Pretext Tasks Selection for Multitask Self-Supervised Speech Representation Learning

Salah Zaiem[1], **Titouan Parcollet** [2], **Slim Essid** [1], **Abdel Heba** [3]

[1]Télécom Paris, [2]Avignon Université, [3]IRIT,
zaiemsalah@gmail.com

Abstract

Through solving pretext tasks, self-supervised learning leverages unlabeled data to extract useful latent representations replacing traditional input features in the downstream task. In audio/speech signal processing, a wide range of features where engineered through decades of research efforts. As it turns out, learning to predict such features (a.k.a pseudo-labels) has proven to be a particularly relevant pretext task, leading to useful self-supervised representations which prove to be effective for downstream tasks. However, methods and common practices for combining such pretext tasks for better performance on the downstream task have not been explored and understood properly. In fact, the process relies almost exclusively on a computationally heavy experimental procedure, which becomes intractable with the increase of the number of pretext tasks. This paper introduces a method to select a group of pretext tasks among a set of candidates. The method we propose estimates calibrated weights for the partial losses corresponding to the considered pretext tasks during the self-supervised training process. The experiments conducted on automatic speech recognition, speaker and emotion recognition validate our approach, as the groups selected and weighted with our method perform better than classic baselines, thus facilitating the selection and combination of relevant pseudo-labels for self-supervised representation learning.

1 Introduction

Self-supervised learning (SSL) methods usually rely on a supervision obtained from the data itself through solving specific pretext tasks leveraging the underlying structure of the considered data (Doersch, Gupta, and Efros 2016; Arandjelovic and Zisserman 2018). This technique is used in various domains including image processing (Misra and Maaten 2020; Jing and Tian 2020; Grill et al. 2020), natural language understanding (Chen et al. 2020b; Du et al. 2020; Lan et al. 2019) or speech and audio processing (Baevski et al. 2020; Liu et al. 2020; Jiang et al. 2020). It offers numerous advantages, such as the independence from labeled data, stronger performance on downstream tasks, more robust models and an easier transfer to low-resource setups (*e.g.*, low-resource languages) (Baevski et al. 2020; Jing and Tian 2020).

The numerous existing SSL approaches are characterized by the nature of the pretext tasks they solve. For instance, common techniques include predictive coding (Baevski et al. 2020; Liu et al. 2020; Song et al. 2020; Zhang et al. 2020; Hsu et al. 2021), pseudo-label learning (Pascual et al. 2019; Ravanelli et al. 2020), auto-encoding (Renshaw et al. 2015; Algayres et al. 2020), generative modelling (Khurana et al. 2020) or contrastive learning (Saeed, Grangier, and Zeghidour 2020; Jiang et al. 2020). More precisely, these pretext tasks may be defined through the choice of pretext labels, hereafter referred to as *pseudo-labels*. The automatic extraction of pseudo-labels for SSL (*i.e.* from the data itself) is common in many application domains, such as computer vision (Noroozi and Favaro 2017; Gidaris, Singh, and Komodakis 2018), music processing (Wu et al. 2021) and speech processing (Pascual et al. 2019; Shukla, Petridis, and Pantic 2020), and is commonly referred to as *multitask self supervised learning*. In the specific context of speech processing, the process of designing pseudo-labels may benefit from decades of research in signal processing. For instance, potential candidates are pitch estimators, energy-based features, voicing state and many more.

As demonstrated by (Pascual et al. 2019), multitask speech representation learning is a powerful tool to build representations that are beneficial for a wide range of distinct downstream tasks, by combining different pseudo-labels which "intuitively" correspond to these tasks. Unfortunately, there is no clear understanding of how these pseudo-labels may interact when optimised together, and therefore, no common practice of how to select groups of pseudo-labels to obtain better performance on a known downstream task. As a matter of fact, this design process has been essentially driven by empirical validation. This empirical approach can rapidly become intractable with modern SSL architectures which may contain hundreds of millions or billions of parameters trained on thousands of hours of speech, not to mention the carbon footprint of such pseudo-label searches. For instance, the self-supervised training of a single state-of-the-art large wav2vec 2.0 model (Baevski et al. 2020) on $53.2k$ hours of speech requires 128 GPUs for 5.2 days.

This work aims at providing a clear, efficient and theoretically motivated procedure for pseudo-label group selection and weighting based on conditional independence (CI). The method presented allows one to design ahead of training the most adapted multitask self-supervised speech representation learning model which perfectly suits the considered downstream tasks. Such an approach may also enable researchers to save a substantial amount of time and computation usually devoted to pseudo-label search. Hence, the contributions of this work are fourfold:

1. Introduce a theoretically motivated and computationally efficient method for the selection of pseudo-label *groups* among a set of candidates and with respect to the considered downstream tasks (Sections 3 and 4).
2. Validate empirically the proposed approach with a first model SSL model relying on different sets of pseudo-labels corresponding to the ones obtained for three considered speech tasks. (Sections 5).
3. Extend our method to the SOTA wav2vec 2.0 to enhance its performance (Section 6).
4. Release the code base developed with SpeechBrain (Ravanelli et al. 2021) for replication and to encourage further investigations.[1]

The conducted experiments demonstrate that the proposed method allows for a more intelligent, *i.e.* better informed, pseudo-label group selection for multitask SSL settings. Indeed, we find that the models built with the proposed method obtain a word error rate and an equal error rate, respectively, 31.6% and 27.4% lower than the baseline, without the need for any empirical search.

2 Related works and motivations

SSL recently became a key component to achieve good performance on downstream tasks especially with low-resource setups, either in speech (Baevski et al. 2020; Conneau et al. 2020), natural language processing (Lan et al. 2019; Chen et al. 2020b) or computer vision (Gidaris et al. 2019; Misra and Maaten 2020; Jing and Tian 2020). Due to its very nature, SSL relies on large amounts of unlabeled data used to train large deep neural networks over long periods of time. It it thus crucial to understand properly what makes a good SSL model to lower the amount of computation and time needed to obtain the best downstream performance.

SSL for Speech. Self-supervised learning for speech has recently enabled researchers to reach state-of-the-art results on various speech processing tasks (Fan et al. 2021). The most successful models rely on predictive and contrastive objectives (Baevski et al. 2020; Chung et al. 2019; Saeed, Grangier, and Zeghidour 2020) performing well across the different tasks even in low-resource settings. This led to the design of different benchmarks evaluating the self-supervised representations in different languages (Yang et al. 2021; Evain et al. 2021). However, in contrast to this proposition, these works have not tried to motivate beforehand the choices made in the self-supervision pipeline.

Understanding SSL. A few works have tried to shed some theoretical light on the mainly empirical field of self-supervised learning. Following the different paradigms in SSL, various tracks have been followed to understand what makes for a good self-supervised representation, exploring different approaches (Lee et al. 2020; Arora et al. 2019; Wei et al. 2020). On the one hand, contrastive learning (Oord, Li, and Vinyals 2018; Chen et al. 2020a) has been advocated both theoretically and empirically to achieve a balance in the mutual information between alternative representations of the data, keeping just enough shared information to keep the class-related content (Tschannen et al. 2020; Tian et al. 2020; Bachman, Hjelm, and Buchwalter 2019). In a recent work from (Li et al. 2021), independence testing has been used to produce better transformations in contrastive learning settings for image representations. Predictive learning, on the other hand, requires the model to predict masked elements in sequential data. This technique is powerful on downstream tasks that can be reduced to a masking problem, as suggested by research on language modeling (Saunshi, Malladi, and Arora 2020). However, all these works have been focusing on computer vision or text-related applications, and none of them addressed the multi-tasked self supervision problem.

Multi-task self-supervised learning. While the literature on multi-tasking in self-supervised learning remains scarce, it has been shown in classic supervised learning settings, that through estimates of similarity between tasks or thorough empirical testing, several tasks can take advantage of being solved with a common encoder (Zamir et al. 2018; Dwivedi and Roig 2019; Shafey, Soltau, and Shafran 2019; Chen et al. 2015). More specifically, combining pretext tasks with SSL has been mainly explored in computer vision and speech (Pascual et al. 2019; Ravanelli et al. 2020). Pretext tasks such as Jigsaw (Doersch, Gupta, and Efros 2016), colourisation and rotation (Gidaris, Singh, and Komodakis 2018) have been combined successfully to improve downstream performance (Kim et al. 2018; Shin'ya Yamaguchi et al.). The two closest works to our line of research are from Lee et al. (2020) and Doersch, Zisserman, and Deepmind (2017). The former shows that a theoretical link can be established between conditional independence and an improvement of the performance on the downstream task, while the latter proposes to select layers from a multitask self-supervised encoder according to the pretext task to be solved. However, in both cases, the studies do not offer practical and theoretical solutions to select groups of pseudo-labels to build an adapted SSL model that will perform well on the considered downstream tasks.

