Self-supervised Neural Factor Analysis for Disentangling Utterance-level Speech Representations

Weiwei Lin [*1] **Chenhang He** [*2] **Man-Wai Mak** [1] **Youzhi Tu** [1]

Abstract

Self-supervised learning (SSL) speech models such as wav2vec and HuBERT have demonstrated state-of-the-art performance on automatic speech recognition (ASR) and proved to be extremely useful in low label-resource settings. However, the success of SSL models has yet to transfer to utterance-level tasks such as speaker, emotion, and language recognition, which still require supervised fine-tuning of the SSL models to obtain good performance. We argue that the problem is caused by the lack of disentangled representations and an utterance-level learning objective for these tasks. Inspired by how HuBERT uses clustering to discover hidden acoustic units, we formulate a factor analysis (FA) model that uses the discovered hidden acoustic units to align the SSL features. The underlying utterance-level representations are disentangled from the content of speech using probabilistic inference on the aligned features. Furthermore, the variational lower bound derived from the FA model provides an utterance-level objective, allowing error gradients to be backpropagated to the Transformer layers to learn highly discriminative acoustic units. When used in conjunction with HuBERT's masked prediction training, our models outperform the current best model, WavLM, on all utterance-level non-semantic tasks on the SUPERB benchmark with only 20% of labeled data.

1. Introduction

Supervised learning has driven the development of speech technologies for two decades. However, annotating speech data is considerably more challenging than other modalities. For example, automatic speech recognition (ASR) and language identification require linguistic knowledge. For speaker and emotion recognition, label ambiguity and human error are hard to avoid. Self-supervised learning (SSL) promises a prospect of learning without labeled datasets. SSL speech models such as wav2vec (Schneider et al., 2019; Baevski et al., 2020b) and HuBERT (Hsu et al., 2021a) have profoundly changed the research landscape of ASR. By training on a large amount of unlabeled speech to learn a general representation and then fine-tuning with a small amount of labeled data, SSL models demonstrated state-of-the-art performance and proved to be very resource efficient in low label-resource settings (Hsu et al., 2021a; Baevski et al., 2020b).

The success of wav2vec and HuBERT attracts researchers to apply SSL to other speech tasks (Wang et al., 2021). For this purpose, Speech processing Universal PERformance Benchmark (SUPERB) for SSL models was proposed in (Yang et al., 2021). The tasks include content-based classifications, such as ASR, phoneme recognition, and intent classification, and utterance-level discriminative tasks, such as speaker recognition, diarization, and emotion recognition. SUPERB focuses on reusability of SSL features. Thus all tasks must share the same SSL model. Only the classification heads are learned using labeled data for a specific task. This encourages learning task-agnostic features for downstream tasks. Recently, a NOn-Semantic Speech benchmark (NOSS) that specifically designed for utterance-level tasks was proposed in (Shor et al., 2020). Using a triplet-loss unsupervised objective, they were able to exceeds the state-of-the-art performance on a number of transfer learning tasks.

Although it has been shown that SSL features can outperform hand-crafted features for almost all tasks (Yang et al., 2021) under the SUPERB protocols, the performance of supervised downstream models are still far behind the fully supervised or find-tuned models in utterance-level tasks, suggesting that directly using the SSL features to train the

[*] Equal contribution [1] Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China. [2] Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China.. Correspondence to: Weiwei Lin <weiwei.lin@connect.polyu.hk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

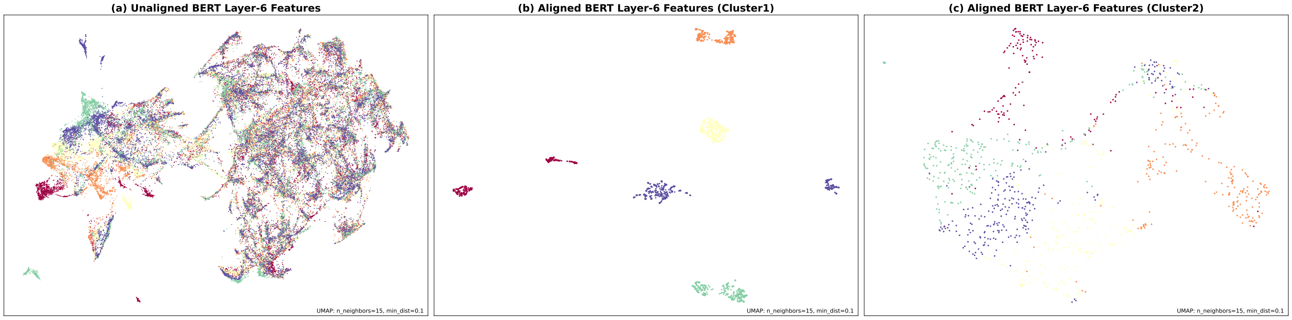

Figure 1. Scatter plots of UMAP embeddings of Transformer features from HuBERT. Different colors represent different speakers. "Aligned" means that the frames were aligned using K-means.

downstream models is not enough. Besides, the labeled datasets in these tasks are considerably large. Using SSL models with little labeled data has yet to be explored for these tasks. This has led us to search for a more appropriate representation and an utterance-level self-supervised learning objective for these tasks.

But, can an SSL model trained for frame-wise discrimination benefits utterance-level discrimination? We believe so. As shown in (Lei et al., 2014), a DNN trained for phoneme classification can be used for training a powerful speaker verification system. The key is in frame alignments. Averaging frame-level features cannot produce a good utterance representation because content variations within an utterance is too structural to be treated as Gaussian. To demonstrate this, we randomly selected 200 recordings from 5 speakers in the LibriSpeech (Panayotov et al., 2015) test set and extracted speech features from the sixth Transformer layer of a HuBERT model. The UMAP (McInnes et al., 2018) embeddings of the features are plotted in Figure 1(a). Different colors in the figure represent different speakers. We cannot see any apparent speaker clusters in Figure 1(a). If the content variations within an utterance are Gaussian, we should see blob-like speaker clusters. One way to reduce content variations is to align frames according to phoneme-like units. However, the existing frame aligners either require supervised learning such as phoneme classification DNNs (Lei et al., 2014) or not amenable to stochastic gradient descent training such as Gaussian mixture models (GMM). Inspired by HuBERT's use of K-means to discover hidden acoustic units, we propose aligning the frames using K-means. To this end, we trained a K-means model with 100 clusters on the LibriSpeech training set and used it to label the test set recordings. Then, we randomly selected two K-means clusters and only kept the frames assigned to these two clusters. The results are presented in Figures 1(b) and (c). As we can see, the speaker clusters are clearly revealed with the help of K-means alignments.

