al., 2024; Liu et al., 2023; Nam et al., 2023) or finetuning

(Mun et al., 2022). There is also work on disentangling already trained speaker embeddings, requiring no modification to the model architecture (Mohebbi et al., 2024; Noé et al., 2022; Williams and King, 2019).

Chapter 3

Background

This chapter serves as a technical introduction to the field of automatic speaker verification, the subject of this thesis. In Section 3.1, we introduce the general automatic speaker verification pipeline, for a better understanding of how current systems work and what we aim to investigate and improve. In Section 3.2, we go over commonly used techniques to improve robustness against irrelevant variability in various parts of the speaker verification pipeline. These techniques can partly mitigate entangledness in currently used systems.

3.1 Speaker verification pipeline

In a typical automatic speaker verification system, first an utterance is preprocessed and features are extracted from it, then the features pass through a deep neural network, outputting a speaker embedding, which finally can be compared to another speaker embedding by a similarity measure to obtain a score (Mohd Hanifa et al., 2021; Jakubec et al., 2024). This pipeline, consisting of distinct steps, is depicted in Figure 3.1. If the obtained score is above a certain threshold, the utterances are predicted to originate from the same speaker, and if it is below that threshold, they are predicted to be from different speakers (Bai and Zhang, 2021). Preprocessing, feature extraction and the deep neural network are together often referred to as the *front-end*, and the similarity measure is referred to as the *back-end* (Bai et al., 2022; Li et al., 2022a; Zeng et al., 2024).

To train the deep neural network that outputs the speaker embeddings, the target task is initially speaker identification, with the objective of predicting the identifier of the speaker, for a dataset with a closed set of speakers (Bai and Zhang, 2021; Desplanques et al., 2020). When the model has converged, the output layer is removed, and the representations from the

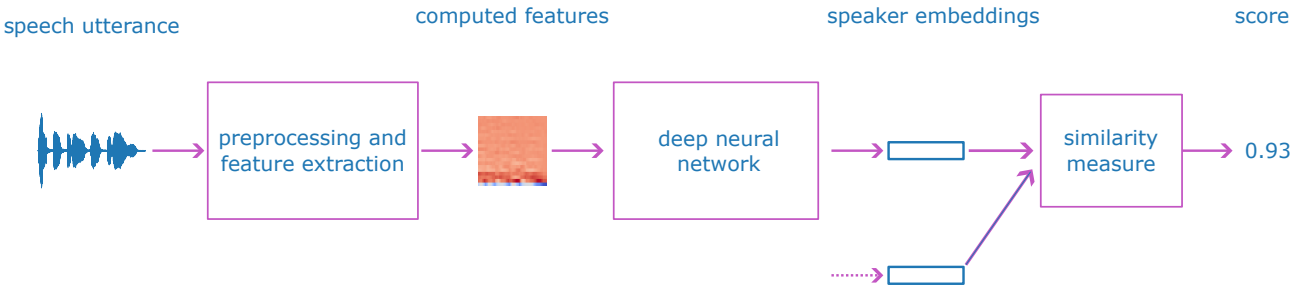


Figure 3.1: **Automatic speaker verification pipeline.** Features are extracted from an utterance and input into a deep neural network, which outputs a speaker embedding. Embeddings from two utterances are compared by a similarity measure to obtain a score which indicates whether the utterances come from the same speaker. For illustrative purposes, the figure shows examples of a waveform, computed features and a similarity score.

layer preceding the output layer are used as speaker embeddings. These speaker embeddings can also represent speakers outside the training set. Alternatively, a system can be trained with a verification-based objective function (Bai et al., 2022; Li et al., 2017), but this is not as common, and it is harder to construct enough data trials and select those that are effective in training the system (Bai and Zhang, 2021). Moreover, systems have been proposed that do not use a separate feature extraction step either, going in one stage from input utterances to a final score (Heigold et al., 2016; Jung et al., 2019). As the field currently stands, however, most automatic speaker verification approaches use a pipeline such as depicted in Figure 3.1, as do the models considered in this work. The next three sections further discuss each of the three parts of the pipeline.

Preprocessing and feature extraction. In most speaker verification systems, before an utterance is passed through a deep neural network to obtain a speaker embedding, preprocessing and feature extraction algorithms are applied. Preprocessing often includes *voice activity detection* (VAD), pre-emphasis and normalization. An audio file of an utterance might include parts that contain only silence or noise instead of speech. VAD can be used to allow the model to focus on parts containing speech (Hansen and Hasan, 2015; Morrison et al., 2020). Sometimes VAD is done after feature extraction, or not at all, if the data used do not contain noise or silence (Bai et al., 2022; Kang et al., 2022a). Pre-emphasis is a high-pass filter, which is used to compensate for high frequencies lost during speech production (Abdul and Al-Talabani, 2022; Jahangir et al., 2021; Zi and Xiong, 2024). Normalization is applied to remove the effects of variance in amplitude between different recordings, which might be caused by intra-speaker variation or differences between recording sessions (Jahangir et al., 2021; Renisha and Jayasree, 2019).

In the feature extraction step, an algorithm is used to obtain acoustic features from the speech signal. These algorithms do not include trainable weights (Ohi et al., 2021). *Mel-frequency cepstral coefficients (MFCCs)* (Davis and Mermelstein, 1980; Mermelstein, 1976) are a commonly used feature type (Hasan et al., 2004; Tirumala et al., 2017; Morrison et al., 2020). We shortly go over how to compute MFCCs, but for more detail, see Abdul and Al-Talabani (2022) or Morrison et al. (2020). After preprocessing, *framing* is applied. A continuous signal is divided up into several frames of a short duration (e.g., 20 milliseconds), which partly overlap, to allow for a more stationary signal. Subsequently, *windowing* is applied to the frames, to narrow the signal towards the frame's borders. Next, the power spectrum is computed using a Discrete Fourier Transform (DFT). A Mel[1] filter bank is applied to the power spectrum. Finally, a Discrete Cosine Transform (DCT) is applied to the logarithm of the Mel filter bank. The first few DCT coefficients are used as MFCCs. Derivatives and double derivatives of the MFCCs can also be included as features.

Instead of using MFCCs, some speaker verification systems use features from earlier steps in the process. The output of the DFT is called a spectogram, and the logarithm of the Mel filter bank is referred to as F-bank. Both are used as features for speaker verification (Ohi et al., 2021). Figure 3.2 shows the feature extraction process. By using feature extraction, model inputs are more similar to how the human auditory system works than raw waveforms (Tirumala et al., 2017). Furthermore, frequency-based representations such as F-bank and MFCCs are more robust to noise than waveforms or other time-based features (Mehrish et al., 2023).

Deep neural network. Features extracted from an utterance are fed into a deep neural network, which produces a speaker embedding. We briefly introduce the main architecture

[1]The Mel scale is a scale where pitches are perceived by humans to be at an equal distance to each other. It has a logarithmic relation to the frequency scale (Makhoul and Cosell, 1976).

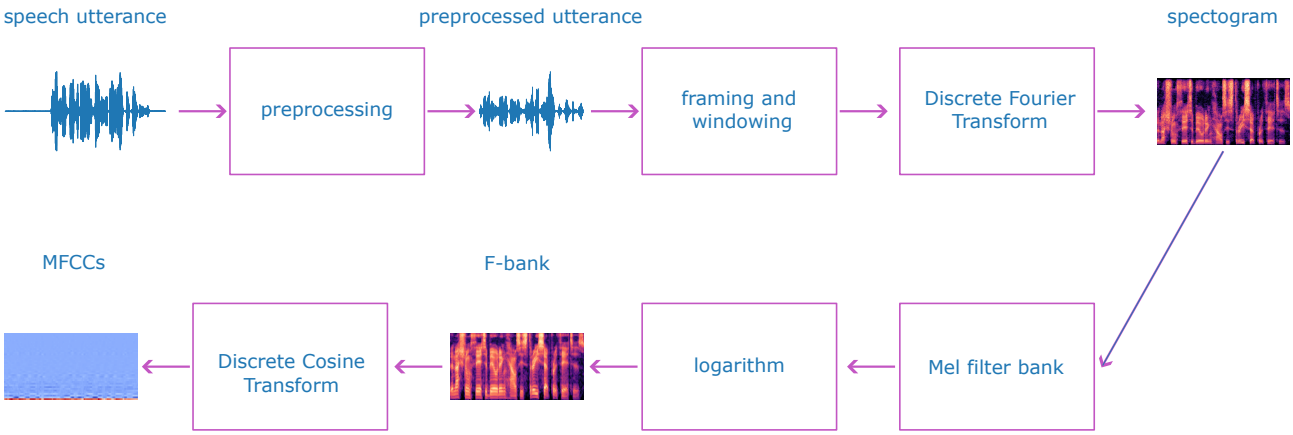


Figure 3.2: **Preprocessing and feature extraction.** Spectograms, F-banks and MFFCs are used in different speaker verification systems.

types that are used in automatic speaker verification. For more details on different models, see Section 4.3. The first speaker embedding based on deep learning that exceeded the previous state of the art was x-vector[2], based on a Time-Delay Neural Network (TDNN) (Waibel et al., 1989) (Wang et al., 2024b). ECAPA-TDNN (Desplanques et al., 2020), one of the most successful speaker verification models in recent years (Brydinskyi et al., 2024; Guo et al., 2024; Wang et al., 2024a) is an improvement of x-vector. Additionally, ResNet (He et al., 2016), a model introduced for image recognition that introduces residual learning to convolutional networks, has also been applied to speaker recognition (Villalba et al., 2020; Wang et al., 2024b; Zeinali et al., 2019). Finally, the Transformer (Vaswani et al., 2017), introduced in natural language processing and emphasizing attention, is another architecture commonly used in speaker verification systems (Sang et al., 2023; Wang et al., 2024b).

Regardless of the model architecture, the neural network takes features for each frame of an utterance as input. To obtain a speaker embedding from frame representations, some kind of pooling must be used to aggregate activations from different frames. Many different temporal pooling mechanisms have been proposed, some using statistics such as mean and standard deviation, and others using a learned attention mechanism (Bai and Zhang, 2021; Jakubec et al., 2024; Wang et al., 2024b).

An important aspect of a neural network architecture is its loss function. Most speaker verification systems use a classification-based loss, being trained on the task of speaker identification rather than speaker verification (Bai and Zhang, 2021). Usually this loss is applied on the segment level, after temporal pooling, although d-vector (Variani et al., 2014), one of the earliest automatic speaker verification systems using deep learning, calculates the loss per frame (Jakubec et al., 2024; Wang et al., 2024b). The exact choice of loss function has an effect on both accuracy and training time (Jakubec et al., 2024). In general, variations on Softmax are used, with different variants encouraging decreasing within-speaker variance or working better with training sets with a great number of classes (Bai et al., 2022; Jakubec et al., 2024). It has been shown that using a Softmax variant that imposes a margin between classes can greatly improve performance over regular Softmax (Xiang et al., 2019).

Similarity measure. Speaker embeddings obtained from the neural network are compared using a similarity measure. Most commonly, cosine similarity is used (Wang et al., 2024b). Alternatively, Probabilistic Linear Discriminant Analysis (PLDA) (Ioffe, 2006; Prince and Elder,

[2]Note that the term 'x-vector' is sometimes used to refer to deep speaker embeddings generally (Gu et al., 2023; Morrison et al., 2020; Pappagari et al., 2020). We use it exclusively to refer to the model architecture introduced by Snyder et al. (2017).

2007) is applied. PLDA is a trainable dimensionality reduction technique that is used to emphasize speaker-discriminative information and discard dimensions containing speaker-independent information, before computing a similarity score (Jakubec et al., 2024). While these are the main similarity measures, other options have been explored, including neural back-ends (Zeng et al., 2024). It has been shown theoretically that PLDA achieves better results than cosine similarity in most situations, and that the latter can be regarded as an approximation that is computionally less expensive to compute (Wang, 2020). In forensic applications of speaker verification, PLDA is preferred above cosine similarity. Before reporting forensic results, the PLDA score should be calibrated, to be interpretable as a likelihood ratio (Morrison et al., 2020).

3.2 Robustness of speaker verification systems

Ideally, speaker embeddings used in speaker verification should have large inter-speaker variability and small intra-speaker variability. To achieve this, systems must be robust against irrelevant variability (Hansen and Hasan, 2015; Wang et al., 2024b). Different types of techniques have been employed to contribute towards this goal, at different parts of the pipeline. Because these techniques ultimately have the same goal as we have with disentanglement, that is, allowing automatic speaker verification systems to better focus on speaker and ignore other aspects of the signal, we briefly review them here.

Signal cleaning and data augmentation. Speech is often distorted by noise and reverberation. It is possible to preprocess utterances to obtain a cleaner signal, to decrease the effects of these distortions on speaker verification (Bai and Zhang, 2021). Deep learning techniques can be used to enhance speech quality, separate speech from noise and/or dereverberate a recording (Wang and Chen, 2018).

Rather than removing noise and reverb from input utterances, another option is to augment the training set with noise and reverb, to obtain speaker embeddings that are more robust to these distortions (Bai and Zhang, 2021). In speaker verification, two datasets are commonly used for this: MUSAN (Snyder et al., 2015) contains music, speech and noise samples, and RIR (Ko et al., 2017) contains room impulse responses with which realistic reverb can be added to a signal (Desplanques et al., 2020; Sang et al., 2023; Snyder et al., 2015; Zhang et al., 2021).

Features. Features extracted from an utterance can be modified by domain mismatch compensation techniques (Hansen and Hasan, 2015; Morrison et al., 2020). For example, *cepstral-mean substraction* (Atal, 1974) calculates the average values of each feature, and substracts those means from the frame features. The idea behind this technique is that channel effects originating for example from the microphone are invariant over time (as the same microphone is used for the entire utterance) (Morrison et al., 2020). *Feature warping* (Pelecanos and Sridharan, 2001), a more complex mismatch compensation technique, instead assumes that channel effects can slowly change throughout the recording (Morrison et al., 2020).

Deep neural network. At the stage of the deep neural network that extracts the speaker embedding, different network architectures, pooling strategies or loss functions might be employed to improve robustness (Bai and Zhang, 2021; Wang et al., 2024b). A popular technique is adversarial training, with the domain-adversarial network (DANN) (Ganin et al., 2016) in particular being used a lot in speaker verification (Bai and Zhang, 2021).

Similarity measure. Before employing a similarity measure, linear discriminant analysis (LDA) (Fisher, 1936, 1940) might be applied, which reduces the dimensionality of speaker em-

beddings and emphasizes inter-speaker variance (Kabir et al., 2021; Kelly et al., 2019; Morrison et al., 2020). LDA can be used in combination with any similarity measure.

At the back-end, PLDA can be used to emphasize speaker-related information (Jakubec et al., 2024). PLDA has been extended to improve calibration on datasets that have different properties than the training data (Ferrer et al., 2022). In another line of work, uncertainty estimates have been added to PLDA (Wang et al., 2023a) and cosine similarity (Wang and Lee, 2024).

Chapter 4

Methods

This chapter introduces the methods used in our experiments. In Section 4.1, we introduce minimum description length probing, our main interpretability method. We also explored the use of a layer similarity analysis technique, see Appendix A.1. In Section 4.2, we explain our disentanglement approach. Finally, in Sections 4.3 and 4.4, we introduce the investigated models and the used datasets, respectively.

4.1 Minimum description length probing

In this work, we use minimum description length probing to investigate the information represented by speaker embeddings. This section introduces the general probing approach, describes the specifics of minimum description length probing, and explains the design choices of probe structure and complexity, included baselines and probe input.

General probing approach. Probing can be used to investigate specific attributes represented by a model, using a classification task designed to require these attributes being embedded (Belinkov and Glass, 2019). This approach was applied in natural language processing papers by Köhn (2015) and Gupta et al. (2015), before being formalised as *auxiliary prediction tasks* by Adi et al. (2017) and as *diagnostic classifiers* by Hupkes et al. (2018)[1]. Conneau et al. (2018) introduced the now standard term *probing*. By probing speaker verification models we can look specifically into the extent to which speaker, content and channel information are encouraged or suppressed in speaker embeddings.

A probing task requires a classification dataset representing the attribute of interest. For example, to probe representation of content, we use a dataset $\mathbf{C} = \{\mathbf{X}, \mathbf{Y}\}$ with $\mathbf{X}$ a set of recordings of sentences uttered by different speakers, and $\mathbf{Y}$ the identifiers of the sentences spoken, such that y_i is the identifier of the sentence uttered in $\mathbf{x}_i$ for each i. We pass $\mathbf{X}$ through our speaker verification system of interest, and extract the utterance representations $\mathbf{h}_i$ for each $\mathbf{x}_i \in \mathbf{X}$. When we probe a model layer that uses frame-based representations, average pooling is used to get the utterance representation. This results in the probing dataset $\mathbf{U} = \{\mathbf{H}, \mathbf{Y}\}$, where each $\mathbf{h}_i \in \mathbf{H}$ is the representation of $\mathbf{x}_i$, and each $y_i \in \mathbf{Y}$ its label. This dataset is used to train and evaluate a classification model called the *probe*. Performance of this probe on the dataset, usually evaluated by classification accuracy, is considered to be indicative of how well the attribute of interest is represented in the probed representations. Figure 4.1 illustrates the probing approach.

Minimum description length probing. Various improvements to the general probing approach have been proposed, in response to critique with regards to its robustness and

[1]Both of these papers had preprint or workshop versions published in 2016.

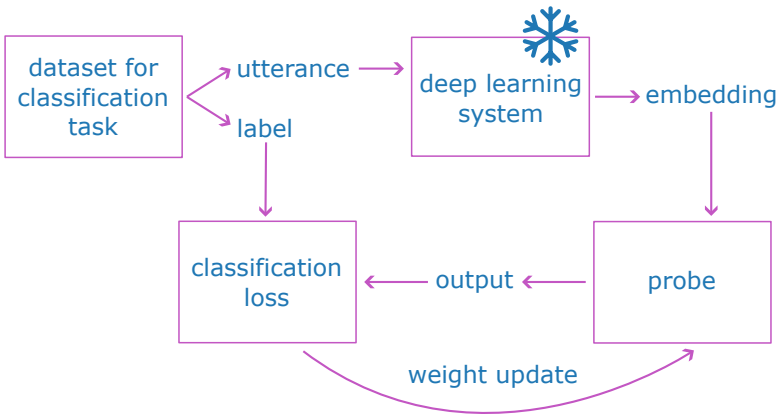


Figure 4.1: **Probing approach.** Inputs from a classification dataset for a given attribute are passed through a frozen model of interest, and representations from the model are used to train a simple classifier, called *probe*. Performance of the probe is taken to be indicative of how well the attribute of interest is represented in the probed representations.

interpretability (Belinkov, 2022). In our probing experiments, we use minimum description length probing, as proposed by Voita and Titov (2020). Minimum description length (MDL), or codelength, is a concept from information theory that refers to the amount of bits required to transmit data labels when the inputs are known. Voita and Titov (2020) compare MDL probing against standard probing that uses accuracy as its evaluation metric, and show that minimum description length probing is more robust against variation in probe architecture, dataset size and random seeds, as well as distinguishing better between trained models and randomly intialised baselines. Besides taking into account final probe performance, codelength also reflects the amount of data required to learn the task. MDL probing has been used in recently published works (Aghazadeh et al., 2022; Fierro et al., 2024; Waldis et al., 2024), but to our knowledge, it has not been applied to speaker verification.