Group feature selection. Finally, feature selection, and especially group feature selection is another close and inspiring field given the problem we consider. The relationship and interactions between features have been largely investigated in the supervised learning literature (Guyon and Elisseeff 2003). This led to multiple solutions to the feature group selection problem, including LASSO-based techniques (Yuan and Lin 2006), or multiple kernel formulations (Sonnenburg et al. 2006; Rakotomamonjy et al. 2007). However, these works do not involve any self-supervision, and links between feature selection and self-supervision design and pretext-task selection are yet to be proved. We will further consider these lines of works for concurrent baselines.

With this work, we aim at shortening the process of de-

[1] https://github.com/iclrsubmitter22/iclr_2022_submission

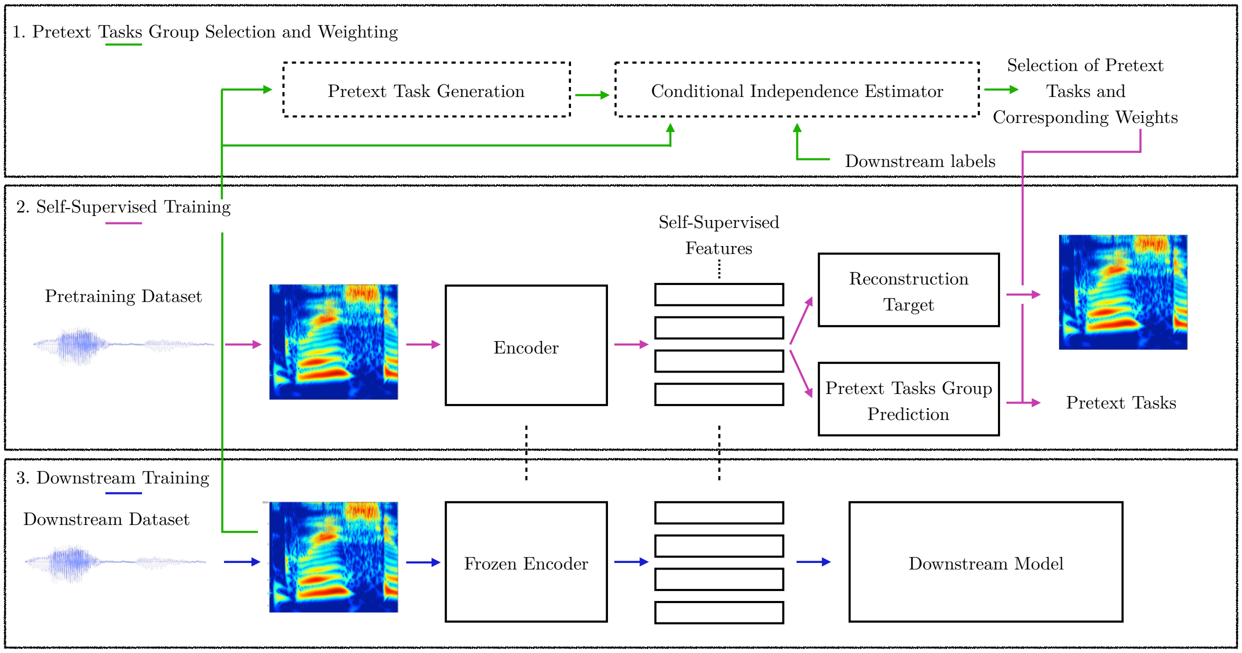

Figure 1: Illustration of the training pipeline. The three steps are depicted: 1. Selecting the group of pseudo-labels and their corresponding weights; 2. SSL training with the selected pretext task; 3. Training on the downstream task with the pretrained SSL model.

signing SSL models while giving insights on the pseudo-label importance and the underlying mechanisms between pretext and downstream tasks at the same time. We decided to experiment with speech due to the lack of literature on this domain for multitask SSL, and for the various pseudo-labels available, which are based on decades of signal processing research. The whole pipeline starting from the acoustic feature extraction to the downstream task scoring follows three major steps summarized in Figure 1. First, for every downstream task, our method produces a pretext task selection and weighting. Then, a SSL model is trained, before being used as a feature extractor front-end in a downstream model to solve the considered task.

3 Conditional independence for utility estimation

As a first step, we require a function that estimates the utility of learning to solve a pretext-task to improve the performance on the downstream task. We use an estimation of the conditional independence between the pseudo-label values and the downstream data points given the downstream labels. Hereafter, we explain the theoretical motivations and describe the computation steps.

3.1 Problem definition and intuition

Let X, Y and Z be, respectively, the downstream data points, their downstream labels and their pseudo-labels. Let also $\mathcal{C}$ be the set of possible downstream classes. As an example, if one considers speaker recognition as a downstream task, X would be the speech samples, Y the speaker IDs, $\mathcal{C}$ the set of unique speaker IDs, and Z a computed signal feature, such as the fundamental frequency.

As stated in Section 2, Lee et al. (2020) linked the utility of a pseudo-label (Z) to the conditional independence (CI) between Z and X given Y. The approach prescribes that, given the labels Y, one may seek to *quantify how much it is possible to predict the pseudo-labels Z without knowing much about X*. The authors bounded, under certain assumptions, the downstream classifier error with a function of the downstream training set size, and a measure of the CI. More precisely, the main theorem shows that the bounding function decreases linearly with the downstream-task dataset size (M) and quadratically with the CI, thus making it a potential estimator for pseudo-label utility.

The proposed function depends on the final downstream task to be solved. This is motivated by two main reasons. First, it can be seen through the large literature on feature selection for various speech or computer vision tasks (Liu et al. 2020; Serizel et al. 2017; Schuller et al. 2007; Wang et al. 2019), that different tasks require the description of different aspects of the data. This suggests that different downstream tasks may perform better after different pre-trainings. A second argument is the difficulty to evaluate representations quality intrinsically, *i.e.* independently from the choice of a particular downstream task. A few metrics and tests (Schatz et al. 2013; Carlin et al. 2011; Lakhotia et al. 2021) have been proposed for speech, but the correlation between these and downstream-task performance has not been clearly identified (Algayres et al. 2020; Gump, Hsu, and Glass 2020).

The principal issue with CI is the difficulty of computing good estimates of how much two variables are independent given a third one on realistic data (Shah and Peters 2018). In a previous work (not cited to respect blind-reviewing process), we proposed a simple way to get an estimation of the conditional independence. This method has proven effective for individual pretext task selection, as the utility estimator correlated highly with the final downstream performances. In our case, the considered pseudo-labels are not independent of the speech samples, as they are signal features. The approach resorts to performing classic independence testing on data sliced by the downstream classes, to check whether this dependence remains given the downstream labels.

3.2 Conditional independence estimator computation

This section details the computation of the conditional independence estimate that is used as a measure of pseudo-label utility. Let $X = \{x_i\}_{i \in \{0,...,M\}}$ with M being the cardinal of X and x_i data samples (*e.g.* Mel-band spectrogram for audio and speech). Every sample x_i has a corresponding downstream label y_i and an automatically generated pseudo-label z_i. We assume that y_i is discrete reducing the task to a classification problem such as with speaker ID for speaker recognition. We also assume that for every pretext-task Z, a single z_i value corresponds to each x_i. In our case, z_i values are the mean of the frame-wise pseudo-label values.

For independence testing, we decided to rely on the kernel-based Hilbert Schmidt Independence Criterion (HSIC) (Gretton et al. 2007) for two reasons. First, HSIC has already proven successful for textual data in testing statistical dependence between translated sentences (Gretton et al. 2007). Second, kernel-based techniques facilitate the handling of multivariate and varying-length data such as speech, as the estimation then boils down to the computation of a similarity measure between the considered variables.