Specifically, we propose using the offline K-means model in HuBERT training to align the speech features. K-means is conceptually simple and amenable to the mini-batch training (Sculley, 2010). During HuBERT training, the K-means model is updated iteratively, which means the aligners can be gradually improved as well. With the K-means aligned features, we then decompose the utterance-level variations into a set of cluster-dependent loading matrices and a compact utterance-level vector. The utterance-level representation can be extracted using probabilistic inference on the aligned features. Finally, instead of using the EM algorithm to train the FA model as in many traditional FA approaches (Dehak et al., 2010), we derived an utterance-level learning objective using the variational lower bound of the data likelihood. This allows gradients to be back-propagated to the Transformer layers to learn more discriminative acoustic features. Our experiments show that this objective can significantly improve the performance of SSL models on utterance-level tasks.

2. Related Work

Self-supervised Learning for Speech The majority of SSL approaches rely on pretext tasks, tasks that are not necessarily the direct objective but learning them can capture a high-level structure in the data (Devlin et al., 2019; Chen et al., 2020; Doersch et al., 2015). In the speech community, some early attempts used multiple tasks as the learning pretexts (Pascual et al., 2019; Ravanelli et al., 2020). An increasingly popular pretext is to use a context encoder to encode information about past frames to predict or reconstruct future frames, as pioneered by contrastive predictive coding (CPC) (Oord et al., 2018). This line of work includes wav2vec (Schneider et al., 2019), which encodes raw waveform to perform frame differentiation, and autoregressive predictive coding (Chung & Glass, 2020) which uses an autoregressive model to predict future frames. Some researchers found that it is helpful to perform the frame

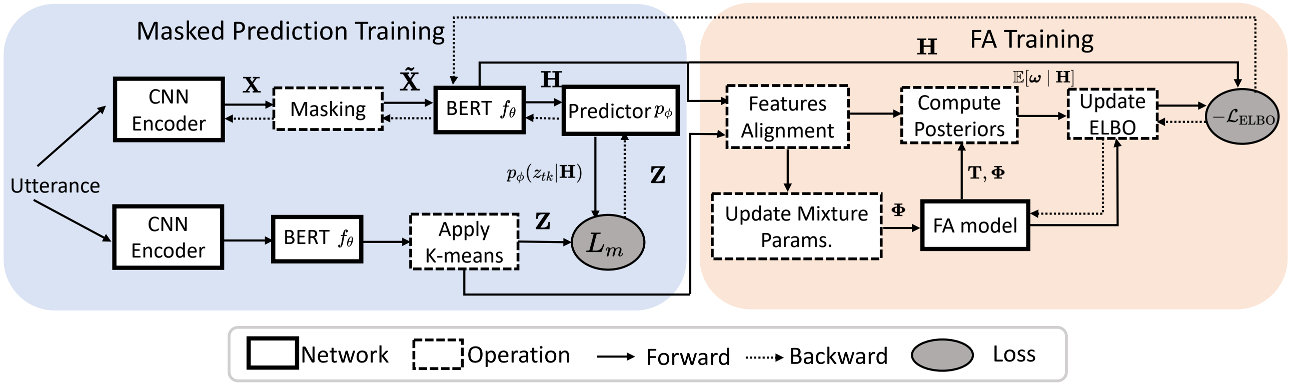

Figure 2. Training of the HuBERT variant of our neural factor analysis model. The dashed arrows represent gradient pathways. For the details of the learning algorithm, the reader may refer to Algorithm 1.

discrimination on quantized representations (Baevski et al., 2020a; Ling et al., 2020). Later, Transformers were used to encode both future and past contexts to perform frame discrimination, as in wav2vec 2.0 (Baevski et al., 2020b) and Mockingjay (Liu et al., 2020).

More recently, the Hidden-Unit BERT (HuBERT) was proposed for self-supervised speech representation learning (Hsu et al., 2021a). Different from explicit frame-wise discrimination in wav2vec and its variants, HuBERT is trained to perform masked prediction of pseudo labels given by an inferior HuBERT model from the previous optimization step. Later, multi-layer masked prediction losses were added to the intermediate layers of HuBERT to further strengthen the representation (Wang et al., 2022). In ContentVec (Qian et al., 2022), the authors improved HuBERT's performance for content-related tasks by disentangling speaker information from content information using voice conversion units. WavLM (Chen et al., 2022), on the other hand, was proposed to improve both content-related tasks and utterance-level tasks by adding utterance mixing during training and gated relative position bias to the Transformer.

Factor Analysis Factor analysis (FA) and probabilistic models in general have wide applications in machine learning (Bishop & Nasrabadi, 2006; Murphy, 2012). Before the advent of deep learning, there had been several successes of FA models in speaker verification, face recognition, and ECG signal classification, including joint-factor analysis (Kenny et al., 2007), probabilistic linear discriminative analysis (Prince & Elder, 2007), and most famously i-vector (Dehak et al., 2010). The FA models generally assume that there is a latent variable responsible for generating the observation vectors. Different relationships between the observation vectors and the latent variable result in different FA models, such as one-to-one mapping between the observation and the latent variable in probabilistic PCA and many observations to one latent variable in i-vector and JFA. Noticeably most of these FA models are applied to raw input or hand-craft features such as natural images or mel-frequency cepstral coefficients (MFCCs). One exception is PLDA in speaker verification, which is applied to neural speaker embeddings or i-vectors.

Utterance-level Speech Tasks Utterance-level speech tasks include speaker recognition (Tu et al., 2022), emotion recognition (Wani et al., 2021), and language identification (Li et al., 2013). They are an important part of intelligent speech systems. Besides their respective applications, they are essential for semantic and generative tasks like ASR and text-to-speech (TTS) synthesis. For example, multilingual ASR and speech translation often require language identification as the first step (Radford et al., 2022). Multi-speaker TTS and voice conversion systems rely on speaker recognition models to extract speaker information (Jia et al., 2018; Qian et al., 2019). Solving these utterance-level tasks often involves different model architectures and domain knowledge.

3. Methodology

In this section, we will introduce our neural factor analysis (NFA) in the context of HuBERT. NFA aims to disentangle utterance-level information such as speaker identity, emotional state, and language from frame-wise content information such as phonemes. Figure 2 shows the training procedure of the HuBERT variant of our NFA model. The learning objective we are about to derive can be used in any SSL model, such as wav2vec and its variants, as long as frame assignments are provided. NFA can learn various utterance-level representations, such as speaker identities, emotion states, and language categories. We will refer to them as utterance-level identities in the remaining paper.