Voita and Titov (2020) propose two ways to compute codelength: *variational code* and *online code*, and show that both agree in results. Variational code requires using a Bayesian probe and a variational loss, whereas online code can be used with standard probe architectures and objectives. Hence, we implement online code. In the information-theoretic intuition behind online code, person A and person B both have a set of input points $\mathbf{H}$. Person A also has the corresponding labels $\mathbf{Y}$, and wants to communicate them to person B while transmitting as little information as possible, which can be done by compressing the data. In this case, a probe is used to compress the data. Person A and person B agree on a probe architecture, hyperparameters and seeds. Then, person A sends a small subset of the data labels to person B and both train a probe on these labels (the trained probes will be identical due to the previous agreements). Person A uses the probe to compress a larger subset of the untransmitted labels, and transmit it to person B. Both train a probe on all the data transmitted so far, and the process is repeated, until all the labels have been communicated. Figure 4.2 shows a schematic diagram of the process.

Online code is implemented as follows. Let $\mathbf{U} = \{\mathbf{H}, \mathbf{Y}\}$ be a probe training dataset and $\mathbf{E} = \{\mathbf{H}', \mathbf{Y}'\}$ an evaluation dataset, constructed from distinct subsets of a classification dataset. Out of the total dataset $\mathbf{U}$, n increasingly large subsets $\mathbf{T}_i = \{\mathbf{H}_i, \mathbf{Y}_i\}$ are taken, where $\mathbf{T}_i \subseteq \mathbf{T}_{i+1}$. Following Voita and Titov (2020), $n = 11$, $\mathbf{T}_1$ constitutes 0.1% of the dataset, and $\mathbf{T}_{11}$ is the full training set, with each $\mathbf{T}_i$ containing roughly twice as many training examples as $\mathbf{T}_{i-1}$. Then, on each $\mathbf{T}_i$ for $i < n$, a probe $p_{\theta_i}(y|\mathbf{h})$ is trained until convergence, measured by evaluating the probe on $\mathbf{E}$ after each epoch, and stopping when performance does not improve further. Then, the probe is evaluated on $\mathbf{T}_{i+1} \setminus \mathbf{T}_i$, i.e., on examples in $\mathbf{T}_{i+1}$ not present in $\mathbf{T}_i$. The total codelength is now given by the sum of the cross entropy losses of each probe on its

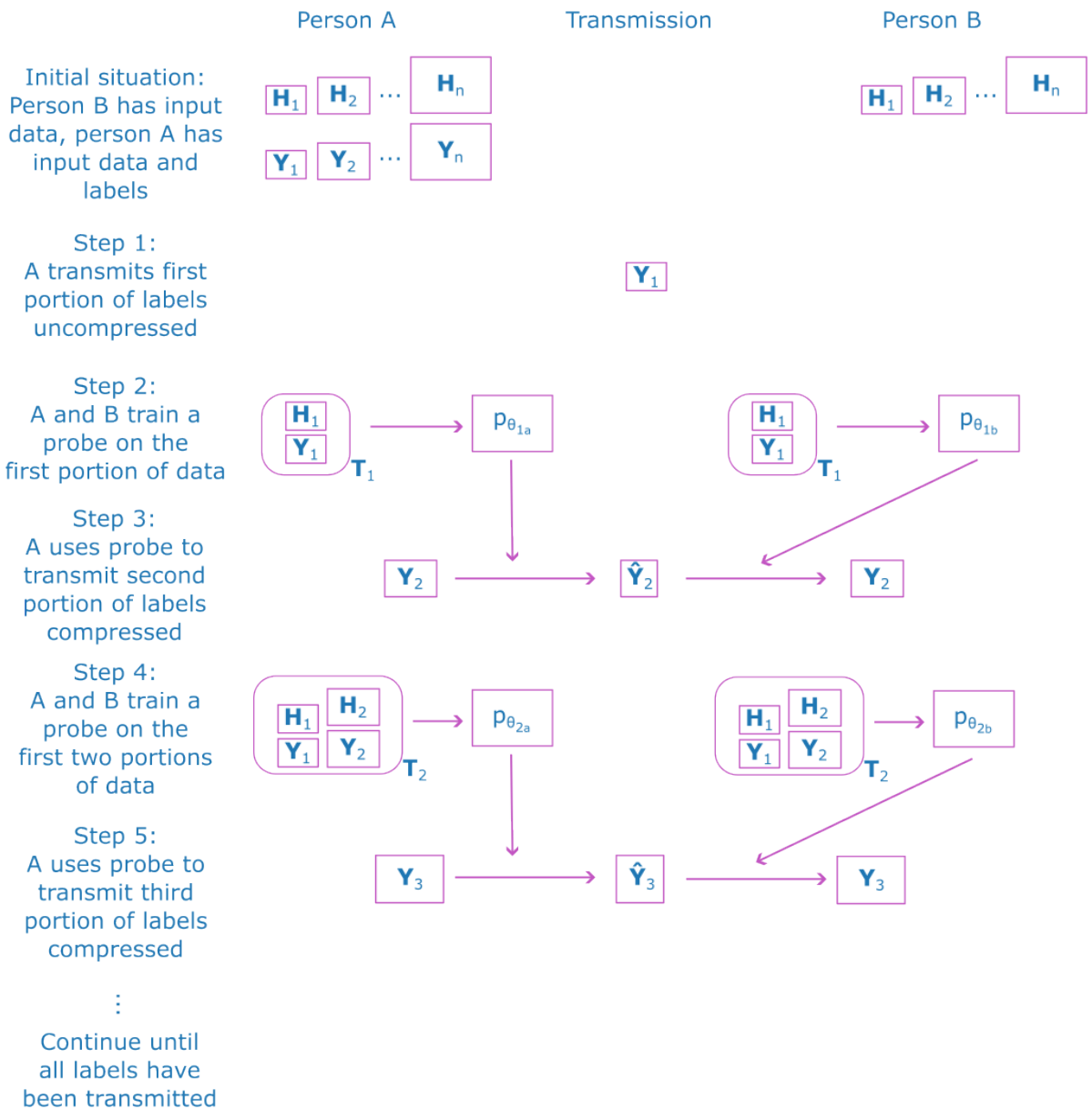


Figure 4.2: **Online code.** A probe is used to compress information.

evaluation set[2], plus the uniform codelength of $\mathbf{T}_1$—that is, the codelength of transmitting the data without using a probabilistic model. The total codelength reflects how well the probe can learn the task, as well as how much data is required for it.

Codelength inherently depends on the amount of training examples in a dataset. Thus, we opt to report *compression*, which is obtained by dividing the uniform codelength of $\mathbf{U}$ by its codelength. For a classification task with K classes, compression for a probe training set $\mathbf{U} = \{\mathbf{H}, \mathbf{Y}\}$ can be calculated as[3]

$$compression(\mathbf{U}) = \frac{n\_ex(\mathbf{U}) \times \log K}{n\_ex(\mathbf{T}_1) \times \log K - \sum_{i=1}^{n-1} \sum_{(\mathbf{h}_j, y_j) \in \mathbf{T}_{i+1} \setminus \mathbf{T}_i} \log p_{\theta_i}(y_j | \mathbf{h}_j)}, \tag{4.1}$$

with $n\_ex(\mathbf{D})$ representing the amount of examples in some dataset $\mathbf{D}$. As a sanity check, we also calculate accuracy of the final trained probe on the evaluation set. Algorithm 1 shows pseudocode of the full probing approach.

Probe structure and complexity. It is a matter of debate what the structure of a probe should look like, and how complex it should be. One of the studies first proposing probing recommends using linear classifiers (Hupkes et al., 2018). If a probe is too complex, its performance might not reflect what is encoded in the representations trained on, but rather learn the task itself (Hewitt and Liang, 2019). However, Pimentel et al. (2020) argue that there is no difference between the probe learning the task and the representations encoding the structure, and claim that more complex probes are better.

By using minimum description length probing and reporting codelength compression, the decision of probe structure and complexity is less important, as codelength not only reflects final performance of a trained probe on some task but also how easily the task is learned (Voita and Titov, 2020). Further, Voita and Titov (2020) show that codelength compression is stable across different probe hyperparameter settings.

In most previous research utilising probing on speaker embeddings, a multi-layer perceptron with a single hidden layer is used (Chiu et al., 2025; Raj et al., 2019; Wang et al., 2017), which we adopt in this research. Following Raj et al. (2019), we use a hidden layer size of 500 and ReLU activations between the layers.

Controls and baselines. When designing a probing task, it is important to use baselines to aid in interpreting the results (Belinkov, 2022). Despite this, the lion's share of research investigating speaker embeddings using probing (Chiu et al., 2025; Peri et al., 2020a; Raj et al., 2019; Wang et al., 2017; Zhao et al., 2022) does not provide any baselines or controls, except for comparing against chance level. We use different baselines to provide context necessary to determine whether performance should be considered high or low.

To get an upper bound, or skyline, on how well content and channel are represented in speaker verification models, we compare against probes trained on embeddings from speech models trained on a general pre-training objective, without finetuning for speaker verification. The assumption is that finetuning on speaker verification should not increase representation of content and channel.

To get a lower bound, we use two baselines. For the first, we train on embeddings from speaker verification models that have had their weights randomised, assuming that with randomised weights, the models should not increase or suppress content and channel in any particular way. This baseline was recommended by Chrupała et al. (2020), after Zhang and Bowman

[2]Note that this 'evaluation set' is actually a subset of the training set $\mathbf{U}$, and the actual evaluation set $\mathbf{E}$ is only used to determine when to stop training the probe.

[3]As compression is a ratio, the base of the logarithm does not affect the results. However, the logarithm base in the uniform codelength should be the same as that used in calculation of the cross entropy loss.

Algorithm 1: **Minimum description length probing algorithm.** For simplicity, we leave out probing different model layers. For some dataset $\mathbf{D}$, we use $\mathbf{D}^{0:9}$ to represent the first 10 data points of $\mathbf{D}$. We use $n\_ex(\mathbf{D})$ to represent the amount of examples in $\mathbf{D}$.

Input: A model M, a training dataset $\mathbf{A} = \{\mathbf{X}, \mathbf{Y}\}$ and an evaluation dataset $\mathbf{B} = \{\mathbf{X}', \mathbf{Y}'\}$, both with K classes.

Output: Compression and accuracy of a probe trained on embeddings of $\mathbf{A}$ obtained from M.

▷ Obtain embeddings from model

$\mathbf{H} \leftarrow M(\mathbf{X})$

$\mathbf{H}' \leftarrow M(\mathbf{X}')$

$\mathbf{U} \leftarrow \{\mathbf{H}, \mathbf{Y}\}$

$\mathbf{E} \leftarrow \{\mathbf{H}', \mathbf{Y}'\}$

▷ Create data subsets

$l \leftarrow n\_ex(\mathbf{U})$

$P \leftarrow \{0.001, 0.002, 0.004, 0.008, 0.016, 0.032, 0.0625, 0.125, 0.25, 0.5, 1\}$

for $i = 1, \ldots, 11$ **do**

    $\mathbf{T}_i \leftarrow \mathbf{U}^{0:P_i \times l}$

    if $i < 11$ **then**

        $\mathbf{V}_i \leftarrow \mathbf{U}^{P_i \times l : P_{i+1} \times l}$

    end if

end for

▷ Train and evaluate probes

codelength $\leftarrow 0$

for $i = 1, \ldots, 11$ **do**

    Initialise probe p

    repeat

        Train p for one epoch

        Evaluate loss of p on $\mathbf{E}$

    until evaluation loss does not improve anymore

    if $i < 11$ **then**

        codelength $\leftarrow$ *codelength* + **Evaluate** loss of p on $\mathbf{V}_i$

    else

        accuracy $\leftarrow$ **Evaluate** accuracy of p on $\mathbf{E}$

    end if

end for

▷ Return results

codelength $\leftarrow$ *codelength* $+ n\_ex(\mathbf{V}_1) \times \log_2 K$

uniform_codelength $\leftarrow n\_ex(\mathbf{U}) \times \log_2 K$

compression $\leftarrow$ *uniform_codelength* / *codelength*

return *compression*, *accuracy*

(2018) found that with enough training data, a probe can learn tasks using embeddings from an untrained baseline. Second, we train a probe on the preprocessed audio features directly, without putting them through a model, as previously done in Ashihara et al. (2024). This allows us to see what is easily extractable from the input features, before a complex neural network emphasises some aspects of the signal and suppresses others.

Probe input. When we probe speaker embeddings, we can simply use the speaker embedding as the input to the probe. However, in other experiments we probe the final layer of a general pretrained speech model, or we probe an earlier layer of a speech or speaker model. In these cases, we may get representations per frame rather than per utterance. To be comparable to the experiments on speaker embeddings, we construct an utterance representation using mean pooling. Monteiro et al. (2020) showed that the pooling strategy chosen to go from frame representations to utterance representations can have a large effect on performance (in their study, this was shown for language identification). However, using one of the pooling strategies investigated in Monteiro et al. (2020) would require learning pooling parameters while training the probe, which is infeasible in our setup due to constraints on memory and computation.

4.2 Disentanglement using a two-stage information bottleneck approach

For disentanglement, we adapt the approach proposed by Mohebbi et al. (2024), which uses a Variational Information Bottleneck to learn disentangled encodings on top of representations from frozen models. Their two-stage approach aims to improve disentanglement of the attribute trained for in stage 1 from the attribute trained on in stage 2. We opt for this supervised approach because it has been shown that unsupervised approaches do not reliably lead to disentangled representations (Locatello et al., 2019). Furthermore, it allows improving disentanglement of speaker embeddings without training or finetuning any speaker verification models, and is architecture-agnostic. Because the code from Mohebbi et al. (2024) is publicly available, we can reliably reproduce the proposed approach. This section introduces the Variational Information Bottleneck and outlines the used two-stage approach. It also makes explicit where our approach differs from the one originally proposed, explains how we evaluate and analyse the experiments' results and mentions what controls we use.

Variational Information Bottleneck. We use a Variational Information Bottleneck (VIB) to encourage representation of a target attribute, and discourage representation of irrelevant information. The VIB was introduced by Alemi et al. (2017), and computes a lower bound on the Information Bottleneck (IB) as introduced by Tishby et al. (2000). When we have a representation h (e.g., a speaker embedding) and its target variable y (e.g., a speaker id), the goal of the IB is to find an encoding z of x for which z is maximally expressive about y but minimally expressive about x. More formally, the objective is

$$I(z, y) - \beta I(z, x),$$

where $I(a, b) = \mathop{\mathbb{E}}_{p(a,b)} \log \frac{p(a,b)}{p(a)p(b)}$ is the mutual information of a and b, and $\beta > 0$ is a constant or variable which determines the tradeoff between keeping more information about y in z and less information about x in z. The IB is computationally difficult, and the VIB uses the reparameterisation trick (Kingma and Welling, 2014) to make it tractable.

The VIB uses an encoder $p_\theta(z|h)$ and a decoder $p_\phi(y|z)$, modeled by neural networks. We parameterise the decoder as $p_\theta(z|h) = \mathcal{N}(z|\mu_\theta(h), \Sigma_\theta(h))$, learning a distribution that

stochastically maps an input h to a latent representation z. The mutual information between the encoding z and the output y is lower-bounded by

$$I(z, y) \geq \mathop{\mathbb{E}}_{z \sim p_\theta(z|h)} \log p_\phi(y|z),$$

which is the cross entropy loss. The mutual information between the encoding z and the input h is upper-bounded by

$$I(z|h) \leq D_{\mathrm{KL}}(\mathcal{N}(z|\mu_\theta(h), \Sigma_\theta(h)) \;||\; \mathcal{N}(\mathbf{0}, \mathbf{I})),$$

with $\mathbf{I}$ the identity matrix. The intuition behind the KL divergence term is as follows: the more similar $p_\theta(z|h)$ is to the standard normal distribution, the less informative z is of h. In the remainder of this section we use the standard IB objective to illustrate the disentanglement framework, as it is easier to understand than the VIB.

The (V)IB has been used in other disentanglement frameworks. Wang et al. (2023b) use it to develop an attribution method, generating better explanations for vision-language models. Information Bottlenecks are also used in neural speech synthesis frameworks, to disentangle acoustic features that can then be controlled separately (Mehrish et al., 2023). Qian et al. (2019) disentangle speaker-dependent information from speaker-independent information, and Qian et al. (2020) disentangle speech into content, timbre, pitch and rhythm, with the goal of style transfer for voice conversion in both studies. In speaker representation disentanglement, Li et al. (2023a) and Mun et al. (2022) both use Contrastive Log-ratio Upper Bound (CLUB) (Cheng et al., 2020), which computes an upper bound on mutual information, and can also be applied in an Information Bottleneck.

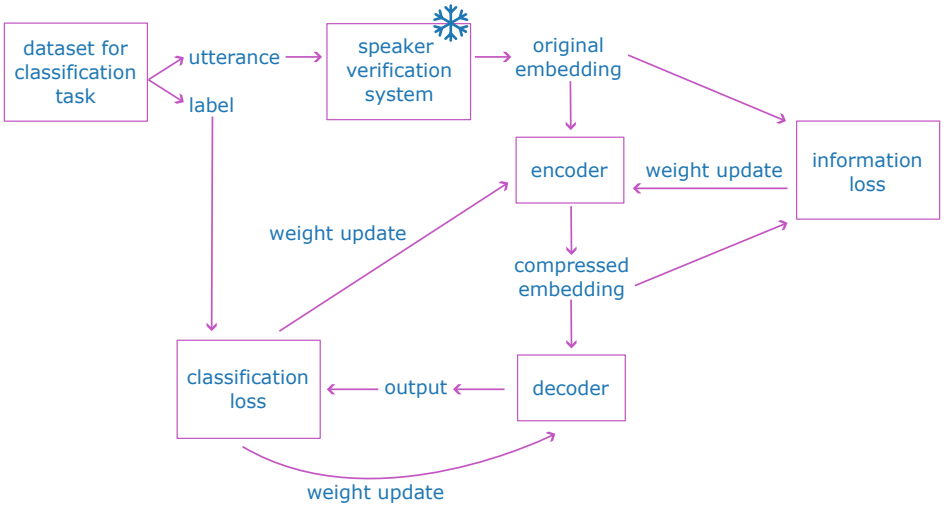
Two-stage disentanglement framework. Similar to probing (see Section 4.1), the disentanglement framework adapted from Mohebbi et al. (2024) trains new models on top of embeddings from frozen speaker verification models. The main difference with probing is that the VIBs trained have an encoder-decoder structure, where the encoder learns a compressed representation of the input embedding, and the decoder performs classification, rather than using a single classifier network. The goal of this approach is to train an encoder that can be used on top of speaker embeddings to obtain embeddings that are more disentangled than the original ones. The framework uses two stages, which are illustrated in Figure 4.3 and further explained below.

In stage 1, a classification dataset representing an attribute of interest is used to obtain representations from a speaker verification system. A VIB is trained on these representations. Because the VIB maximizes $I(Z, Y)$ while minimizing $I(Z, X)$, it is encouraged to retain only the information relevant for the target task in the generated embeddings. In our experiments, for each investigated model, we train two VIBs in stage 1: one on content classification, and one on channel classification.

