Informed Speech Self-supervised Representation Learning

Mohamed Salah Zaiem

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2024IPPAT009

Thèse de doctorat

Informed Speech Self-supervised Representation Learning

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat: Signal, Images, Automatique et robotique

Thèse présentée et soutenue à Palaiseau, le 25 Mars 2024, par

MOHAMED SALAH ZAIEM

Composition du Jury :

Emmanuel Vincent Directeur de Recherches, Inria	Président/Examinateur
Hung-yi Lee Associate Professor, National Taiwan University	Rapporteur
Anthony Larcher Professeur, Le Mans Université	Rapporteur
Karen Livescu Professor, Toyota Technological Institute at Chicago	Examinateur
Shinji Watanabe Associate Professor, Carnegie Mellon University	Examinateur
Hervé Bredin Chargé de Recherches, CNRS	Examinateur
Slim, Essid Professeur, Télécom Paris	Directeur de thèse
Titouan, Parcollet Research Scientist, Samsung AI	Co-directeur de thèse

Abstract

Feature learning has been driving machine learning advancement with the recently proposed methods getting progressively rid of hand-crafted parts within the transformations from inputs to desired labels. The availability of large non-annotated raw audio and speech corpora encouraged the development of approaches exploiting these datasets in the feature learning process. Self-supervised learning has emerged within this context, allowing the processing of unlabeled data towards better performance on low-labeled tasks. It relies upon a wide set of so-called pretext tasks allowing for learning the structure of human speech. Recently, self-supervised learning methods have been building robust representations easily mappable to various desired labels such as phonetic, speaker, or emotion-related identities. However, with recent advancements focusing on scale and breadth-first exploration, the field still lacks informed and motivated best practices during pretraining, evaluation, and downstream usage of self-supervised models.

The first part of this doctoral work is aimed at motivating the choices in the speech self-supervised pipelines learning the unsupervised representations. In this thesis, I first show how conditional-independence-based scoring can be used to efficiently and optimally select pretraining tasks tailored for the best performance on a target task. After developing an estimator of conditional independence for speech data, I show its utility in two settings; first in the selection and weighting of multiple pretext-labels, and second in the view-creation policies in contrastive learning approaches.

The second part of this manuscript studies the evaluation and usage of pretrained self-supervised representations. I explore, first, the robustness of current speech self-supervision benchmarks to changes in the downstream modeling choices, diagnosing efficiency, generalization, and performance issues related to using limited-capacity probes. I, then, design and evaluate methods for the downstream fine-tuning of self-supervised encoders towards two main objectives: generalization, especially to out-of-distribution samples, and inference efficiency.

Keywords: speech processing, deep learning, supervised learning, self-supervised learning.

Résumé

L'apprentissage des caractéristiques a été un des principaux moteurs des progrès de l'apprentissage automatique. Les méthodes récemment proposées se sont débarassées progressivement des caractéristiques non-apprises dans la transformations des entrées en étiquettes souhaitées. La disponibilité de vastes corpus audio et vocaux bruts non annotés a encouragé le développement d'approches exploitant ces ensembles de données dans le processus d'apprentissage des caractéristiques. L'apprentissage auto-supervisé est apparu dans ce contexte, permettant le traitement de données non annotées en vue d'une meilleure performance sur des tâches faiblement étiquetées. Il repose sur un large éventail de tâches dites prétextes permettant d'apprendre la structure de la parole humaine. Ainsi, les méthodes d'apprentissage auto-supervisé ont permis de construire des représentations robustes pouvant facilement être associées à diverses étiquettes souhaitées, telles que le contenu phonétique ou émotionnel ou les identités de locuteur. Toutefois, malgré les progrès récents axés sur la mise à l'échelle, le domaine manque encore de bonnes pratiques informées et motivées lors du pré-entraînement, de l'évaluation et de l'utilisation en aval des modèles auto-supervisés.

La première partie de mon travail de doctorat vise à motiver les choix dans les méthodes d'apprentissage auto-supervisé de la parole qui apprennent les représentations non supervisées. Dans cette thèse, je montre d'abord comment une fonction basée sur l'indépendance conditionnelle peut être utilisée pour sélectionner efficacement et de manière optimale des tâches de pré-entraînement adaptées à la meilleure performance sur une tâche cible. Après avoir développé un estimateur de l'indépendance conditionnelle pour les données vocales, je montre son utilité dans deux contextes : d'abord dans la sélection et la pondération de multiples étiquettes de prétexte, et ensuite dans les méthodes de création de vues dans les approches d'apprentissage contrastif.

La deuxième partie de mon travail de doctorat étudie l'évaluation et l'utilisation de représentations auto-supervisées pré-entraînées. J'explore d'abord la robustesse des benchmarks actuels d'auto-supervision de la parole aux changements dans les choix de modélisation en aval, en diagnostiquant les problèmes d'efficacité, de généralisation et de performance liés à l'utilisation d'architectures à capacité limitée. Ensuite, je conçois et évalue des méthodes pour l'entraînement en aval des encodeurs auto-supervisés avec deux objectifs principaux : la généralisation, en particulier sur des échantillons hors distribution, et l'efficacité au cours de l'inférence.

Mots clés : traitement de la parole, apprentissage profond, apprentissage supervisé, apprentissage auto-supervisé.

Résumé Substantiel

La parole est le moyen de communication central et spécifique de l'humanité. Si la grande majorité des enfants humains maîtrisent naturellement, rapidement et aisément son utilisation au bout de quelques années, la compréhension et la synthèse machine des signaux de parole est une tâche complexe encore non résolue dans plusieurs cadres et cas d'usage.

L'utilisation et la compréhension par la machine des signaux de parole impliquent une numérisation du signal analogique d'entrée. Cette numérisation permet de passer de l'onde physique de pression que capte les microphones numériques à une représentation vectorielle sous forme d'amplitude de signal échantillonnée souvent à 16000, 22050 ou 48000 échantillons par seconde de signal audio, qu'on appelle souvent la forme d'onde (mais qui n'en est qu'une version échantillonnée). Sous cette forme, le signal est historiquement difficilement décodable par les algorithmes d'apprentissage statistique. C'est pourquoi les premiers modèles de traitement du signal de parole ont utilisé des représentations intermédiaires sous forme de spectrogrammes temps-fréquence basés sur des transformées de Fourier du signal d'entrée.

Ces représentations ont été à la base des méthodes d'apprentissage machine pour le traitement des signaux de paroles depuis les années 60. Vers le début des années 2010, le nouveau paradigme d'apprentissage des caractéristiques a incité les chercheurs à passer outre les représentations et caractéristiques non apprises ou dites faites-main. Ainsi, des modèles dits de bout-en-bout ont essayé de s'affranchir des représentations spectrales, et apprendre la fonction de classification ou de transcription directement dans l'espace des formes d'onde échantillonnées.

Une deuxième révolution, celle de l'apprentissage non-supervisé, permettant l'exploitation des larges sources de données brutes et de s'émanciper du coût de l'annotation humaine, a mené au sujet de cet ouvrage : l'apprentissage auto-supervisé. Nous le définissons comme l'ensemble des techniques non-supervisés qui permettent d'apprendre des représentations

et caractéristiques intermédiaires facilitant, à travers leur utilisation comme entrée des modèles, la résolution par la suite des tâches par apprentissage statistique. Pour le traitement du signal de parole, les approches auto-supervisés permettent d'apprendre des représentations de la parole où le contenu phonétique, émotionnel ou le timbre vocal sont plus facilement accessibles, permettant de meilleures performances sur les tâches de transcription et de reconnaissance de l'émotion ou du locuteur.

Précisément, ces représentations sont apprises à travers la résolution, dans un premier temps, de tâches dites prétextes. Ces tâches sont différentes des tâches classiques dans le traitement de la parole qui nécessitent des annotations manuelles. Les étiquettes sont générées automatiquement permettant l'apprentissage non-supervisé de la tâche. Les représentations apprises sont, dans un deuxième temps, passées à des classifieurs les utilisant pour résoudre les tâches d'intérêts, dites tâches en aval. Les représentations auto-supervisées ont permis de diminuer l'apport nécessaire d'annotations, produisant des modèles capables d'atteindre des performances de généralisation très raisonnables avec très peu de données supervisées.

L'adoption des modèles de représentations auto-supervisées sur le large ensemble des tâches traitant du signal de parole a été très rapide. Les avancées et découvertes se sont faites principalement dans une direction en largeur, ajoutant de nouvelles tâches comme la traduction ou la reconnaissance vocale des langues peu dotées. En parallèle, les approches visant à développer les modèles de représentation se sont concentrées autour de la mise à l'échelle des entraînements sur des jeux de données brutes de plus en plus étendu de l'ordre de la dizaine de millions d'heures.

Ce manuscrit décrit des travaux qui essaient d'apporter une meilleure compréhension des raisons du succès des approches auto-supervisées, et d'en tirer les meilleures pratiques lors de l'utilisation en aval des représentations auto-supervisées. Tout en validant par l'expérience les approches et questionnnements théoriques proposés, nos travaux essayent d'automatiser et de motiver un maximum de choix dans la structure d'un pré-entrainement auto-supervisé. Plus particulièrement, nous nous sommes intéressés à l'entraînement, l'évaluation et l'amélioration des performances sur les tâches en aval des représentations auto-supervisées. Nous montrons, dans ce qui suit, que les techniques proposées permettent un développement plus efficace des modèles de représentations de parole ainsi que des modèles finaux plus robustes et généralisants.

D'abord, dans **le chapitre 1**, nous exposons une large revue de littérature des modèles auto-supervisés de parole, permettant de comprendre le contexte dans lequel s'inscrivent

les travaux qui vont être présentés, et de mieux apprécier leur apport à l'état de l'art. Partant des représentations spectrales du signal de parole et passant par les représentations apprises supervisées basées sur des filtres convolutionnels, ce chapitre explique le contexte d'apparition des modèles auto-supervisés à l'intersection de l'apprentissage de caractéristiques et de la révolution non-supervisée. Dans la suite, et après la description des différentes approches majeures et la classification des modèles les plus populaires dans ce cadre, nous dressons une liste des critères désirables dans ces modèles de représentation. Cette liste est utilisée comme fil conducteur expliquant nos démarches suivantes. Enfin, la littérature des travaux qui ont comme objectif chacune des caractéristiques est étudiée.

Le chapitre 2 étudie un choix crucial dans le pipeline d'auto-supervision : la conception de la tâche prétexte. Les travaux présentés dans ce chapitre dévoilent un lien entre l'indépendance conditionnelle entre les étiquettes de la tâche prétexte et les échantillons de parole en aval étant donné les étiquettes en aval. Ils montrent que, pour une tâche d'intérêt en aval (reconnaissance de la parole par exemple), plus cette indépendance est élevée, plus la performance obtenue sur la tâche en aval à l'aide de représentations auto-supervisées apprises sur la tâche en aval considérée est élevée. Cela nous a permis d'attribuer un score aux tâches prétextes en vue d'une meilleure performance sur les tâches en aval dignes d'intérêt. Nous nous appuyons sur cette notation pour élaborer une approche de sélection et de pondération des tâches prétextes multitâches. Cette méthode est, dans un deuxième temps, étendue avec succès aux paramètres d'apprentissage contrastif pour l'augmentation automatique des données. Précisément, dans le cadre de l'apprentissage contrastif, la tâche prétexte est définie par le choix des distorsions qui permettent la création de deux vues à partir d'une même entrée. Nous montrons que notre méthode de création de vue, reposant sur l'indépendance conditionnelle, mène à des gains de performance en aval. Globalement, cette méthode est validée sur quatre tâches en aval : la reconnaissance de la parole, la vérification du locuteur, la reconnaissance des émotions et l'identification de la langue.