3.1. HuBERT

Consider an acoustic sequence $\mathbf{X}$ of T frames. We denote $\mathcal{M} \subset \{1, \ldots, T\}$ as the index set indicating the frames in $\mathbf{X}$ to be masked. Define $\tilde{\mathbf{X}} = \text{mask}(\mathbf{X}, \mathcal{M})$ as the masked version of $\mathbf{X}$, where the masked $\mathbf{x}_t$ $(t \in \mathcal{M})$ is replaced by a mask embedding. The BERT encoder $f_{\boldsymbol{\theta}}(.)$ takes as input the masked sequence $\tilde{\mathbf{X}}$ and outputs a feature sequence $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_T]$. Let us introduce a K-dimensional binary random variable $\mathbf{y}_t$ for frame t having a 1-of- K representation, where $y_{tk} \in 0, 1$ and $\sum_k y_{tk} = 1$. Denote the output of the predictor as $q_\phi \left(y_{tk} \mid \mathbf{H}\right)$. Given the target distribution for the masked frames $p\left(y_{tk}\right)$, the cross-entropy can be computed as:

$$L_m(\mathbf{H}, \mathcal{M}) = -\sum_{t \in \mathcal{M}} \sum_k p\left(y_{tk}\right) \log q_\phi\left(y_{tk} \mid \mathbf{H}\right) \quad (1)$$

However, we do not have access to the target distribution $p\left(y_{tk}\right)$. HuBERT solves this problem by iterative clustering to obtain the frame label z_{tk} as a surrogate for $p\left(y_{tk}\right)$, where $z_{tk} \in 0, 1$ and $\sum_k z_{tk} = 1$. With the frame label z_{tk}, the cross-entropy loss can be re-written as:

$$L_m(\mathbf{H}, \mathbf{Z}, \mathcal{M}) = -\sum_{t \in \mathcal{M}} \sum_k z_{tk} \log q_\phi\left(y_{tk} \mid \mathbf{H}\right) \quad (2)$$

At first, the cluster assignments are obtained by running *K*-means clustering on MFCCs. Then the model is updated by minimizing the masked prediction loss. New cluster assignments are obtained by running *K*-means on the updated features at the Transformer layer. The learning process then proceeds with new cluster assignments $\{\mathbf{z}_t\}$. The masked prediction and cluster refinement are performed iteratively. The blue area in Figure 2 illustrates HuBERT's masked prediction training.

3.2. Utterance-level Representation Learning via Neural Factor Analysis

Figure 1 shows that the K-means alignments can reveal meaningful speaker information. One simple way to obtain the utterance-level representation is to average the aligned frames in each cluster and concatenate the results. The probabilistic model for such approach can be written as follows:

$$\mathbf{h}_t^i \sim \sum_{k=1}^K z_{tk}^i \mathcal{N}\left(\boldsymbol{\mu}_k + \mathbf{w}_k^i, \boldsymbol{\Sigma}_k\right), \quad (3)$$

where $\mathbf{h}_t^i$ is the Transformer layer features from the utterance i, $z_{tk}^i \in \{0, 1\}$ is the frame label assigned by K-means, $\boldsymbol{\mu}_k$ is the k-th cluster center, $\boldsymbol{\Sigma}_k$ is the covariance matrix of the k-th cluster, and $\mathbf{w}_k^i$ is the utterance identity in the k-th cluster. The concatenation of $\mathbf{w}_k^i$, i.e. $[\mathbf{w}_1^i, \ldots \mathbf{w}_K^i]$, can be used as utterance identity representation. However, its dimension scales linearly with K. Instead, we decompose $\mathbf{w}_k^i$ into the product of a cluster-dependent loading matrix $\mathbf{T}_k$ and utterance identity vector $\boldsymbol{\omega}^i$ for more compact representation:

$$\mathbf{h}_t^i \sim \sum_{k=1}^K z_{tk}^i \mathcal{N}\left(\boldsymbol{\mu}_k + \mathbf{T}_k \boldsymbol{\omega}^i, \boldsymbol{\Sigma}_k\right). \quad (4)$$

Specifically, we train a K-means model using the Transformer layer features to produce $\{\boldsymbol{\mu}_k\}$, which can be viewed as *content representations* of the speech. Then, we run K-means to produce frame labels $\{z_{tk}^i\}$ and calculate $\{\boldsymbol{\Sigma}_k\}$ and cluster weight prior $\{\pi_k\}$ for the K clusters, which we denoted as $\boldsymbol{\Phi} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \,| k = 1, \ldots, K\}$. With cluster parameters and frame labels $\{z_{tk}^i\}$, we only have one set of parameters $\{\mathbf{T}_k\}$ and one latent variable $\boldsymbol{\omega}^i$ left in the model, which is a problem that can be solved with the expectation-maximization (EM) algorithm.

Given a sequence of frame-level features $\mathbf{H}^i = \{\mathbf{h}_1^i, \ldots, \mathbf{h}_T^i\}$, the frames labels (alignments) $\mathbf{Z}^i = \{z_{tk}^i | t = 1, \ldots, T; k = 1, \ldots, K\}$, and cluster parameters $\boldsymbol{\Phi}$, we can use the EM algorithm to find $\mathbf{T} = \{\mathbf{T}_k | k = 1, \ldots, K\}$. In the E-step, we compute the posterior of utterance identity $\boldsymbol{\omega}^i$:

$$p_{\mathbf{T}}\left(\boldsymbol{\omega}^i | \mathbf{H}^i; \mathbf{Z}^i, \boldsymbol{\Phi}\right) = \frac{\prod_{t=1}^T p_{\mathbf{T}}\left(\mathbf{h}_t^i | \boldsymbol{\omega}^i; \mathbf{z}_{t\bullet}^i\right) p\left(\boldsymbol{\omega}^i\right)}{\int \prod_{t=1}^T p_{\mathbf{T}}(\mathbf{h}_t^i | \boldsymbol{\omega}^i; \mathbf{z}_{t\bullet}^i) \mathrm{d}\boldsymbol{\omega}^i}, \quad (5)$$

where $\mathbf{z}_{t\bullet}^i = \{z_{tk}^i\}_{k=1}^K$ and $p_{\mathbf{T}}\left(\boldsymbol{\omega}^i | \mathbf{H}^i; \mathbf{Z}^i, \boldsymbol{\Phi}\right)$ is the probability distribution of $\boldsymbol{\omega}^i$ conditioned on $\mathbf{H}^i$ given $\mathbf{Z}^i$ and $\boldsymbol{\Phi}$. Because the alignments $\mathbf{Z}^i$ and the cluster parameters $\boldsymbol{\Phi}$ are fixed while optimizing the likelihood, we drop the dependency when expressing the posterior for simplicity.