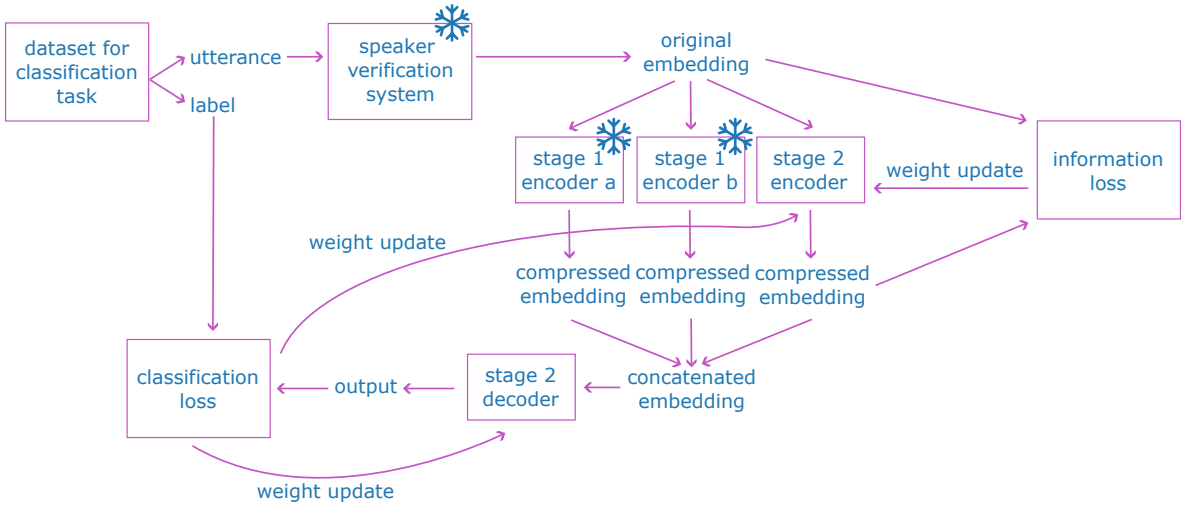
Stage 2 is similar to stage 1, and again we train a VIB on representations from a speaker verification system, to perform a classification task. However, in stage 2 the embedding from the trained encoder is concatenated with embeddings from the (frozen) encoders trained in stage 1, before being fed to the decoder. This way, the decoder can use information extracted by the previously trained encoders. This changes the objective to

$$I(Z', Y) - \beta I(Z, X),$$

with Z' the concatenation of the embeddings from the previously trained encoders and the embedding from the newly trained encoder. Thus, the decoder can use the information extracted by the stage 1 encoders 'for free', as they do not contribute to $I(Z, X)$. This encourages the stage 2 VIB to suppress any information that is also present in the embeddings from the stage



(a) **Disentanglement stage 1.** An encoder-decoder VIB model is trained to generate a compressed embedding that is minimally expressive about the input embeddings (by the information loss) while being maximally expressive about the target label (by the classification loss).



(b) **Disentanglement stage 2.** A VIB is trained similarly to stage 1, but the decoder receives a concatenated embedding from its encoder and frozen encoders from stage 1, encouraging the stage 2 encoder to discard any information that is also extracted by the stage 1 encoders.

Figure 4.3: **Disentanglement framework.** A two-stage approach is used to improve disentanglement, adapted from Mohebbi et al. (2024).

1 VIBs, improving disentanglement. In our experiments, in stage 2, we train a VIB on speaker classification, concatenating embeddings from the frozen content and channel VIBs from stage 1. Note that this concatenation is only done while training the stage 2 VIB, and in evaluation only the embeddings from its encoder are used.

Differences from original approach. In adapting the approach from Mohebbi et al. (2024), we make the following changes. First, in the original research, a weighted layer average of frame representations is taken as an input to the encoders. As we want our approach to work on speaker embeddings from models of which hidden representations might not be accessible, we take just the speaker embedding as input instead. Second, because the speaker embedding is utterance-based, we do not perform time-pooling, as is done by Mohebbi et al. (2024) in stage 2. Third, in both stages, we use an utterance-based task, whereas Mohebbi et al. (2024) use a frame-based transcription task in stage 1. Finally, stage 2 takes two stage 1 embeddings in the concatenation instead of one.

Evaluation and analysis. To evaluate the trained encoders, we use minimum description length probing, comparing performance of the compressed embeddings and the original embeddings. We also evaluate speaker verification performance on VoxCeleb1 (see Section 4.4), reporting Equal Error Rate (EER), the standard metric in speaker verification research (Hansen and Hasan, 2015; Wang, 2020). For a set of output scores and corresponding target labels, the EER represents the error rate at the threshold where the false positive rate is (roughly) equal to the false negative rate (Kabir et al., 2021).

In analysing the trained VIBs' performance, it can be useful to see what embeddings contribute most to the decoder's output in stage 2. Thus we visualize the decoder's weights in one of our experiments. It is also insightful to see what encoded examples look like, and we visualize samples of the encoders' encodings in another experiment.

4.3 Investigated models

We perform experiments on six speaker verification models, using trained checkpoints obtained from HuggingFace (Wolf et al., 2020). We also include two models trained with a general speech prediction objective.

SpeechBrain models. We use three models from the SpeechBrain toolkit (Ravanelli et al., 2021). These models were trained on VoxCeleb1 (Nagrani et al., 2017) and VoxCeleb2 (Nagrani et al., 2020b), two datasets containing clips from celebrities (see also Section 4.4). As the backend, cosine distance is used. The models use F-bank features with a frame-length of 25 milliseconds at a 10 millisecond shift. To be able to probe different model layers, we create a fork of the SpeechBrain toolkit[4] that allows returning hidden representations for the models investigated. The following models were used:

- x-vector[5] (Snyder et al., 2017, 2018) was one of the first successful deep neural networks for speaker verification. It is a Time Delay Neural Network (Waibel et al., 1989) that consists of feed-forward layers operating on speech frames, a statistics pooling layer that aggregates frame-level representations into a segment-level representation, and feed-forward layers that operate on the segment-level representation. x-vector uses the standard Softmax loss. As implemented in SpeechBrain, x-vector uses 24-dimensional F-bank features.

- ResNet[6] (He et al., 2016) is a model introduced for image recognition, that has also been applied to speaker recognition (Villalba et al., 2020; Wang et al., 2024b; Zeinali et al., 2019). It is a convolutional neural network (Krizhevsky et al., 2012) that includes residual connections. As implemented in SpeechBrain, ResNet uses Additive Angular Margin Softmax (Deng et al., 2019; Xiang et al., 2019) as its loss function, and 80-dimensional F-bank features as input features.

- ECAPA-TDNN[7] (Desplanques et al., 2020) is an evolution of x-vector. Previous improvements had incorporated attention mechanisms (Okabe et al., 2018), which Desplanques et al. (2020) further improve. Secondly, Squeeze-Excitation blocks (Hu et al., 2020) are added, allowing to model global channel interdependencies. Finally, whereas in x-vector, each layer only gets the input from the previous layer, ECAPA-TDNN aggregates feature maps from all preceding layers. ECAPA-TDNN, like ResNet, uses Additive Angular Margin Softmax, and 80-dimensional F-bank input features, as implemented in SpeechBrain.

[4] https://github.com/607GitHub/speechbrain
[5] https://huggingface.co/speechbrain/spkrec-xvect-voxceleb
[6] https://huggingface.co/speechbrain/spkrec-resnet-voxceleb
[7] https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

UniSpeech models. We use two models from the UniSpeech (Wang et al., 2021b) family. Both model architectures use self-supervised learning to learn speech processing. The models were finetuned for various tasks, among which speaker verification. Both architectures are based on HuBERT (Hsu et al., 2021). HuBERT uses clustering to obtain targets for self-supervision, and masks hidden units. It uses a convolutional feature encoder, which uses 39-dimensional MFCCs (13 coefficients with derivatives and double derivatives) initially, but in later stages uses latent features learnt by an earlier model checkpoint. The output of the feature encoder serves as input to a Transformer (Vaswani et al., 2017) encoder. The models were trained on LibriLight (Kahn et al., 2020), GigaSpeech (Chen et al., 2021) and VoxPopuli (Wang et al., 2021a). All of these datasets were made to improve automatic speech recognition, and contain mostly unlabeled data. To finetune on speaker verification, VoxCeleb1 was used, and Additive Margin Softmax (Wang et al., 2018; Xiang et al., 2019) as the loss function. For the UniSpeech models, we investigate both the general pretrained models and models finetuned for speaker verification. The following models were used:

- WavLM[8] (Chen et al., 2022a) masks speech segments with noise and other utterances during pretraining, with the goal of improving performance on tasks such as speaker identification, besides automatic speech recognition.
- UniSpeech-SAT[9] (Chen et al., 2022b) ('SAT' standing for *speaker-aware pre-training*) uses an utterance-wise contrastive loss, and, similar to WavLM, sometimes mixes another utterance with the utterance under consideration, the former with the goal of improving performance on tasks such as speaker verification, and the latter to benefit tasks such as speaker diarization.

TitaNet[10]. Finally, we use TitaNet (Koluguri et al., 2022), which is part of the NeMo toolkit (Kuchaiev et al., 2019). It is based on the ContextNet (Han et al., 2020) architecture, which uses a convolutional encoder that utilises Squeeze-Excitation blocks. TitaNet adds channel attention to the decoder to obtain an utterance-level embedding. Like the SpeechBrain models, TitaNet takes 80-dimensional F-bank input features with a frame-length of 25 ms at a 10 ms shift, and uses Additive Angular Margin Softmax as the loss function. To be able to probe different model layers, we create a fork of the NeMo toolkit[11] that allows returning TitaNet's hidden representations.

4.4 Datasets

This section introduces the corpora used in our experiments, used to create classification datasets for **speaker**, **content** and **channel**. In our main interpretability experiments, we generate datasets based on corpora consisting of real recordings. In our disentanglement experiments, we generate synthetic datasets, employing corpora of real recordings in speech synthesis and augmentation. We also use a standard dataset for speaker verification evaluation.

British Isles[12]. To train a probe to predict **speaker**, we want a dataset containing a fixed set of speakers, with some other condition differing. To train a probe to predict **content**,

[8] https://huggingface.co/microsoft/wavlm-base-plus-sv and https://huggingface.co/microsoft/wavlm-base-plus-sv

[9] https://huggingface.co/microsoft/unispeech-sat-base-plus and https://huggingface.co/microsoft/unispeech-sat-base-plus-sv

[10] https://huggingface.co/nvidia/speakerverification_en_titanet_large

[11] https://github.com/607GitHub/NeMo

[12] Name coined by us, as the corpus was not released named.

we want a corpus of utterances containing a fixed set of sentences, with some other condition differing. The corpus released by Demirsahin et al. (2020), originally created to study English accents in the British Isles, offers both. This corpus features 120 speakers reading out lines from a script. The scripts include 50 sentences that are spoken by approximately every speaker. Thus we create a **speaker** probing task where the probe predicts the identifier of the speaker and a **content** probing task where it predicts the identifier of the spoken sentence, using the same data.

When constructing a dataset for speaker identification, it is crucial that channel information cannot serve as an indicator for speaker identity. In British Isles, there is little channel variation, as all recordings were made using the same equipment, and 101 out of the 120 speakers were recorded in the same room. All utterances were recorded in quiet conditions. Because the same sentences are spoken by all speakers, content cannot be an indicator for speaker, and vice versa.

VOiCES. To train a probe to predict **channel**, we want a dataset that has a controlled set of channel settings, with some other condition differing. The Voices Obscured in Complex Environmental Settings (VOiCES) corpus (Richey et al., 2018) realises this by playing back clean speech in different channel conditions. Utterances from the LibriSpeech corpus (Panayotov et al., 2015), which consists of audio books recorded by different speakers, were played back in four different rooms and with four different background noise settings, and recorded with microphones in five different positions. We only use recordings from the closest and furthest microphones, resulting in $4 \times 4 \times 2 = 32$ channel settings, which we use to create a classification task where the probe predicts the identifier of the **channel**.

VOiCES contains both speaker and content variability. As each utterance is recorded in all channel settings, neither attribute can be an indicator of channel.

SCC. For our disentanglement experiments, we construct *SCC*, a novel dataset in which **S**peaker, **C**ontent and **C**hannel all vary. By letting all three considered attributes vary for each target attribute, it becomes more important for the VIB encoders to disentangle the target attribute from the other attributes.

We generate utterances with XTTS-v2[13] (Casanova et al., 2024), a zero-shot text-to-speech model. XTTS-v2 can generate a voice based on a few seconds of speech. We use this to clone voices from a corpus containing recordings of human utterances, ensuring consistent variation in voice generation. We get our speakers from the CSTR VCTK corpus (Veaux et al., 2013; Yamagishi et al., 2019), which includes 110 native speakers of English with different accents reading lines from a script.

We select 100 speakers from VCTK to supply the voices, using the longest line common to all speakers. This line is shared with British Isles, but this should not have an effect on generalisation, as we only use the utterances to clone the voice, and do not use their content. For the content, we use the LJ Speech corpus (Ito and Johnson, 2017), which is based on the LibriVox project (Kearns, 2014). LJ Speech includes 50 read passages, and we select two lines from each. Finally, we augment generated utterances with one of 100 different channel settings: 5 noise settings (*quiet*, *pop music*, *classical music*, *chatter* and *misc*), with audio files taken from MUSAN (Snyder et al., 2015), 10 simulated reverb settings (*stairway*, *office*, *meeting*, *booth*, *aula*, *lecture*, *smallroom1*, *mediumroom1*, *largeroom1*, *largeroom2*), with room impulse reponses from RIR (Ko et al., 2017), and either no filter or a telephone-like bandpass filter.

For each target attribute, we generate a dataset using all 100 of the possible settings for the target attribute, and 15 each for the other attributes, resulting in 22500 examples per dataset. We use different datasets to train VIBs and to probe VIBs, generating 6 datasets in total. For better generalisation, the datasets have no examples in common, as can be seen in

[13] https://huggingface.co/coqui/XTTS-v2

Table 4.1: **SCC dataset setup.** 100 settings were selected for each of the attributes speaker, content and channel. Two datasets were generated for each task, each dataset containing 22500 combinations of speaker, content and channel id. No examples are shared between datasets. In practice, the class ids were shuffled.

(a) **SCC datasets for disentanglement.**

Task	Speaker ids	Content ids	Channel ids
Speaker	1-100	1-15	1-15
Content	1-15	1-100	16-30
Channel	16-30	16-30	1-100

(b) **SCC datasets for probing.**

Task	Speaker ids	Content ids	Channel ids
Speaker	1-100	31-45	31-45
Content	31-45	1-100	46-60
Channel	46-60	46-60	1-100

Table 4.1, which shows the SCC setup. We prompt XTTS-v2 once for every speaker-sentence pair within a dataset, and augment the generated signal with the different channel settings. Thus, the clean signal is identical within a speaker-content pair. In channel augmentation, we select used files stochastically. For example, the setting which augments with *pop music* and *aula* has different music playing in different examples, and room impulse responses taken from different positions in the aula. By not augmenting with the same data each time, we make the channel classification task more challenging, with the goal of obtaining a more generalised representation of channel.

VoxCeleb1. Besides using probing, we also evaluate our disentangled representations on the target task of speaker verification. We use VoxCeleb1 (Nagrani et al., 2017), an audio-visual dataset comprised of crops of fragments of videos from YouTube[14] depicting celebrities, labeled for speaker. In this work, we only use the audio component. To test a model's performance, we use the main test set of VoxCeleb1, commonly referred to as *Vox1-O*. All investigated model checkpoints have been pre-trained or finetuned on the development set of VoxCeleb1, so there should be no domain mismatch for any of them. As the test set is released separately from the development set, none of the models should have seen the instances evaluated on during training.

[14] `https://youtube.com`, a video sharing website.

Chapter 5

Experiments and results

In this chapter, we explain our experiments and interpret the results. In Section 5.1, we show final layer probing results of all investigated models, in Section 5.2, we show the results per layer, and in Section 5.3 we test generalisation to other datasets. In Section 5.4, we experiment with further disentanglement of content and channel, in Section 5.5, we use visualisations to further investigate the results, and Section 5.6 shows evaluation of our investigated and trained models on speaker verification.

5.1 Probing final layer

For every model under consideration, we use minimum description length probing for speaker, content and channel classification tasks, to investigate how well these three attributes of utterances are represented by the models. For the speaker verification models, we probe the speaker embedding, and for the general speech models, we probe the final layer, mean pooled over the frame dimension.

For reliable speaker verification, we want speaker to be well-represented in the speaker embeddings, and content and channel to be poorly represented. Earlier research has shown that x-vector represents content and channel, besides representing speaker (Peri et al., 2020a; Raj et al., 2019; Zhao et al., 2022). Zhao et al. (2022) reach the same conclusion for ResNet and ECAPA-TDNN. However, none of these studies compare their results to probing baselines. Ashihara et al. (2024) find content to be represented in ECAPA-TDNN better than in a feature baseline, but in their approach representations from all layers can be used by the probe, rather than only the speaker embedding.

Our final layer probing results, evaluated on British Isles and VOiCES as introduced in Section 4.4, can be found in Table 5.1. We use codelength compression as the metric in our experiments, see Section 4.1. For results using accuracy, see Tables A.1 and A.2. For the **speaker** classification task, all models outperform the baselines by a large margin. Moreover, as expected, we observe that the models trained or finetuned for speaker verification represent speaker more clearly than the models with a general speech prediction objective.

In the **content** classification task, all speaker verification models encode the task markedly worse than the general speech models. We note that x-vector represents content a lot better than its baselines. All other speaker verification models, however, are roughly on par with their baselines. Moreover, for most, probing performance does not exceed chance level, indicating very little representation of content. Probes trained on ECAPA-TDNN reach compression below chance level, which we think is likely caused by idiosyncracies of the dataset, and might not generalise.

Speaker verification models represent less **channel** than the general models. However, the difference is not as pronounced as was the case for content. Further, x-vector represents

Table 5.1: **Minimum description length probing results for speaker, content and channel classification tasks, on the final layers of different models**. Compression is reported, where higher compression indicates the information required for the probing task being better represented by the representations. At chance level, compression is 1. Compression is averaged over three runs, with standard deviation shown in parentheses. We include two baselines: probing randomly initialised versions of the used model architectures, and probing the input features. Due to time constraints, for TitaNet, we only use the feature baseline, as we consider it to be most informative.

(a) **Probing results on a speaker classification task.** Higher compression is better, as we want speaker verification models to clearly represent speaker.

Model	Compression	Random baseline	Feature baseline
WavLM (general)	3.09 (0.054)	1.13 (0.003)	2.27 (0.160)
WavLM (SV)	6.33 (0.158)	1.00 (0.000)	2.27 (0.160)
UniSpeech-SAT (general)	3.11 (0.111)	1.09 (0.009)	2.27 (0.160)
UniSpeech-SAT (SV)	5.59 (0.211)	1.00 (0.000)	2.27 (0.160)
ECAPA-TDNN	**11.78** (0.996)	2.54 (0.262)	2.43 (0.029)
x-vector	7.70 (0.640)	1.43 (0.069)	2.03 (0.012)
ResNet	**12.56** (0.929)	1.09 (0.013)	2.43 (0.029)
TitaNet	3.99 (0.004)	-	2.43 (0.029)

(b) **Probing results on a content classification task.** Lower compression is better, as we want speaker verification models to disregard content variation.