Computation steps. The estimation of the CI of a pseudo-label Z for a downstream task (X, Y) consists of three steps. We start by splitting the data samples X according to the downstream (discrete) classes. Then, we compute for every downstream class $c \in \mathcal{C}$, the kernel matrices K_c and L_c representing the similarity measures for the data samples, and the pseudo-labels, respectively. Finally, we perform the independence test for every split group using K_c and L_c and aggregate the estimates with a mean weighted with the number of samples per downstream class. Thus, for two speech samples x_i and x_j, holding two pseudo-label values z_i and z_j, the coefficients of the similarity matrices K_c and L_c are computed as follows:

$$\begin{aligned} K_{ij} &= K(x_i, x_j) = \cos(GD(x_i), GD(x_j)) \\ L_{ij} &= RBF(z_i, z_j), \end{aligned} \tag{1}$$

with $GD(.)$ the Gaussian Downsampling function (more details in the appendix .7) , $\cos(.,.)$ the cosine similarity, and $RBF(.,.)$ the Radial Basis Function kernel, defined as:

$$\begin{aligned} \cos(x, x') &= \frac{trace(x^T x')}{||x||.||x'||} \\ RBF(z, z') &= \exp(-\frac{||z-z'||^2}{2\sigma^2}) \end{aligned} \tag{2}$$

where σ is the width of the RBF kernel and $trace(.)$ the sum of elements of the main diagonal. Note that we compute the matrices K_c and L_c, for each group of samples sharing the same downstream class $c \in C$. Hence, K_c and L_c correspond to the definitions above, but restricted to the points with c as a downstream label. For each downstream class c, and as in (Gretton et al. 2007), the HSIC value is given by:

$$HSIC_c(X, Z) = \frac{1}{n_c^2} trace(K_c H_c L_c H_c), \tag{3}$$

with $H_c = I_{n_c} - \frac{1}{n_c} 1_{n_c} 1_{n_c}^T$, n_c being the number of points with downstream label c, and 1_{n_c} a vector of ones of size $n_c \times 1$.

The HSIC value is non-negative and corresponds to the Hilbert norm of their cross-covariance matrix. It is used to characterize the independence of the two considered quantities. Intuitively, the HSIC value is high if samples similar in K_c are similar in L_c. Therefore, the lower this value is, the more independent the two arguments of HSIC are and the better the pseudo-label should be for self-supervision before fine-tuning on the downstream class. The final value for a given pseudo-label and a downstream task is expressed as:

$$HSIC(X, Z|Y) = \frac{1}{M} \sum_{c \in \mathcal{C}} HSIC_c(X, Z) \times n_c. \tag{4}$$

4 Pretext task group selection and weighting

While we now are able to predict the utility of every considered pretext task independently, the next step remains to define a clever way to combine them optimally within the same pre-training process. Hence, we introduce a method to select a group of pseudo-labels and weight their respective losses to increase or decrease their importance in the self-supervised representation. Such an optimisation of the latent representation is expected to provide better downstream performance. Our method minimises the conditional dependence between the resulting group pretext task, entailing the prediction of a selected pseudo-label group and the downstream samples given the downstream labels.

Given a set of k possible pseudo-labels $(Z_i)_{i \in [0,k]}$, we seek to estimate the parameters $(\lambda_i)_{i \in [0,k]}$ weighting the loss of every pseudo-label k during the pre-training phase. Hence, we define the pseudo-label Z_λ as an orthogonal concatenation of $(Z_i)_{i \in [0,k]}$ weighted with $(\lambda_i)_{i \in [0,k]}$:

$$Z_\lambda = (\lambda_1 Z_1, ..., \lambda_k Z_k).$$

The custom conditional HSIC computation pipeline described above is fully differentiable with respect to $(\lambda_i)_{i \in [0,k]}$ as proved in appendix .1. In the HSIC computation, the data similarity matrices $\{K_c\}_{c \in C}$ are independent of Z and therefore of λ. Only the pseudo-label similarity matrices $\{L_c\}_{c \in C}$ are changed. For every downstream class c, L_c is defined as:

$$[L_c]_{i,j} = RBF((Z_\lambda)_i, (Z_\lambda)_j)$$
$$= \exp(\frac{-1}{2\sigma^2}\sum_{h=1}^{k}\lambda_h||z_{h,i} - z_{h,j}||_2^2)$$

with $z_{h,i}$ denotes the mean value of the h-th pseudo-label for the i-th data point in the dataset.

4.1 Constraints on the weights

The conditional-independence based utility estimator must be optimized with respect to the weighting parameters $(\lambda_i)_{i\in[0,k]}$ and three constraints.

First, the parameters $(\lambda_i)_{i\in[0,k]}$ must be positive, as they are used as weights for the corresponding losses. A negative weighting loss would lack interpretability as it could imply that the self-supervised model should "unlearn" the corresponding pretext task. This may be the case for adversarial learning methods trying to unlearn harming features, but we are not considering this case in the present work.

Second, the value of the weights must be high enough. Indeed, the presented method for estimating the conditional independence assumes that the considered pseudo-label Z is not independent of X. It is fortunately true for speech features, as Z is a function of X, but not always valid. For instance, with $(\lambda_i)_{i\in[0,k]} = 0$, the utility estimator would be null and thus the lowest (*i.e.* the best), but it would break the assumption of non independence between Z and X. Furthermore, the $HSIC$ value decreases with positive decreasing values of $(\lambda_i)_{i\in[0,k]}$ and we thus need to ensure that the sum of the weights is significantly higher than zero, or it would mean that we are not really doing multi-task learning as the losses of the pseudo-labels would be barely considered.

Finally, for a fair comparison between the weighting choices during the optimization, the sum of the weights should remain constant. In the following, the sum of the $(\lambda_i)_{i\in[0,k]}$ is fixed to one and the optimisation problem can be summarized as follows:

$$\begin{aligned}\min_{\lambda\in R^k} \quad & HSIC(Z_\lambda, X, Y)\\ \text{s.t.} \quad & Z_\lambda = (\lambda_1 Z_1, ..., \lambda_k Z_k),\\ & \lambda_i \geq 0,\ \forall\, i \in [0,k], \textstyle\sum_i \lambda_i = 1.\end{aligned}$$

To minimize the estimator quantity while respecting the constraints, the weights used in the computation of the CI value are the softmax output of free learnable parameters $(W_i)_{i\in[0,k]}$. Softmax enforces the conditions while being differentiable and the problem becomes:

$$\begin{aligned}\min_{W\in R^k} \quad & HSIC(Z_\lambda, X, Y),\\ \text{s.t.} \quad & Z_\lambda = (\lambda_1 Z_1, ..., \lambda_k Z_k),\\ \text{and } & \lambda = Softmax(W)\end{aligned}$$

4.2 Weights sparsity

Another trait that is desirable for the weighting vector is sparsity. If a few pseudo-labels are not needed for the given downstream task, they would rather be discarded than given a low weight. First, this would save computation time including the extraction of the pseudo-labels, and their extraction and prediction during the self-supervised training process. Second, it would help with transparency to understand what features are included or not in the latent space. This sparsity property is also related to feature selection such as with LASSO (Yuan and Lin 2006). To ensure the sparsity of the output weighting vector, while maintaining the desired property of differentiability, we choose the sparsemax function (Martins and Astudillo 2016) to replace the softmax in the second minimization problem.

5 Experimental study

This section details the experiments validating the introduced utility estimator. It describes the selection and weighting processes (Section 5.1), the self-supervised learning models (Section 5.2), the downstream tasks (Section 5.3), and the obtained results (Section 5.4).

5.1 Group selection and weighting

Individual pseudo-labels of interest are obtained with the OpenSmile library (Eyben, Wöllmer, and Schuller 2010). We decided to focus on markers mostly related to prosody and spectral descriptors as the signal processing literature commonly associates them to the three considered downstream tasks (*i.e.* speech, speaker and emotion recognition). Selected pseudo-labels include: *Loudness, F0, Voicing, α Ratio* (Sundberg and Nordenberg 2006), *Zero Crossing Rate, L1 Norm of Rasta Spectrum* (Hermansky et al. 1992), *log of Harmonicity to Noise Ratio* (Murphy and Akande 2005). Then, and according to Figure 1 (*step 1*), we group these pseudo-labels based on their weights, *i.e.* by optimising (4.1) to obtain the λ values associated to each one of them.