In the M-step, we choose the $\mathbf{T}$ that maximize the expected log-likelihood:

$$\arg\max_{\mathbf{T}} \sum_{i=1}^I \mathbb{E}_{p_{\mathbf{T}'}(\boldsymbol{\omega}^i | \mathbf{H}^i)}\left[\log p_{\mathbf{T}}\left(\mathbf{H}^i, \boldsymbol{\omega}^i\right)\right], \quad (6)$$

where $\mathbf{T}^{'}$ is the loading matrix from the previous M-step (or randomly initialized). Eq. 6 has a closed-form solution. After the matrix $\mathbf{T}$ is found, the mean of the posterior $\mathbb{E}[\boldsymbol{\omega}|\mathbf{H}]$ is used as the utterance identity representation.

$$\mathbb{E}[\boldsymbol{\omega}|\mathbf{H}] = (\mathbf{I} + \sum_k^K \mathbf{T}_k^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1} \mathbf{T}_k)^{-1} \sum_k^K \mathbf{T}_k^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1} \sum_t (\mathbf{h}_t - \boldsymbol{\mu}_k). \quad (7)$$

Learning via gradient on ELBO There are two limitations to learning matrix $\mathbf{T}$ using the EM algorithm. First, the EM algorithm limits the possibility of large-scale training. In Eq. 6, the loading matrix $\mathbf{T}$ is estimated using the

Algorithm 1 Training procedure of the proposed NFA model

```
Initialize: BERT parameters $\boldsymbol{\theta}$, predictor parameters $\phi$, Loading matrix $\mathbf{T}$, Initial cluster labels $\{\mathbf{Z}^i\}_{i=1}^I$.
for $n \leftarrow 0$ to $N$ iterations do
    Input: CNN encoder output $\{\mathbf{X}^i\}_{i=1}^I$, masking index set $\mathcal{M}$.
    if $n > 0$ then
        Run $K$-means on the BERT features to obtain frame labels $\{\mathbf{Z}^i\}_{i=1}^I$
    end if
    Use the alignments $\{\mathbf{Z}^i\}_{i=1}^I$ and Transformer features $\{\mathbf{H}^i\}_{i=1}^I$ to compute cluster parameters $\mathbf{\Phi}$.
    for $i \leftarrow 1$ to $I$ do
        # Forward Pass
        Mask the encoder output $\tilde{\mathbf{X}}^i = \text{mask}(\mathbf{X}^i, \mathcal{M})$.
        Calculate BERT output $\mathbf{H}^i = f_{\boldsymbol{\theta}}(\tilde{\mathbf{X}}^i)$
        Calculate the posteriors of the latent factor (Eq. 5) and use them to update the ELBO $\mathcal{L}_{\text{ELBO}}\left(\mathbf{H}^i; \mathbf{T}\right)$ (Eq. 10).
        # Backward Pass
        Calculate the gradients on cross entropy loss $L_m(\mathbf{H}^i, \mathbf{Z}^i, \mathcal{M})$.
        Calculate the ELBO gradients with respect to $\mathbf{T}$ (Eq. 11).
        Calculate the ELBO gradients with respect to the Transformer parameters $\boldsymbol{\theta}$ (Eq. 13)).
        Update $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\mathbf{T}$ using gradient descent.
    end for
end for
Return $\boldsymbol{\theta}$, $\mathbf{T}$
```

whole training set, contrary to the stochastic update in modern DNN training. Another disadvantage is the separation between the Transformer layers and the FA model during training, which prevents the possibility of joint optimization of the matrix $\mathbf{T}$ and Transformer layers' parameters $\boldsymbol{\theta}$.

We aim to derive a learning rule that is amenable to stochastic updates and allows joint optimization of the FA model and the Transformer layers. As a latent variable model, the log-likelihood of our FA model can be written as (Bishop & Nasrabadi, 2006; Kingma & Welling, 2013):

$$\log p_{\mathbf{T}}\left(\mathbf{H}^i\right) = D_{\text{KL}}\left(q(\boldsymbol{\omega}^i) \| p_{\mathbf{T}}(\boldsymbol{\omega}^i|\mathbf{H}^i)\right) + \mathcal{L}_{\text{ELBO}}\left(\mathbf{H}^i; \mathbf{T}\right), \tag{8}$$

where $\mathcal{L}_{\text{ELBO}}\left(\mathbf{H}^i; \mathbf{T}\right)$ is called the evidence lower bound (ELBO). $D_{\text{KL}}\left(q(\boldsymbol{\omega}^i) \| p_{\mathbf{T}}(\boldsymbol{\omega}^i|\mathbf{H}^i)\right)$ is the KL-divergence between the approximate posterior $q(\boldsymbol{\omega}^i)$ and true posterior $p_{\mathbf{T}}(\boldsymbol{\omega}^i|\mathbf{H}^i)$. Minimizing KL or maximizing the ELBO can both increase the log-likelihood. In the case of our model, minimizing the KL is easy as the posterior of $\boldsymbol{\omega}$ is tractable, which gives rise to the E-step in Eq. 5. To optimize the ELBO, we need to re-write Eq. 8 as:

$$\mathcal{L}_{\text{ELBO}}\left(\mathbf{H}^i; \mathbf{T}\right) = \mathbb{E}_{q(\boldsymbol{\omega}^i)}\left[-\log q(\boldsymbol{\omega}^i) + \log p_{\mathbf{T}}(\mathbf{H}^i, \boldsymbol{\omega}^i)\right]. \tag{9}$$

Because we already know the closest ELBO to likelihood is when $q(\boldsymbol{\omega}^i)$ equals to the posterior $p_{\mathbf{T}}\left(\boldsymbol{\omega}^i \mid \mathbf{H}^i\right)$, Eq. 9 can be written as:

$$\mathbb{E}_{p_{\mathbf{T}'}(\boldsymbol{\omega}^i|\mathbf{H}^i)}\left[-\log p_{\mathbf{T}'}\left(\boldsymbol{\omega}^i \mid \mathbf{H}^i\right) + \log p_{\mathbf{T}}(\mathbf{H}^i, \boldsymbol{\omega}^i)\right], \tag{10}$$

where $\mathbf{T}^{'}$ is the loading matrix from the last update. We can see the first term is a constant with respect to $\mathbf{T}$. Therefore, the gradient of the lower-bound with respect to $\mathbf{T}$ is:

$$\frac{d\mathcal{L}_{\text{ELBO}}}{d\mathbf{T}} = \nabla_{\mathbf{T}} \mathbb{E}_{p_{\mathbf{T}'}(\boldsymbol{\omega}^i|\mathbf{H}^i)}\left[\log p_{\mathbf{T}}\left(\mathbf{H}^i, \boldsymbol{\omega}^i\right)\right]. \tag{11}$$