Model	Compression	Random baseline	Feature baseline
WavLM (general)	12.51 (3.312)	1.12 (0.003)	0.98 (0.028)
WavLM (SV)	**1.00** (0.001)	1.00 (0.000)	0.98 (0.028)
UniSpeech-SAT (general)	14.85 (0.085)	1.07 (0.002)	0.98 (0.028)
UniSpeech-SAT (SV)	**1.00** (0.000)	1.00 (0.000)	0.98 (0.028)
ECAPA-TDNN	**0.90** (0.004)	1.45 (0.038)	1.04 (0.002)
x-vector	2.78 (0.204)	1.26 (0.074)	1.04 (0.002)
ResNet	1.12 (0.003)	1.08 (0.013)	1.04 (0.002)
TitaNet	**1.01** (0.018)	-	1.04 (0.002)

(c) **Probing results on a channel classification task.** Lower compression is better, as we want speaker verification models to disregard channel variation.

Model	Compression	Random baseline	Feature baseline
WavLM (general)	3.82 (0.066)	1.43 (0.014)	1.61 (0.016)
WavLM (SV)	1.56 (0.014)	1.09 (0.004)	1.61 (0.016)
UniSpeech-SAT (general)	3.51 (0.042)	1.34 (0.006)	1.61 (0.016)
UniSpeech-SAT (SV)	1.50 (0.010)	1.11 (0.022)	1.61 (0.016)
ECAPA-TDNN	**1.34** (0.015)	2.57 (0.093)	1.90 (0.017)
x-vector	3.39 (0.062)	2.00 (0.098)	1.38 (0.007)
ResNet	1.93 (0.043)	1.32 (0.007)	1.90 (0.017)
TitaNet	2.14 (0.071)	-	1.90 (0.017)

channel a lot better than its baselines, even getting close to the performance of the general speech models. All other models are roughly on par with the baselines, except for ECAPA-TDNN. The later model represents channel less than its baselines, although still above chance level.

Overall, with the exception of x-vector, the probed speaker verification models appear to encourage speaker information without increasing representation of content and channel, with ECAPA-TDNN even suppressing content and channel compared to its baselines. We note that in most experiments, especially for the speaker and channel classification tasks, our baselines exceed chance level. For the feature baselines, this can be explained by the features having been designed to work well for speaker recognition tasks. The acoustic info captured can be extracted by a speaker verification model, and evidently also by a simple probe. For the random baselines, some of the useful parts of the input signal may pass through in an untrained model, or even be emphasized due to the design of the model architecture.

5.2 Probing per layer

For speaker verification, only the final layer (i.e., the speaker embedding) makes a difference for the system's decision. It can however be insightful to see to what degree different layers represent different attributes. Ashihara et al. (2024) find that ECAPA-TDNN encodes more content in the earlier layers, and that speaker is most clearly represented in the final layer. WavLM[1] is found to represent speaker best in the lower layers.

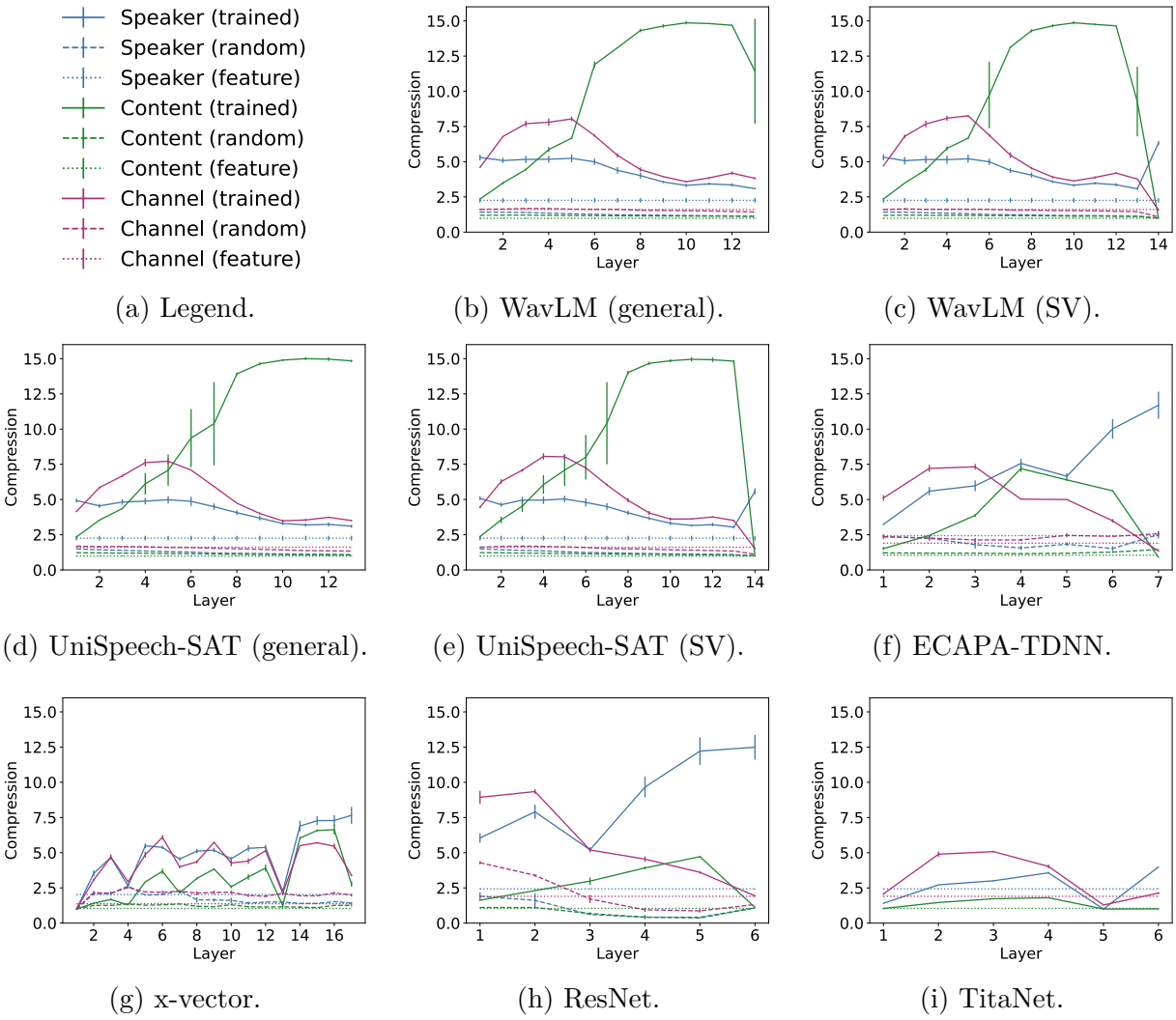
Probing results per layer can be found in Figure 5.1. Across all speaker verification models, the final layer plays a great role in suppressing content and channel information. In the earlier layers, compression for both content and channel tasks often exceeds the baselines, and only by the final layer it gets on par or below. For WavLM and UniSpeech-SAT, this is to be expected, because the models were pretrained with a general speech prediction objective, before finetuning for speaker verification. Indeed we observe that the differences between the general and finetuned models are very small, when leaving the speaker embedding layer out of the equation (see also Appendix A.1). However, the models trained with a speaker identification objective show a similar pattern, although not as extreme. TitaNet slightly deviates from the pattern, showing greatly decreased compression on content, channel *and* speaker after the pooling layer. The linear transformation to the speaker embeddings increases extractability again, focusing on speaker, although channel increases again slightly as well. While x-vector, as previously reported, stays above the baselines even in the embedding layer, we observe here too that content and channel are suppressed at the final layer compared to the layers preceding it.

5.3 Generalisation to different datasets

In our main probing experiments, we have represented each attribute by one dataset (speaker and content by British Isles, channel by VOiCES), assuming that it represents the given attribute well. To test whether results generalise to other datasets, we probe ECAPA-TDNN using SCC, which is based on different corpora. Furthermore, we probe for speaker using VCTK, and we probe for content and channel using an augmented dataset based on LJ Speech, similar to how channel amplification was done for SCC. We note that the augmented LJ Speech only includes a single speaker. Figure 5.2 shows the results.

In general, we see similar patterns to the main probing experiments, including the final layer being important for suppression of content and channel. However, when using LJ Speech

[1] In Ashihara et al. (2024) the `large` checkpoint is used, whereas we investigate the `base-plus` checkpoint.

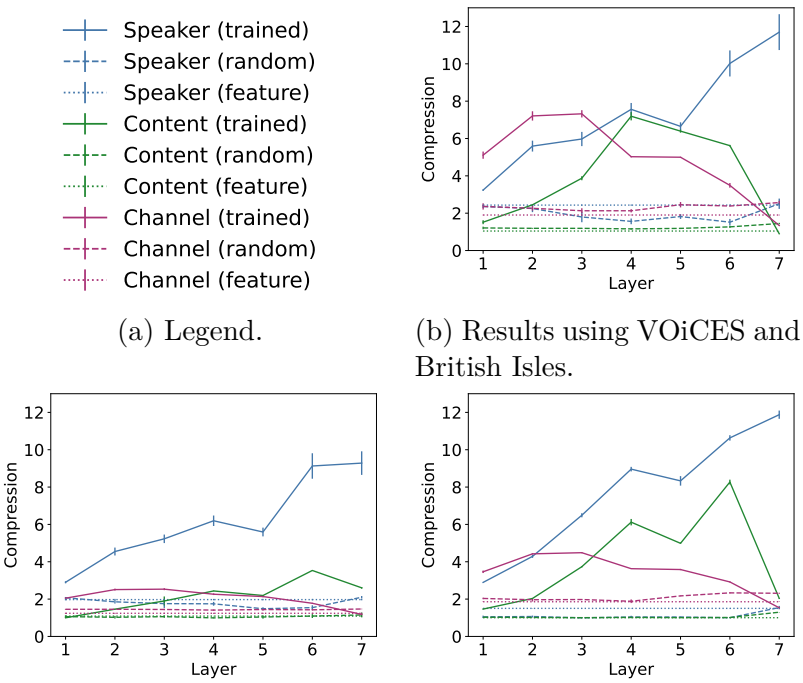


(a) Legend. (b) WavLM (general). (c) WavLM (SV).

(d) UniSpeech-SAT (general). (e) UniSpeech-SAT (SV). (f) ECAPA-TDNN.

(g) x-vector. (h) ResNet. (i) TitaNet.

Figure 5.1: **Probing results per layer.** Compression is reported on speaker, content and channel classification tasks. Higher compression indicates the information required for the probing task being better represented. For speaker verification, compression for speaker should be high while compression for content and channel should be low. As baselines, we probe a randomly initialised version of the used models, and we probe the input features. The latter is shown as a horizontal line because there is no concept of different layers.



(a) Legend.

(b) Results using VOiCES and British Isles.

(c) Results using VCTK and augmented LJ Speech.

(d) Results using SCC.

Figure 5.2: **Probing results per layer for ECAPA-TDNN.** We compare compression on different datasets.

or SCC, representation of content stays above the baselines. For both of these datasets, the best content compression is found at layer 6, whereas it is at layer 4 for British Isles. For LJ Speech in particular, it is remarkable that content representation stays above the baselines, as in general compression of content and channel is significantly lower than it is using British Isles and VOiCES. It is plausible that there is less decrease in probing performance in the final layer because of the absence of speaker variation, leaving the content and channel information more accessible in the embeddings. Although the results in Figure 5.2 only show a clear change to content representation, in earlier experiments, on a different split of the data, we also noticed channel compression staying above the baselines in the final layer. These results suggest that while representation of content and channel is suppressed by the final layers of speaker verification models, it might still have an impact on speaker verification.

5.4 Training and probing VIBs

After probing existing speaker verification models, we examine whether improved disentanglement can be achieved. As introduced in Section 4.2, we investigate the two-stage Variational Information Bottleneck (VIB) approach proposed by Mohebbi et al. (2024) as a means to further disentangle speaker embeddings. As motivated below, we focus on WavLM (general), ECAPA-TDNN and x-vector. For each, a hyperparameter search was conducted. For the hyperparameters used, see Appendix A.3.

WavLM. To validate our disentanglement method, we run experiments on WavLM, not finetuned for speaker verification. As demonstrated above, speaker embeddings are already quite disentangled compared to speech models trained with a general speech prediction objective, making it harder to evaluate the approach using speaker verification models.

We train stage 1 VIBs for content and channel encoding, and a stage 2 VIB for speaker encoding, which gets access to the embeddings from the stage 1 content and channel VIBs. As an ablation experiment, we also train a stage 1 VIB for speaker encoding, to confirm that the two-stage approach improves disentanglement. Results can be found in Table 5.2a. Our best settings for stage 1 show improved disentanglement for each attribute. When a VIB is trained on some target attribute, probing performance increases for that attribute and decreases for the other attributes, compared to probing the embeddings from the original model. The stage 2 VIB, trained for speaker encoding, also shows improved disentanglement compared to WavLM. However, disentanglement in stage 2 is inferior to disentanglement in the stage 1 VIB trained for speaker. We also evaluate our VIBs by probing representations of British Isles and VOiCES, the datasets used in our main interpretability experiments, as displayed in Table 5.2b. While some disentanglement is still shown, the performance for the target attribute is inferior to probing the original embeddings. It seems that the trained VIBs generalise poorly to other domains, compared to the original model.

ECAPA-TDNN. We repeat our disentanglement experiments on ECAPA-TDNN, which we found to be the most disentangled speaker verification model, to check if further disentanglement can be obtained. We train stage 1 VIBs for speaker, content and channel encoding, and a stage 2 VIB for speaker encoding, which gets access to the embeddings from the stage 1 content and channel VIBs. Tables 5.2c and 5.2d show the results. Our stage 1 speaker VIB shows improved disentanglement, with speaker compression increasing slightly compared to the original embeddings, and content and channel compression decreasing to chance level. However, the stage 1 VIBs trained for content and channel show decreased compression for all three attributes. Moreover, for the stage 1 content VIB, speaker compression exceeds content compression. Possibly, because content representation is already very low in ECAPA-TDNN, it is hard to train a VIB that encodes it. For the stage 2 speaker VIB, content and channel compression decrease to chance level, as for the stage 1 speaker VIB, but speaker compression does not exceed that of the original embeddings. In this case, stage 2 not improving performance is to be expected, as the stage 1 content and channel VIBs do not show successful disentanglement. When we probe the trained VIBs on British Isles and VOiCES, we again find that results do not generalise. Remarkably, we now obtain better speaker performance for the VIBs not trained on speaker than for the speaker VIBs. For content and channel representation, chance level is barely exceeded.

x-vector. Motivated by the hypothesis that content and channel are too suppressed in ECAPA-TDNN to train a VIB for these attributes, we repeat our experiments with its predecessor, x-vector. As shown in Sections 5.1 and 5.2, x-vector represents content and channel more than other speaker verification models. Results can be found in Tables 5.2e and 5.2f. Each VIB shows successful disentanglement, but the stage 2 speaker VIB does not improve over the stage 1 speaker VIB. However, for both VIBs trained on speaker, probing performance on content and channel decreases to chance level, indicating successful disentanglement. Although again, compression on all attributes decreases when probing using British Isles and VOiCES, the attribute trained for remains best represented, similar to WavLM and contrary to ECAPA-TDNN.

5.5 Visualisations of trained VIBs

Prompted by reflection on our disentanglement results, we form hypotheses about the behaviour of our VIBs, as explained below. To investigate these, we visualize the weights of VIBs' decoders and encodings of VIBs' encoders.

Table 5.2: **Probing results for VIBs trained on representations from WavLM, ECAPA-TDNN and x-vector.** We train stage 1 VIBs on speaker, content and channel encoding, and a stage 2 VIB on speaker encoding. Probing results of the original model's embeddings are also provided. Compression (chance level at 1, higher indicates better representation) for each probing task is averaged over three runs, with standard deviation shown in parentheses. For successful disentanglement, we want compression of the attribute trained for to increase compared to the original embeddings, and compression for both other attributes to decrease. The VIBs were trained on SCC. Probing was done on SCC and on British Isles and VOiCES.

(a) WavLM on SCC.

Encoder	Speaker	Content	Channel
WavLM (general)	2.45 (0.042)	8.29 (0.618)	3.65 (0.069)
Stage 1 speaker VIB	**2.74** (0.009)	1.25 (0.012)	1.00 (0.001)
Stage 1 content VIB	1.00 (0.000)	**19.74** (0.459)	1.05 (0.000)
Stage 1 channel VIB	1.00 (0.000)	1.08 (0.006)	**3.85** (0.029)
Stage 2 speaker VIB	2.65 (0.005)	1.28 (0.009)	1.00 (0.000)

(b) WavLM on British Isles and VOiCES.

Encoder	Speaker	Content	Channel
WavLM (general)	**3.09** (0.054)	**12.51** (3.312)	**3.82** (0.066)
Stage 1 speaker VIB	1.60 (0.022)	1.23 (0.011)	1.17 (0.005)
Stage 1 content VIB	1.03 (0.006)	8.06 (0.135)	1.25 (0.012)
Stage 1 channel VIB	1.36 (0.019)	1.66 (0.032)	1.95 (0.032)
Stage 2 speaker VIB	1.56 (0.033)	1.26 (0.006)	1.16 (0.005)

(c) ECAPA-TDNN on SCC.

Encoder	Speaker	Content	Channel
ECAPA-TDNN	**11.88** (0.222)	**2.04** (0.015)	**1.53** (0.020)
Stage 1 speaker VIB	**12.02** (0.330)	1.00 (0.000)	1.00 (0.000)
Stage 1 content VIB	1.10 (0.009)	1.05 (0.003)	1.00 (0.001)
Stage 1 channel VIB	1.07 (0.004)	1.00 (0.000)	1.44 (0.008)
Stage 2 speaker VIB	9.84 (0.205)	1.00 (0.000)	1.00 (0.000)

(d) ECAPA-TDNN on British Isles and VOiCES.

Encoder	Speaker	Content	Channel
ECAPA-TDNN	**11.78** (0.996)	0.90 (0.004)	**1.34** (0.015)
Stage 1 speaker VIB	1.04 (0.014)	**1.00** (0.000)	1.00 (0.000)
Stage 1 content VIB	1.16 (0.017)	**1.00** (0.000)	1.00 (0.001)
Stage 1 channel VIB	1.36 (0.021)	**1.00** (0.000)	1.02 (0.004)
Stage 2 speaker VIB	1.03 (0.016)	**1.00** (0.000)	1.00 (0.001)

(e) x-vector on SCC.

Encoder	Speaker	Content	Channel
x-vector	6.85 (0.214)	2.34 (0.106)	2.90 (0.079)
Stage 1 speaker VIB	**9.11** (0.081)	1.00 (0.000)	1.00 (0.002)
Stage 1 content VIB	1.03 (0.001)	**2.56** (0.031)	1.03 (0.004)
Stage 1 channel VIB	1.00 (0.000)	1.00 (0.000)	**3.37** (0.032)
Stage 2 speaker VIB	**8.97** (0.129)	1.00 (0.000)	1.00 (0.001)

(f) x-vector on British Isles and VOiCES.