Ensuite, **le chapitre 3** offre une étude critique de la façon dont les représentations auto-supervisées ont été évaluées dans la littérature sur le traitement de la parole. Avec le nombre croissant de représentations auto-supervisées proposées, il était nécessaire de disposer d'un benchmark complet sur diverses tâches vocales en aval afin de guider les chercheurs et les praticiens souhaitant utiliser ces représentations pour leurs problèmes. Les principaux classements de la communauté ont fixé les conditions d'entraînement en aval, à savoir l'architecture du décodeur en aval, pour chaque tâche considérée, et ont évalué les représentations auto-supervisées figées avec celles-ci. Nous évaluons la

robustesse des classements actuels aux changements dans le choix des décodeurs en aval. Les résultats obtenus montrent que les méthodes actuelles d'évaluation sont très sensibles au choix des architectures en aval. Ceci nous pousse à nous interroger sur la validité des choix populaires en termes d'architectures en aval. Ces choix mettent souvent en avant la simplicité, comme critère de sélection de décodeurs en aval. Dans une deuxième partie, et sur la base des résultats obtenus et de quelques propriétés souhaitées d'un benchmark utile, nous présentons quatre arguments en faveur de décodeurs en aval plus complexes.

Le chapitre 4 décrit des méthodes visant à renforcer trois propriétés souhaitées des modèles utilisant des représentations pré-entraînées auto-supervisées : l'adaptation à des domaines non vus pendant le pré-entraînement auto-supervisé, des inférences efficaces en termes de calcul et des capacités de généralisation sur des échantillons de test en dehors de la distribution d'entraînement. La première s'appuie sur l'approche basée sur l'indépendance conditionnelle développée dans le chapitre 2. Pour réduire les coûts d'inférence, nous explorons les options de réduction des séquences et des réseaux proposées dans la littérature. Enfin, nous étudions le rôle de l'oubli de la tâche de pré-entraînement dans la perte de performance de la généralisation et les moyens de réduire cet oubli pour une meilleure reconnaissance vocale hors domaine. L'ensemble de ces travaux donne des indications et des bonnes pratiques couvrant les principaux aspects de l'utilisation de l'auto-supervision en aval.

Enfin, dans la conclusion, nous commençons par récapituler les principales contributions des travaux présentés. Ces contributions sont agrémentés par les lignes de codes et des jeux de données que nous avons publiés et partagés avec la communauté pour la reproductibilité de nos travaux, et faciliter l'utilisation par des tierces de nos approches. Une deuxième partie explore les pistes pour des travaux futurs. Deux sujets nous semblent très dignes d'intérêt. D'abord, et avec les quantités toujours plus massives de données audio disponibles et les larges coûts de calcul qu'elles engendrent, la sélection automatique des données d'entraînement est un sujet qui devient indispensable. Elle permet de se débarrasser, sans efforts manuels, des données de mauvaise qualité. Nous proposons deux types d'approches de sélection qui nous semblent prometteuses, classées selon qu'elles soient dépendantes des modèles ou pas. Ensuite, nous discutons de l'apparition récente des modèles discrets de représentations de la parole et leurs possibles utilisations pour des tâches génératives (tâches dont la sortie est de l'audio, comme la conversion de voix par exemple). Nous décrivons l'utilité de ces représentations dans le cadre d'un

paradigme dit de régénération, et proposons différentes pistes qui permettrait d'améliorer les modèles discrets actuels.

Acknowledgements/Remerciements

I am always sadly uncomfortably shy with this exercise although occasions to recognize and be grateful for the role others played in our life are scarce. I will, thus, try to keep it short. I apologize in advance to all the people who were there to help me, support me, or make these years more enjoyable both professionally and personally, but who are not cited here. If you happen to read these lines and feel forgotten, I will happily offer drinks we share when we meet. Cette partie est rédigée en Français et en Anglais, selon la personne ou le groupe de personnes à remercier.

First, I want to thank the jury for their time and efforts in reading the thesis, attending the defense, and coming up with the set of questions that made the discussion after the defense appreciable and fertile for me and the attendance. Thanks then to Hung-Yi Lee and Anthony Larcher, my two rapporteurs, and to Hervé Bredin, Karen Livescu, Shinji Watanbe and Emmanuel Vincent.

Je tiens ensuite à remercier mes deux encadrants, Slim Essid et Titouan Parcollet, qui m'ont introduit aux usages de la recherche scientifique. Les années de thèse sont courtes devant tout ce qu'il y à apprendre et tout ce qu'on souhaite en faire, merci d'avoir été d'excellents transmetteurs . Merci Slim pour avoir essayé de me transmettre ton recul et ta vision des liens entre apprentissage machine et traitement du signal. Merci Titouan pour m'avoir aidé à apprendre à prioriser, à être rigoureux dans les protocoles expérimentaux et dans le code, à remettre en cause ce qui me semblait simple ou évident et à toujours prendre avec précaution les nouveaux résultats, positifs ou négatifs. J'ai eu la chance d'avoir deux super superviseurs sur le plan humain. J'en ressens aujourd'hui le privilège et j'en suis extrêmement reconnaissant.

Thanks to all the other senior members who have been supervising me during shorter parts of this thesis, and from whom I also learned a lot. Starting with Emmanuel Dupoux during my master's thesis who introduced me to speech research, Félix de Chaumont-Quitry and Zalán Borsos who were my internship supervisors at Google in Zurich who guided me in my first speech synthesis experience, Mirco Ravanelli and Cem Subakan at Mila that oriented my latest pieces of research, and Lucas Ondel Yang that introduced me

to the joys of lattice-based speech recognition. I also want to thank Gaël, Matthieu, and Geoffroy for their multiple observations and questions during the lab seminars, you have been during these years a formidable and attentive audience for my first presentations.

Let me also thank my two thesis committee members, Laurent Besacier and Devon Hjelm. Laurent made me meet Titouan when he was not able to supervise me anymore, and that happened to be a tremendous suggestion. Devon, not coming from a speech background, made me think about my approaches outside of the speech modality, and always came up with observations opening new ideas and questions that shaped future works.

I want to thank all my colleagues, from the people I collaborated with in Paris, Zurich, Montréal or Le Mans during the short JSALT summer, to the ones that were there for a drink, discussions, and PhD discussions and interrogations. I would still say that PhDs, especially in France, are quite solitary adventures, but you contributed to making it less solitary. Special thanks to David, who was always there, even when the lab rang hollow ("sonnait creux") after COVID times. Special thanks to Robin also, who was my supervisor in my first speech internship before we became friends, co-authors, and co-supervisors.

Finally, a more personal note. With its ups and downs, the thesis adventure quickly becomes overwhelming and this is where people from outside that world are needed. I want to first thank my parents and my sisters for their continuous support. Je remercie mes parents, Hédi et Hayet, pour m'avoir encouragé, poussé et appris les valeurs de travail et de sérieux nécessaires à cette entreprise, et à toutes les autres de ma vie. Je remercie aussi mes sœurs Meriem et Yesmine, plus proches géographiquement, qui étaient là quand les temps étaient un peu durs et qu'il me fallait en parler. Je tiens à remercier tous les amis qui m'ont accompagné durant cette période et qui je l'espère m'accompagneront encore pour la suite. Les compagnons de thèse, d'abord, Vincent et Achille, dont les multiples débats ont formé la meilleure courbe d'apprentissage de l'esprit critique et de l'art subtile de la persuasion et de la mauvaise foi. Lancelot, le coloc du début de thèse, et la team avec Andrea et Constant scellée autour de l'amour d'un soir pour le Mac Ambrose. Enfin, je remercie les "Outliers" et le "Lounge", et surtout les membres de l'intersection Sami et Brahim et les discussions quotidiennes aussi légères qu'imprévisibles.

Enfin, je remercie Alice, avec qui la vie, et même la thèse, est plus douce. Merci de m'avoir si souvent aidé à sortir la tête du guidon et à profiter des petits et grands plaisirs de la vie.

Publications

* denotes shared first authorship.

Included in the main (in order of appearance)

- **Zaiem, S.**, Parcollet, T., Essid, S. (2021). Conditional independence for pretext task selection in Self-supervised speech representation learning. *Proc. Interspeech 2021,* 2851-2855, doi: 10.21437/Interspeech.2021-1027
- **Zaiem, S.**, Parcollet, T., Essid, S., Heba, A. (2022). "Pretext Tasks Selection for Multitask Self-Supervised Audio Representation Learning," in *IEEE Journal of Selected Topics in Signal Processing*, 2022, doi: 10.1109/JSTSP.2022.3195430. Impact Factor : 7.695
- **Zaiem, S.,** Parcollet, T., Essid, S. (2022). Automatic Data Augmentation Selection and Parametrization in Contrastive Self-Supervised Speech Representation Learning. *Proc. Interspeech 2022*, 669-673, doi: 10.21437/Interspeech.2022-10191
- **Zaiem, S.**, Kemiche, Y., Parcollet, T., Essid, S., & Ravanelli, M. (2023). Speech Self-Supervised Representation Benchmarking: Are We Doing it Right? *in Proc. Interspeech 2023*
- **Zaiem, S.**, Kemiche, Y., Parcollet, T., Essid, S., & Ravanelli, M. (2023). Speech Self-Supervised Representations Benchmarking: a Case for Larger Probing Heads. *Currently under review.*
- **Zaiem, S.**, Parcollet, T., & Essid, S. (2023). Automatic Data Augmentation for Domain Adapted Fine-Tuning of Self-Supervised Speech Representations. in *Proc. Interspeech 2023*
- **Zaiem, S.**, Algayres, R., Parcollet, T., Essid, S., & Ravanelli, M. (2023). Fine-tuning Strategies for Faster Inference using Speech Self-Supervised Models: A Comparative Study. *ICASSP 2023*-IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Not included in the manuscript (in reverse order of publication year)

- Della Libera, L., Mousavi, P., **Zaiem, S.**, Subakan, C., & Ravanelli, M. (2023). CL-MASR: A Continual Learning Benchmark for Multilingual ASR. ArXiv. /abs/2310.16931
- Wright, G. A., Cappellazzo, U., **Zaiem, S.**, Raj, D., Yang, L. O., Falavigna, D., & Brutti, A. (2023). Training dynamic models using early exits for automatic speech recognition on resource-constrained devices. ArXiv. /abs/2309.09546
- Malard, H., **Zaiem, S.**, & Algayres, R. (2023). Big model only for hard audios: Sample dependent Whisper model selection for efficient inferences. ArXiv. /abs/2309.12712
- Abdallah, A. A*., Kabboudi, A., Kanoun, A., & **Zaiem, S.*** (2023). Leveraging Data Collection and Unsupervised Learning for Code-switched Tunisian Arabic Automatic Speech Recognition. ArXiv. /abs/2309.11327
- Algayres, R., Ricoul, T., Karadayi, J., Laurençon, H., **Zaiem, S.**, Mohamed, A., Sagot, B., & Dupoux, E. (2022). DP-Parse: Finding Word Boundaries from Raw Speech with an Instance Lexicon. *Transactions of the Association for Computational Linguistics*, 10, 1051–1065. https://doi.org/10.1162/tacl_a_00505
- Gao, Y., Parcollet, T., **Zaiem, S.**, Fernandez-Marques, J., de Gusmao, P. P. B., Beutel, D. J., & Lane, N. D. (2021). End-to-End Speech Recognition from Federated Acoustic Models. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 7227-7231, doi: 10.1109/ICASSP43922.2022.9747161.
- Algayres, R., **Zaiem, S.**, Sagot, B., Dupoux, E. (2020). [Evaluating the reliability of acoustic speech embeddings.](http://arxiv.org/abs/2007.13542) *Proc. Interspeech, 2020-October*, 4621–4625.
- **Zaiem, M. S.**, & Bennequin, E. (2019). Learning to Communicate in Multi-Agent Reinforcement Learning : A Review. ArXiv Preprint. http://arxiv.org/abs/1911.05438
- **Zaiem, S.**, & Sadat, F. (2018). Sequence to Sequence Learning for Query Expansion. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019*, http://arxiv.org/abs/1812.10119

Contents

List of figures

List of tables

0 Introduction

When you set out on your journey to Ithaca, pray that the road is long, full of adventure, full of knowledge.