The gradient with respect to the Transformer features $\frac{d\mathcal{L}_{\text{ELBO}}}{d\mathbf{H}^i}$ involves both terms in Eq. 10:

$$\nabla_{\mathbf{H}^i} \mathbb{E}_{p_{\mathbf{T}'}(\boldsymbol{\omega}^i|\mathbf{H}^i)}\left[-\log p_{\mathbf{T}'}\left(\boldsymbol{\omega}^i \mid \mathbf{H}^i\right) + \log p_{\mathbf{T}}(\mathbf{H}^i, \boldsymbol{\omega}^i)\right]. \tag{12}$$

By applying the chain rule, we can obtain the gradient with respect to the Transformer parameters $\boldsymbol{\theta}$:

$$\frac{d\mathcal{L}_{\text{ELBO}}}{d\boldsymbol{\theta}} = \frac{d\mathcal{L}_{\text{ELBO}}}{d\mathbf{H}^i}\frac{d\mathbf{H}^i}{d\boldsymbol{\theta}}. \tag{13}$$

Eq. 13 shows that we can backpropagate the gradient of ELBO back to the Transformer layers. The total loss of our NFA model is:

$$\sum_i \left(L_m(\mathbf{H}^i, \mathbf{Z}^i, \mathcal{M}) - \lambda \mathcal{L}_{\text{ELBO}}\left(\mathbf{H}^i; \mathbf{T}\right)\right). \tag{14}$$

Therefore, in addition to HuBERT's mask prediction and self-training, in each forward pass, we will compute the posteriors $p_{\mathbf{T}}\left(\boldsymbol{\omega}^i \mid \mathbf{H}^i\right)$ (Eq. 5) given a sequence of BERT features and frame labels produced by K-means. Then, we use the posteriors to evaluate the gradient with respect to $\mathbf{T}$ to update the loading matrix and the gradient with respect to BERT features $\mathbf{H}^i$ to update the SSL model parameters $\boldsymbol{\theta}$. Algorithm 1 summarizes the whole training procedure of our NFA.

4. Experiments

In this section, we will evaluate the proposed NFA model's performance on three kinds of utterance-level speech tasks, namely speaker, emotion, and language recognition, by comparing it to SSL models such as wav2vec2.0, HuBERT, and WavLM. Note that the NFA can use both HuBERT and wav2vec2.0 architecture as long as frame labels are provided.

4.1. Tasks, Datasets, Baselines, and Implementation

Speech Tasks and Datasets The speech tasks that we will evaluate include:

- Automatic speaker verification (ASV or SV), speaker identification (SID), and speaker diarization (SD). We followed the SUPERB protocol (Yang et al., 2021) using the VoxCeleb1 (Nagrani et al., 2017) training split to train the model and used the test split to evaluate speaker verification performance. Note that the reported ASV downstream model in (Yang et al., 2021) is a deep neural network (Snyder et al., 2018) trained on SSL features (Yang et al., 2021). The evaluation metric is equal error rate (EER) (the lower, the better). For speaker identification, we used the VoxCeleb1 train-test split provided by the SUPERB organizer. The evaluation metric is accuracy. For SID, the SUPERB downstream model is a linear classifier trained on averaged SSL features. Speaker diarization is to segment and label a recording according to speakers. We followed the SUPERB protocol using the LibriSpeech (Panayotov et al., 2015) splits for training and evaluation. The SUPERB downstream model is a recurrent neural network. The evaluation metric is diarization error rate (DER) (the lower, the better)

- Emotion recognition (ER). We used IEMOCAP (Busso et al., 2008) dataset. Following the same protocol as SUPERB, we dropped the unbalance emotion classes to leave the neutral, happy, sad, and angry classes. The evaluation metric is accuracy. The SUPERB downstream model is a linear classifier trained on averaged SSL features.

- Language identification (LID). Language identification is not included in the SUPERB benchmark. We included it because it is also an important utterance-level task. The dataset we used is the the Common Language dataset prepared by (Sinisetty et al., 2021), which includes 45 languages with 45.1 hours of recordings. On average, each language has one-hour recordings.[1] The downstream baseline is a linear classifier trained on averaged SSL features.

Pre-trained models The pre-trained models we used in this paper include HuBERT (Hsu et al., 2021a), WavLM (Chen et al., 2022), and wav2vec2-XLS-R (Babu et al., 2022). HuBERT and WavLM models were used in speaker and emotion evaluation. Because language identification requires models trained on multi-lingual data, wav2vec2-XLS-R was used.

Implementation details. The HuBERT and Wav2vec2-based NFA models were trained on LibriSpeech using the model checkpoints provided by fairseq. The language identification NFA models were trained on the Common Language dataset using the XLS-R checkpoint. λ in Eq. 14 is set to 0.01 for all models. After the optimization steps in Algorithm 1 were done, we re-trained the loading matrix $\mathbf{T}$ for each task with EM using unlabeled task-related data. Other than specifically stated, the acoustic features were extracted from layer 6 for the base SSL models (HuBERT, WavLM, and Wav2Vec2-XLS-R) and layer 9 for the large SSL models. The number of clusters in K-means is 100, and the rank of loading matrix dimension is 300 for all NFA models. After utterance-level representations have been extracted using Eq. 7, we used the simple logistic classifier in sklearn (Pedregosa et al., 2011) for SID, ER, and LID. For speaker verification, we used the PLDA backend. For SD, we used linear discriminant analysis (LDA) to reduce the dimension to 200 and then used agglomerative hierarchical clustering to produce speaker assignments. Note that all our downstream methods are linear models.

4.2. SUPERB Experiments

In this section, we evaluate the NFA's performance on SUPERB tasks (Yang et al., 2021; Chen et al., 2022). Besides the standard speaker-related and emotion recognition, we also included language identification (LID) on Common Langue (Sinisetty et al., 2021). For LID, we followed the same protocol as other SUPERB tasks, i.e., the SSL models' weights were frozen, and only linear models were trained with labeled data without data augmentation. To give a better idea of the expected performance of each task in unrestricted settings, we also included the results using the fine-tuned SSL models on the ASV and ER tasks and the current best result in the Common Language dataset reported by other researchers.