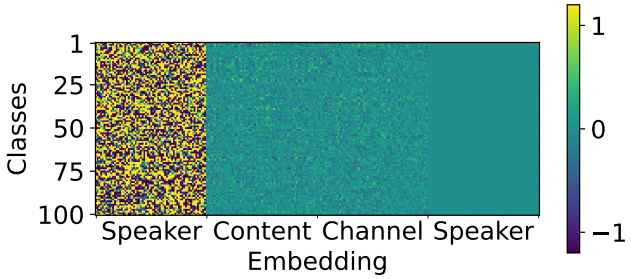
Encoder	Speaker	Content	Channel
x-vector	**7.70** (0.640)	**2.78** (0.204)	**3.39** (0.062)
Stage 1 speaker VIB	1.98 (0.072)	1.00 (0.000)	1.02 (0.006)
Stage 1 content VIB	1.09 (0.006)	1.19 (0.013)	1.09 (0.007)
Stage 1 channel VIB	1.50 (0.087)	1.01 (0.001)	1.50 (0.061)
Stage 2 speaker VIB	1.95 (0.001)	1.00 (0.000)	1.02 (0.002)

Decoder weights. As shown above, speaker representation of the stage 2 speaker VIBs does not exceed that of the stage 1 speaker VIBs. This could be caused by the stage 2 VIB's decoder being able to get enough speaker information from the stage 1 embeddings. In this case, the VIB could remove relevant information from the stage 2 speaker embedding, resulting in a weaker representation of speaker. To investigate this hypothesis, we visualize the weights of stage 2 VIBs' decoders.

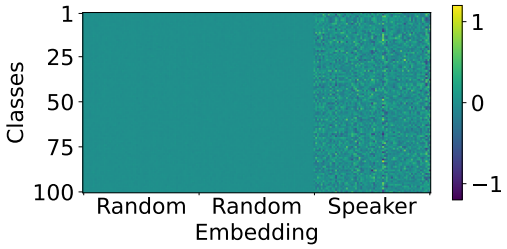
First we perform two control experiments, to verify that our VIBs and visualization method work as expected. We train a stage 2 speaker VIB on WavLM that receives the stage 1 speaker VIB embedding, besides the stage 1 content and channel VIB embeddings. Figure 5.3a shows that the decoder relies heavily on the stage 1 speaker embedding and also learns to use the stage 1 content and channel embeddings, but, as expected, does not use the newly trained stage 2 speaker embedding. We also train a stage 2 speaker VIB on WavLM that receives random vectors instead of stage 1 VIB embeddings. Figure 5.3b shows that in this case, the decoder learns to exclusively pay attention to the newly trained encoding, as expected.

Figures 5.3c to 5.3e show the weights of our regular trained stage 2 speaker VIBs. For WavLM, the decoder learns to take all of the embeddings into account equally, both the newly trained speaker embedding and the content and channel embeddings from stage 1. For ECAPA-TDNN and x-vector, the decoder puts less weight on the channel embedding. This aligns with probing results, as for both models, the stage 1 channel VIB represents speaker slightly worse than the stage 1 content VIB. However, for both models, the newly trained speaker embedding plays an important role. Thus, we conclude that the cause of stage 2 speaker VIBs not showing better probing performance than stage 1 speaker VIBs is not caused by the necessary information being available in the concatenated stage 1 embeddings. Possibly, stage 2 does not improve performance in general, as Mohebbi et al. (2024) do not include an ablation experiment.

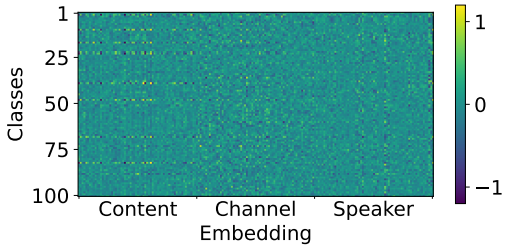
Encoder samples. As shown earlier, probing results on trained VIBs do not generalise well to other datasets. This is the case for all models, but in particular for the VIBs trained on ECAPA-TDNN, where the VIBs trained on content and channel obtain higher speaker compression than the VIBs trained on speaker. This could be caused by the encodings from the VIB being informative when used on the corpus trained on, but not when used to encode



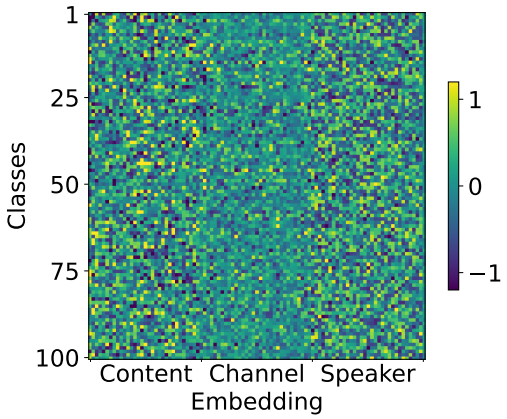
(a) Control WavLM stage 2 VIB that gets a stage 1 speaker VIB embedding.



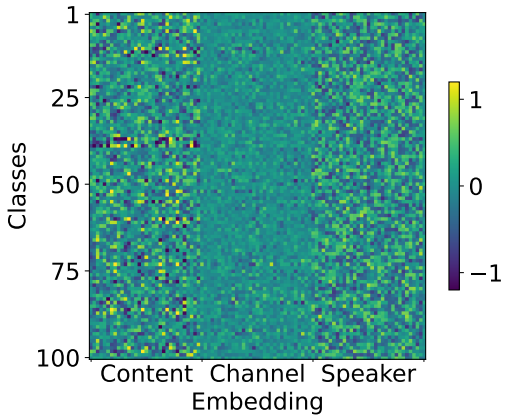
(b) Control WavLM stage 2 VIB that gets two random vectors.



(c) WavLM stage 2 speaker VIB.



(d) ECAPA-TDNN stage 2 speaker VIB.



(e) x-vector stage 2 speaker VIB.

Figure 5.3: **Decoder weights of stage 2 VIBs**. We show the stage 2 speaker VIBs for WavLM, ECAPA-TDNN and x-vector, which get access to content and channel embeddings from stage 1. Two control stage 2 speaker VIBs for WavLM are also provided, one that additionally gets access to a stage 1 speaker embedding, and one that gets access to randomised values instead of trained embeddings. The right-most column in each plot represents the bias vector. Weight values that are outside $(-1.2, 1.2)$ get clipped, to allow better visibility of small weight differences.

examples from other datasets. To investigate this, we plot sample encodings from VIBs on the different datasets, comparing examples within a class and examples across classes.

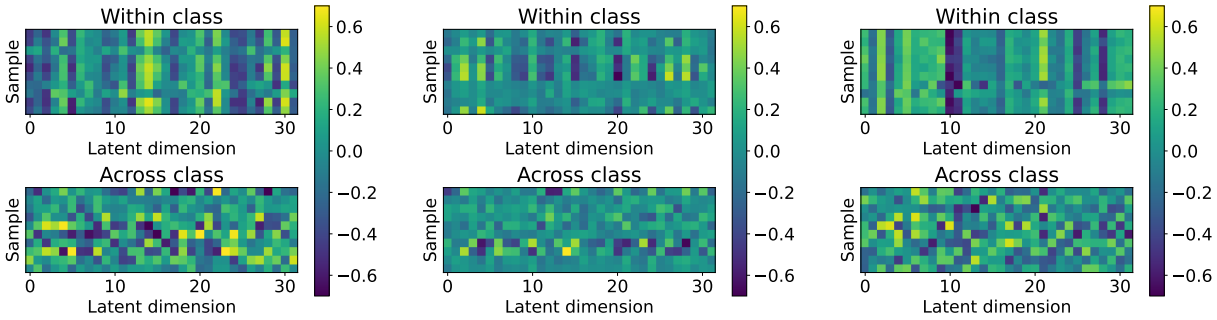
We show encoder samples from stage 1 x-vector and ECAPA-TDNN VIBs, on SCC and British Isles and VOiCES, in Figure 5.4. For x-vector, when using SCC, encodings for examples within a class are generally fairly similar, whereas they are different across classes. When using British Isles and VOiCES, variance within encodings decreases, with latent values getting closer to 0. The difference between within class and across class conditions decreases, aligning with the decreased probing performance found. Similar results are observed for speaker, content and channel tasks.

For ECAPA-TDNN, we see different patterns than for x-vector. For content and channel VIBs, within class samples do not appear more similar than across class samples, matching poor probing performance of these VIBs. Here, when probing on the other datasets, behaviour does not noticeably change. For speaker, examples from SCC generate very similar encodings within class and very different embeddings across classes, accounting for the high speaker compression found. However, examples from British Isles generate extremely similar encodings, whether within class or across class. These encodings are close to 0 and seem to not represent any extractable information, explaining the large drop in probing performance. These results also clarify that speaker compression on content and channel VIBs exceeds that of the speaker VIB, because the former show variation that might be more or less useful to a probe, whereas the latter exhibits very little variation at all.

Overall, we note that the more disentangled a VIB's results are on SCC, the closer to 0 its encodings become when using other datasets. This is likely caused by the information loss, which aims to get the encoding as close to the 0 vector as possible. The task loss ensures that information required for the task is still kept, but we find that this information is specific to the corpus.

5.6 Speaker verification performance

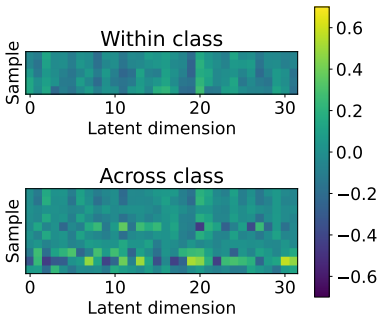
We evaluate our ECAPA-TDNN and x-vector VIBs on speaker verification, comparing to the models investigated. Results can be found in Table 5.3. WavLM, UniSpeech-SAT and x-vector get higher EER than reported by their authors, which is discussed in Section 6.2. All ECAPA-TDNN VIBs get performance close to chance level. The VIBs trained on speaker do worst. This is caused by the embeddings for most utterances being almost identical, regardless of the target label, similar to what is shown in Figure 5.4j (like British Isles and VOiCES, VoxCeleb is based on a different corpus than SCC). However, the VIBs trained on x-vector embeddings show better EER when they are trained on speaker, although still far from the performance of the original embeddings.



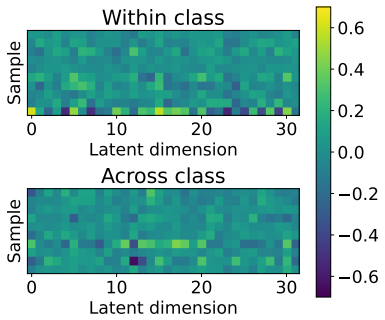
(a) x-vector speaker, SCC.

(b) x-vector content, SCC.

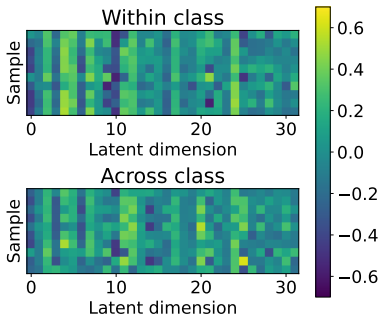
(c) x-vector channel, SCC.



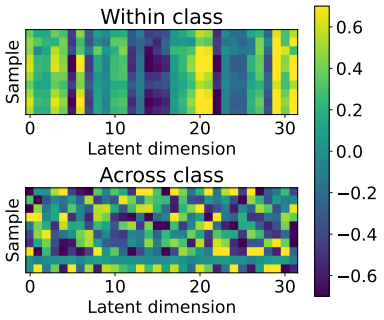
(d) x-vector speaker, British Isles.



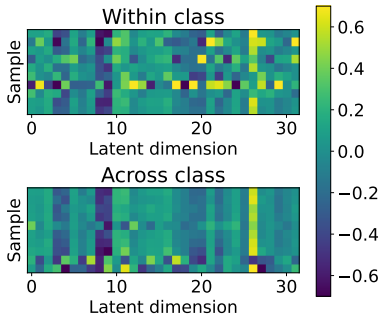
(e) x-vector content, British Isles.



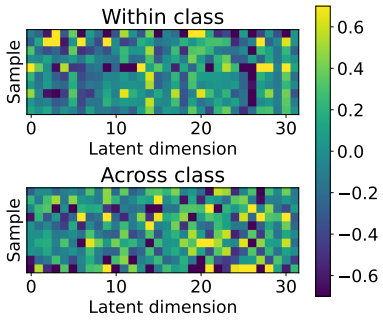
(f) x-vector channel, VOiCES.



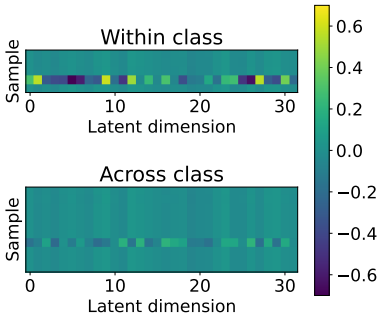
(g) ECAPA-TDNN speaker, SCC.



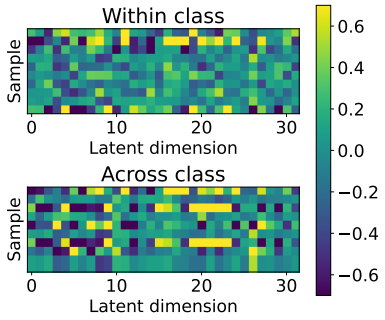
(h) ECAPA-TDNN content, SCC.



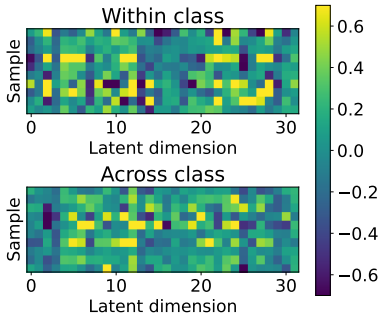
(i) ECAPA-TDNN channel, SCC.



(j) ECAPA-TDNN speaker, British Isles.



(k) ECAPA-TDNN content, British Isles.



(l) ECAPA-TDNN channel, VOiCES.

Figure 5.4: **Encoder samples of stage 1 x-vector and ECAPA-TDNN VIBs.** In the 'within class' condition, all samples have the same target label, in the 'across class' condition, all samples have different target labels. For SCC, we use the splits that are used for probing. We take 10 random samples for each condition, except when less than 10 examples with the same label are in the dataset. Values outside $(-0.7, 0.7)$ get clipped, to allow better visibility of small differences.

Table 5.3: **Speaker verification results of investigated models and VIBs.** Models are evaluated on VoxCeleb1. Lower EER indicates better speaker verification performance. Chance level is 50%.

(a) Original models.

Model	EER (%)
WavLM (SV)	4.93
UniSpeech-SAT (SV)	5.18
ECAPA-TDNN	0.90
x-vector	8.87
ResNet	1.04
TitaNet	**0.83**

(b) Trained VIBs.

Model	VIB	EER (%)
ECAPA-TDNN	Stage 1 speaker	48.94
	Stage 1 content	44.25
	Stage 1 channel	41.98
	Stage 2 speaker	49.51
x-vector	Stage 1 speaker	30.55
	Stage 1 content	42.85
	Stage 1 channel	41.23
	Stage 2 speaker	**28.72**

Chapter 6

Discussion and conclusions

This chapter aims to give a broader view of our work, putting it in context. Section 6.1 repeats our findings and emphasizes important results. In Section 6.2, we touch on some potential limitations of our work. Section 6.3 suggests directions for future research. Section 6.4 gives a statement on ethical considerations regarding this study, and Section 6.5 wraps up by giving a short summary of this thesis.

6.1 Contributions

In this work, we study to what degree content and channel are encouraged or suppressed in embeddings from models for automatic speaker verification, and attempt to further disentangle content and channel from speaker embeddings, without decreasing speaker verification performance. This section repeats and elaborates on our findings.

Representation of content and channel in speaker embeddings. In our interpretability experiments, we find that, with the exception of x-vector, all investigated speaker verification models encourage speaker information without increasing content and channel representation, in the case of ECAPA-TDNN even suppressing content and channel compared to its baselines. Most earlier research (Peri et al., 2020a; Raj et al., 2019; Zhao et al., 2022) does not use randomly initialised or feature baselines, increasing concerns regarding the validity of probing results. Ashihara et al. (2024) compare to a feature baseline and find content to be represented by ECAPA-TDNN, but they look at all model layers, rather than just the speaker embedding. As we found in our experiments, the middle layers of ECAPA-TDNN represent content above the level of the baselines, explaining the earlier finding.

Embeddings from randomly initialised versions of the speaker verification models, as well as the input features of the models, regularly achieve probing performance above chance level, justifying comparison to such baselines instead. However, possibly content and channel can be suppressed further relative to the baselines, as we see ECAPA-TDNN do in some of our experiments, and chance level can be reached for both attributes.

Our results show that x-vector is influenced by content and channel to a much greater degree than other contemporary speaker verification models. This result is significant because x-vector is used in forensic practice (VOCALISE (Alexander et al., 2016; Kelly et al., 2019), for example, is a tool used in forensic research that provides an x-vector model (van der Vloed and Cambier-Langeveld, 2023)). If LDA, PLDA or other techniques to improve robustness are employed, the impact of content and channel on the final likelihood ratio might be reduced. However, it would be worth exploring the replacement of x-vector by ECAPA-TDNN, as we show it to be less affected by content and channel. ECAPA-TDNN has been successfully applied to forensic datasets, and found to improve speaker verification performance over x-vector (Sigona

and Grimaldi, 2024; Sztahó and Fejes, 2023).

Disentanglement of content and channel. Using the two-stage VIB approach proposed by Mohebbi et al. (2024), we are able to successfully train disentangled embeddings for speaker, content and channel on top of WavLM, a general speech model, as well as on x-vector. For ECAPA-TDNN, we do not obtain a disentangled representation of content and channel, which could be explained by content and channel representation being too low in ECAPA-TDNN to successfully learn to encode.

We find that the second stage of the two-stage approach does not improve results over the first stage. We show that this is not caused by too much speaker information being contained by the content and channel embeddings. It is uncertain whether this is unique to our application of the approach, as Mohebbi et al. (2024) do not include an ablation experiment.

Results on trained VIB encodings of all three models do not generalise well when probed with datasets from different corpora than those used for training. In the case of WavLM and x-vector, disentanglement is still shown, but we observe lower compression on all three attributes, compared to the original embeddings. We show that the VIB encoders that are well disentangled on the corpus we train on, exhibit less variance when used on other corpora. It is likely that the VIB approach suppresses information that is not necessary for the specific classification task and corpus, although it is relevant for the target attribute in general. It is worth noting that Mohebbi et al. (2024) use a transcription task for training a content representation, which we consider to be a more challenging task than sentence classification. However, it is also not clear whether the results in Mohebbi et al. (2024) generalise, as probing performance is evaluated on data that are also used in VIB training, in both stages. Evaluation on VoxCeleb shows that VIBs trained using our approach do not obtain satisfactory speaker verification performance.

Novel dataset for classification of speaker, content and channel. For our disentanglement experiments, we construct SCC, a novel synthetic dataset based on four existing corpora. We believe this dataset to be the first that displays controlled variation of both speaker, content and channel. We provide six splits, two each for training, development and testing. SCC contains 135,000 examples in total, with 100 classes each for speaker, content and channel classification.

6.2 Limitations

In this section we address some possible limitations concerning different aspects of our approach.

Underperforming speaker verification models. When evaluating our investigated models using VoxCeleb, some models obtain a much higher EER than reported by its authors. In particular, x-vector is expected to reach 3.2%[1] compared to our 8.87%, and WavLM is expected to reach 0.84% (Chen et al., 2022a), compared to our 4.93%. UniSpeech-SAT also achieves worse performance than expected[2]. We have not been able to find the cause of these discrepancies. Brydinskyi et al. (2024), who evaluate speaker verification models on a novel dataset, also find surprisingly poor performance for WavLM compared to the other models they evaluate, among which TitaNet and ECAPA-TDNN.