- Constantin Cavafy (Ithaca)

In recent years, the dynamism of machine learning has manifested in remarkable breakthroughs, revolutionizing diverse modalities. From image recognition to natural language processing, the efficacy of machine learning algorithms has been increasingly contributing to advancements in technology and science.

Traditionally, these breakthroughs have often been anchored in supervised learning, relying heavily on meticulously annotated datasets. However, it is an expensive and time-consuming process that limits the scalability of machine-learning applications. Especially for voice and speech applications, it hinders the ability to extend to more languages and keeps a large part of the globe excluded from speech technology advancements.

As the demand for broader applications of machine learning and speech technologies grows, the limitations of supervised learning become increasingly evident. Enter the paradigm of unsupervised learning with algorithms able to learn disentangled representations from unlabeled data. These approaches, learning meaningful insights from patterns in raw speech, mirror an important aspect of human learning, drawing parallels with the manner in which infants acquire abilities in their environment without explicit instruction, or at least with a reduced amount of it.

Self-supervised learning is a sub-family of unsupervised approaches. They allow taking advantage of the large available unlabeled corpora, but also from algorithmic advances in supervised learning. Indeed, they model unsupervised representation learning through solving a task in a supervised fashion approach, but with automatically generated labels. They are, today, very popular in various applications, across almost all data modalities. A thorough definition will be given in Section 0.2.

The first chapter describes the context leading to the development of self-supervised models in the speech-processing community, and the motivations that pushed us toward working on the questions detailed in the next chapters. It ends with a summary of our contributions answering the raised questions.

0.1 Context

Self-supervised learning (SSL) has emerged as the intersection between two blooming ideas: feature learning in end-to-end (E2E) approaches and unsupervised learning.

0.1.1 Feature Learning

Feature learning, a by-product of the deep learning advent, has seen many machine learning researchers and practitioners explore ways to reduce hand-crafted or human priors in the design of the representations fed to further statistical models. Originally, audio model front-ends have been relying on time-frequency handcrafted variants of spectrograms, generally Mel-scaled or in the cepstral domain (such as Mel-frequency Cepstral Coefficients or MFCCs) (Furui, 1981). These spectral representations, motivated initially by psycho-acoustic or bio-acoustic findings (Fechner, 1966), have been a driving source of improvements and progress for speech or speaker recognition during the 20th century. At the eventual cost of interpretability or modularity, substantial gains in performances have been attained by allowing models more degrees of freedom, either in the feature learning process or in the decoding or label output phase (Sainath et al., 2012). In speech processing, this has been witnessed mainly through two aspects. First, feature engineering, which has been relying almost exclusively on hand-crafted spectral-based features, has been more and more driven by learnable interfaces between the signal represented as a sampled raw waveform and final objectives such as textual transcriptions or diarization. From learnable audio front-ends relying on the same filtering-based approaches equipped with a few learnable parameters (Ravanelli & Bengio, 2018), to fully learnable heavily-parametrized convolution-based deep neural networks (Zeghidour et al., 2018), audio front-ends have been heavily impacted by the feature learning and deep-learning revolution.

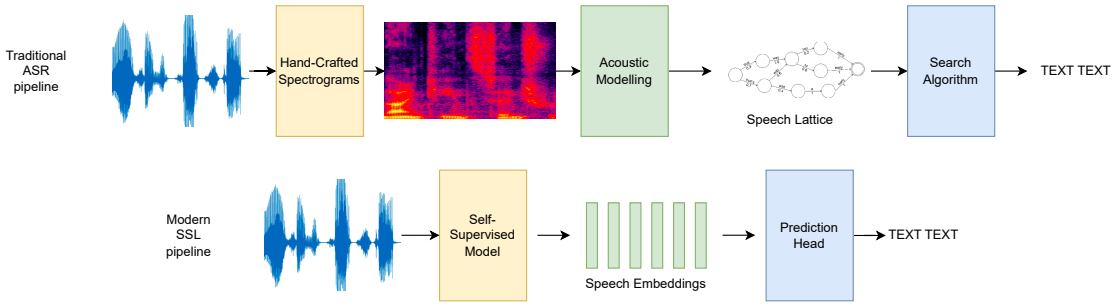

Fig. 1.: Comparison between different components in a traditional automatic speech recognition (ASR) pipeline and to “end-to-end" self-supervision-based ones.

This revolution did not only change front-ends but also led to very different back-ends impacting acoustic modeling and prediction heads. In the first decade of the century, speech recognition approaches were mainly relying on Gaussian mixtures for acoustic modeling and finite-state machines for decoding. Modeling was also structured with generative Hidden Markov Models (HMMs) or discriminative Conditional Random Fields (CRFs). In the second decade, following the modality-agnostic trend described above, Gaussian mixtures have been replaced by deep neural networks (DNN) in so-called hybrid HMM-DNN approaches. This change has been pushed by popular speech processing toolkits, mainly Kaldi (Povey et al., 2011) offering efficient, state-of-the-art, and easily deployable hybrid pipelines.

In an automatic speech recognition pipeline, the acoustic model would assign to every speech frame, *i.e.* short speech windows, probabilities over the predictable phonemes or characters. During inference, a search step over the predicted lattice allows for the generation of a probable sequence (the most probable if the search algorithm is exhaustive). Large research efforts have been deployed to find appropriate prediction network topologies, state reweightings, or faster search algorithms. Again following the end-to-end trend, these have started to be replaced with less interpretable and prior-heavy prediction heads. From the seminal work of Alex Graves and collaborators on Connectionist Temporal Classification (CTC) (Graves, 2012) to attention-based decoders trained with cross-entropy loss (Good, 1952), the speech recognition field has been moving to lattice-free and search-free approaches, slowly losing interpretability and modularity in the same process.

Research in self-supervised representation learning of speech representations has risen within this general data-centered trend with embeddings **learned** from the data. Figure 1 shows how modern end-to-end[1] speech recognition using self-supervision pipelines compare to traditional ones. It shows how these representations can replace traditional spectrogram-based representations and parts of acoustic modeling. However, given the data-hunger of deep learning approaches, those methods would not have had this success if not for the availability of large speech corpora. Even more abundant today than labeled speech corpora is unlabeled speech data, which leads us to the second rising idea: Unsupervised representation learning.

0.1.2 Unsupervised Representation Learning

The availability of large collections of unlabeled utterances in all modalities from speech samples to images pushed the research community towards exploring ways to exploit them aiming for better performance on common tasks solved with labeled data. “Unsupervised" comes here in contrast to “supervised" learning, *i.e.* learning algorithms necessitating human labels, such as textual transcription of the audio inputs in the case of automatic speech recognition.

Unsupervised representation learning is considered at the core of human learning processes, with infants learning world representations mainly through natural interactions with their environment (Zaadnoordijk et al., 2022). For instance, the first studies have focused on the use of child-directed speech for unsupervised speech learning (Vallabha et al., 2007). It relies on finding the common patterns among different versions of the same semantic objects (words, visual or physical objects...). As an example, studies have shown that babies develop the ability to recognize word identities around the age of 9 months (Dupoux, 2018).

Unsupervised learning is not a new idea *per se*. Auto-encoding approaches have been developed and shown effective in a few settings starting from the nineties, with Helmholtz (Dayan et al., 1995) and restricted Boltzmann machines (Hinton, 2012). Auto-encoding approaches aim to learn “useful" (generally for dimensionality reduction, topic modeling, or sampling) representations through learning to map the inputs to a lower-dimension space allowing to generate the input from the compressed representation. With limited

[1] One may argue that the presence of pretrained SSL modules makes it non-end-to-end. As E2E is an ill-defined concept, we see no need to delve into this discussion.

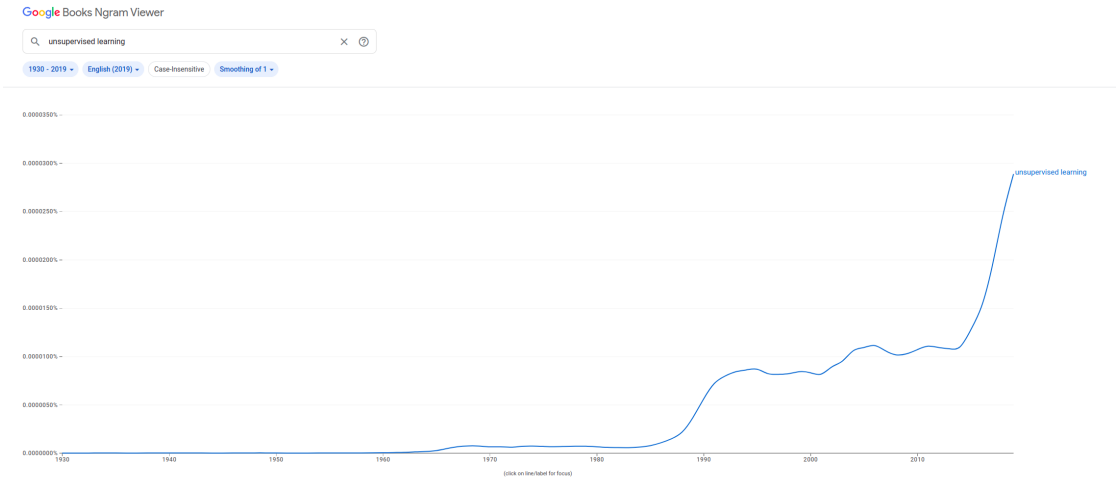

Fig. 2.: Trendiness of the expression “unsupervised learning". A small start in the nineties before an explosion in the last decade.

training datasets and auto-encoding leading to mitigated downstream performance, the trend has been steadily decelerating in the first decade of this century, after an impressive in the nineties, as shown in Figure 2.

However, the recent availability and release of large and relatively clean unlabeled datasets, such as Libri-Light (Kahn et al., 2020) for English speech samples, increased the motivation to exploit these, nurturing research in unsupervised learning. Techniques, inspired by child learning, tried to find recurring patterns within the raw data samples and learn useful representations from these recurring patterns (Lavechin et al., 2020).

One of the first notable breakthroughs of unsupervised representation learning can be found in the continuous word representations learned with Word2Vec (Mikolov et al., 2013). Through solving the task of predicting missing words in context or upcoming words, it has been shown that using large datasets of raw text allows models to learn semantically rich representations leading to better text classification performances even with reduced labeled datasets. Word2Vec appears again within the intersection of feature learning, as it replaced with learned word embeddings classic statistical representations such as Bag-of-Words (BoW) or Latent Semantic Analysis (LSA) (Dumais, 2004), and unsupervised learning as these embeddings are learned without human annotation relying on large unlabeled datasets. More precisely, the Word2Vec approach can be classified in a sub-family of unsupervised learning algorithms, called Self-supervised learning which is the topic of this thesis. It is time to give it a proper definition.

0.2 Self-Supervised Learning

Definitions of self-supervision in the literature may vary, but we will give it a common one that is coherent with the works, experiments, and results described in this manuscript. A point of agreement is that self-supervision is a sub-family of unsupervised learning methods, in the sense that self-supervised methods aim to exploit unlabeled data. A first distinguishing aspect lies in the concept of pretext-tasks. SSL allows learning unsupervised representations through solving tasks, and copying supervised settings, but with automatically generated labels instead of human-provided ones. A pretext-task is thus a learning task, with labels or objectives that can be defined without specific human annotation.

A second distinction lies in the objective of the learning phase. While unsupervised approaches may aim for clustering or sampling from the data distribution, self-supervision almost exclusively targets representation learning[2], *i.e.* learning representations that are useful for other tasks. This implies that these methods are generally evaluated in two steps. First, the representations are learned through solving the pretext-task on the unlabeled sets, in what is called the “upstream" learning phase. In a second phase, the “downstream" one, these representations are evaluated on their ability to improve performance on common tasks, in a supervised setting, compared to classic representations.