Comparative baselines follow common feature group selection strategies or natural intuitions. The first one simply bundles all the pseudo-labels together without any weighting (*i.e.* $\lambda = 1$ for all pseudo-labels) as proposed for PASE (Pascual et al. 2019). As SSL group pretext-task selection is yet to be fully explored, the two other baselines come from the feature selection literature as it represents the closest field with well established techniques. The CI-based pseudo-label selection is thus compared to Recursive Feature Elimination (RFE, (Guyon et al. 2002)) and Maximum Relevance Minimum Redundancy (MRMR, (Peng, Long, and Ding 2005)). The maximum relevance minimum redundancy (MRMR) technique (Peng, Long, and Ding 2005) relies on the Conditional Independence based estimator. It is a close to a naive selection of the best pretext tasks according to the CI based criterion, but it furthermore penalizes the mutual information between the selected pretext tasks. More precisely, we select the group of pseudo-labels $(Z_i)_{i\in[0,p]}$ maximizing :

$$MRMR(Z) = \frac{-1}{p}\sum_{i\in[0,p]} HSIC(X, Z_i|Y) - \frac{1}{\binom{p}{2}}\sum_{i<j} I(Z_i, Z_j)$$

Recursive Feature Elimination (RFE) (Guyon et al. 2002) relies on a classifier that provides information concerning the importance of a given feature in the decision. This classifier is first trained with the whole set of pseudo-labels as features, and the least important feature is eliminated. The process is repeated until only 4 pseudo-labels are kept. We use the scikit-learn implementation with the C-Support Vector Classification as the the classifier providing the feature importance values, with the default hyperparameters.

Table 1: Weights for every pretext-tasks in every considered experiment. With techniques only leading to a selection of pretext tasks (without weights) a unitary weight is assigned for the selected tasks and zero for the non selected. We can see in this table the zeros induced by the Sparsemax function.

Selection	α-zero	F0	Loudness	Audspec Rasta	ZCR	log HNR	Voicing
All	1	1	1	1	1	1	1
VC RFE	1	1	0	0	1	0	1
VC MRMR	1	0	0	1	0	1	0
VC Sparsemax	0.28	0.26	0	0	0	0.4544	0
VC Softmax	0.27	0.11	0.18	0.04	0.06	0.31	0.03
Libri RFE	1	0	0	0	1	1	1
Libri MRMR	0	1	0	1	0	1	1
Libri Sparsemax	0.30	0.37	0	0.06	0	0.27	0
Libri Softmax	0.28	0.47	0.07	0.04	0.02	0.08	0.04
IEMOCAP Sparse.	0.16	0.22	0	0.14	0.12	0.17	0.19

Table 2: Results observed with the proposed selection strategies on the two considered downstream tasks. Word Error Rate (WER) and Equal Error Rate (EER) are expressed in percentage and used for LibriSpeech 100 hours and VoxCeleb1 respectively (*i.e.* lower is better). ASR results are given with and without Language Modeling (LM). All SSL models contain $16.3M$ neural parameters.

Selections	**LibriSpeech** *(WER % ↓)*		**VoxCeleb1** *(EER % ↓)*
	No LM	*LM*	
All	21.98	11.70	11.90
MRMR	18.94	10.36	10.56
RFE	20.02	11.42	11.91
Softmax	**13.17**	**8.00**	9.24
Sparsemax	17.18	10.41	**8.63**

Table 1 shows the workers selected and their corresponding weights for every experiment. Noise (HNR) seems to be the most important information to learn to predict for speaker recognition while fundamental frequency is privileged for ASR.

5.2 Self-supervised training

In the second step of Figure 1, the SSL model learns to predict the selected pseudo-labels. For every one of those, the loss is multiplied by the corresponding assigned weight. Based on previous work conclusions (Ravanelli et al. 2020; Jiang et al. 2020) and apart from the considered pretext task, the network learns to reconstruct the input Mel spectrograms, and to compute 40-dimensional Mel-Frequency Cepstral Coefficients (MFCC) feature vectors. These targets are usually kept to avoid information loss harming heavily downstream performance and are used in all our experiments. For a given pretext-tasks weighting vector $(\lambda_i)_{i\in[0,k]}$, the self-supervised loss is defined as:

$$L_{SSL} = MSE_{mel} + MSE_{mfcc} + \sum\nolimits_{i=1}^{k} \lambda_i \ell_1(Z_i), \quad (5)$$

with MSE the classic mean squared error computed for Mel spectra (MSE_{mel}) and MFCC (MSE_{mfcc}), and $\ell_1(Z)$ the ℓ_1-loss of the pretext task related to pseudo-label Z.

Prior to extending our method to state-of-the-art architectures such as wav2vec 2.0 that are particularly costly to train, we propose to first employ a PASE-like model to empirically validate the approach. Hence, the encoder is composed of three distinct parts: a VGG-like feature extractor (Simonyan and Zisserman 2015), a bidirectional LSTM, and a two-layered dense neural network. All the details of the architecture are given in the appendix .4. Then, and inspired by (Ravanelli et al. 2020), the encoder is followed by simple one-layered predictors voluntarily limited in capacity.

SSL dataset. The SSL model is optimised on the training set of the English Common Voice dataset (version 5.1, 700 hours of training, (Ardila et al. 2020)). Common Voice is a collection of speech utterances from worldwide users recording themselves from their own devices. Hence, the closeness to natural settings makes it a suitable choice for self-supervised learning. 700 hours of speech is a relatively small amount compared to what is generally used for state-of-the-art SSL models. However, we believe it is a sound choice as this is generally greater than what is typically available in SSL use-cases like low-resource languages. We decided to not use the LibriSpeech dataset for pre-training as it is part of our downstream evaluation protocol hence alleviating a strong bias.

5.3 Downstream tasks

Our proposed pseudo-label selection strategy is compared with the two baselines on two different downstream tasks leading to different groups of pseudo-labels: automatic speech recognition (ASR, with LibriSpeech 100 hours) and speaker recognition (SR, with VoxCeleb 1). Datasets and downstream architectures are inspired from the SUPERB benchmark (Yang et al. 2021) for self-supervised learning representations and carefully described in Appendix .5. Prior to downstream training, the SSL model are frozen to be used as a feature extractor with the new pipeline that is task-dependent. We do not use any data augmentation for a pristine comparison of the learned models.

5.4 Results

Baselines are respectively referred to as "*All*", "*RFE*" and "*MRMR*". All the details about the selection and weights are available in table 1. First, it is clear from the results reported in Table 2 that, for the considered downstream tasks, the two introduced strategies (*Sparsemax* and *Softmax*) perform better than the group selection baselines with a gain of 3.28 of EER for *Sparsemax* against the *RFE* approach on VoxCeleb, and 8.81 of WER with *Softmax* compared to the *All* baseline. Interestingly, simply bundling all the pseudo-labels together may lead to poor performance as observed on LibriSpeech with a very high 21.98% of WER obtained. Hence, *intuitively* building sets of labels could be

Table 3: Results observed retraining the Wav2vec2 model with and without weighted pretext tasks using the sparsemax method. “Fr.” and “Fine.” also respectively refer to Frozen and Finetuned settings. Adding selected pretext tasks improves the donwstream performance on all three considered tasks. All models contain $100M$ neural parameters.

Selections	**LibriSpeech** *(WER % ↓)*		**VoxCeleb1** *(EER % ↓)*		**IEMOCAP** *(Acc % ↑)*	
	Fr.	Fine.	Fr.	Fine.	Fr.	Fine.
wav2vec 2.0 *BASE*	17.93	10.21	7.20	5.35	56.6	74.0
wav2vec 2.0 *BASE* + multitask SSL	**16.70**	**9.18**	**6.57**	**5.30**	**59.5**	74.0

harmful for the final representation. This motivates the need for a better pseudo-label selection strategy such as the one introduced in this work, as the WER dropped to 13.17%. As a comparison, the exact same architecture trained with Mel spectra only (*i.e.* no SSL) obtains a WER of 17.3% without LM. Hence, our method even further decreases the WER while being only pretrained with a reasonable amount of data (*i.e.* only 700 hours compared to a few thousands for common SSL techniques (Baevski et al. 2020)). As expected, introducing the joint decoding with a language model strongly decreases the WER but also introduces a bias in our comparison as probabilities are smoothed with a third-party neural model. Nevertheless, and even in this scenario, our weighting strategy outperforms all the baselines. In the context of speaker recognition, *Sparsemax* beats *Softmax* with an EER 0.61 lower.