The results are presented in Table 1. As observed in the table, NFA significantly outperforms all SSL models across ASV, SD, SID, and LID. NFA performs only marginally worse than the self-supervised Conformer (Shor et al., 2020), which has been specifically designed for utterance-level tasks. In speaker verification, the relative EER reduction is 40% when compared with the WavLM, the previous

[1] https://huggingface.co/datasets/common_language

Table 1. Results on SUPERB and language identification tasks.

Tasks Metrics	ASV EER ↓	SD DER ↓	SID Acc ↑	ER Acc ↑	LID Acc ↑
wav2vec2.0 Large (Yang et al., 2021)	5.65	5.62	86.14	65.64	-
Supervised Finetuning (Wang et al., 2021)	4.46	-	-	64.2	
NFA (wav2vec2-based)	4.02	2.83	96.3	73.4	
HuBERT Large (Yang et al., 2021)	5.98	5.75	90.33	67.62	-
WavLM Large (Chen et al., 2022)	3.77	3.24	95.49	70.62	-
Supervised Finetuning HuBERT Large(Wang et al., 2021)	2.36	-	-	72.7	
NFA (HuBERT-based)	**2.26**	**1.84**	**98.1**	78.1	-
Conformers (Shor et al.)	-	-	-	**79.2**	-
wav2vec2-XLS-R	-	-	-	-	80.4
ECAPA-TDNN	-	-	-	-	84.9
NFA (XLS-R-based)	-	-	-	-	**86.3**

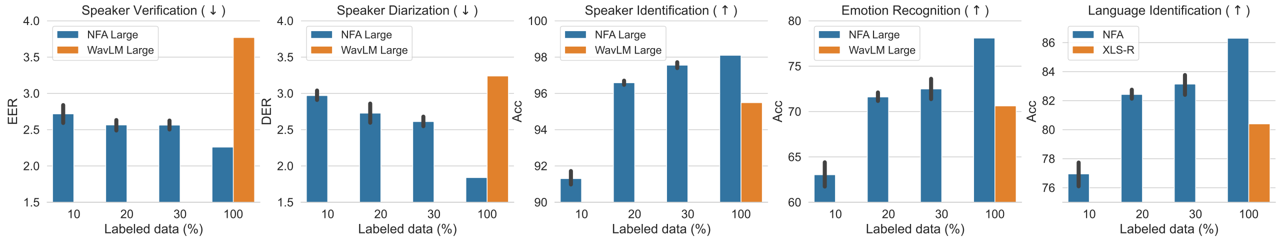

Figure 3. Bar plots of SSL models' performance in low label-resource settings.

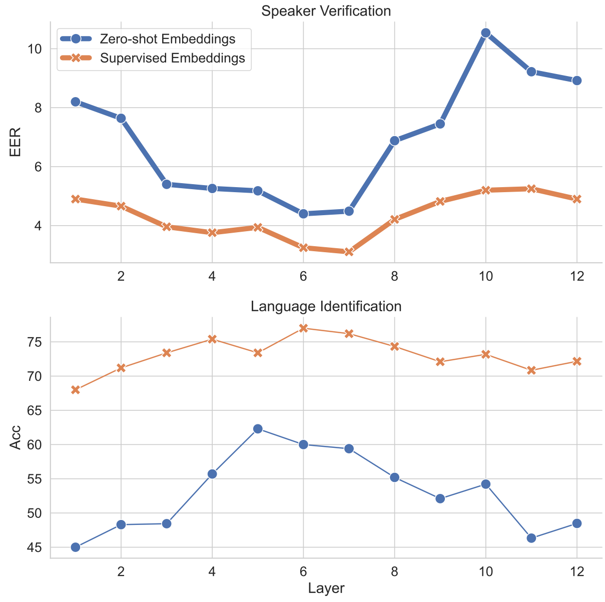

Figure 4. NFA embeddings' zero-shot performance on speaker verification and language ID.

best model on utterance-level tasks. It is worth noting that WavLM's ASV baseline used a DNN network trained on the Transformer features, but we only use linear models. Our models even perform better than the fully fine-tuned models in (Wang et al., 2021) in both ASV and ER tasks. For LID, our XLS-R-based NFA performs better than the best-reported result on Common Language by SpeechBrain (Ravanelli et al., 2021).

4.3. Downstream Low Label-resource Experiments

One of the most attractive features of wav2Vec and HuBERT is their performance on low label-resource ASR. The resource efficiency of these models enables the potential development of many low label-resource languages and speech tasks where labeled data are hard to collect. In this section, we evaluate NFA performance in low label-resource settings. To this end, we divided the labeled dataset in the speaker recognition, emotion recognition, and language identification tasks into 10%, 20%, and 30% subsets as low label-resource settings. For ASV, SD, SID, and ER, we extracted the embeddings from a large Hubert-based NFA model. For LID, we used the embeddings from the XLS-R-based NFA model. WavLM Large and XLS-R were used as performance references. To reduce the performance deviation in the division, we ran each partition five times and reported

Table 2. Zero-shot speaker verification performance on different domains. The metric is the equal error rate.

Dataset	LibriSpeech	VoxCeleb	VOiCES
I-VECTOR	11.2	15.8	22.3
HUBERT	28.7	32.1	34.5
NFA	**3.98**	**9.32**	**12.32**
HUBERT LARGE	30.21	26.88	37.45
NFA LARGE	**2.87**	**7.92**	**12.02**

the results. The loading matrices in the NFA models were trained using the entire unlabeled dataset. The results are presented in Figure 3.

We can see that even with only 10% of labeled data for the downstream models, NFA's performance in ER, SID, and LID is very close to the WavLM and XLS-R. For ASV and SD, our method already outperforms the WavLM models trained on fully labeled data. With 20% labeled data, NFA already outperforms WavLM and XLS-R on all tasks. This shows the high resource efficiency of our NFA models.

4.4. Zero-Shot Speaker Verification

In Figure 1, we observe that by clustering and aligning the Transformer features, speaker information can be revealed. This is all done without labeled data. But how discriminative these unsupervised learned embeddings are? We will evaluate NFA embeddings' zero-shot performance quantitatively in this section. Specifically, we evaluated NFA models on zero-shot speaker verification. After we extracted the utterance-level representations using Eq. 7, we directly used cosine similarity to obtain verification scores without any supervised training (the models were never given speaker information). We evaluated the performance on (1) LibriSpeech, which is considered in-domain data as HuBERT and NFA were trained on this dataset (Panayotov et al., 2015; Hsu et al., 2021a), (2) Voxceleb1-test, a popular speaker verification dataset (Nagrani et al., 2017), and (3) VOiCES (Nandwana et al., 2019), a dataset used to evaluated speaker verification robustness against noise and room reverberation. As a comparison, we also included i-vector (Dehak et al., 2010) and averaged Transformer features (HuBERT rows in Table 2) as baselines.