Figure 3 schematises the two phases discussed above. The right branch showing the downstream training, represents the classical supervised setting, with the considered downstream task learned on the available annotated training data. In this setting, the self-supervised encoder, represented here with a yellow box and whose parameters are trained on the unlabeled data samples, would generally be replaced with an encoder trained only on the downstream data points, and usually following spectral feature extraction.

The reason behind the success and the wide adoption of self-supervised representations is the gains in performance reached, especially in low-resource scenarios, compared to traditional supervised-only pipelines. Manual annotation is a costly and imprecise endeavor, especially in the case of complicated tasks such as diarization. Specifically concerning speech, collecting annotated data, for all languages, in all recording settings,

[2]There are very few exceptions to this, such as what is called “self-supervised speaker verification".

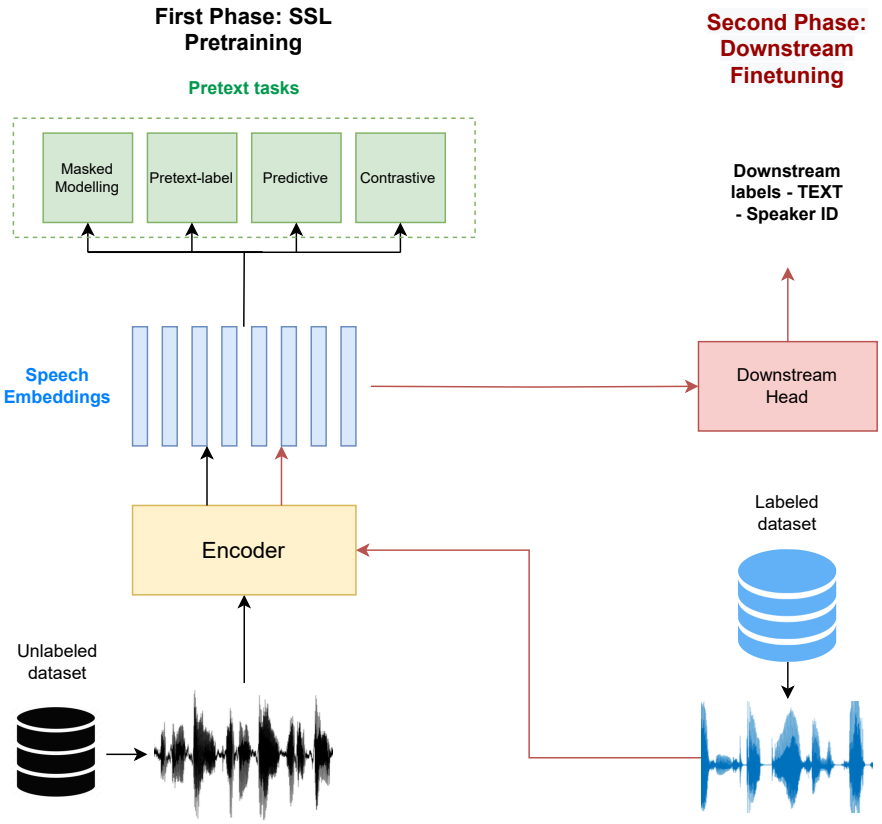

Fig. 3.: Two phases of speech self-supervised experiments. Black arrows represent upstream and red ones downstream training. Self-supervised learning has allowed substantial performance gains especially with reduced labeled datasets.

is very costly, and, thus, reasonable speech recognition performance was only attained on the small set of popular, and consequently financially profitable, languages.

Formally, given a set of speech utterances (U) (for unlabeled), with every speech utterance $u = (u_i)_{i\in[1,T]}$ composed of T samples. The input u may be the raw sampled waveform, or frame-segmented spectral representations, with a large trend towards getting rid of the spectral representations in recent works. The goal of the self-supervised training, *i.e.* the first phase in Figure 3, is to learn a function we will call $e(.)$ for encoder, that maps a given audio sample $u = (u_i)_{i\in[1,T]}$ to a representation $h = (h_i)_{i\in[1,T/k]}$ with k a downsampling factor, and $\forall i \in [1, T/k]$, h_i is a vector of a chosen dimension d ($k = 320$ and $d = 768$ or $d = 1024$ are common choices in the literature).

This function $e(.)$ is generally learned through solving a pretext-task. Let us call Z the pretext-labels, so that every speech sample $u = (u_i)_{i\in[1,T]}$ is mapped, in the pretraining phase, to z. z here may have different time-granularities, for instance, be constant over the speech sample so that the model would only predict one value, or have the same granularity as u or h. As an example, in the HuBERT model (Hsu, Tsai, et al., 2021), $z = (z_i)_{i\in[1,T/k]}$ has the same time-dimension as the encoded representation h with every z_i corresponding to a cluster ID, *i.e.* a single integer.

A projection head, a function we will call $d(.)$ in the following, mapping the latent representations h to the pretext-labels z, is learned in the pretraining phase. Both these functions are generally modeled as neural networks with a deep network for the encoder and usually a shallow one for the projection head. Let $\mathcal{L}$ be a loss function over the z space, θ_e and θ_d the weights of the encoder and the projection head, and n the number of samples in the training set. The pretraining phase aims at finding:

$$\theta_e^*, \theta_d^* = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(z_i, d_\theta(e_\theta(u_i))). \tag{0.1}$$

Again, for the specific case of HuBERT, $\mathcal{L}$ is a cross-entropy classification loss.

After the first pretraining phase, the projection head is generally discarded and the encoder representations are used to solve the downstream task. Precisely, in the second phase, given a downstream annotated dataset (X, Y) composed of m speech samples $X = (x_i)_{i\in[1,m]}$ and their corresponding labels $Y = (y_i)_{i\in[1,m]}$ (for instance speaker IDs in speaker recognition settings), a downstream head is trained to map the representations $h = e(x)$ to their corresponding downstream labels y. While we have only been discussing the last output of the encoder, it is not uncommon to use various layers of the encoder for downstream purposes.

Self-supervised Learning Popularity

In the last years, mainly after the Wav2Vec2.0 (Baevski, Zhou, et al., 2020) release in 2020, and its impressive results obtained on speech recognition tasks, self-supervised representations have become extremely popular within the speech processing community. From the speech technology point of view, self-supervised speech representations have been used on almost all the tasks generally tackled by the community: from speech transcription, speaker-related questions, emotion recognition (Y. Wang et al., 2021),

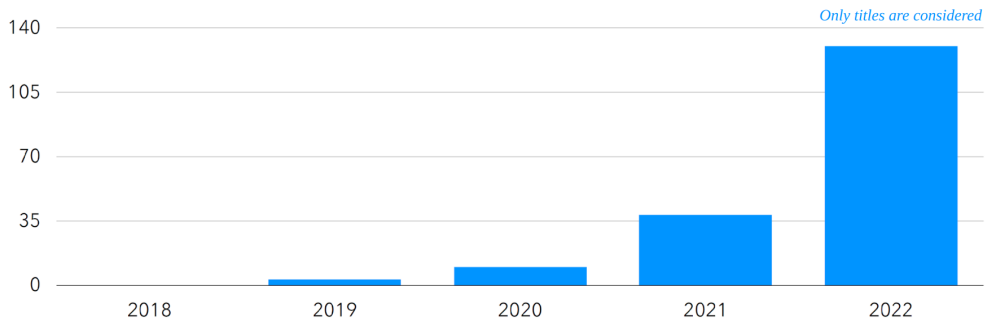

Fig. 4.: Evolution of the number of occurrences of self-supervision-related terms in paper titles at INTERSPEECH, showing the trend of using them on various speech tasks.

speech translation (Zanon Boito et al., 2022), and more recently, speech synthesis, accent classification (Zuluaga-Gomez, Ahmed, et al., 2023), pronunciation assessment (E. Kim et al., 2022), and the list may go on...

But its impact did not stop at speech technology and also concerned speech science. Self-supervised representations have been used to understand and model how humans learn to understand and produce speech in different settings: phonemic/phonetic differences, bilingual children, second languages, and comparison with cerebral settings...

In these years, replacing hand-crafted spectrograms with the most recent self-supervised representation has been a low-hanging fruit many researchers and speech practitioners quickly tried to grab. Speech conferences have been flooded with works building on these representations. This popularity appears in the rising number of papers at InterSpeech, the main annual conference for research in speech science and technology, with “self-supervision" appearing in the title already, as shown in Figure 4. These numbers do not even properly represent the popularity of these models. While this use was an event worth being in the title in the first years, self-supervised representations can now appear as a simple hyperparameter or input in papers barely mentioning their use. “HuBERT" or “WavLM", two self-supervised models, may appear in a results table in a row between “LogMel Spectrograms" or “MFCC". They have become the go-to representations for every speech practitioner suffering from reduced training datasets.

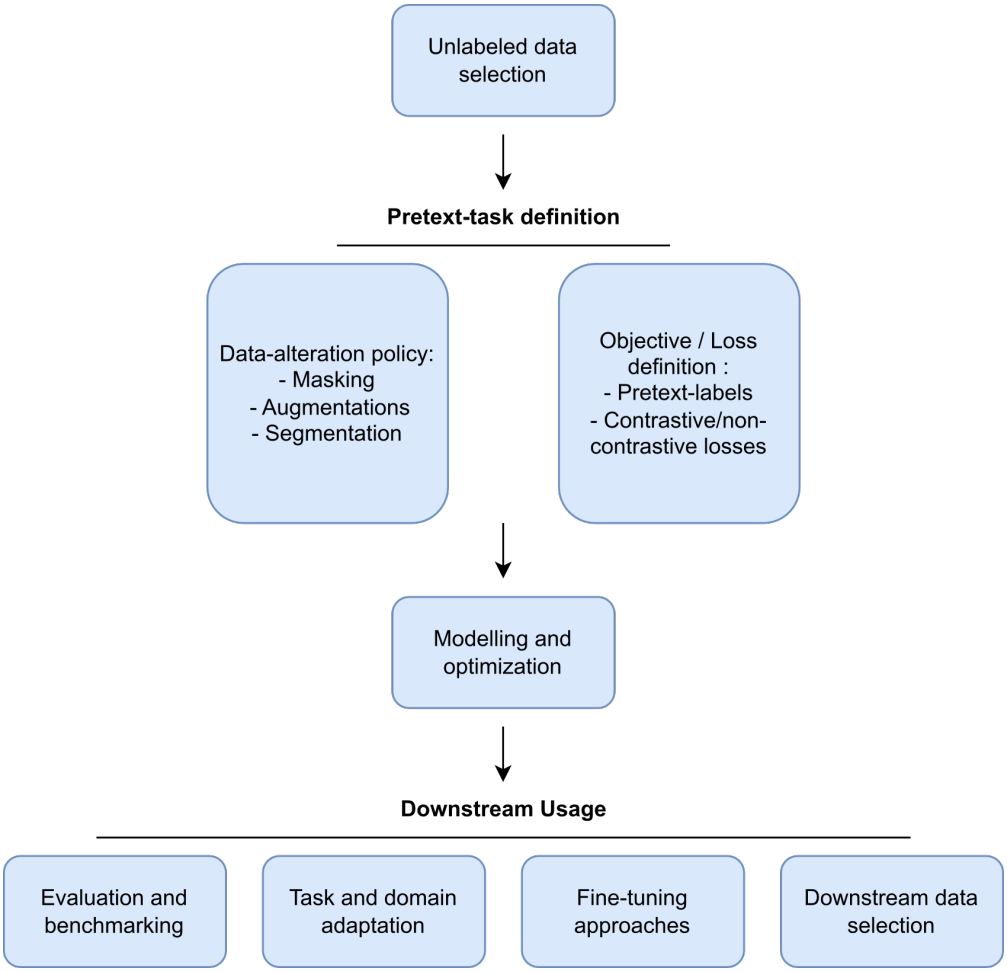

Fig. 5.: Boxes representing the different questions implying choices in the development of a self-supervision full pipeline.