6 Extending wav2vec 2.0 to multitask SSL

To the best of our knowledge, multi-task speech representation learning has not been scaled to a point where it could represent a state-of-the-art alternative. Contrastive predictive coding (Oord, Li, and Vinyals 2018) based techniques like wav2vec 2.0 (Baevski et al. 2020), on the other hand, currently trust most of the leaderboards for speech-related tasks. Recently, (Sadhu et al. 2021) showed that combining a consistency loss and contrastive predictive coding improves the results of the wav2vec 2.0 architectures in noisy conditions. Following this idea, we propose to further validate our selection method with an extension of wav2vec 2.0 to multitask SSL to demonstrate its scaling capabilities. Hence, the training loss is extended to:

$$L_{SSL} = L_{W2V} + \sum\nolimits_{i=1}^{k} \lambda_i \ell_1(Z_i). \tag{6}$$

We use the standard *BASE* wav2vec 2.0 first described in (Baevski et al. 2020) as a SSL model and train it with the same Common Voice dataset. The pre-training pipeline is implemented within SpeechBrain. The trained *BASE* model has been compared to one obtained with the official Fairseq implementation from (Baevski et al. 2020), and results are strictly equivalent. The entire recipe alongside with the large set of hyperparameters needed to properly train a wav2vec 2.0 model are released in our anonymous repository and will be made available with SpeechBrain afterwards.

To further investigate our methodology, we extend this analysis to emotion recognition (ER) with the IEMOCAP (Busso et al. 2008) dataset, as prosody has shown important in ER literature (Luengo et al. 2005). We follow the SUPERB benchmark conventions (Yang et al. 2021) both at the data and downstream architecture levels. Hence, and conversely to the previous experiments, the ASR system solely optimises the CTC criterion over characters. For each of the three tasks (*i.e.* ASR, SV, ER) we compare the standard *BASE* wav2vec 2.0 model with one trained following the sparsemax selection of multitask SSL. Sparsemax is chosen over softmax because it enforces the sparsity criterion and removes completely a few pseudo-labels from the training, which is one of the objectives of this work. As for the other experiments, the exact weights of each pseudo-label are reported in table 1. Each wav2vec 2.0 model required 24 NVIDIA Tesla V100 GPUs to train for 150 epochs (40 hours). Finally, we also propose to compare frozen and unfrozen (*i.e.* where the wav2vec 2.0 encoder is fine-tuned with the downstream task) SSL models.

It is clear from the results reported in Table 3 that our approach improves the performance over the standard wav2vec 2.0 framework for every considered downstream task. Here, it is worth noting that the difference in performance compared to the literature mostly comes from the training conditions. For instance, wav2vec 2.0 is commonly pre-trained with larger models on LibriSpeech to achieve lower WER on this dataset. We decided to avoid this stick to CommonVoice in the pre-training phase to prevent any biases in the evaluation of our method.

7 Further discussions

Computational efficiency. Efficiency is one of the key motivations of this work, and the gain in time observed with our approach is considerable. The K and L matrices used for the CI estimate are only computed for the downstream datasets. But since these datasets may get bigger and bigger, we can sample among the considered downstream classes to keep the computations tractable and quick. For instance, the CI testing of a considered pretext task (*i.e.* pseudo-labels selection) takes less than half an hour on 20 CPUs whether it be for LibriSpeech or VoxCeleb. This is to be compared to 40 hours of GPU training (*i.e.* 24 Nvidia Tesla V100 for wav2vec 2.0 *BASE*) for a pre-training experiment. The exhaustive search for proper pseudo-label weighting is even more dramatically computationally and energy consuming, if we considered s values per pseudo-label in a grid search involving k pseudo-labels, we would need s^k experiments.

Pseudo-label interactions. We conduct a brief analysis of the evolution of the CI estimate as a function of the weights. To understand the interactions between pseudo-labels, studying the evolution of the CI estimate as a function of the weights shows which pseudo-labels seem interchangeable, which ones are complementary and which ones seem only harmful to the considered downstream task. Figure 2 shows the CI estimates for weighted combinations of groups of three pseudo-labels. As the weights sum up to one, two pretext tasks' values are shown on the x and y axes, while the value of the remaining one, whose name is in the title, is equal to $1 - x - y$. For instance, at the origin point $(0, 0)$, only the third pseudo-label is selected with a weight equal to one, while its weight is equal to zero on the hypotenuse of the right triangle. Figure 2 illustrates that the relationship leading to a lower CI-based utility estimator is not always straightforward. For instance, if we consider the second plot on the second row (*i.e.* α*-ratio, F0, logHNR*), we can see that selecting only one element is always worse than selecting a weighted concatenation, because the areas around the origin and the points $(1, 0)$ and $(0, 1)$ are brighter than the central

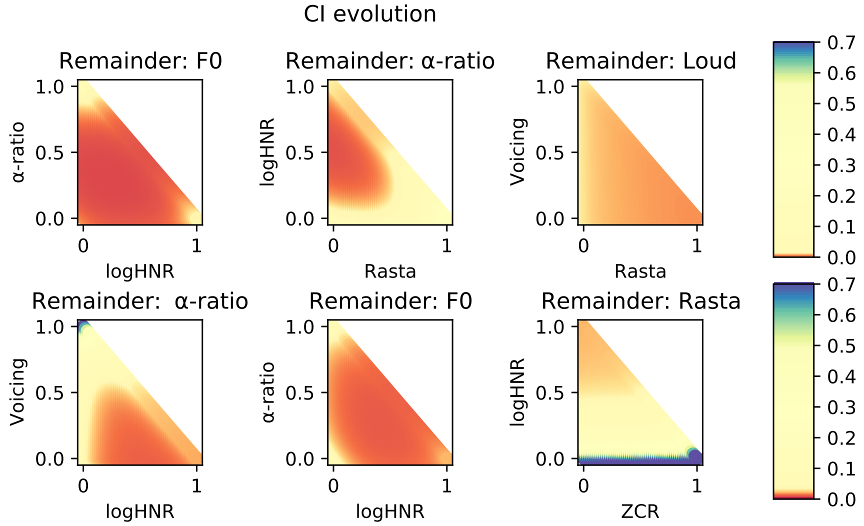

Figure 2: CI-Based utility estimator as a function of the weighting for groups of three pseudo-labels. Top line is for Librispeech, while the bottom one is for VoxCeleb. Three pseudo-labels are presented on every plot, one on the x-axis, one on the y-axis and one that is equal to $1 - x - y$ (hence being called the remainder) and whose name is on the title. Every point in the triangle corresponds to a pretext task that is the weighted combination of the three considered pseudo-labels. For instance, in the top left corner, the point $(0.5, 0.3)$ correspond to the CI value of a pretext task weighting logHNR with 0.5, α-ratio with 0.3 and F0 with 0.2.

area. These results, and especially the two first triangles on every row, suggest that the selection of good sets of pretext-tasks does not only rely on the selection of individually good tasks, but that tasks interfere in a non trivial way.

8 Conclusion

In this work, we introduce a method to quickly and simply combine pseudo-labels into a useful pretext task for multitask self-supervised learning settings. Our approach allows for an optimal selection of pseudo-labels following a cheap optimisation process drastically decreasing the time and compute needed to design the best performing multitask SSL model. Our method is validated on three speech-related downstream tasks and outperforms common pseudo-label selection strategies when combined with simple and state-of-the-art SSL models. This opens a range of possibilities for finding and selecting new pretext tasks in self-supervised learning for speech or other types of data.

9 Acknowledgements

This work has been partly funded by L'Agence de l'innovation de Defense.

References

Algayres, R.; Zaiem, M. S.; Sagot, B.; and Dupoux, E. 2020. Evaluating the reliability of acoustic speech embeddings. In *INTERSPEECH 2020 - Annual Conference of the International Speech Communication Association*. Shanghai / Vitrtual, China.

Arandjelovic, R.; and Zisserman, A. 2018. Objects that Sound. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ardila, R.; Branson, M.; Davis, K.; Henretty, M.; Kohler, M.; Meyer, J.; Morais, R.; Saunders, L.; Tyers, F. M.; and Weber, G. 2020. Common Voice: A Massively-Multilingual Speech Corpus. arXiv:1912.06670.

Arora, S.; Khandeparkar, H.; Khodak, M.; Plevrakis, O.; and Saunshi, N. 2019. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. *36th International Conference on Machine Learning, ICML 2019*, 2019-June: 9904–9923.

Bachman, P.; Hjelm, R. D.; and Buchwalter, W. 2019. *Learning Representations by Maximizing Mutual Information across Views*. Red Hook, NY, USA: Curran Associates Inc.

Baevski, A.; Zhou, H.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Busso, C.; Bulut, M.; Lee, C.-C.; Kazemzadeh, E. A.; Provost, E. M.; Kim, S.; Chang, J. N.; Lee, S.; and Narayanan, S. S. 2008. IEMOCAP: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42: 335–359.

Carlin, M.; Thomas, S.; Jansen, A.; and Hermansky, H. 2011. Rapid Evaluation of Speech Representations for Spoken Term Discovery. 821–824.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A Simple Framework for Contrastive Learning of Visual Representations. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 1597–1607. PMLR.

Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. 2020b. Big Self-Supervised Models are Strong Semi-Supervised Learners.

Chen, Z.; Watanabe, S.; Erdogan, H.; and Hershey, J. 2015. Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In *INTERSPEECH*.

Chung, Y.-A.; Hsu, W.-N.; Tang, H.; and Glass, J. 2019. An Unsupervised Autoregressive Model for Speech Representation Learning. arXiv:1904.03240.

Conneau, A.; Baevski, A.; Collobert, R.; Mohamed, A.; and Auli, M. 2020. Unsupervised Cross-lingual Representation Learning for Speech Recognition. *CoRR*, abs/2006.13979.

Doersch, C.; Gupta, A.; and Efros, A. A. 2016. Unsupervised Visual Representation Learning by Context Prediction. arXiv:1505.05192.

Doersch, C.; Zisserman, A.; and Deepmind. 2017. Multi-task Self-Supervised Visual Learning. Technical report.

Du, J.; Grave, E.; Gunel, B.; Chaudhary, V.; Celebi, O.; Auli, M.; Stoyanov, V.; and Conneau, A. 2020. Self-training Improves Pretraining for Natural Language Understanding. arXiv:2010.02194.

Dwivedi, K.; and Roig, G. 2019. Representation Similarity Analysis for Efficient Task taxonomy & Transfer Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June: 12379–12388.

Evain, S.; Nguyen, H.; Le, H.; Boito, M. Z.; Mdhaffar, S.; Alisamir, S.; Tong, Z.; Tomashenko, N.; Dinarelli, M.; Parcollet, T.; Allauzen, A.; Esteve, Y.; Lecouteux, B.; Portet, F.; Rossato, S.; Ringeval, F.; Schwab, D.; and Besacier, L. 2021. LeBenchmark: A Reproducible Framework for Assessing Self-Supervised Representation Learning from Speech. arXiv:2104.11462.

Eyben, F.; Wöllmer, M.; and Schuller, B. 2010. Opensmile: The Munich Versatile and Fast Open-Source Audio Feature Extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, 1459–1462. New York, NY, USA: Association for Computing Machinery. ISBN 9781605589336.

Fan, Z.; Li, M.; Zhou, S.; and Xu, B. 2021. Exploring wav2vec 2.0 on speaker verification and language identification. arXiv:2012.06185.

Gidaris, S.; Bursuc, A.; Komodakis, N.; Pérez, P.; and Cord, M. 2019. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8059–8068.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised Representation Learning by Predicting Image Rotations. *CoRR*, abs/1803.07728.

Graves, A. 2012. Connectionist temporal classification. In *Supervised Sequence Labelling with Recurrent Neural Networks*, 61–93. Springer.

Gretton, A.; Fukumizu, K.; Teo, C. H.; Song, L.; Schölkopf, B.; and Smola, A. 2007. A Kernel Statistical Test of Independence.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Gump, M.; Hsu, W.-N.; and Glass, J. 2020. Unsupervised Methods for Evaluating Speech Representations.

Guyon, I.; and Elisseeff, A. 2003. An Introduction of Variable and Feature Selection. *J. Machine Learning Research Special Issue on Variable and Feature Selection*, 3: 1157 – 1182.

Guyon, I.; Weston, J.; Barnhill, S.; and Vapnik, V. 2002. Gene Selection for Cancer Classification Using Support Vector Machines. *Machine Learning*, 46: 389–422.

Hermansky, H.; Morgan, N.; Bayya, A.; and Kohn, P. 1992. RASTA-PLP speech analysis technique. volume 1, 121 – 124 vol.1. ISBN 0-7803-0532-9.

Holzenberger, N.; Du, M.; Karadayi, J.; Riad, R.; and Dupoux, E. 2018. Learning Word Embeddings: Unsupervised Methods for Fixed-size Representations of Variable-length Speech Segments. In *Interspeech 2018*, Proceedings of Interspeech 2018. Hyderabad, India: ISCA.

Hsu, W.-N.; Tsai, Y.-H. H.; Bolte, B.; Salakhutdinov, R.; and Mohamed, A. 2021. Hubert: How Much Can a Bad Teacher Benefit ASR Pre-Training? In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6533–6537.

Ioffe, S. 2006. Probabilistic Linear Discriminant Analysis. In Leonardis, A.; Bischof, H.; and Pinz, A., eds., *Computer Vision – ECCV 2006*, 531–542. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-33839-0.

Jiang, D.; Li, W.; Cao, M.; Zhang, R.; Zou, W.; Han, K.; and Li, X. 2020. Speech SIMCLR: Combining Contrastive and Reconstruction Objective for Self-supervised Speech Representation Learning. arXiv:2010.13991.

Jing, L.; and Tian, Y. 2020. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Khurana, S.; Laurent, A.; Hsu, W.-N.; Chorowski, J.; Lancucki, A.; Marxer, R.; and Glass, J. 2020. A Convolutional Deep Markov Model for Unsupervised Speech Representation Learning. arXiv:2006.02547.

Kim, D.; Cho, D.; Yoo, D.; and Kweon, I. S. 2018. Learning Image Representations by Completing Damaged Jigsaw Puzzles. *CoRR*, abs/1802.01880.

Lakhotia, K.; Kharitonov, E.; Hsu, W.-N.; Adi, Y.; Polyak, A.; Bolte, B.; Nguyen, T.-A.; Copet, J.; Baevski, A.; Mohamed, A.; and Dupoux, E. 2021. Generative Spoken Language Modeling from Raw Audio. arXiv:2102.01192.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Lee, J. D.; Lei, Q.; Saunshi, N.; and Zhuo, J. 2020. Predicting What You Already Know Helps: Provable Self-Supervised Learning. arXiv:2008.01064.

Li, Y.; Pogodin, R.; Sutherland, D. J.; and Gretton, A. 2021. Self-Supervised Learning with Kernel Dependence Maximization. arXiv:2106.08320.

Liu, A. T.; Yang, S.-w.; Chi, P.-H.; Hsu, P.-c.; and Lee, H.-y. 2020. Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Luengo, I.; Navas, E.; Hernáez, I.; and Sánchez, J. 2005. Automatic emotion recognition using prosodic parameters. 493–496.

Lüscher, C.; Beck, E.; Irie, K.; Kitza, M.; Michel, W.; Zeyer, A.; Schlüter, R.; and Ney, H. 2019. RWTH ASR Systems for LibriSpeech: Hybrid vs Attention. *Interspeech 2019*.

Martins, A. F. T.; and Astudillo, R. F. 2016. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, 1614–1623. JMLR.org.

McAuliffe, M.; Socolof, M.; Mihuc, S.; Wagner, M.; and Sonderegger, M. 2017. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. 498–502.

Misra, I.; and Maaten, L. v. d. 2020. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6707–6717.

Murphy, P.; and Akande, O. 2005. Cepstrum-Based Harmonics-to-Noise Ratio Measurement in Voiced Speech. In Chollet, G.; Esposito, A.; Faundez-Zanuy, M.; and Marinaro, M., eds., *Nonlinear*

Speech Modeling and Applications, 199–218. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-31886-6.

Nagrani, A.; Chung, J. S.; and Zisserman, A. 2017. VoxCeleb: A Large-Scale Speaker Identification Dataset. *Interspeech 2017*.

Noroozi, M.; and Favaro, P. 2017. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. arXiv:1603.09246.

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

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: An ASR corpus based on public domain audio books. 5206–5210.

Pascual, S.; Ravanelli, M.; Serrà, J.; Bonafonte, A.; and Bengio, Y. 2019. Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks. arXiv:1904.03416.

Peddinti, V.; Povey, D.; and Khudanpur, S. 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In *INTERSPEECH*.

Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8): 1226–1238.

Rakotomamonjy, A.; Bach, F.; Canu, S.; and Grandvalet, Y. 2007. More Efficiency in Multiple Kernel Learning. *Proceedings of the 24th International Con- ference on Machine Learning (ICML)*, 227.

Ravanelli, M.; Parcollet, T.; Rouhe, A.; Plantinga, P.; Rastorgueva, E.; Lugosch, L.; Dawalatabad, N.; Ju-Chieh, C.; Heba, A.; Grondin, F.; Aris, W.; Liao, C.-F.; Cornell, S.; Yeh, S.-L.; Na, H.; Gao, Y.; Fu, S.-W.; Subakan, C.; De Mori, R.; and Bengio, Y. 2021. SpeechBrain. https://github.com/speechbrain/speechbrain.