These deviating results might indicate that the models concerned do not behave as intended. However, as these models are publicly available and commonly used[3], we believe the results are

[1] https://huggingface.co/speechbrain/spkrec-xvect-voxceleb

[2] Judging from results on related models; VoxCeleb evaluation of the checkpoint we use is unavailable.

[3] As of June 2025, within the last month, WavLM was downloaded 350,000+ times (https://web.archive.org/web/20250610074658/https://huggingface.co/microsoft/wavlm-base-plus-sv),

significant, regardless of whether the model is performing as intended. Moreover, the other three models evaluated obtain an EER closer to what is reported by the authors: ECAPA-TDNN is expected to reach 0.80%[4] compared to our 0.90%, ResNet is expected to reach 1.05%[5] compared to our 1.04%, and TitaNet is expected to reach 0.66%[6] compared to our 0.83%.

Shortcomings of probing. Probing is a popular interpretability technique, but it has also attracted criticism (Belinkov, 2022). Hyperparameter choices can influence results, results are hard to interpret without controls, datasets might not reflect the target task, and extraction of information from representations does not necessarily indicate that the information is used by the model. We go over these issues in turn.

In response to the challenge of hyperparameter selection, we use minimum description length probing, which has been shown to be robust against many axes of hyperparameter variation (Voita and Titov, 2020). Furthermore, we perform one of the first studies of interpretability of speaker verification that includes controls, comparing probing results against skylines and baselines. With these two decisions, we believe we have taken into account many of the common shortcomings of probing.

We also repeat some of our probing experiments on other corpora, and find that results are not entirely robust. This is exemplary of another issue with probing, which is that probing tasks have to be operationalised as datasets, and the dataset might not reflect the full complexity of the task (Ravichander et al., 2021). Although we find similar trends across different datasets, it is possible that the classification task does not truly reflect the target attribute, because performance on our datasets might be correlated with other information. In particular, different sentences have different lengths, and as such, when an embedding encodes sentence length (which was shown to be the case for some speaker verification models by Raj et al. (2019) and Wang et al. (2017)), this might be enough for a probe to be able to show improved sentence classification compared to the baselines, without the embeddings necessarily representing any of the more complex aspects of content. However, in this study, we make no claims about what specific aspects of content and channel are represented in speaker embeddings. Indeed, any effect of intra-speaker variation on speaker verification is undesirable.

Finally, a common concern is that information extracted by a probe is not necessarily used in model predictions (Belinkov, 2022; Ravichander et al., 2021). Indeed, we cannot say for certain that, for example, ECAPA-TDNN uses content information to arrive at its final speaker embedding, solely because content is well represented in the middle layers. The applicability of this concern to speaker embeddings, our object of research, depends on what similarity measure is used. When cosine similarity is used to compare different speaker embeddings, any information represented by the embeddings will have an effect on the score, because cosine similarity is an unparameterised function. When PLDA is used, however, it is possible that it can learn to disregard content and channel information that is represented by the embeddings.

Set-up of SCC. For our disentanglement experiments, we synthesize a novel dataset, SCC. Models might perform differently on real recorded data than on synthesized and/or augmented data. We hope to have created a reasonably realistic dataset, by basing synthesis and augmentation on corpora of recorded naturalistic sounds. Furthermore, we also evaluate our trained models using non-synthesized datasets.

UniSpeech-SAT 3000+ times (https://web.archive.org/web/20250610074726/https://huggingface.co/microsoft/unispeech-sat-base-plus-sv), and x-vector 15,000+ times (https://web.archive.org/web/20250610074815/https://huggingface.co/speechbrain/spkrec-xvect-voxceleb).

[4] https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

[5] https://huggingface.co/speechbrain/spkrec-resnet-voxceleb

[6] https://huggingface.co/nvidia/speakerverification_en_titanet_large

We use SCC both for training and probing VIBs, to evaluate disentanglement within the VIB training domain. To get improved generalisation, we provide two sets of splits that do not contain the same examples. However, both sets do use the same classes. Thus, to illustrate, when using SCC for disentanglement and probing of content, the probe is learning to classify the same 100 sentences that the VIB has seen during training. This likely inflates probing results. It would have been ideal to select different classes from the same corpus for the probing and disentanglement datasets, to have the domain equal but still evaluate generalisation. When training a model using SCC, we advise also evaluating it using a different dataset, as we do in this work.

Options for disentanglement. The literature for disentanglement of neural representations, and even speaker verification disentanglement specifically, is very broad. In this work, we adopt one approach, and do not obtain improved speaker verification performance. However, it is possible that an approach based on Mohebbi et al. (2024) could work better with a different task operationalisation, or with different hyperparameter settings than we experiment with. Moreover, we do not make any general statements about the possibility of further disentangling speaker verification embeddings without decreasing speaker verification. We show that ECAPA-TDNN demonstrates excellent disentanglement, but as probing results are often above chance level and sometimes above the level of our baselines, there could still be room for improvement using another approach.

6.3 Future directions

To complement our findings, we suggest three directions for future research.

Role of content in middle layers. A question that might be raised is why the middle layers of ECAPA-TDNN, ResNet and TitaNet improve content representation, before it being suppressed again in the final layer. Our hypothesis is that the models learn to extract phonetic information in the middle layers, in order to be able to learn phonetically-dependent speaker features. For example, different speakers might produce certain vowels differently. It has been shown that incorporating phonetic information can improve speaker verification performance (Liu et al., 2018, 2022; Wang et al., 2019). To study if and how speaker representation depends on content representation in middle layers, amnesic probing (Elazar et al., 2021) might be used, which removes information extracted by a probe from the representation and observes the effect on task performance.

Effect of (P)LDA. The speaker verification systems investigated in this research use cosine similarity as a back-end. However, PLDA is also commonly used, and sometimes the similarity measure is preceded by LDA. Both of these techniques aim to maximise inter-speaker variance while reducing dimensionality. We expect that this aids suppression of content and channel information. Future research could explore how LDA and PLDA affect representation of channel and content.

Multiple disentangled embeddings for separate attributes. In this work, our goal is obtaining speaker embeddings that focus on speaker information as much as possible, disregarding variability in content and channel. However, content and channel information might both contribute to a correct judgment in speaker verification. Different people use different vocabularies, so similarity between lexical content of two utterances can point to whether they were uttered by the same person or not. Similarity in channel information might represent that two utterances were recorded from the same phone, or in the same place, both of which increase the likelihood that the utterances originate from the same person.

It would be worthwhile to train different disentangled embeddings for separate attributes. These embeddings should not be built on top of speaker verification models, as they suppress a lot of this information, but trained from scratch or using general pretrained speech models. Future work could aim to develop a speaker verification tool which provides not one but three similarity scores or likelihood ratios, expressing the similarity in speaker, content and channel. An aggregate score could be formed by weighting the three aspects, or the judgment could be left to the user. In either case, this would result in more interpretable speaker verification that can make use of different kinds of information in the signal.

6.4 Ethical considerations

In this work, we have not conducted any experiments with human participants, nor collected novel data. The work might however be affected by issues in existing datasets. In particular, data for VoxCeleb, which was used for evaluation of speaker verification performance, was collected without consent of the included individuals, and contains representational biases (Huh et al., 2024; Leschanowsky et al., 2025). We decided to use VoxCeleb in spite of these issues to be able to compare to other research, as it is the standard in the field (Wang et al., 2024b).

6.5 Conclusion

We find that embeddings for automatic speaker verification represent less content and channel than was suggested by previous research. x-vector forms an exception, and we suggest replacing it by ECAPA-TDNN, a modern alternative that gets better speaker performance and achieves the highest degree of disentanglement in our experiments. We apply a two-stage Variational Information Bottleneck approach towards further disentangling ECAPA-TDNN embeddings. The results support the conclusion that content and channel are hardly represented in ECAPA-TDNN embeddings. We suggest that there might be a place for content and channel information in speaker verification, and propose creating a system that combines disentangled speaker, content and channel representations.

Chapter 7

Bibliography

Zrar Kh. Abdul and Abdulbasit K. Al-Talabani. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access*, 10:122136–122158, 2022. doi: 10.1109/ACCESS.2022.3223444.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BJh6Ztuxl.

Ehsan Aghazadeh, Mohsen Fayyaz, and Yadollah Yaghoobzadeh. Metaphors in pre-trained language models: Probing and generalization across datasets and languages. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2037–2050, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.144. URL https://aclanthology.org/2022.acl-long.144/.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyxQzBceg.

Anil Alexander, Oscar Forth, Alankar Aryal Atreya, and Finnian Kelly. VOCALISE: A forensic automatic speaker recognition system supporting spectral, phonetic, and user-provided features. In *Odyssey 2016*, 2016.

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. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10166–10170, 2024. doi: 10.1109/ICASSP48485.2024.10446422.

B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *The Journal of the Acoustical Society of America*, 55 (6):1304–1312, 06 1974. ISSN 0001-4966. doi: 10.1121/1.1914702. URL https://doi.org/10.1121/1.1914702.

Zhongxin Bai and Xiao-Lei Zhang. Speaker recognition based on deep learning: An overview. *Neural Networks*, 140:65–99, 2021. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2021.03.004. URL https://www.sciencedirect.com/science/article/pii/S0893608021000848.

Zhongxin Bai, Jianyu Wang, Xiao-Lei Zhang, and Jingdong Chen. End-to-end speaker verification via curriculum bipartite ranking weighted binary cross-entropy. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1330–1344, 2022.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 04 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00422. URL https://doi.org/10.1162/coli_a_00422.

Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 04 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00254. URL https://doi.org/10.1162/tacl_a_00254.

Imen Ben-Amor and Jean-François Bonastre. Ba-lr: Binary-attribute-based likelihood ratio estimation for forensic voice comparison. In *2022 International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6, 2022. doi: 10.1109/IWBF55382.2022.9794542.

Imen Ben-Amor, Jean-François Bonastre, Benjamin O'Brien, and Pierre-Michel Bousquet. Describing the phonetics in the underlying speech attributes for deep and interpretable speaker recognition. In *Interspeech 2023*, pages 3207–3211, 2023. doi: 10.21437/Interspeech.2023-1648.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Teva Merlin, Javier Ortega-García, Dijana Petrovska-Delacrétaz, and Douglas A Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, 2004(101962), 2004.

Vitalii Brydinskyi, Yuriy Khoma, Dmytro Sabodashko, Michal Podpora, Volodymyr Khoma, Alexander Konovalov, and Maryna Kostiak. Comparison of modern deep learning models for speaker verification. *Applied Sciences*, 14(4), 2024. ISSN 2076-3417. doi: 10.3390/app14041329. URL https://www.mdpi.com/2076-3417/14/4/1329.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and Julian Weber. XTTS: a massively multilingual zero-shot text-to-speech model. In *Interspeech 2024*, pages 4978–4982, 2024. doi: 10.21437/Interspeech.2024-2016.

Christophe Champod and Didier Meuwly. The inference of identity in forensic speaker recognition. *Speech Communication*, 31(2):193–203, 2000. ISSN 0167-6393. doi: https://doi.org/10.1016/S0167-6393(99)00078-3. URL https://www.sciencedirect.com/science/article/pii/S0167639399000783.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. Gigaspeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. In *Interspeech 2021*, pages 3670–3674, 2021. doi: 10.21437/Interspeech.2021-1965.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022a. doi: 10.1109/JSTSP.2022.3188113.

Sanyuan Chen, Yu Wu, Chengyi Wang, Zhengyang Chen, Zhuo Chen, Shujie Liu, Jian Wu, Yao Qian, Furu Wei, Jinyu Li, and Xiangzhan Yu. Unispeech-Sat: Universal speech representation learning with speaker aware pre-training. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6152–6156, 2022b. doi: 10.1109/ICASSP43922.2022.9747077.

Xianhong Chen and Changchun Bao. Phoneme-unit-specific time-delay neural network for speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1243–1255, 2021. doi: 10.1109/TASLP.2021.3065202.

Zhengyang Chen, Shuai Wang, Yanmin Qian, and Kai Yu. Channel invariant speaker embedding learning with joint multi-task and adversarial training. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6574–6578, 2020. doi: 10.1109/ICASSP40776.2020.9053905.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. CLUB: A contrastive log-ratio upper bound of mutual information. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1779–1788. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/cheng20b.html.

Aemon Yat Fei Chiu, Paco Kei Ching Fung, Roger Tsz Yeung Li, Jingyu Li, and Tan Lee. Probing speaker-specific features in speaker representations, 2025. URL https://arxiv.org/abs/2501.05310.

Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee. Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. In *Interspeech 2018*, pages 501–505, 2018. doi: 10.21437/Interspeech.2018-1830.

Grzegorz Chrupała, Bertrand Higy, and Afra Alishahi. Analyzing analytical methods: The case of phonology in neural models of spoken language. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4146–4156, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.381. URL https://aclanthology.org/2020.acl-main.381/.

Yu-An Chung, Yonatan Belinkov, and James Glass. Similarity analysis of self-supervised speech representations. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3040–3044, 2021. doi: 10.1109/ICASSP39728.2021.9414321.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198/.

S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980. doi: 10.1109/TASSP.1980.1163420.

Isin Demirsahin, Oddur Kjartansson, Alexander Gutkin, and Clara Rivera. Open-source multi-speaker corpora of the English accents in the British isles. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6532–6541, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.804/.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech 2020*, pages 3830–3834, 2020. doi: 10.21437/Interspeech.2020-2650.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 03 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00359. URL https://doi.org/10.1162/tacl_a_00359.

Xin Fang, Liang Zou, Jin Li, Lei Sun, and Zhen-Hua Ling. Channel adversarial training for cross-channel text-independent speaker recognition. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6221–6225, 2019. doi: 10.1109/ICASSP.2019.8682327.

Luciana Ferrer, Mitchell McLaren, and Niko Brümmer. A speaker verification backend with robust performance across conditions. *Computer Speech & Language*, 71:101258, 2022. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2021.101258. URL https://www.sciencedirect.com/science/article/pii/S0885230821000656.

Constanza Fierro, Nicolas Garneau, Emanuele Bugliarello, Yova Kementchedjhieva, and Anders Søgaard. MuLan: A study of fact mutability in language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 762–771, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.67. URL https://aclanthology.org/2024.naacl-short.67/.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

Ronald A Fisher. The precision of discriminant functions. *Annals of Eugenics*, 10(1):422–429, 1940.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL http://jmlr.org/papers/v17/15-239.html.

Bin Gu, Jie Zhang, and Wu Guo. A dynamic convolution framework for session-independent speaker embedding learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:3647–3658, 2023. doi: 10.1109/TASLP.2023.3313431.

Jingxiang Guo, Jinxuan Zhu, Sixu Lin, and Feng Shi. ECAPA-TDNN embeddings for speaker recognition. In *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 1488–1491, 2024. doi: 10.1109/AINIT61980.2024.10581514.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. Distributional vectors encode referential attributes. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1002. URL https://aclanthology.org/D15-1002/.

Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu. ContextNet: Improving convolutional neural networks for automatic speech recognition with global context. In *Interspeech 2020*, pages 3610–3614, 2020. doi: 10.21437/Interspeech.2020-2059.

John H.L. Hansen and Hynek Bořil. On the issues of intra-speaker variability and realism in speech, speaker, and language recognition tasks. *Speech Communication*, 101:94–108, 2018. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2018.05.004. URL https://www.sciencedirect.com/science/article/pii/S0167639317303849.

John H.L. Hansen and Taufiq Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99, 2015. doi: 10.1109/MSP.2015.2462851.

Md. Rashidul Hasan, Mustafa Jamil, Md. Golam Rabbani, and Md. Saifur Rahman. Speaker identification using mel frequency cepstral coefficients. In *3rd International Conference on Electrical and Computer Engineering*, pages 565–568, 2004.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 5115–5119. IEEE Press, 2016. doi: 10.1109/ICASSP.2016.7472652. URL https://doi.org/10.1109/ICASSP.2016.7472652.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Qian-Bei Hong, Chung-Hsien Wu, and Hsin-Min Wang. Decomposition and reorganization of phonetic information for speaker embedding learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1745–1757, 2023. doi: 10.1109/TASLP.2023.3267833.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/0a0a0c8aaa00ade50f74a3f0ca981ed7-Paper.pdf.

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. doi: 10.1109/TASLP.2021.3122291.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2011–2023, 2020. doi: 10.1109/TPAMI.2019.2913372.

Jaesung Huh, Joon Son Chung, Arsha Nagrani, Andrew Brown, Jee-weon Jung, Daniel Garcia-Romero, and Andrew Zisserman. The VoxCeleb speaker recognition challenge: A retrospective. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3850–3866, 2024. doi: 10.1109/TASLP.2024.3444456.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926, 2018.

Sergey Ioffe. Probabilistic linear discriminant analysis. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 531–542, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33839-0.

Keith Ito and Linda Johnson. The LJ Speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

Rashid Jahangir, Ying Wah Teh, Henry Friday Nweke, Ghulam Mujtaba, Mohammed Ali Al-Garadi, and Ihsan Ali. Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges. *Expert Systems with Applications*, 171:114591, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.114591. URL https://www.sciencedirect.com/science/article/pii/S0957417421000324.

Maros Jakubec, Roman Jarina, Eva Lieskovska, and Peter Kasak. Deep speaker embeddings for speaker verification: Review and experimental comparison. *Engineering Applications of Artificial Intelligence*, 127:107232, 2024. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.107232. URL https://www.sciencedirect.com/science/article/pii/S0952197623014161.

Jee-weon Jung, Hee-Soo Heo, Ju-ho Kim, Hye-jin Shim, and Ha-Jin Yu. RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification. In *Interspeech 2019*, pages 1268–1272, 2019. doi: 10.21437/Interspeech.2019-1982.

Muhammad Mohsin Kabir, M. F. Mridha, Jungpil Shin, Israt Jahan, and Abu Quwsar Ohi. A survey of speaker recognition: Fundamental theories, recognition methods and opportunities. *IEEE Access*, 9:79236–79263, 2021. doi: 10.1109/ACCESS.2021.3084299.

J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. Libri-Light: A benchmark for ASR with limited or no supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673, 2020. doi: 10.1109/ICASSP40776.2020.9052942.

Jingu Kang, Jaesung Huh, Hee Soo Heo, and Joon Son Chung. Augmentation adversarial training for self-supervised speaker representation learning. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1253–1262, 2022a. doi: 10.1109/JSTSP.2022.3200915.

Woo Hyun Kang, Sung Hwan Mun, Min Hyun Han, and Nam Soo Kim. Disentangled speaker and nuisance attribute embedding for robust speaker verification. *IEEE Access*, 8:141838–141849, 2020. doi: 10.1109/ACCESS.2020.3012893.

Woohyun Kang, Md Jahangir Alam, and Abderrahim Fathan. MIM-DG: Mutual information minimization-based domain generalization for speaker verification. In *Interspeech 2022*, pages 3674–3678, 2022b. doi: 10.21437/Interspeech.2022-142.

Jodi Kearns. Librivox: Free public domain audiobooks. *Reference Reviews*, 28(1):7–8, 2014.

Finnian Kelly, Oscar Forth, Samuel Kent, Linda Gerlach, and Anil Alexander. Deep neural network based forensic automatic speaker recognition in VOCALISE using x-vectors. In *Audio Engineering Society Conference: 2019 AES International Conference on Audio Forensics*. Audio Engineering Society, 2019.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Conference proceedings: papers accepted to the International Conference on Learning Representations (ICLR) 2014*, 2014. URL https://arxiv.org/abs/1312.6114.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224, 2017. doi: 10.1109/ICASSP.2017.7953152.