0.3 Motivation

Figure 5 is a tentative to regroup in a single plot all the questions involving manual choices in the definition of a self-supervision pipeline. Let us quickly go through the different components. The first box concerns choosing, among the available data, appropriate subsets leading to the best representations. Not explored in this work, we will discuss ideas about this phase in Chapter 5. The second level concerns the definition of the pretext-task. We divide it into two parts. First, input data alteration as a means to enforce invariances in the learned representations with augmentations is ubiquitous in contrastive

self-supervised approaches. Alterations here also include any masking, re-ordering, or re-organization policy that may be needed to implement the defined pretext-task.

This brings us to the third box which is surely the one that has had the most attention in the speech self-supervision literature. Discrete auto-encoding, multi-task learning, contrastive or predictive approaches; the literature abounds in examples of pretext labels, losses, and methods leading to the definition of a pretext-task for speech self-supervised learning. The fourth box describes all the techniques, generally inspired by the success in classic supervised settings, around modeling the learned functions and minimizing the defined losses. This part is not actively discussed in the main chapters of this thesis. Following the majority of recent works, independently of their domain of application, speech self-supervised modeling networks converged towards using Transformers architectures, with the corresponding training methods for optimization and learning rate handling.

Finally, the fourth and final level concerns the downstream exploitation of the learned representations. As expressed before, the field has witnessed large efforts towards applying or adapting these representations to different tasks. Non-focusing on any task or dataset in particular, despite a drift towards speech recognition in the last chapter, this thesis explores task-agnostic ideas, trying to improve our understanding of why these techniques work and to give informed progress tracks. This level involves questions about evaluating these models, the best ways of fine-tuning them for a given task, and adapting the model to the task and its acoustic conditions.

The works presented in this thesis try to dissect the elements in these boxes, with the aim of shedding some light on the best practices that should be followed in every box. Precisely, the goal has been to take some distance from task-oriented applications towards deriving motivated insights, techniques, and rules for 5 out of the 8 boxes in Figure 5; namely data-augmentation policies, objective definition, evaluation, and benchmarking of these methods and finally downstream fine-tuning and adaptation. The main reason motivating this work is that, despite the self-supervision exploding popularity and successful use described in the previous section, the field still lacks “intelligent" explanations behind this success, and good practices for evaluation and usage.

0.3.1 Lack of Fundamental Insights

The second motivation is the lack of fundamental insights behind this success and a deep understanding of the underlying mechanisms. It is important, first, to give credit to what

has been properly done. First, various pretext-tasks have been proposed for training speech self-supervised approaches from bottleneck auto-encoding techniques (Algayres et al., 2020) and multi-task learning (Pascual et al., 2019b), to contrastive predictive coding (Schneider et al., 2019) and teacher-student approaches (Baevski et al., 2022). Successful paradigms, such as heavy data augmentations, or sequence-masking, were kept and built upon in the next iterations. Second, following NLP's so-called "BERTology" (Devlin et al., 2019; Rogers et al., 2021), in-depth layer-wise, domain-wise, phonetic and linguistic, probing of the content of these representations has been performed by speech researchers after the outbreak of these self-supervised models (Pasad et al., 2021). However, the field, following a general deep-learning-related trend, has been building only on empirically-motivated ideas. There have been no real efforts to explain why a given method leads to downstream gains.

Four reasons can be given for this absence. First, formal and quantified non-empirical justifications have been hardened with the deep learning era, where interpretable and theoretically justified approaches have been replaced with large black boxes encompassing hard-to-parse computations. This, as said previously, has been a general trend in the deep-learning landscape across various application domains.

Second, state-of-the-art self-supervised trainings are, at least, an order of magnitude more costly in terms of computations than classic supervised settings. At the end of 2023, replicating trainings leading to the best performing "Large" versions of popular self-supervised models is a prowess almost unattainable for academic players. Even, smaller "Base" versions, trained "only" on the full training sets of LibriSpeech, would generally require a week of training on a dozen of high-performing GPUs. Even in the case of availability of these, the impact of large batch sizes, documented in the literature, leads to a performance gap between models developed using large and very large memory GPUs. More concerning, experiments led on smaller models and smaller datasets do not seem to be able to predict the performance post-scaling to large ones, making costly trainings the only reliable way of exploration. The high cost of training a single model, combined with the need for numerous trials on large sets to explore the utility of a given approach, made training self-supervised models a domain restricted to a few, mainly industrial, entities in the speech research landscape.

The third reason is partly a consequence of the second one. Restricting self-supervision to a few players restricts mechanically the number of explored tracks. One of these tracks, is further scaling, in terms of computations and data. Motivated by tremendous success in other domains, and in supervised settings, this has been the main lever for performance

gains in the last years for self-supervision. Wav2Vec 2.0 (Baevski, Zhou, et al., 2020), with its 300 million-parameter encoder trained on $60k$ hours of speech data, was seen as a mastodon in the speech community in 2020. Since then, recent models have exceeded a billion parameters and were trained on more than 650k hours of labeled speech data or more than 10 million hours of multilingual unlabeled speech data (Y. Zhang et al., 2023).

The fourth and final reason is more mundane. The exploding usage of self-supervised representations should not hide the fact that the first very efficient methods are slightly more than three years old. The first years have naturally focused on more low-hanging fruits, such as use on new tasks or scaling described before. The coming years may be ones where the community looks more for higher fruits. This has been seen in other machine learning domains, where the larger sizes of the research communities led to shorter low-hanging fruit life duration.

0.3.2 Research Questions

This work addresses essentially the following research objective: to seek a better and deeper understanding of the reasons for the success of self-supervised approaches and to recommend the best ways to build self-supervision-based speech processing pipelines, as we believe they will cover a large share of future pipelines. Specifically, the works described in this document aimed to provide answers to the following interrogations:

- What is the link between pretext-task choice and downstream task performance? Can we automatically find optimal pretext-tasks towards better performance on a given downstream task?
- The two-step evaluation of self-supervised representations requires new evaluation methods compared to classic supervised settings. How to build robust evaluation methods, covering the large spectrum of speech tasks and usages?
- Compared to hand-crafted representations, using self-supervised encoders implies a multitude of choices and costs, including efficiency and domain shifts. What are the costs and problems raised by the usage of this new technology? Given these costs and downfalls, what are the best practices to solve them in self-supervised pipelines?

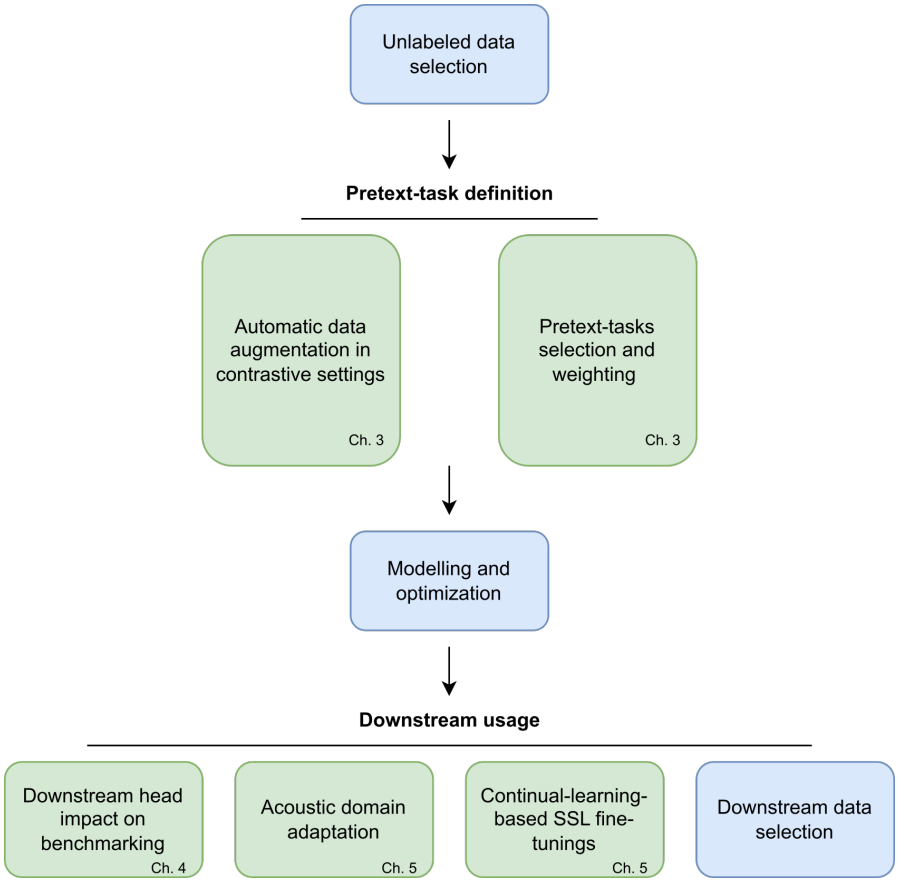

Fig. 6.: Main contributions of this work within the SSL framework defined in Figure 5. While the first and fourth boxes have not been properly explored, we discuss them in Chapter 5. Green boxes encapsulate our main contributions.

The answer to these questions drove the efforts in the core chapters of this manuscript and shaped the contributions of this work. Let us go through these contributions.

0.4 Contributions

Figure 6 summarizes in the green boxes the main contributions of this work. The first and second green boxes concern insights on training self-supervised models. The final one tackles how pretrained models should be used for downstream fine-tuning. We replaced the titles given in Figure 5 with our contributions on the given part. The remaining blue boxes are those not discussed in the core works of this document.

Chapter 2 investigates a crucial choice in the self-supervision pipeline: pretext-task design. Works presented in this chapter unveil a link between the conditional independence between the pretext-task labels and the downstream speech samples given the downstream labels. It shows that, given a downstream task of interest (speech recognition for instance), the higher this independence[3], the higher the performance obtained on the downstream task using self-supervised representations learned on the considered downstream task. This allowed us to score pretext-tasks towards better performance on downstream tasks of interest. We build on this scoring a multi-task pretext-task selection and weighting approach. This method is, in a second time, successfully extended to contrastive learning settings for automatic data augmentation. The method is validated on four downstream tasks: speech recognition, speaker verification, emotion recognition and language identification.

Chapter 3 gives a critical look at the way self-supervised representations have been evaluated in the speech literature. With the growing number of proposed self-supervised representations, comprehensive benchmarks on various downstream speech tasks were needed to guide researchers and practitioners wanting to use these for their problems. Main benchmarks in the community fixed the downstream training conditions, namely the downstream head, for each considered task, and evaluated the frozen self-supervised representations with these. We evaluate how robust the current rankings are to changes in the choice of the downstream heads. Based on the obtained results, and a few desired properties of probing, we give four arguments for larger downstream heads.

Chapter 4 describes methods to enforce three desired properties of full self-supervision-based models given pretrained representations: adaptation to domains unseen during self-supervised pretraining, computationally efficient inferences, and out-of-domain generalization abilities. The first one builds on the conditional-independence-based approach developed in Chapter 2. To reduce inference costs, we explore sequence and network shrinking options in the literature. Finally, we explore the role of forgetting the pretraining task in losing generalization performance and ways to reduce this forgetting for better out-of-domain speech recognition. Combined, these works give insights and good-practices covering the main aspects of self-supervision downstream usage.

[3]While independence is generally a binary value/concept, we deal here with estimates that are not.

1 Related Works

It is a laborious madness and an impoverishing one, the madness of composing vast books - setting out in five hundred pages an idea that can be perfectly related orally in five minutes. The better way to go about it is to pretend that those books already exist, and offer a summary, a commentary on them.

- Jorge Luis Borges (Fictions- The Garden of Forking Paths)

This chapter sets the historical stage leading to the appearance and success of self-supervised representations for speech. Starting from traditional speech representations, and following the advancement of the feature-learning trend, it describes, subsequently, efforts produced in self-supervised learning across modalities. Finally, it outlines works on improving speech self-supervised models by dividing them into a list of desired characteristics and describing techniques to enforce those. This chapter offers the keys to an in-depth understanding of the global context surrounding the works described in further chapters, allowing a better appreciation of the positioning of this work and its contributions.