Ravanelli, M.; Zhong, J.; Pascual, S.; Swietojanski, P.; Monteiro, J.; Trmal, J.; and Bengio, Y. 2020. Multi-task self-supervised learning for Robust Speech Recognition. arXiv:2001.09239.

Renshaw, D.; Kamper, H.; Jansen, A.; and Goldwater, S. 2015. A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge. In *INTERSPEECH*.

Sadhu, S.; He, D.; Huang, C.-W.; Mallidi, S. H.; Wu, M.; Rastrow, A.; Stolcke, A.; Droppo, J.; and Maas, R. 2021. wav2vec-C: A Self-Supervised Model for Speech Representation Learning. In *Proc. Interspeech 2021*, 711–715.

Saeed, A.; Grangier, D.; and Zeghidour, N. 2020. Contrastive Learning of General-Purpose Audio Representations.

Saunshi, N.; Malladi, S.; and Arora, S. 2020. A Mathematical Exploration of Why Language Models Help Solve Downstream Tasks. *CoRR*, abs/2010.03648.

Schatz, T.; Peddinti, V.; Bach, F.; Jansen, A.; Hermansky, H.; and Dupoux, E. 2013. Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline. In *INTERSPEECH 2013 : 14th Annual Conference of the International Speech Communication Association*, 1–5. Lyon, France.

Schuller, B.; Vlasenko, B.; Minguez, R.; Rigoll, G.; and Wendemuth, A. 2007. Comparing one and two-stage acoustic modeling in the recognition of emotion in speech. In *2007 IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, 596–600.

Serizel, R.; Bisot, V.; Essid, S.; and Richard, G. 2017. Acoustic Features for Environmental Sound Analysis. In Virtanen, T.; Plumbley, M. D.; and Ellis, D., eds., *Computational Analysis of Sound Scenes and Events*, 71–101. Springer International Publishing AG.

Shafey, L. E.; Soltau, H.; and Shafran, I. 2019. Joint Speech Recognition and Speaker Diarization via Sequence Transduction. *CoRR*, abs/1907.05337.

Shah, R.; and Peters, J. 2018. The Hardness of Conditional Independence Testing and the Generalised Covariance Measure. *Annals of Statistics*, 48.

Shin'ya Yamaguchi, S.; Kanai, T.; Shioda, S.; Takeda, N.; and Tokyo, J. ???? Multiple Pretext-Task for Self-Supervised Learning via Mixing Multiple Image Transformations. Technical report.

Shukla, A.; Petridis, S.; and Pantic, M. 2020. Learning Speech Representations from Raw Audio by Joint Audiovisual Self-Supervision.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Snyder, D.; Garcia-Romero, D.; and Povey, D. 2015. Time Delay Deep Neural Network-Based Universal Background Models for Speaker Recognition. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 92–97.

Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; and Khudanpur, S. 2018. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5329–5333.

Song, X.; Wang, G.; Wu, Z.; Huang, Y.; Su, D.; Yu, D.; and Meng, H. 2020. Speech-XLNet: Unsupervised Acoustic Model Pretraining For Self-Attention Networks. arXiv:1910.10387.

Sonnenburg, S.; Rätsch, G.; Schäfer, C.; and Schölkopf, B. 2006. Large Scale Multiple Kernel Learning. *J. Mach. Learn. Res.*, 7: 1531–1565.

Sundberg, J.; and Nordenberg, M. 2006. Effects of vocal loudness variation on spectrum balance as reflected by the alpha measure of long-term-average spectra of speech. *The Journal of the Acoustical Society of America*, 120: 453–7.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What Makes for Good Views for Contrastive Learning? In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 6827–6839. Curran Associates, Inc.

Tschannen, M.; Djolonga, J.; Rubenstein, P. K.; Gelly, S.; and Lucic, M. 2020. On Mutual Information Maximization for Representation Learning. In *8th International Conference on Learning Representations (ICLR)*.

Wang, X.; Yu, F.; Wang, R.; Darrell, T.; and Gonzalez, J. E. 2019. TAFE-Net: Task-Aware Feature Embeddings for Low Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wei, C.; Shen, K.; Chen, Y.; and Ma, T. 2020. Theoretical Analysis of Self-Training with Deep Networks on Unlabeled Data. *CoRR*, abs/2010.03622.

Wu, H.-H.; Kao, C.-C.; Tang, Q.; Sun, M.; McFee, B.; Bello, J. P.; and Wang, C. 2021. Multi-Task Self-Supervised Pre-Training for Music Classification. arXiv:2102.03229.

Yang, S.; Chi, P.-H.; Chuang, Y.-S.; Lai, C.-I. J.; Lakhotia, K.; Lin, Y. Y.; Liu, A. T.; Shi, J.; Chang, X.; Lin, G.-T.; Huang, T.-H.; Tseng, W.-C.; tik Lee, K.; Liu, D.-R.; Huang, Z.; Dong, S.; Li, S.-W.; Watanabe, S.; Mohamed, A.; and yi Lee, H. 2021. SUPERB: Speech processing Universal PERformance Benchmark. arXiv:2105.01051.

Yuan, M.; and Lin, Y. 2006. Model Selection and Estimation in Regression With Grouped Variables. *Journal of the Royal Statistical Society Series B*, 68: 49–67.

Zamir, A. R.; Sax, A.; Shen, W. B.; Guibas, L. J.; Malik, J.; and Savarese, S. 2018. Taskonomy: Disentangling Task Transfer Learning. *CoRR*, abs/1804.08328.

Zhang, Y.; Qin, J.; Park, D. S.; Han, W.; Chiu, C.-C.; Pang, R.; Le, Q. V.; and Wu, Y. 2020. Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition.

.1 Differentiability proof

We want to show that the utility estimate is differentiable with respect to the weighting parameters $(\lambda_i)_{i\in[0,k]}$. Since the final estimate is a weighted mean of the in-class independent tests, the problem boils down to showing that within a downstream class c, $HSIC_c(X, Z_\lambda)$ is differentiable. Let us recall the definition of the considered quantities:

$$HSIC_c(X, Z_\lambda) = \frac{1}{n_c^2} trace(K_c H_c L_c H_c) \qquad (7)$$

where K_c and H_c are independent of λ and L_c coefficients are defined as:

$$[L_c]_{i,j} = RBF((Z_\lambda)_i, (Z_\lambda)_j)$$
$$= \exp(\tfrac{-1}{2\sigma^2} \sum_{h=1}^{k} \lambda_h ||z_{h,i} - z_{h,j}||_2^2)$$

Therefore for $p \in [0, k]$:

$$\tfrac{\partial HSIC_c(X,Z_\lambda)}{\partial \lambda_p} = \tfrac{1}{n_c^2} \sum_{i,j} \tfrac{\partial(trace(K_c H_c L_c H_c)}{\partial [L_c]_{i,j}} \tfrac{\partial [L_c]_{i,j}}{\partial \lambda_p}$$
$$= \tfrac{1}{n_c^2} \sum_{i,j} (H_c^T K_c^T H_c^T)_{i,j} \tfrac{-||z_{p,i} - z_{p,j}||_2 \, [L_c]_{i,j}}{2\sigma^2}$$

This allowed us to minimize the conditional-independence based utility estimator according to the weighting values.

.2 Considered signal features and descriptions

Table 4 contains the descriptions of the signal features used as pseudo-labels in this work.

.3 Sparsemax initialization

When initialized with random parameters W, and if one parameter is high enough compared to the other, leading with the Sparsemax function to a weighting value close to 1, we observed that the minimization process falls into local minima selecting only one pseudo-label with weight 1. To avoid this, we initialize all the free parameters W with the same unitary value to which we add some Gaussian noise. Hence, $W_{init} = (1) + N(0, \epsilon)$ with $\epsilon = 0.05$.

.4 Training and architectures details

All the considered audio files are sampled at 16kHz. We feed the SSL models with 80-band Mel spectrograms, with 25ms windows and 10ms stride. To every Mel band corresponds a learned vector of size 256 obtained at the output of the SSL model. So if the input spectrogram is of size (N, 80) with N the number of frames, the representation fed to the downstream pipeline is of size (N, 256). All models including SSL and downstream ones are developed with SpeechBrain (Ravanelli et al. 2021).