Arne Köhn. What's in an embedding? analyzing word embeddings through multilingual evaluation. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1246. URL https://aclanthology.org/D15-1246/.

Nithin Rao Koluguri, Taejin Park, and Boris Ginsburg. TitaNet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8102–8106, 2022. doi: 10.1109/ICASSP43922.2022.9746806.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Varun Krishna and Sriram Ganapathy. Towards the next frontier in speech representation learning using disentanglement, 2024. URL https://arxiv.org/abs/2407.02543.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, Patrice Castonguay, Mariya Popova, Jocelyn Huang, and Jonathan M. Cohen. NeMo: a toolkit for building ai applications using neural modules, 2019. URL https://arxiv.org/abs/1909.09577.

Shakti Kumar, Jithin Pradeep, and Hussain Zaidi. Learning robust latent representations for controllable speech synthesis. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3562–3575, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.312. URL https://aclanthology.org/2021.findings-acl.312/.

Yoohwan Kwon, Soo-Whan Chung, and Hong-Goo Kang. Intra-class variation reduction of speaker representation in disentanglement framework. In *Interspeech 2020*, pages 3231–3235, 2020. doi: 10.21437/Interspeech.2020-2075.

Anna Leschanowsky, Casandra Rusti, Carolyn Quinlan, Michaela Pnacek, Lauriane Gorce, and Wiebke Hutiri. A data perspective on ethical challenges in voice biometrics research. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 7(1):118–131, 2025. doi: 10.1109/TBIOM.2024.3446846.

Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. Deep speaker: an end-to-end neural speaker embedding system, 2017. URL https://arxiv.org/abs/1705.02304.

Jianchen Li, Jiqing Han, Shiwen Deng, Tieran Zheng, Yongjun He, and Guibin Zheng. Mutual information-based embedding decoupling for generalizable speaker verification. In *Interspeech 2023*, pages 3147–3151, 2023a. doi: 10.21437/Interspeech.2023-1314.

Jianchen Li, Jiqing Han, Fan Qian, Tieran Zheng, Yongjun He, and Guibin Zheng. Distance metric-based open-set domain adaptation for speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2378–2390, 2024. doi: 10.1109/TASLP.2024.3389646.

Kai Li, Masato Akagi, Yibo Wu, and Jianwu Dang. Segment-level effects of gender, nationality and emotion information on text-independent speaker verification. In *Interspeech 2020*, pages 2987–2991, 2020. doi: 10.21437/Interspeech.2020-1700.

Lin Li, Fuchuan Tong, and Qingyang Hong. When speaker recognition meets noisy labels: Optimizations for front-ends and back-ends. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1586–1599, 2022a. doi: 10.1109/TASLP.2022.3169977.

Pengqi Li, Lantian Li, Askar Hamdulla, and Dong Wang. Reliable visualization for deep speaker recognition. In *Interspeech 2022*, pages 331–335, 2022b. doi: 10.21437/Interspeech.2022-926.

Pengqi Li, Lantian Li, Askar Hamdulla, and Dong Wang. Visualizing data augmentation in deep speaker recognition. In *Interspeech 2023*, pages 2243–2247, 2023b. doi: 10.21437/Interspeech.2023-1298.

Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5670–5679. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/yingzhen18a.html.

Weiwei Lin, Chenhang He, Man-Wai Mak, and Youzhi Tu. Self-supervised neural factor analysis for disentangling utterance-level speech representations. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21065–21077. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lin23e.html.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable AI: A review of machine learning interpretability methods. *Entropy*, 23(1), 2021. ISSN 1099-4300. doi: 10.3390/e23010018. URL https://www.mdpi.com/1099-4300/23/1/18.

Tianchi Liu, Rohan Kumar Das, Kong Aik Lee, and Haizhou Li. Neural acoustic-phonetic approach for speaker verification with phonetic attention mask. *IEEE Signal Processing Letters*, 29:782–786, 2022. doi: 10.1109/LSP.2022.3143036.

Tianchi Liu, Kong Aik Lee, Qiongqiong Wang, and Haizhou Li. Disentangling voice and content with self-supervision for speaker recognition. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 50221–50236. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9d276b0a087efdd2404f3295b26c24c1-Paper-Conference.pdf.

Yi Liu, Liang He, Jia Liu, and Michael T. Johnson. Speaker embedding extraction with phonetic information. In *Interspeech 2018*, pages 2247–2251, 2018. doi: 10.21437/Interspeech.2018-1226.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4114–4124. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/locatello19a.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Chau Luu, Steve Renals, and Peter Bell. Investigating the contribution of speaker attributes to speaker separability using disentangled speaker representations. In Hanseok Ko and John H. L. Hansen, editors, *Proceedings of Interspeech 2022*, pages 610–614. ISCA, September 2022. doi: 10.21437/Interspeech.2022-10643.

Yi Ma, Shuai Wang, Tianchi Liu, and Haizhou Li. ExPO: Explainable phonetic trait-oriented network for speaker verification. *IEEE Signal Processing Letters*, pages 1–5, 2025. doi: 10.1109/LSP.2025.3530850.

J. Makhoul and L. Cosell. LPCW: An LPC vocoder with linear predictive spectral warping. In *ICASSP '76. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 466–469, 1976. doi: 10.1109/ICASSP.1976.1170013.

Marco Matassoni, Seraphina Fong, and Alessio Brutti. Speaker anonymization: Disentangling speaker features from pre-trained speech embeddings for voice conversion. *Applied Sciences*, 14(9), 2024. ISSN 2076-3417. URL https://www.mdpi.com/2076-3417/14/9/3876.

Ambuj Mehrish, Navonil Majumder, Rishabh Bharadwaj, Rada Mihalcea, and Soujanya Poria. A review of deep learning techniques for speech processing. *Information Fusion*, 99:101869, 2023. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2023.101869. URL https://www.sciencedirect.com/science/article/pii/S1566253523001859.

Zhong Meng, Jinyu Li, Zhuo Chen, Yang Zhao, Vadim Mazalov, Yifan Gong, and Biing-Hwang Juang. Speaker-invariant training via adversarial learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5969–5973, 2018. doi: 10.1109/ICASSP.2018.8461932.

Zhong Meng, Yong Zhao, Jinyu Li, and Yifan Gong. Adversarial speaker verification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6216–6220, 2019. doi: 10.1109/ICASSP.2019.8682488.

Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Status Report on Speech Research: A Report on the Status and Progress of Studies on the Nature of Speech, Instrumentation for Its Investigation, and Practical Applications*, 47: 91–103, 1976.

Rafizah Mohd Hanifa, Khalid Isa, and Shamsul Mohamad. A review on speaker recognition: Technology and challenges. *Computers & Electrical Engineering*, 90:107005, 2021. ISSN 0045-7906. doi: https://doi.org/10.1016/j.compeleceng.2021.107005. URL https://www.sciencedirect.com/science/article/pii/S0045790621000318.

Hosein Mohebbi, Grzegorz Chrupała, Willem Zuidema, Afra Alishahi, and Ivan Titov. Disentangling textual and acoustic features of neural speech representations, 2024. URL https://arxiv.org/abs/2410.03037.

Joao Monteiro, Md Jahangir Alam, and Tiago Falk. On the performance of time-pooling strategies for end-to-end spoken language identification. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3566–3572, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.438/.

Geoffrey Stewart Morrison, Ewald Enzinger, Daniel Ramos, Joaquín González-Rodríguez, and Alicia Lozano-Díez. Statistical models in forensic voice comparison. In *Handbook of Forensic Statistics*, pages 451–497. Chapman and Hall/CRC, 2020.

Sung Hwan Mun, Min Hyun Han, Minchan Kim, Dongjune Lee, and Nam Soo Kim. Disentangled speaker representation learning via mutual information minimization. In *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 89–96, 2022. doi: 10.23919/APSIPAASC55919.2022.9979966.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A large-scale speaker identification dataset. In *Interspeech 2017*, pages 2616–2620, 2017. doi: 10.21437/Interspeech.2017-950.

Arsha Nagrani, Joon Son Chung, Samuel Albanie, and Andrew Zisserman. Disentangled speech embeddings using cross-modal self-supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6829–6833, 2020a. doi: 10.1109/ICASSP40776.2020.9054057.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027, 2020b. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2019.101027. URL https://www.sciencedirect.com/science/article/pii/S0885230819302712.

Kihyun Nam, Youkyum Kim, Jaesung Huh, Hee-Soo Heo, Jee weon Jung, and Joon Son Chung. Disentangled representation learning for multilingual speaker recognition. In *Interspeech 2023*, pages 5316–5320, 2023. doi: 10.21437/Interspeech.2023-1603.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KJNcAkY8tY4.

Paul-Gauthier Noé, Mohammad Mohammadamini, Driss Matrouf, Titouan Parcollet, Andreas Nautsch, and Jean-François Bonastre. Adversarial disentanglement of speaker representation for attribute-driven privacy preservation. In *Interspeech 2021*, pages 1902–1906, 2021. doi: 10.21437/Interspeech.2021-1712.

Paul-Gauthier Noé, Andreas Nautsch, Driss Matrouf, Pierre-Michel Bousquet, and Jean-François Bonastre. A bridge between features and evidence for binary attribute-driven perfect privacy. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3094–3098, 2022. doi: 10.1109/ICASSP43922.2022.9746114.

Abu Quwsar Ohi, M. F. Mridha, Md. Abdul Hamid, and Muhammad Mostafa Monowar. Deep speaker recognition: Process, progress, and challenges. *IEEE Access*, 9:89619–89643, 2021. doi: 10.1109/ACCESS.2021.3090109.

Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. Attentive statistics pooling for deep speaker embedding. In *Interspeech 2018*, pages 2252–2256, 2018. doi: 10.21437/Interspeech.2018-993.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

Raghavendra Pappagari, Tianzi Wang, Jesus Villalba, Nanxin Chen, and Najim Dehak. X-vectors meet emotions: A study on dependencies between emotion and speaker recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7169–7173, 2020. doi: 10.1109/ICASSP40776.2020.9054317.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Jason Pelecanos and Sridha Sridharan. Feature warping for robust speaker verification. In *Proceedings of 2001 A Speaker Odyssey: The Speaker Recognition Workshop*, pages 213–218. European Speech Communication Association, Crete, Greece, 2001. URL https://eprints.qut.edu.au/10408/.

Raghuveer Peri, Haoqi Li, Krishna Somandepalli, Arindam Jati, and Shrikanth Narayanan. An empirical analysis of information encoded in disentangled neural speaker representations. In *The Speaker and Language Recognition Workshop (Odyssey 2020)*, pages 194–201, 2020a. doi: 10.21437/Odyssey.2020-28.

Raghuveer Peri, Monisankha Pal, Arindam Jati, Krishna Somandepalli, and Shrikanth Narayanan. Robust speaker recognition using unsupervised adversarial invariance. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6614–6618, 2020b. doi: 10.1109/ICASSP40776.2020.9054601.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.420. URL https://aclanthology.org/2020.acl-main.420/.

Simon J.D. Prince and James H. Elder. Probabilistic linear discriminant analysis for inferences about identity. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. doi: 10.1109/ICCV.2007.4409052.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. AutoVC: Zero-shot voice style transfer with only autoencoder loss. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5210–5219. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/qian19c.html.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox. Unsupervised speech decomposition via triple information bottleneck. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7836–7846. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/qian20a.html.

Xiaoyi Qin, Na Li, Weng Chao, Dan Su, and Ming Li. Cross-age speaker verification: Learning age-invariant speaker embeddings. In *Interspeech 2022*, pages 1436–1440, 2022. doi: 10.21437/Interspeech.2022-648.

Leyuan Qu, Taihao Li, Cornelius Weber, Theresa Pekarek-Rosin, Fuji Ren, and Stefan Wermter. Disentangling prosody representations with unsupervised speech reconstruction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:39–54, 2024. doi: 10.1109/TASLP.2023.3320864.

Desh Raj, David Snyder, Daniel Povey, and Sanjeev Khudanpur. Probing the information encoded in x-vectors. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 726–733, 2019. doi: 10.1109/ASRU46091.2019.9003979.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin,

William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. Speechbrain: A general-purpose speech toolkit, 2021. URL https://arxiv.org/abs/2106.04624.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.295. URL https://aclanthology.org/2021.eacl-main.295/.

G. Renisha and T. Jayasree. Cascaded feedforward neural networks for speaker identification using perceptual wavelet based cepstral coefficients. *Journal of Intelligent & Fuzzy Systems*, 37(1):1141–1153, 2019. doi: 10.3233/JIFS-182599. URL https://doi.org/10.3233/JIFS-182599.

Douglas Reynolds. Gaussian mixture models. In Stan Z. Li and Anil Jain, editors, *Encyclopedia of Biometrics*, pages 659–663. Springer US, Boston, MA, 2009. ISBN 978-0-387-73003-5. doi: 10.1007/978-0-387-73003-5_196. URL https://doi.org/10.1007/978-0-387-73003-5_196.

Colleen Richey, Maria A. Barrios, Zeb Armstrong, Chris Bartels, Horacio Franco, Martin Graciarena, Aaron Lawson, Mahesh Kumar Nandwana, Allen Stauffer, Julien van Hout, Paul Gamble, Jeffrey Hetherly, Cory Stephenson, and Karl Ni. Voices obscured in complex environmental settings (VOiCES) corpus. In *Interspeech 2018*, pages 1566–1570, 2018. doi: 10.21437/Interspeech.2018-1454.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Mufan Sang, Yong Zhao, Gang Liu, John H.L. Hansen, and Jian Wu. Improving transformer-based networks with locality for automatic speaker verification. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10096333.

R. Sharma, D. Govind, J. Mishra, A. K. Dubey, K. T. Deepak, and S. R. M. Prasanna. Milestones in speaker recognition. *Artificial Intelligence Review*, 57:58, 2024.

Francesco Sigona and Mirko Grimaldi. Validation of an ECAPA-TDNN system for forensic automatic speaker recognition under case work conditions. *Speech Communication*, 158: 103045, 2024. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2024.103045. URL https://www.sciencedirect.com/science/article/pii/S0167639324000177.

David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A music, speech, and noise corpus, 2015. URL https://arxiv.org/abs/1510.08484.

David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *Interspeech 2017*, pages 999–1003, 2017. doi: 10.21437/Interspeech.2017-620.

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, 2018. doi: 10.1109/ICASSP.2018.8461375.

Dávid Sztahó and Attila Fejes. Effects of language mismatch in automatic forensic voice comparison using deep learning embeddings. *Journal of Forensic Sciences*, 68(3):871–883, 2023. doi: https://doi.org/10.1111/1556-4029.15250. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1556-4029.15250.

Jianwei Tai, Xiaoqi Jia, Qingjia Huang, Weijuan Zhang, Haichao Du, and Shengzhi Zhang. SEEF-ALDR: A speaker embedding enhancement framework via adversarial learning based disentangled representation. In *Proceedings of the 36th Annual Computer Security Applications Conference*, ACSAC '20, page 939–950, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450388580. doi: 10.1145/3427228.3427274. URL https://doi.org/10.1145/3427228.3427274.

Naohiro Tawara, Atsunori Ogawa, Tomoharu Iwata, Marc Delcroix, and Tetsuji Ogawa. Frame-level phoneme-invariant speaker embedding for text-independent speaker recognition on extremely short utterances. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6799–6803, 2020. doi: 10.1109/ICASSP40776.2020.9053871.

Sreenivas Sremath Tirumala, Seyed Reza Shahamiri, Abhimanyu Singh Garhwal, and Ruili Wang. Speaker identification features extraction methods: A systematic review. *Expert Systems with Applications*, 90:250–271, 2017. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2017.08.015. URL https://www.sciencedirect.com/science/article/pii/S0957417417305535.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000. URL https://arxiv.org/abs/physics/0004057.

Andros Tjandra, Ruoming Pang, Yu Zhang, and Shigeki Karita. Unsupervised learning of disentangled speech content and style representation. In *Interspeech 2021*, pages 4089–4093, 2021. doi: 10.21437/Interspeech.2021-1936.

Fuchuan Tong, Siqi Zheng, Haodong Zhou, Xingjia Xie, Qingyang Hong, and Lin Li. Deep representation decomposition for rate-invariant speaker verification. In *The Speaker and Language Recognition Workshop (Odyssey 2022)*, pages 228–232, 2022. doi: 10.21437/Odyssey.2022-32.

Youzhi Tu, Man-Wai Mak, and Jen-Tzung Chien. Variational domain adversarial learning for speaker verification. In *Interspeech 2019*, pages 4315–4319, 2019. doi: 10.21437/Interspeech.2019-2168.

David van der Vloed and Tina Cambier-Langeveld. How we use automatic speaker comparison in forensic practice. *The International Journal of Speech, Language and the Law*, 29(2):201–224, 2023. doi: 10.1558/ijsll.23955. URL https://doi.org/10.1558/ijsll.23955.

Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056, 2014. doi: 10.1109/ICASSP.2014.6854363.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Christophe Veaux, Junichi Yamagishi, and Simon King. The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–4, 2013. doi: 10.1109/ICSDA.2013.6709856.

Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Leibny Paola García-Perera, Fred Richardson, Réda Dehak, Pedro A. Torres-Carrasquillo, and Najim Dehak. State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations. *Computer Speech & Language*, 60:101026, 2020. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2019.101026. URL https://www.sciencedirect.com/science/article/pii/S0885230819302700.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL https://aclanthology.org/2020.emnlp-main.14/.

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, 1989. doi: 10.1109/29.21701.

Andreas Waldis, Yufang Hou, and Iryna Gurevych. Dive into the chasm: Probing the gap between in- and cross-topic generalization. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2197–2214, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.146/.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.80. URL https://aclanthology.org/2021.acl-long.80/.

Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang. UniSpeech: Unified speech representation learning with labeled and unlabeled data. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10937–10947. PMLR, 18–24 Jul 2021b. URL https://proceedings.mlr.press/v139/wang21y.html.

Chunli Wang, Linming Xu, Hongxin Zhu, and Xiaoyang Cheng. Robustness study of speaker recognition based on ECAPA-TDNN-CIFG. *Journal of Computational Methods in Sciences and Engineering*, 24(4-5):3287–3296, 2024a. doi: 10.3233/JCM-247581. URL https://journals.sagepub.com/doi/abs/10.3233/JCM-247581.

DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726, 2018. doi: 10.1109/TASLP.2018.2842159.

Dong Wang. A simulation study on optimal scores for speaker recognition. *EURASIP Journal on Audio, Speech, and Music Processing*, 2020(18), 2020. doi: 10.1186/s13636-020-00183-3. URL https://doi.org/10.1186/s13636-020-00183-3.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018. doi: 10.1109/LSP.2018.2822810.

Jia Wang, Tianhao Lan, Jie Chen, Chengwen Luo, Chao Wu, and Jianqiang Li. Phoneme-aware adaptation with discrepancy minimization and dynamically-classified vector for text-independent speaker verification. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 6737–6745, New York, NY, USA, 2022a. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548240.

Qing Wang, Wei Rao, Pengcheng Guo, and Lei Xie. Adversarial training for multi-domain speaker recognition. In *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5, 2021c. doi: 10.1109/ISCSLP49672.2021.9362053.