1.1 Representations for Machine Learning

Representations in machine learning serve as the critical bridge between raw data and meaningful insights. They capture essential features and patterns, enabling algorithms to understand and learn from complex information. Well-designed representations enhance model efficiency and effectiveness, facilitating more accurate predictions and

generalization to unseen data. In essence, the quality of representations directly influences the success and robustness of machine learning models. This has been roughly stated in an equation by Domingos (2012):

$$\text{Machine Learning} = \text{Representation} + \text{Objective} + \text{Optimization}.$$

This thesis in general and this chapter in particular focus on the first element of the sum.[1] It describes efforts in representation learning, and the evolution of the speech representations, learned or not, used for common speech processing tasks.

If (X, Y) is a labeled dataset, composed of raw inputs X and human-provided labels Y, the inputs X are generally mapped to a representation $R = f(X)$ that conditions the performance of a learned mapping $g : R \to Y$, linking the representations with their final labels. The function f can take several forms. If it is the identity function, then we would say that the mapping g is learned on the raw input, in an “end-to-end" fashion. As previously indicated, this is more and more the case since the feature learning revolution.

But before this recent trend, f would generally consist of a sequence of hand-crafted processing steps, aiming to inject human domain knowledge and priors into the representations. We will discuss in detail these approaches for speech in Section 1.2. In computer vision, for instance, the use of local filters for image classification derives from the known importance of edge detection and localization invariance for determining the labels. The main goals of these steps are generally to reduce the quantity of “noise" inherent to data samples, to enforce a few desired invariances, and to normalize the inputs for better processing by the following statistical models.

Scale-invariant feature transform (SIFT) (Lowe, 2004) is an interesting example to dissect. It has been widely used for image processing in the first decade of this century. It identifies distinctive points in an image, regardless of their size or orientation, making it robust to scale and rotation changes. Precisely, SIFT operates by first identifying keypoint locations using scale-space extrema, and then describing these keypoints using histograms of gradient orientations. This allows it to find and match features between images, making it a valuable tool in tasks like object recognition and image stitching.

[1] It is funny to note that, given the feature-learning trend, the first part now involves Objectives and Optimization as well.

In modern pipelines, it is common that f is learned along with g through minimizing a defined loss $\mathcal{L}(g(f(X)), Y)$. How separable g and f are is unclear in “end-to-end" approaches as f may only consist of basic preprocessing steps such as decoding (in the case of audio encoded audio-samples), resampling, or normalization.

Improving the representations has been one of the main tracks for progress in machine learning. Better feature-extraction tricks, properly enforcing the appropriate invariances, or better disentangling the signal components, are approaches often widely adopted by the research communities. In a review, Bengio *et al.* (2013) have listed a set of desired properties and outcomes of representation learning. This list included disentangling explanatory factors, potentially with a hierarchical organization, smoothness (*i.e.* for two input points x and x', if $x \approx x'$ then $f(x) \approx f(x')$), natural clustering... We will discuss in a further section, which characteristics are upheld by modern self-supervised representations, and define a new set of desired properties.

The next two sections will focus on speech processing, detailing the evolution of speech representations, from hand-crafted spectral representations to learned ones with different degrees of learning and imposed priors. We will afterwards, get back to a modality-independent discussion when going into self-supervision approaches on unlabeled datasets.

1.2 Hand-crafted Spectral Representations for Speech

Mel-scaled spectrograms have been the main representation used for speech and non-speech audio tasks almost since the dawn of this research field. From the sampled one-dimensional raw waveform obtained from recording devices, a Fourier transform is applied on sliding analysis windows of the original signal to obtain the two-dimensioned time-frequency Short-Term-Fourier-Transform (STFT) representation. The power spectrogram, the basis of almost all the popular representations, is the (squared) modulus of the STFT. One of the main paradigms for the development of hand-crafted spectral representations was to draw inspiration from the human auditory system to design features for audio processing pipelines. Mel-scaling (Stevens & Volkmann, 1940), still widely used in the speech community is one facet of these inspirations. Inspired by psycho-acoustic experiments, it processes the spectrogram through a set of filters, called mel-filterbanks, possibly designed as triangular filters. These filters are narrow at low frequencies and get wider at higher ones reflecting human perception of pitch.

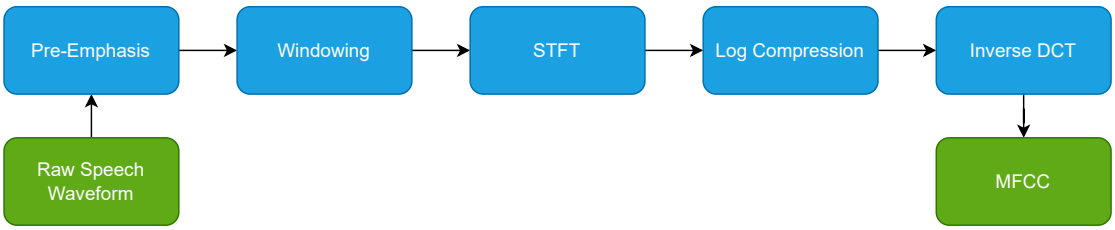

Fig. 1.1.: MFCC computation steps from left to right. Every step involves its share of choices and hyperparameters.

What follows the Mel-scaling generally depends on the extraction toolkit, the research laboratory practices, or the considered speech task. Figure 1.1 shows the computation steps of Mel-frequency cepstral coefficients (MFCCs). Variations of the MFCCs have been an intensive research question in the first decade of this century, from replacing log compression (Schluter et al., 2007) to replacing the mel-scale (Umesh et al., 1999).

Various studies have shown that hand-crafted feature-based approaches have the advantage of more stable trainings compared to fully learned pipelines (Haider et al., 2023). But while log-Mel spectrograms are still very widely used in the audio and speech communities, mainly for tasks with sufficient data such as English Speech-to-Text (Radford et al., 2023), they suffer from the hand-crafted features common limitations. First, all the choices are regularly challenged by new approaches and hard-coded replacements, with even serious doubts cast over the Mel scale defining experiments reproducibility (Fechner, 1966). Second, while the human biases injected may be suitable for speech recognition or other understanding tasks, others may necessitate more fine-grained frequency bins or a focus on different parts of the audible spectrum. Finally, approaches in different modalities have shown that learning more parameters can lead to an increase in performance, as part of the feature learning trend.

1.3 Learnable Front-ends

The aforementioned reasons, including the feature-learning revolution, witnessed in the last two decades and discussed in Section 0.1.1, led to attempts to replace the old log-Mel-based approaches with learned representations.

Replacing the spectrograms with learnable front-ends reduces the human intervention and priors to the choice of the modeling approach and loss functions. With the feature-learning trend, one-dimensional convolutional front-ends, able to learn local filterbanks similar to the hand-crafted ones, were their first replacements (Palaz et al., 2015; Zeghidour et al., 2018). The convolutional front-ends impose the locality of the learned features, while the filter size and the stride inject human knowledge and priors about signal stationarity and dynamics (Schneider et al., 2019). Today, in contrast with supervised settings, these convolutional front-ends are very popular in self-supervised ones, although recent works have been criticizing the memory over-head they come with (Parcollet et al., 2023).

Parameter-Efficient Approaches

A second approach consists in incorporating learned parameters in the feature extraction phase, with the aim of learning interpretable parameters within the spectral-based framework (Ravanelli & Bengio, 2018). The main mechanism is to make classic feature extraction pipelines differentiable according to their main parameters (these were generally hyper-parameters fixed through trials) such as the Mel-scale parameters. This offers two advantages: first, a reduced number of learned parameters, which supposedly leads to better generalization, with the cost of instability. Second, by focusing on a few parameters within theoretically motivated frameworks, the learned filters are easily interpretable, offering explanations for the possible performance gains. To get a good understanding of these ideas, we will discuss two examples: SincNet (Ravanelli & Bengio, 2018) and Learnable Audio Front-ends (LEAF) (Zeghidour, Teboul, et al., 2021).

SincNet features were first introduced for speaker recognition purposes. Sine cardinal filters replace the Fourier transform in the classic Mel spectrograms, but with the additional twist of having for each filter, two learnable parameters: the central frequency and the bandwidth of the sine cardinal filter. Then, if N filters ($N = 80$ in the original paper) are applied on the raw waveform, it only implies $2N = 160$ learnable parameters for the feature extraction. It has since been successfully applied on a wide range of speech and non-speech audio tasks for instance in bio-acoustics, and even on cerebral signals (Fainberg et al., 2019; Zeng et al., 2019).

LEAF (Zeghidour, Teboul, et al., 2021) goes a little bit further. First, the Sine cardinal filters are replaced with Gabor ones. Gabor filters in LEAF, similarly to band-pass filters, allow two learnable parameters per filter again, the min and the max cut-off frequencies. These filters are followed with low-pass filtering and per-channel normalization, both of

which include learnable parameters, making all the components of the feature-extraction pipeline learnable.

However, while these approaches allow learning audio representations better tailored to the considered task, they rely exclusively on annotated speech datasets for transcription, classification, or generation tasks. As said in the introduction, the last decade witnessed the release of large unlabeled datasets, as an indirect result of the Internet explosion. Self-supervision allowed the use of these datasets to reduce the quantity of annotation needed to reach state-of-the-art performance.

1.4 Self-supervised Learning: Historical Progress

As we already defined self-supervised learning in Section 0.2, this section will mainly cover a few historical aspects, especially in non-speech modalities. The wide use of SSL in various fields makes an exhaustive review of the SSL techniques on non-speech data too large for the scope of this work. We will, thus, focus in the two next sections on techniques that inspired similar approaches for speech, and on the main trends.

Let us note first that a close sibling to self-supervision is transfer learning. It is very similar in the fact that it generally involves two training phases with a first learning step on one initial domain (generally the one with more labeled data) and a fine-tuning on the target domain. The main difference with SSL lies in the fact that the first training also involves an annotated dataset. Generally, the two trainings are for the same task and would share the same objective. Transfer learning approaches, studying which data to use, and how to avoid forgetting the first phases and overfitting on the generally reduced target datasets, have been abundant (Bell et al., 2020).

Another close sibling is self-training. Similarly to self-supervised training, it enables the use of both unlabeled and labeled data. In self-training settings, a model is trained initially on a limited set of labeled data. The trained model is then used to make predictions on unlabeled data, and the most confident predictions are added to the labeled dataset. This process is iteratively repeated, with the model being retrained on the expanding dataset. In contrast with self-supervised learning, the multiple phases share the same training objective but with pseudo-labels obtained using a model from a previous step. We will not delve into these two siblings in this work.

1.4.1 Masking Approaches for Sequential Data

For textual data, the first widely used model falling within the boundaries of our definition of self-supervision may be, as said in the introduction, Word2Vec (Mikolov et al., 2013). It has enabled learning word-level representations through the pretext-task of masked language modeling. Interesting, now infamous, semantic properties and linear interpolations of these representations have been largely exposed and commented on. Its downstream performance gains, mainly for text classification, represented a turning point in natural language processing research (Pennington et al., 2014).

More recently, with the success of transformer-based language modeling, internal representations of the language models have been used intensively for other text-related tasks. This use falls again perfectly within the definition of SSL given in Section 0.2. Today, apart from the thriving language modeling applications for text and code generation and conversational agents, a big industry around selling self-supervised text embeddings, learned through language modeling pretext-tasks, exists mainly for intent and text classification purposes.

Let us define the language modeling task; it will enable us to grasp later the similarities it has with recent speech self-supervised approaches. Let S be a sequence of tokens $t_1, t_2, \ldots, t_n$ where each t_i belongs to a token vocabulary V. Tokens are generally subword character sequences obtained through Byte-Pair Encoding (BPE) (Sennrich et al., 2016). The language modeling task involves estimating the probability of observing a given sequence of tokens S. This can be denoted as $P(S)$, the probability of the entire sequence.