The results are presented in Table 2. Without supervision, simple averaging the Transformer features cannot produce useful speaker representations. It even performs worse than i-vector, a non-DNN approach. NFA embeddings, however, achieve an EER of 3.98% on LibriSpeech without any supervised training. This suggests that during self-supervised learning, the model has already learned to differentiate speakers, which also empirically demonstrates that the NFA model can disentangle speaker information from the content information. However, when evaluated on VoxCeleb1 and VOiCES, the performance of zero-shot SV dropped significantly. This may be because VoxCeleb1 and VOiCES are real-world speech datasets containing spontaneous speech and environmental noise. NFA and HuBERT were pre-trained on a read speech dataset. The domain discrepancy in SSL models can have a significant impact on the downstream tasks, as mentioned in (Hsu et al., 2021b). Another interesting observation is that scaling the model size improves the zero-shot SV performance, as shown when using HuBERT Large and NFA large models.

4.5. Layer-wise Representation Evaluation

Because our NFA models show excellent zero-shot performance, we can use them to evaluate the discrimination power from each Transformer layer before supervised learning is applied. We extracted the acoustic features from Layer 1 to Layer 12 of the Transformer in the NFA model to conduct zero-shot speaker verification and language identification. For language identification, we used top-1 accuracy as the metric. Then, we used the labeled data to train an LDA on top of NFA embeddings to compare the results. The results are presented in Figure 4.

The blue lines in Figure 4 show that under zero-shot settings, both speaker and language discriminative abilities increase from Layer 1 up to Layer 6. Then, the features from the deeper layers have poorer performance. This is largely consistent with the supervised baselines (orange lines), with Layer 7 obtaining the lowest speaker verification error and Layer 6 having the highest language identification top-1 accuracy in supervised settings. This shows that our NFA models' zero-shot performance can be a reliable predictor of supervised performance.

4.6. Gradient-based Learning Versus EM

To assess whether gradient-based learning has an edge over the Expectation-Maximization (EM) method, we extracted HuBERT features and separately trained a factor analysis model using EM. The results are displayed in Table 3. We observe that gradient-based optimization consistently outperforms EM-based I-vector trained on HuBERT features. This suggests that jointly training the NFA model with the SSL model can yield more potent feature representations than training the two modules independently.

4.7. Impact on ASR

The ultimate goal of a self-supervised learning (SSL) speech model is to utilize a single backbone model for all downstream tasks. Consequently, it's critical that the NFA model does not compromise performance on content-based tasks

Model Checkpoint	Optimization	ASV (EER) ↓	ER (ACC) ↑	Lang. ID (ACC) ↑
HuBERT-Large	EM	2.54%	73.4%	-
HuBERT-Large-NFA	Gradient	2.26%	78.1%	-
Wav2vec-XLS-R	EM	-	-	83.6%
Wav2vec-XLS-R-NFA	Gradient	-	-	86.3%

Table 3. The performance of gradient-based learning versus EM.

Models	WER
Base HuBERT	6.42
Base NFA	6.31
Large HuBERT	3.62
Large NFA	3.66

Table 4. ASR performance on LibriSpeech clean subset.

such as ASR. To ensure this, we compared the performance of the NFA and the large NFA model against HuBERT on the LibriSpeech clean subset. The results, as shown in Table 4, demonstrate that the NFA and large NFA models perform on par with HuBERT. This confirms that our NFA model does not sacrifice performance on content-based tasks.

5. Conclusions

In this paper, we proposed a novel self-supervised speech model for utterance-level speech tasks. Instead of using frame-wise discrimination loss alone, we introduced an utterance-level learning objective based on factor analysis and feature disentanglement. Through extensive experiments, we demonstrate that our NFA model can significantly improve SSL models' performance on utterance-level discriminative tasks without supervised fine-tuning. The zero-shot and low label-resource experiments also show the data efficiency of our approach, which to the best of our knowledge, has yet been shown for utterance-level tasks. This can significantly benefit the utterance-level speech classification tasks where labeled data is hard to obtain, such as speaker recognition for low label-resource languages (Thanh et al., 2021), depression speech detection (Ma et al., 2016), children speech processing (Shahnawazuddin et al., 2021), speech disorder diagnosis (Alhanai et al., 2017), and classifying intelligibility for disordered speech (Venugopalan et al., 2021).

Our findings also shed some insights into speech SSL learning itself. Currently, the frame-wise discriminative SSL models are often thought of as acoustic unit discovery models. Little has been considered for utterance-level identity discovery such as speaker information in self-supervised learning. As we show in Section 4.4, SSL can perform very well on speaker verification with supervision, which suggests speaker-related information is also discovered during the self-supervised learning stage. This is encouraging as it shows that SSL learning can discover multiple hidden information in the speech that can benefit a wide range of speech tasks.

A significant limitation of the NFA model lies in its performance with out-of-domain data. As observed in Section 4.4, NFA's performance significantly deteriorates when evaluated on out-of-domain data. This observation underscores the persistent challenge of achieving robust zero-shot performance in SSL models. Another limitation of NFA pertains to the types of signals it can effectively disentangle. While the NFA model showcases impressive feature disentanglement capabilities across several utterance-level tasks, it's worth noting that it does not disentangle different types of utterance-level information from one another. For instance, it does not separate speaker information from emotional states. For such nuanced tasks, we continue to rely on downstream models to achieve this level of disentanglement. In future research, we intend to explore methodologies that could disentangle different types of utterance-level information during the self-supervised learning stage.

References

Alhanai, T., Au, R., and Glass, J. Spoken language biomarkers for detecting cognitive impairment. In *Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 409–416, 2017.

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. XLS-R: self-supervised cross-lingual speech representation learning at scale. In *Proc. Interspeech 2022*, pp. 2278–2282, 2022.

Baevski, A., Schneider, S., and Auli, M. vq-wav2vec: Self-supervised learning of discrete speech representations. In *Proc. International Conference on Learning Representations, ICLR*, 2020a.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Proc. Advances in Neural Information Processing Systems*, 2020b.

Bishop, C. M. and Nasrabadi, N. M. *Pattern Recognition and Machine Learning*, volume 4. 2006.

Busso, C., Bulut, M., Lee, C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. IEMOCAP: interactive emotional dyadic motion capture database. *Lang. Resour. Evaluation*, 42(4):335–359, 2008.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., Yu, X., and Wei, F. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE J. Sel. Top. Signal Process.*, 16(6):1505–1518, 2022.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. In *Proc. International Conference on Machine Learning, ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, 2020.