Qiongqiong Wang and Kong Aik Lee. Cosine scoring with uncertainty for neural speaker embedding. *IEEE Signal Processing Letters*, 31:845–849, 2024. doi: 10.1109/LSP.2024.3375080.

Qiongqiong Wang, Kong Aik Lee, and Tianchi Liu. Incorporating uncertainty from speaker embedding estimation to speaker verification. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023a. doi: 10.1109/ICASSP49357.2023.10097019.

Shuai Wang, Yanmin Qian, and Kai Yu. What does the speaker embedding encode? In *Interspeech 2017*, pages 1497–1501, 2017. doi: 10.21437/Interspeech.2017-1125.

Shuai Wang, Johan Rohdin, Lukáš Burget, Oldřich Plchot, Yanmin Qian, Kai Yu, and Jan Černocký. On the usage of phonetic information for text-independent speaker embedding extraction. In *Interspeech 2019*, pages 1148–1152, 2019. doi: 10.21437/Interspeech.2019-3036.

Shuai Wang, Zhengyang Chen, Kong Aik Lee, Yanmin Qian, and Haizhou Li. Overview of speaker modeling and its applications: From the lens of deep speaker representation learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:4971–4998, 2024b. doi: 10.1109/TASLP.2024.3492793.

Xin Wang, Chuan Xie, Qiang Wu, Huayi Zhan, and Ying Wu. A novel phoneme-based modeling for text-independent speaker identification. In *Interspeech 2022*, pages 4775–4779, 2022b. doi: 10.21437/Interspeech.2022-617.

Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9677–9696, 2024c. doi: 10.1109/TPAMI.2024.3420937.

Ying Wang, Tim G. J. Rudner, and Andrew G Wilson. Visual explanations of image-text representations via multi-modal information bottleneck attribution. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 16009–16027. Curran Associates, Inc., 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/339caf45a6fa281cae8adc6465343464-Paper-Conference.pdf.

Yuheng Wei, Junzhao Du, Hui Liu, and Zhipeng Zhang. CentriForce: Multiple-domain adaptation for domain-invariant speaker representation learning. *IEEE Signal Processing Letters*, 29:807–811, 2022. doi: 10.1109/LSP.2022.3154237.

Jennifer Williams and Simon King. Disentangling style factors from speaker representations. In *Proceedings Interspeech 2019*, volume 2019-September of *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 3945–3949. ISCA, September 2019. doi: 10.21437/Interspeech.2019-1769.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Similarity analysis of contextual word representation models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.422. URL https://aclanthology.org/2020.acl-main.422/.

Xiaoliang Wu, Chau Luu, Peter Bell, and Ajitha Rajan. Explainable attribute-based speaker verification, 2024. URL https://arxiv.org/abs/2405.19796.

Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu. Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1652–1656, 2019. doi: 10.1109/APSIPAASC47483.2019.9023039.

Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK corpus: English multi-speaker corpus for CSTR Voice Cloning Toolkit (version 0.92), 2019.

Jiadi Yao, Hong Luo, Jun Qi, and Xiao-Lei Zhang. Interpretable spectrum transformation attacks to speaker recognition systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1531–1545, 2024. doi: 10.1109/TASLP.2024.3364100.

Lu Yi and Man-Wai Mak. Disentangled speaker embedding for robust speaker verification. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7662–7666, 2022. doi: 10.1109/ICASSP43922.2022.9747778.

Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot. BUT system description to VoxCeleb speaker recognition challenge 2019, 2019. URL https://arxiv.org/abs/1910.12592.

Chang Zeng, Xiaoxiao Miao, Xin Wang, Erica Cooper, and Junichi Yamagishi. Joint speaker encoder and neural back-end model for fully end-to-end automatic speaker verification with multiple enrollment utterances. *Computer Speech & Language*, 86:101619, 2024. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2024.101619. URL https://www.sciencedirect.com/science/article/pii/S0885230824000020.

Haoran Zhang, Yuexian Zou, and Helin Wang. Contrastive self-supervised learning for text-independent speaker verification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6713–6717, 2021. doi: 10.1109/ICASSP39728.2021.9413351.

Jian Zhang, Liang He, Xiaochen Guo, and Jing Ma. A study on visualization of voiceprint feature. In *Interspeech 2023*, pages 2233–2237, 2023. doi: 10.21437/Interspeech.2023-1286.

Jian Zhang, Jing Ma, Xiaochen Guo, Lin Li, and Liang He. A speaker recognition method based on stable learning. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10221–10225, 2024. doi: 10.1109/ICASSP48485.2024.10446329.

Kelly Zhang and Samuel Bowman. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary syntactic task analysis. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi, editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5448. URL https://aclanthology.org/W18-5448/.

Zifeng Zhao, Ding Pan, Junyi Peng, and Rongzhi Gu. Probing deep speaker embeddings for speaker-related tasks, 2022. URL https://arxiv.org/abs/2212.07068.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, June 2016.

Jianfeng Zhou, Tao Jiang, Lin Li, Qingyang Hong, Zhe Wang, and Bingyin Xia. Training multi-task adversarial network for extracting noise-robust speaker embedding. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6196–6200, 2019. doi: 10.1109/ICASSP.2019.8683828.

Tianyan Zhou, Yong Zhao, and Jian Wu. ResNeXt and Res2Net structures for speaker verification. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 301–307, 2021. doi: 10.1109/SLT48900.2021.9383531.

Yunfei Zi and Shengwu Xiong. Short-duration speaker verification by joint filter superposition-based multi-dimensional central difference feature extraction and Res2Block-based bidirectional sampling. *IEEE Transactions on Consumer Electronics*, 70(3):5128–5141, 2024. doi: 10.1109/TCE.2024.3411116.

Appendix A

Appendix

A.1 Similarity analysis

We do experiments using linear centered kernel alignment (LinCKA) (Kornblith et al., 2019; Nguyen et al., 2021), a layer similarity analysis technique, which was used to investigate similar models before by Ashihara et al. (2024). For our experiments, we use our British Isles, VOiCES and LJ Speech training splits. As we do not obtain any reliable conclusions that add much value to our experiments, we leave the similarity analysis out of the thesis.

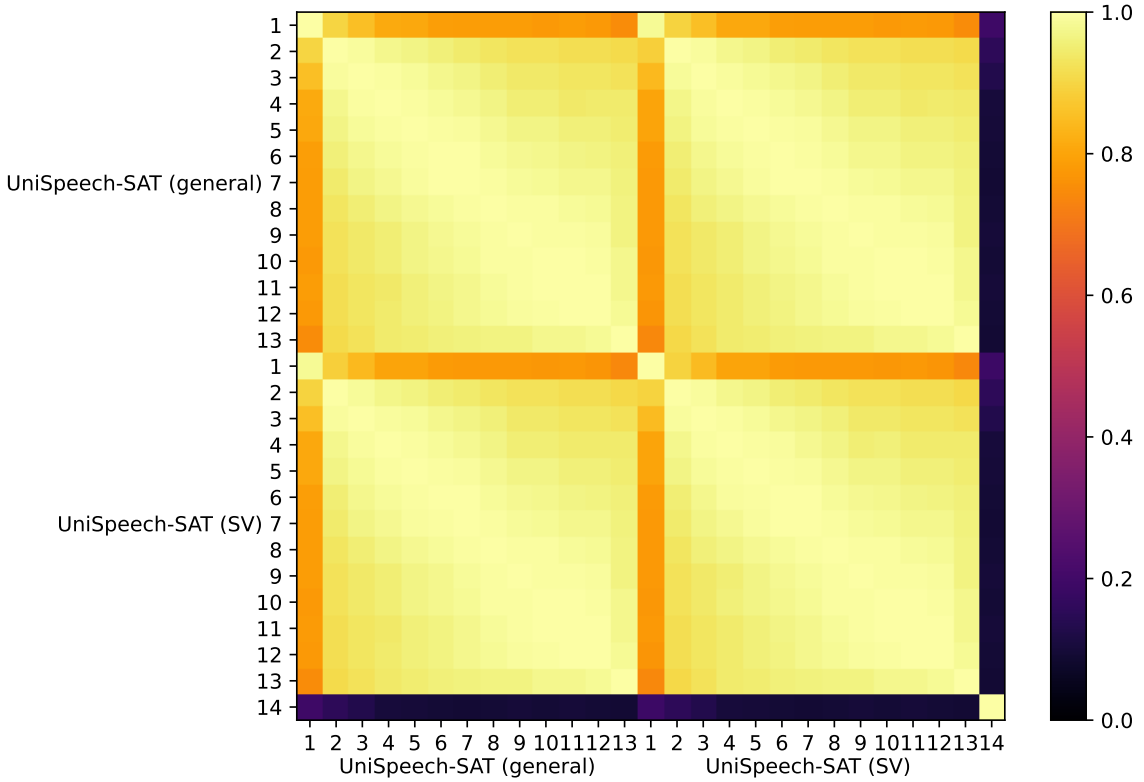
In Figure A.1a we compare both investigated versions of UniSpeech-SAT (results for WavLM look very similar). We see that representations do not change much throughout, except for the first layer being a bit more different, and the speaker embedding in the finetuned version being very different. We also confirm that there is barely any difference between the general and finetuned versions of the model, which is also suggested by the probing results.

In Figure A.1b we compare ECAPA-TDNN and ResNet. We note that for both, the pooled representations (the final two layers) are quite different from the earlier layers. However, for ResNet we do not see as big of a difference between the last two layers as might be expected based on our probing results, which show that the final layer has a great impact on representation of content and channel. We see that within ECAPA-TDNN, layers 1-3 and layers 3-5 are similar to each other, but layers 1 and 2 are fairly different from layers 4 and 5. Layer 3 corresponds to the point just before content representation exceeds channel representation, in our probing experiments. However, we do not see the same pattern in other models, so it is hard to say whether these observations are related. It is noteable that layer 4 and 5 are found to be so similar, despite the latent dimensionality increasing from 512 dimensions to 1536 dimensions per frame between those layers. Finally, we note that layers 1 and 3 of ResNet are fairly similar to layers 1-3 of ECAPA-TDNN, but do not find a likely cause.

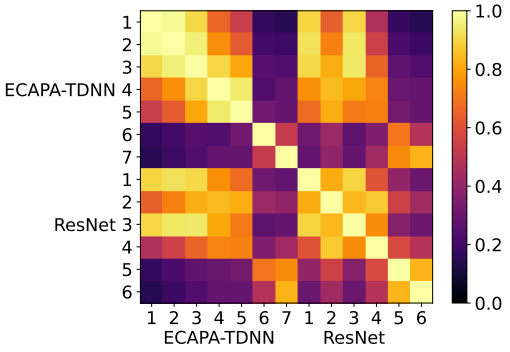
Figure A.1c shows analysis of TitaNet, ECAPA-TDNN and three types of input features. We see that the F-bank features are not similar to any model layers, but the MFCC features are. This is surprising, considering that both models use the F-bank features. TitaNet's pooling layers (5 and 6) do not follow the pattern of ECAPA-TDNN and ResNet. Both layers after pooling are more similar to earlier layers in the model than to each other. In particular, the layer directly after pooling is most similar to the early layers, and the speaker embedding is most similar to the later layers.

A.2 Other probing metrics

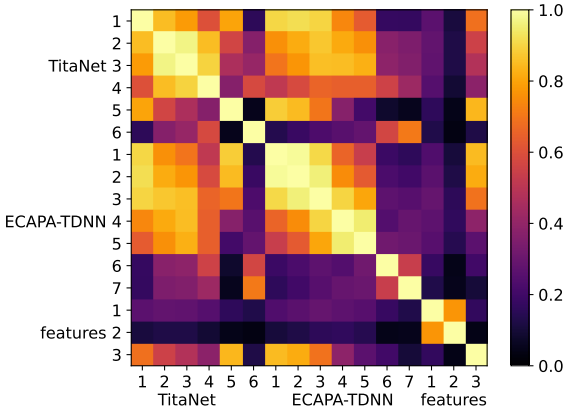
In our interpretability experiments, we report codelength compression of probes trained using MDL probing. Table A.1 shows accuracy of the trained probes. We also run experiments using linear probes, reporting accuracy in Table A.2.



(a) UniSpeech-SAT, general and finetuned.



(b) ECAPA-TDNN and ResNet.



(c) TitaNet, ECAPA-TDNN and input features.

Figure A.1: **Layer similarity plots across different models.** For 'features', layer 1 indicates the spectogram, layer 2 indicates the F-bank as used by ECAPA-TDNN, ResNet and TitaNet, and layer 3 indicates the MFCCs as used by WavLM and UniSpeech-SAT.

Table A.1: **Probing results for speaker, content and channel classification tasks, on the final layers of different models**. Compare Table 5.1. Here, accuracy is reported of the best MLP probe trained on half of the train set, and evaluated on the test set.

(a) **Probing results on a speaker classification task.** Higher accuracy is better, as we want speaker verification models to clearly represent speaker.

Model	Accuracy	Random baseline	Feature baseline
WavLM (general)	0.97 (0.000)	0.38 (0.005)	0.90 (0.003)
WavLM (SV)	**1.00** (0.001)	0.01 (0.006)	0.90 (0.003)
UniSpeech-SAT (general)	0.96 (0.001)	0.33 (0.008)	0.90 (0.003)
UniSpeech-SAT (SV)	**1.00** (0.001)	0.01 (0.001)	0.90 (0.003)
ECAPA-TDNN	**0.99** (0.002)	0.98 (0.005)	0.88 (0.000)
x-vector	**1.00** (0.001)	0.92 (0.004)	0.64 (0.002)
ResNet	**1.00** (0.000)	0.26 (0.003)	0.88 (0.000)
TitaNet	**1.00** (0.000)	-	0.88 (0.000)

(b) **Probing results on a content classification task.** Lower accuracy is better, as we want speaker verification models to disregard content variation.

Model	Accuracy	Random baseline	Feature baseline
WavLM (general)	1.00 (0.000)	0.19 (0.005)	0.13 (0.011)
WavLM (SV)	0.10 (0.012)	0.02 (0.001)	0.13 (0.011)
UniSpeech-SAT (general)	1.00 (0.000)	0.16 (0.003)	0.13 (0.011)
UniSpeech-SAT (SV)	**0.06** (0.005)	0.02 (0.004)	0.13 (0.011)
ECAPA-TDNN	0.31 (0.008)	0.60 (0.020)	0.13 (0.001)
x-vector	0.92 (0.017)	0.73 (0.005)	0.10 (0.004)
ResNet	0.45 (0.002)	0.28 (0.011)	0.13 (0.001)
TitaNet	0.18 (0.001)	-	0.13 (0.001)

(c) **Probing results on a channel classification task.** Lower accuracy is better, as we want speaker verification models to disregard channel variation.

Model	Accuracy	Random baseline	Feature baseline
WavLM (general)	0.73 (0.002)	0.40 (0.011)	0.39 (0.008)
WavLM (SV)	0.35 (0.003)	0.19 (0.002)	0.39 (0.008)
UniSpeech-SAT (general)	0.71 (0.002)	0.36 (0.011)	0.39 (0.008)
UniSpeech-SAT (SV)	**0.30** (0.013)	0.17 (0.001)	0.39 (0.008)
ECAPA-TDNN	0.36 (0.006)	0.65 (0.006)	0.38 (0.000)
x-vector	0.66 (0.019)	0.52 (0.004)	0.24 (0.001)
ResNet	0.46 (0.002)	0.27 (0.000)	0.38 (0.000)
TitaNet	0.47 (0.001)	-	0.38 (0.000)

Table A.2: **Probing results for speaker, content and channel classification tasks, on the final layers of different models**. Compare Table 5.1. Here, accuracy is reported of a linear probe trained on the full train set, and evaluated on the test set.

(a) **Probing results on a speaker classification task.** Higher accuracy is better, as we want speaker verification models to clearly represent speaker.

Model	Accuracy	Random baseline	Feature baseline
WavLM (general)	0.92 (0.009)	0.30 (0.005)	0.87 (0.016)
WavLM (SV)	0.98 (0.004)	0.01 (0.000)	0.87 (0.016)
UniSpeech-SAT (general)	0.91 (0.011)	0.27 (0.007)	0.87 (0.016)
UniSpeech-SAT (SV)	0.96 (0.009)	0.01 (0.000)	0.87 (0.016)
ECAPA-TDNN	**1.00** (0.000)	0.86 (0.027)	0.81 (0.010)
x-vector	**1.00** (0.001)	0.47 (0.097)	0.66 (0.021)
ResNet	**1.00** (0.000)	0.31 (0.006)	0.81 (0.010)
TitaNet	**1.00** (0.001)	-	0.81 (0.010)

(b) **Probing results on a content classification task.** Lower accuracy is better, as we want speaker verification models to disregard content variation.

Model	Accuracy	Random baseline	Feature baseline
WavLM (general)	1.00 (0.000)	0.16 (0.023)	0.14 (0.005)
WavLM (SV)	**0.07** (0.007)	0.02 (0.001)	0.14 (0.005)
UniSpeech-SAT (general)	1.00 (0.000)	0.13 (0.013)	0.14 (0.005)
UniSpeech-SAT (SV)	**0.06** (0.003)	0.02 (0.006)	0.14 (0.005)
ECAPA-TDNN	0.26 (0.006)	0.44 (0.016)	0.16 (0.005)
x-vector	0.86 (0.011)	0.32 (0.017)	0.15 (0.009)
ResNet	0.54 (0.005)	0.30 (0.007)	0.16 (0.005)
TitaNet	0.32 (0.012)	-	0.16 (0.005)

(c) **Probing results on a channel classification task.** Lower accuracy is better, as we want speaker verification models to disregard channel variation.

Model	Accuracy	Random baseline	Feature baseline
WavLM (general)	0.72 (0.009)	0.40 (0.025)	0.44 (0.005)
WavLM (SV)	**0.36** (0.009)	0.13 (0.034)	0.44 (0.005)
UniSpeech-SAT (general)	0.72 (0.011)	0.37 (0.010)	0.44 (0.005)
UniSpeech-SAT (SV)	**0.35** (0.014)	0.15 (0.033)	0.44 (0.005)
ECAPA-TDNN	0.40 (0.004)	0.64 (0.012)	0.49 (0.007)
x-vector	0.72 (0.003)	0.57 (0.046)	0.30 (0.004)
ResNet	0.47 (0.003)	0.29 (0.002)	0.49 (0.007)
TitaNet	0.46 (0.007)	-	0.49 (0.007)

A.3 Training hyperparameters

In our experiments, we use AdamW (Loshchilov and Hutter, 2019) as our optimiser, with learning rate 0.001. This learning rate is found to give the best results on stage 1 disentanglement of WavLM, compared to 0.01 and 0.0001. While MDL probing should be robust against hyperparameter choices (Voita and Titov, 2020), we do experiment with different batch sizes, because full batch probing gave weak final classifiers, probably because of the removed stochasticity. We find a batch size of 128 to achieve the highest compression for training a probe to predict speaker on WavLM (general) embeddings, and use this batch size in all our experiments.

In our disentanglement experiments, we have separately tuned hyperparameters for the different models. For WavLM, we train with β linearly increasing from 0.1 at the start of training to 1 for the last five epochs. We train for 100 epochs, and use a latent dimensionality of 64. For ECAPA-TDNN, we train with β increasing from 0.05 to 0.5. We train for 200 epochs, and use a latent dimensionality of 32. For x-vector, we use the same hyperparameters as for WavLM except for a latent dimensionality of 32.

Experiments were run on an NVIDIA A100 GPU, and implemented using the PyTorch library (Paszke et al., 2019).