Mathematically, the goal is to maximize the likelihood of the observed sequence of tokens, which is equivalent to finding the parameterized model θ that maximizes the probability of a given sequence of tokens S:

$$P(S;\theta) = P(t_1, t_2, \ldots, t_n; \theta).$$

This can be factorized using the chain rule of probability:

$$P(S;\theta) = P(t_1;\theta) \cdot P(t_2|t_1;\theta) \cdot \ldots \cdot P(t_n|t_1, t_2, \ldots, t_{n-1};\theta).$$

In practice, a recurrent neural network (RNN) or a transformer model, is often used to model these conditional probabilities. The model is trained on a large corpus of raw text to learn the parameters θ that maximize the likelihood of the training sequences. This learned language model can then be used to generate text or evaluate the likelihood of new sequences. Internal representations can be used in a self-supervised flavor for classification tasks.

The large success of the language modeling objective, especially after the introduction of transformers and BERT-like models (Devlin et al., 2019), that is perfectly adapted to the sequential and tokenized aspects of text as a pretext-task, left little room for other self-supervised approaches for text. Contrastive approaches for instance suffered from the difficulty and limitations of applying semantically invariant relevant text alterations.

1.4.2 First Pretext-tasks

Two methods, RotNet (Gidaris et al., 2018b) and JigSaw Puzzles solving (Noroozi & Favaro, 2016) were the main seminal works towards the definition of genuine/original pretext-tasks for image representation learning. The first one exploited the fact that human-captured images tend to depict objects in an “up-standing" position, and made a network learn representations by predicting the angle of rotation of artificially rotated pictures. Pretext labels, here, are the angles of rotation applied, and they are known on unsupervised data points as they are automatically generated to create the task. In the second one, Noroozi and Favaro proposed an approach inspired by the JigSaw game. After dividing a picture into square patches, the pretext-task was to reorder them. The motivation behind these two works is that solving the pretext-tasks implies learning properties such as edges, and object orientation.. and that those learned concepts will offer a useful basis for downstream classification.

1.4.3 Contrastive Learning

The next trend in image self-supervised representation learning was set with the pioneering work of Chen *et al.* (T. Chen et al., 2020) on contrastive learning. Contrastive learning is a self-supervised representation-learning technique that aims to teach a neural network to distinguish between pairs of similar and dissimilar points (images in this case). It does so by embedding images into a high-dimensional feature space, where similar

images are mapped closer together and dissimilar images are pushed apart. This method leverages a contrastive loss function, such as the triplet loss or InfoNCE (Noise-Contrastive Estimation) (Van Den Oord, Vinyals, et al., 2017), to ensure that the network learns to capture meaningful features from the images, enabling applications like image similarity search, object recognition, and clustering.

Precisely, let again $X = (x_i)_{i \in [1,N]}$ be the set of data points. In a labeled setting, we can sample a positive pair (x_i, x_j) (*i.e.* sharing the same label) and a negative pair (x_i, x_k), where x_i, x_j, and x_k are data points selected from X. The goal is to encourage the model to embed similar data points closer and dissimilar data points farther apart in the feature space.

To achieve this, we define an encoding function e that maps each data point x to a feature space usually denoted as Z. The feature representations are obtained as $z_i = e(x_i)$, $z_j = e(x_j)$, and $z_k = e(x_k)$.

A common loss function used in contrastive learning is the triplet loss (Bredin, 2017), which encourages the positive pair to be closer in the feature space than the negative pair:

$$\mathcal{L}(x_i, x_j, x_k) = \max\{0, \alpha + d(z_i, z_k) - d(z_i, z_j)\}, \tag{1.1}$$

where α is a hyper-parameter representing a margin, and $d(z_a, z_b)$ represents a distance metric, such as the Euclidean or the cosine distance, between feature vectors z_a and z_b. The loss $\mathcal{L}$ is minimized during training making similar data points closer and dissimilar ones farther apart in the feature space, improving the quality of representations.

In the SimCLR work, a self-supervised version of this approach (T. Chen et al., 2020), positive pairs are constructed by applying data augmentation techniques to the same original data point x. Given two chains of augmentation a_i and a_j sampled from an augmentation policy A, two augmented versions denoted as $x_i = a_i(x)$ and $x_j = a_j(x)$ are generated, with corresponding feature representations $z_i = e(x_i)$ and $z_j = e(x_j)$.

The goal is to maximize the similarity between z_i and z_j while minimizing similarity with features of samples not generated from x. This is achieved using the Noise-Contrastive Estimation (NCE) loss:

$$\mathcal{L}(x_i, x_j, \text{negatives}) = -\log\left(\frac{\exp(z_i \cdot z_j)}{\exp(z_i \cdot z_j) + \sum_k \exp(z_i \cdot z_k)}\right). \tag{1.2}$$

Where $z_i \cdot z_j$ represents the dot product similarity between the feature vectors z_i and z_j, and negatives refers to the set of negative samples from which we get features denoted as z_k here. The NCE loss aims to learn representations that are invariant to the set of augmentations/alterations present in the considered augmentation policy.

1.4.4 Non-contrastive Learning

Sampling negative pairs for triplet-loss or contrastive approaches has been an extensive field of research leading to substantial gains through careful selection (Robinson et al., 2021). This sensitivity led to research emancipating from negative sampling in what has been called “non-contrastive" self-supervised learning methods. The main seminal work for this trend was Bootstrap Your Own Latent (BYOL) (Grill et al., 2020). It leverages, again as in SimCLR, two views of an image, often obtained through different data alterations. However, BYOL employs a teacher-student approach, where the student network has to produce embeddings similar to the teacher ones, and each network receives a different augmented version of the input. The teacher network, which is an exponential moving average of the weights of the student, provides stable and improved target representations for training. BYOL-like approaches, with additional adjustments, mainly dividing the input image into patches and using Vision-Transformers and patch-level losses and masking (Dosovitskiy et al., 2021), are now state-of-the-art for image self-supervised representation learning (Oquab et al., 2023).

Let us keep the notations introduced in the previous section, with x_1 and x_2 denoting two different versions of the same input x. A student network M_S and a teacher network M_T project those to their corresponding feature vectors $z_1 = M_S(x_1)$ and $z_2 = M_T(x_2)$. The teacher network is updated as an exponential moving average (EMA) of the weights of the student. This EMA is used to provide target representations, enhancing the stability of the training process:

$$\theta_{M_T} \leftarrow \beta \cdot \theta_{M_T} + (1 - \beta) \cdot \theta_{M_S}. \tag{1.3}$$

Here, θ_{M_T} and θ_{M_S} are the parameters of the teacher and student networks, and β is a momentum hyperparameter, a real-value between 0 and 1 controlling how acute teacher updates are.

The objective of BYOL is to maximize the similarity between the projected views of the student and teachers and does not rely on a traditional contrastive loss, and thus does not require negative samples. The model encourages $p(z_1)$ to be similar to z_2, with p a projection-head, leading to the following training loss:

$$\mathcal{L}_{\theta_{M_S}} = ||p(z_1) - z_2||_2^2. \tag{1.4}$$

During training, the back-propagation only updates the weights of the student network, with a stop-gradient applied to the teacher branch. The slow update of the parameters of the teacher encoder helps prevent the model from converging to a degenerate solution ensuring that the representations remain diverse and stable throughout the training process.

1.4.5 Modality-agnostic Approaches

Before delving into the approaches specifically designed and proposed for speech and audio modalities, it is important to note a trend of convergence between the techniques among modalities. The transformer-based unifying trend, in terms of modeling architectures and optimization, is a well-documented phenomenon (Xu et al., 2023). But a similar modality-agnostic trend, centered around predicting masked parts or embeddings of masked parts, is also rising. Predicting masked, thus missing, elements or context, has been a successful pretext-task, independently of the modality. The Data2Vec series (Baevski et al., 2022, 2023) has pushed this trend forward, proposing a pipeline valid for all modalities, with limited differences in pre-processing or encoding front-ends.

1.5 Speech Self-supervised Learning

This section describes the historical motivations, approaches and evolutions in learning unsupervised and self-supervised speech representations. After this, we discuss its impact on speech technologies and how this impact can be evaluated. Finally, a third part decorticates the desired properties of these representations, and the efforts targeting these properties in the literature.

1.5.1 Genesis: Zero-speech Oriented Research

Since the beginning of the 2010 decade, and in an attempt to reproduce the learning abilities of infants, the Zero-speech (Dunbar et al., 2017) community, from the name of a long list of challenges, developed a multitude of approaches for unsupervised speech representation learning. The zero-speech setting is simple. As infants are able to learn to speak without any textual inputs (although with extended non-speech ones), models should be able to do the same. The community has been focusing in the first years on the first abilities infants learn, recognizing words in a speech stream, mainly through two tasks: unsupervised speech segmentation and acoustic word discovery. Speech segmentation is the task consisting of retrieving the word boundaries in an unsegmented speech sentence, while acoustic word discovery is the task of regrouping speech segments within clusters composed of the same word in its different pronunciations.

To perform segmentation or word discovery, notions of segment frequencies and phonetic similarity are needed, and those require an embedding space where speech embeddings representing phonetically similar segments should be close, allowing for pure clusters. The first representations used for these tasks relied on hand-crafted spectral-based features. Mel-frequency cepstral coefficients (MFCC) or Perceptual Linear Prediction (PLP) were the main used spectral features (Kamper et al., 2015; Holzenberger et al., 2018). Different pronunciations of the same word may lead to different speech utterance lengths. Thus, for unsupervised clustering, getting fixed-size embeddings of varying-size speech segments has been, and still is in word acoustic embedding research, a hot topic for the zero-speech and now self-supervision community.

Bottleneck Features

One of the popular learned speech representations used was “Bottleneck-features" (Grezl & Fousek, 2008; Yu & Seltzer, 2011). The idea was to learn a reduced-size representation, in a bottleneck of the whole auto-encoding network. The information bottleneck acts as a feature compressor, selecting the relevant (hopefully mainly phonetic) information needed for signal regeneration. The representations learned in a zero-speech (*i.e.* unsupervised) scope were used already in a self-supervised flavor as defined in Section 0.2, in a two-phased training approach with downstream labeled data. For instance, Yu and Seltzer (2011) showed how unsupervised pretraining of bottleneck features improved the speech

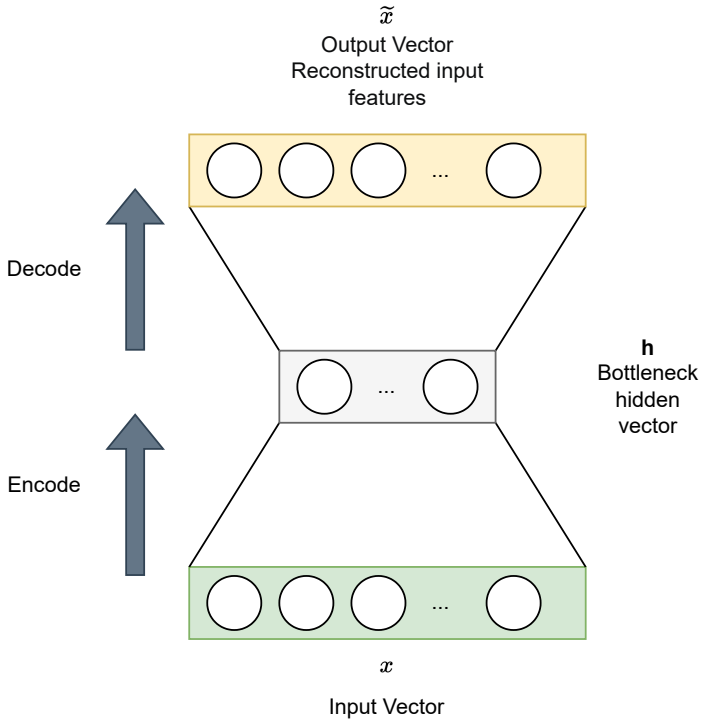

Fig. 1.2.: Schema of a bottleneck feature-extraction network trained with an auto-encoding objective. h is a low-dimension vector supposed to only keep the high-level content needed for regeneration. The figure is adapted from Lee *et al.*(2018)

.

recognition performance on telephonic data. Sainath et al. (2012) showed also how these led to ASR performance gains as well compared to MFCCs and PLPs.