Pretraining of the SSL encoder. The encoder is a succession of 2D CNN layers, LSTM layers and a final dense network. This representation is then fed to one dense layer that predict the selected pretext task labels. There are 3 successive CNN blocks containing each 2 CNN layers with kernel size $(3, 3)$ and 128, 200 and 256 channels for each block respectively. No time pooling is performed in order to preserve the input sequence length. 5 bidirectional LSTM layers of size 256 are then stacked. Finally, a MLP with one hidden layer with 256 neurons. The LeakyReLU activation is used across all the layers except for the LSTM. We use a dropout rate of 0.15 during the training. The AdaDelta optimizer is used to update the weights with an initial learning rate of 1.0, $\rho = 0.8$ and $\epsilon = 10^{-8}$. For every experiment, the SSL model is trained for 10 epochs (leading to the convergence of the validation loss).

.5 Downstream trainings : first experiments

Speaker recognition details. VoxCeleb1 (Nagrani, Chung, and Zisserman 2017) is used for the speaker recognition task. The training set contains $148,642$ utterances from $1,251$ different speakers. To compute the conditional independence estimates while limiting the computational load, we restricted ourselves to the utterances of 50 different speakers (the detailed list is given in the released repository). A standard xvector model (Snyder et al. 2018) is trained following the available VoxCeleb SpeechBrain recipe. The extracted speaker embeddings are tested on the enrol and test splits using PLDA (Ioffe 2006) as a similarity metric. Performance is reported in terms of equal error rate (EER). While architecture details are given in appendix .4, it is worth noticing that the whole pipeline is fully integrated to Speechbrain and can thus easily be extended.

We train an embedding model (XVector) until the validation loss converges, on top of the self supervised representations using 5 successive layers of time-delay neural networks (TDNN) (Peddinti, Povey, and Khudanpur 2015). The number of channels is $(512, 512, 512, 512, 1500)$, with kernel sizes of $(5, 3, 3, 1, 1)$ and dilations of $(1, 2, 3, 1, 1)$. The architecture is inspired by successful works on embeddings for speaker recognition (Snyder, Garcia-Romero, and Povey 2015). The learned embeddings are therefore used on a list of pairs of samples to predict whether they are from the same speaker or not. The details of the recipe can be found in the given GitHub repository. We train every embedding model on 10 epochs with an Adam Optimizer starting with a learning rate of 0.001 decaying linearly to 0.0001.

Speech recognition details. ASR is conducted with the 100-hour clean subset of the LibriSpeech dataset (Panayotov et al. 2015) to simulate the low-resource scenario commonly encountered with SSL settings. CI estimations are obtained with word-level alignments from the *Montreal Forced Aligner* (McAuliffe et al. 2017). The ASR pipeline follows the LibriSpeech recipe of SpeechBrain (Ravanelli et al. 2021) and therefore contains a CRDNN encoder (*i.e.* CNN, RNN, DNN) trained jointly with CTC (Graves 2012) and attention (Lüscher et al. 2019) (details in appendix .4). The decoding process is based on beam-search with and without shallow fusion with a pretrained recurrent language model that is publicly available and obtained from SpeechBrain.[2] Performance is expressed in word error rate (WER).

The CRDNN starts with three CNN blocks composed each with 2 2D CNN layers, layer-normalisation and $(2, 2)$ maxpooling along the frequency dimension. The filter dimensions for each block are $64, 100, 100$. Then, maxpooling of 4 is applied on the time dimension to reduce the sequence length before being fed to the RNN. The latter is made of 5 bidirectional LSTM layers of $1,024$ neurons. Finally two dense layers are connected (with batch-normalisation in between). The LeakyReLU activation function is used across all the layers except for the LSTM. A dropout rate of 0.15 is employed with the encoder. The CTC decoder is a simple dense linear layer of size equal to the vocabulary. The vocabulary

[2] https://huggingface.co/speechbrain/asr-crdnn-rnnlm-librispeech

Table 4: Candidate speech pseudo-labels and descriptions.

Feature	Description
Loudness	Intensity & approx. loudness
F0	Fundamental Frequency
Voicing	Voicing Decision
Alpha Ratio (Sundberg and Nordenberg 2006)	Ratio of spectrum intensity % 1000 Hz
Zero Crossing Rate	Zero crossing number per frame
RastaSpec L1Norm	L1 Norm of Rasta Spectrum (Hermansky et al. 1992)
log HNR (Murphy and Akande 2005)	log of Harmonicity to Noise Ratio

is obtained with byte pair encoding or sub-words units (BPE) and is of size $1,000$. The attentional decoder is a one-layered location-aware GRU ($1,024$ neurons). Then, a beam search of depth 60 is applied to obtain the output transcripts. The model is trained for 30 epochs. The learning rate (1.0) is multiplied with a factor of 0.8 every time the validation loss is not decreasing to ensure an optimal convergence of all the models.

SUPERB settings SUPERB (Yang et al. 2021) is a recent benchmark for self-supervised representations of speech data. We use this benchmark for our experiments in combining wav2vec with our selected pretext tasks. We detail here the downstream models as detailed in the benchmark paper :

Emotion Recognition. IEMOCAP (Busso et al. 2008) is used for the Emotion Recognition (ER) task. 4 classes are considered (neutral, happy, sad, angry), and only the audio data is used. The learned representations are mean-pooled then fed to a final linear classifier to compute a cross-entropy loss. We cross-validate on five folds of the standard splits. The result shown is the average of the five attempts. The evaluation metric is accuracy (ACC).

Automatic Speech Recognition For ASR, the decoder is a vanilla 2-layer 1024-unit BLSTM fed with our self-supervised representations and optimized by CTC loss on characters. We use the same language model for decoding as in the first experiments. LibriSpeech Clean-100 only is used for downstream training.

Speaker Recognition The model and the dataset splits used in the first experiment correspond to the SUPERB ones, so we kept the same settings. The results are therefore comparable.

.6 Intuition around the use of Conditional Independence

To get an intuitive understanding of the motivations of this choice, let us consider the example of image classification as the downstream task, and image colourization as the pretext task. In this case, this pretext task would be suited to the downstream one if the final classification label can help implying the colours. For instance, if there are only two classes ”Blue skies” and ”Yellow deserts”, then colourisation is an interesting pretext task, as knowing the final label helps a lot for the pretext task, independently of the image. However, if all the classes share the same colour palette, colourization may not be an interesting task. (In this simple example, we are ignoring the edge detection aspect of colourization, and only focusing on the colour choice part. Obviously the former aspect plays a big part on why the colourization pretext task has been successful.)

Concerning our estimation method, as the pseudo-labels considered in this work are data features, they are indeed functions of the original data samples. This ensures that the data samples are not independent of the pseudo-labels. The idea behind the estimator of conditional independence is that it will test whether this remains true when the considered points share the same downstream class.

.7 Kernels Used for the similarity matrices

The computation of the similarity matrices used in our kernel-based independence test, requires fixed-size embeddings for the data speech samples. These embeddings allow the use of classic kernels on top. However, in the case of sequential data, as it is the case with audio/speech signals, one may want to avoid the additional burden of learning fixed-size embeddings (for possibly variable-length audio sequences). One possible solution to this, which we conveniently exploited in our application to speech data (see Section 5) is the Gaussian Downsampling method (Holzenberger et al. 2018) detailed thereafter. In this instance, after the Mel spectrogram extraction, a speech sample is a sequence of varying length input feature vectors. Therefore, to obtain fixed size embeddings aggregating the input frame-wise Mel spectrum vectors into a fixed number N of input vectors, N being a fixed hyper-parameter, we first divide the sequence into N equal length segments. Then, in each segment, a Gaussian average of the input spectra is computed around the center of the considered segment with the standard deviation σ_{gd} being another hyper-parameter. Denoting by D the dimension of the input frame-wise Mel spectrum vectors, this leads, for any speech excerpt, to a $N \times D$ tensor, without any training procedure. As in the work presenting the gaussian downsampling method (Holzenberger et al. 2018), we set $N = 20$ and $\sigma_{gd} = 0.07$. For the RBF kernel on the pseudo-labels mean value per file, we fixed he RBF kernel width to $\sigma = 0.05$.

.8 Links with Feature Selection

We also studied the link between classic feature selection and pretext task selection through two experiments. The first one was made to check how hard it was to estimate the utility of a pseudo-label, we computed the mutual information between the pseudo-labels and the downstream labels, and checked how much it would correlate with downstream performance. It led to very low correlation values, with even changing signs between VoxCeleb and LibriSpeech. This seems to indicate that Mutual Information is not related directly to self-supervision utility.

The two feature selection baselines considered (MRMR and RFE) perform worse than the proposed techniques. This suggests that despite the apparent similarity, feature selection and self-supervision pretext task design do not necessarily involve the same mechanisms.