Chung, Y. and Glass, J. R. Generative pre-training for speech with autoregressive predictive coding. In *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 3497–3501, 2020.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2010.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *Proc. International Conference on Computer Vision, ICCV*, pp. 1422–1430, 2015.

Hsu, W., Bolte, B., Tsai, Y. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3451–3460, 2021a.

Hsu, W., Sriram, A., Baevski, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., and Auli, M. Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. In *Proc. Interspeech 2021*, pp. 721–725, 2021b.

Jia, Y., Zhang, Y., Weiss, R. J., Wang, Q., Shen, J., Ren, F., Chen, Z., Nguyen, P., Pang, R., Lopez-Moreno, I., and Wu, Y. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Proc. Advances in Neural Information Processing Systems*, pp. 4485–4495, 2018.

Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Trans. Speech Audio Process.*, 15 (4):1435–1447, 2007.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lei, Y., Scheffer, N., Ferrer, L., and McLaren, M. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1695–1699, 2014.

Li, H., Ma, B., and Lee, K. A. Spoken language recognition: from fundamentals to practice. *Proceedings of the IEEE*, 101(5):1136–1159, 2013.

Ling, S., Liu, Y., Salazar, J., and Kirchhoff, K. Deep contextualized acoustic representations for semi-supervised speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 6429–6433, 2020.

Liu, A. T., Yang, S., Chi, P., Hsu, P., and Lee, H. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 6419–6423, 2020.

Ma, X., Yang, H., Chen, Q., Huang, D., and Wang, Y. Depaudionet: An efficient deep model for audio based depression classification. In *Proc. International Workshop on Audio/Visual Emotion Challenge*, pp. 35–42, 2016.

McInnes, L., Healy, J., Saul, N., and Großberger, L. UMAP: uniform manifold approximation and projection. *J. Open Source Softw.*, 3(29):861, 2018.

Murphy, K. P. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

Nagrani, A., Chung, J. S., and Zisserman, A. Voxceleb: A large-scale speaker identification dataset. In Lacerda, F. (ed.), *Proc. Interspeech*, pp. 2616–2620, 2017.

Nandwana, M. K., Van Hout, J., McLaren, M., Richey, C., Lawson, A., and Barrios, M. A. The voices from a distance challenge 2019 evaluation plan. *arXiv preprint arXiv:1902.10828*, 2019.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 5206–5210, 2015.

Pascual, S., Ravanelli, M., Serrà, J., Bonafonte, A., and Bengio, Y. Learning problem-agnostic speech representations from multiple self-supervised tasks. In Kubin, G. and Kacic, Z. (eds.), *Proc. Interspeech*, pp. 161–165, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Prince, S. J. and Elder, J. H. Probabilistic linear discriminant analysis for inferences about identity. In *Proc. International Conference on Computer Vision*, pp. 1–8, 2007.

Qian, K., Zhang, Y., Chang, S., Yang, X., and Hasegawa-Johnson, M. Autovc: Zero-shot voice style transfer with only autoencoder loss. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proc. International Conference on Machine Learning, ICML*, volume 97, pp. 5210–5219, 2019.

Qian, K., Zhang, Y., Gao, H., Ni, J., Lai, C., Cox, D. D., Hasegawa-Johnson, M., and Chang, S. Contentvec: An improved self-supervised speech representation by disentangling speakers. In *Proc. International Conference on Machine Learning, ICML*, volume 162, pp. 18003–18017, 2022.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. *CoRR*, abs/2212.04356, 2022.

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., and Bengio, Y. Multi-task self-supervised learning for robust speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 6989–6993, 2020.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. wav2vec: Unsupervised pre-training for speech recognition. In Kubin, G. and Kacic, Z. (eds.), *Proc. Interspeech 2019*, pp. 3465–3469, 2019.

Sculley, D. Web-scale k-means clustering. In *Proc. International Conference on World Wide Web*, pp. 1177–1178, 2010.

Shahnawazuddin, S., Ahmad, W., Adiga, N., and Kumar, A. Children's speaker verification in low and zero resource conditions. *Digital Signal Processing*, 116:103115, 2021.

Shor, J., Jansen, A., Han, W., Park, D., and Zhang, Y. Universal paralinguistic speech representations using self-supervised conformers. In *Proc. ICASSP 2022*, pp. 3169–3173.

Shor, J., Jansen, A., Maor, R., Lang, O., Tuval, O., Quitry, F. d. C., Tagliasacchi, M., Shavitt, I., Emanuel, D., and Haviv, Y. Towards learning a universal non-semantic representation of speech. *arXiv preprint arXiv:2002.12764*, 2020.

Sinisetty, G., Ruban, P., Dymov, O., and Ravanelli, M. Commonlanguage, June 2021. URL https://doi.org/10.5281/zenodo.5036977.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. X-vectors: Robust DNN embeddings for speaker recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, 2018.

Thanh, D. V., Viet, T. P., and Thu, T. N. T. Deep speaker verification model for low-resource languages and vietnamese dataset. In *Proc. Pacific Asia Conference on Language, Information and Computation*, pp. 445–454, 2021.

Tu, Y., Lin, W., and Mak, M. A survey on text-dependent and text-independent speaker verification. *IEEE Access*, 10:99038–99049, 2022.

Venugopalan, S., Shor, J., Plakal, M., Tobin, J., Tomanek, K., Green, J. R., and Brenner, M. P. Comparing supervised models and learned speech representations for classifying intelligibility of disordered speech on selected phrases. *arXiv preprint arXiv:2107.03985*, 2021.

Wang, C., Wu, Y., Chen, S., Liu, S., Li, J., Qian, Y., and Yang, Z. Improving self-supervised learning for speech recognition with intermediate layer supervision. In *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 7092–7096, 2022.

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

Wani, T. M., Gunawan, T. S., Qadri, S. A. A., Kartiwi, M., and Ambikairajah, E. A comprehensive review of speech emotion recognition systems. *IEEE Access*, 9: 47795–47814, 2021.

Yang, S., Chi, P., Chuang, Y., Lai, C. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G., Huang, T., Tseng, W., Lee, K., Liu, D., Huang, Z., Dong, S., Li, S., Watanabe, S., Mohamed, A., and Lee, H. SUPERB: speech processing universal performance benchmark. In *Proc. Interspeech 2021*, pp. 1194–1198, 2021.

A. You *can* have an appendix here.

You can have as much text here as you want. The main body must be at most 8 pages long. For the final version, one more page can be added. If you want, you can use an appendix like this one, even using the one-column format.