Since then, multiple models, not oriented necessarily toward zero-speech tasks resolution, have been proposed in the speech literature. The next section covers these through a pretext-task-based classification. This classification relies heavily on the astounding work done in a published review of these works (Mohamed et al., 2022). An important difference with the classification done there is that we did not include a “Predictive approaches" class. Predictive approaches are those where part of the task is to fill in missing parts of the speech utterance. Naturally fitted to sequential tasks, they have been, as in text, a classic almost ubiquitous component of unsupervised speech learning, and have been added to other tasks in almost all the approaches. Thus, we will discuss how it has been added within the different classes, and not consider it as a class of its own. It is also important to note that other classifications are possible, according to the order of

magnitude of data size used, and main downstream tasks. In the following, we will keep using X as the set of speech samples for and e for the encoder function.

1.5.2 Auto-encoding Approaches

We define auto-encoding approaches as approaches where the loss function of the developed model is a distance between the audio input and a reconstructed version of the audio as output. This audio may be represented as its raw waveform, or as its spectral features. In this setting, the model generally consists of an encoder mapping the audio inputs to an intermediate representation, that will be used for downstream training, followed by a decoder mapping the representation back to the audio input, or to an unlearned function of this audio input. Let us call d the decoding function, a_i and a_o two non-learned functions altering the inputs and the outputs of the auto-encoder, and D a metric distance. Thus, the auto-encoding approaches learn representations through minimizing a loss :

$$\mathcal{L}{=}D(d(e(a_i(x))), a_o(x)). \tag{1.5}$$

Bottleneck features, described in the previous section are an example belonging to this category. In that case, the encoder maps the audio features to the bottleneck low-dimension space, while the decoder maps them back to the input features. In this case, a_i and a_o are just the spectral features extraction functions, and D is the Euclidean distance.

Based on auto-encoding, generative approaches, enabling to sample from the considered speech distribution, have been implemented to learn useful speech representations. A classic example is Variational Auto-Encoders (VAE) (Kingma & Welling, 2014). VAEs are designed to learn a probabilistic mapping between high-dimensional data and a lower-dimensional latent space. The encoder network maps input data to a probability distribution in the latent space, typically following a Gaussian distribution. The decoder network then generates data by sampling from this distribution and mapping it back to the original data space. Vector Quantized Variational Autoencoder (VQ-VAE) (Van Den Oord, Vinyals, et al., 2017) combines VAEs and vector quantization to learn a compact and discrete representation of input data. In a VQ-VAE, the encoder maps the input data to a discrete codebook, and the decoder reconstructs the data from these discrete codes. The internal representations of these auto-encoders have been successfully used for common speech tasks (Baevski et al., 2019).

Predictive Auto-Encoding

As said in the introduction of this section, predictive approaches were applied in several SSL settings, and auto-encoding makes no exception. A few methods have introduced masking parts of the audio inputs before feeding them to the encoder, *i.e.* adding a masking part to the a_i function defined above. Mockingjay and TERA (Liu et al., 2020b, 2021) are great examples of applying masking in an auto-encoding setting. Mockingjay employs BERT-like pretraining on Transformer encoders by masking input acoustic features along the time axis and then reconstructing the masked segments. TERA, an extension of Mockingjay, goes a step further by introducing additional masking of frequency bins during the pretraining process.

Discrete CoDecs

Again among the auto-encoding approaches, recent models have been learning discrete speech and audio representations in a CoDec fashion, with compression as the main goal. A seminal work is SoundStream (Zeghidour, Luebs, et al., 2021). It used Residual Vector-Quantization (RVQ) to learn hierarchical discrete representations of audio. These discrete codes allow decent universal audio compression and regeneration with low bitrates. While showing great results for generative tasks, enabling the use of NLP-inspired approaches to learn audio-to-audio or token-to-audio mappings, these representations have been showing low disentanglement leading to low performance in discriminative tasks such as transcription. For instance, the AudioLM model (Borsos et al., 2023), a leading work in audio language modeling and conditional generation, has been using discrete SoundStream audio tokens, solving using these tokens a language modeling task, similar to the one described in Section 1.4.1.

1.5.3 Contrastive Learning for Speech Representations

In contrastive learning settings, as described in Section 1.4.3, models learn the self-supervised representations through the process of discerning a target sample (considered positive) from other distractor samples (regarded as negatives) using an anchor representation. The primary objective of the pretext task is to maximize the similarity in the latent space between the anchor and positive samples while reducing the similarity between

the anchor and negative samples. The task is thus defined by the way anchor elements and negative samples are mined within the data.

Contrastive Approaches

We already described the SimCLR algorithm in Section 1.4.3. A speech version of SimCLR has been proposed (Jiang et al., 2020). It adds a spectrogram reconstruction loss along with the contrastive one, as the latter, while enforcing the desired invariances, is too information-lossy and makes the representation not suitable for ASR. The COLA model (Saeed et al., 2021) targeted classification tasks where the class label is generally constant in one speech utterance, such as speaker recognition or language identification. The anchors were not just augmented versions of a speech segment, but non-overlapping speech segments coming from the same audio file. It led to, at the time it was published, state-of-the-art results on the considered classification tasks.

Contrastive Predictive Approaches

Contrastive Predictive Coding (CPC) (Van Den Oord, Vinyals, et al., 2017) was one of the first works introducing the NEC contrastive loss described in Section 1.4.3 for sequential data. It involves a predictive objective, training a neural network to predict future information from past information within a sequence of data, such as audio or text. However, it differs from traditional auto-encoders in that it uses a contrastive loss function, making the model learn by contrasting the correct prediction with incorrect ones. The task is this way simpler than the regressive one of predicting audio frames.

The CPC approach paved the way to the Wav2Vec series. This series represented a turning moment for speech self-supervision. Given its importance, let us give some details about the loss and the modeling. The W2V2 (Baevski, Zhou, et al., 2020) encoder consists of two parts. First, a convolutional front-end downsamples the raw audio waveform (sampled at 16 kHz) to 50 Hz frame vectors. Then, a contextual module, composed of stacked transformer layers learns contextual representations keeping the same dimension and sampling rate as the input. During training, the output of the convolutional head is quantized, leading to embeddings q_t. The training loss aims for maximizing the similarity between contextual outputs c_t, centered around frame t, and quantized q_t (thus its classification as a contrastive approach). It is defined as :

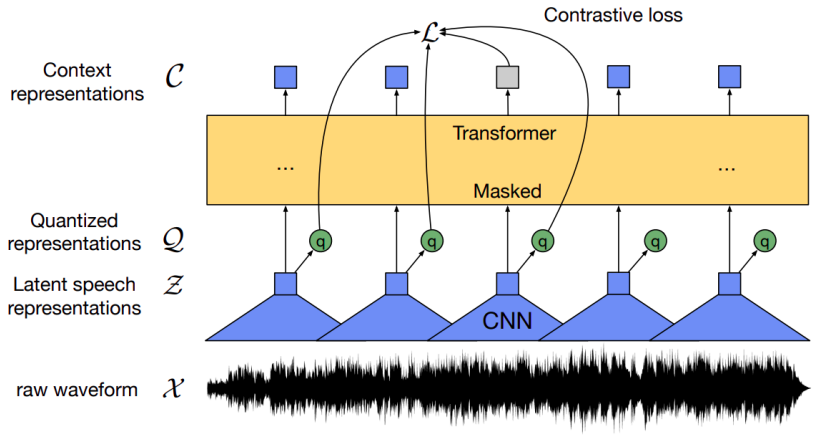

Fig. 1.3.: Figure representing the Wav2Vec 2.0 training components and loss. The NEC-inspired loss aims to maximize the similarity between q and the output c. The output of the convolutional front-end is partly masked. The figure is from Baevski *et al.* (2020)

$$\mathcal{L}_m = -\log \frac{\exp\left(\text{sim}\left(\mathbf{c}_t, \mathbf{q}_t\right)/\kappa\right)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp\left(\text{sim}\left(\mathbf{c}_t, \tilde{\mathbf{q}}\right)/\kappa\right)}, \tag{1.6}$$

with sim the cosine similarity and κ a temperature hyper-parameter. Figure 1.3 shows the different components and the training approach.

1.5.4 Pretext-labeling

Following the notations in Eq (1.5), this class of self-supervised approaches groups all the methods where a_o, the function defining the target output of the decoder, is not merely a spectrogram extraction, but generally a more complex function that could even be learned from the data. In this setting, a_o is a function that maps a speech sample to sample-level or frame-level labels. These labels may be discrete, making the self-supervised training a sequence-to-class or sequence-to-discrete-sequence classification task. They can also be continuous, making the pretraining a regression task. The loss function is not a distance or a similarity measure, but a classic classification loss such as cross-entropy in the case of discrete labels, and regression loss such as mean-squared error in the case of continuous labels.

PASE and PASE+ (Pascual et al., 2019a; Ravanelli et al., 2020b) are pioneering examples of these techniques. They defined a large set of pretext-labels consisting mainly of signal-

processing-based features. These features, widely used in speech processing, such as voicing, pitch and harmonicity, can be automatically extracted with reasonable precision using signal-processing hand-crafted approaches (Mauch & Dixon, 2014). Multiple pretext labels are learned simultaneously based on the same encoded representations through multiple pretext heads or decoders, with each one predicting one of the pretext labels. Thus, the encoder learns to output representations where the pretext-labels are easily distinguishable.

Cluster Identities as Pretext Labels

The biggest success of the idea lies in the HuBERT (Hsu, Tsai, et al., 2021) and WavLM (S. Chen, Wang, et al., 2022) models. Introduced first in the Discrete BERT paper (Baevski & Mohamed, 2020), the idea is to use as frame-level pretext labels, cluster-identities of quantized speech segments or frames. The main intuition is that if clustered properly, quantized spectral features would correlate highly with phoneme identities. This makes the self-supervised training close to a supervised phoneme-level ASR one, where the model learns a mapping between speech sequences and pseudo-phoneme identity. Compared to Discrete-BERT, HuBERT added masking parts of the audio inputs (after the convolutional front-end), making the transformers layers also learn to fill in missing parts. It also improved the correlation between the clusters and phonetic content by bootstrapping the learned representations to create better clusters. This is done by using features extracted from intermediate layers for clustering. The WavLM encoder is learned with HuBERT clusters as the target labels and can be thus seen as a form of (non-shrinking) distillation of HuBERT. The main difference is in this case the use of data augmentation alterating the inputs, making the representations more noise-invariant.

More recently, Best-RQ (Chiu et al., 2022) showed that the performance did not highly depend on the quality of the quantization, as they proposed a model learning to predict random discrete projections of the acoustic features. While the model has not been released, a Best-RQ model, pretrained on very large unlabeled and multilingual data, has led to state-of-the-art ASR performance, on a large number of languages (Y. Zhang et al., 2023) (represented by the corresponding FLEURS dataset (Conneau et al., 2023)). The first work presented in this manuscript in Chapter 2, builds upon these approaches and aims to develop a deeper understanding of the link between pretext-labels and downstream performance.

1.5.5 Teacher-student Approaches

Teacher-student approaches for self-supervised learning have been widely used on different modalities since the BYOL seminal work described in Section 1.4.4. Let us provide a reminder through the scope of the Equation (1.5). In this case, a_o, the function applied on the input to create the target, is a learned “teacher" neural network. The goal of the encoder-decoder neural network to be learned is to output representations that are similar to the ones output by the teacher. The teacher is also learned during training, as it generally consists in a moving average of the student as written in Equation (1.3).

Speech and audio versions of BYOL have been proposed (Niizumi et al., 2021; Elbanna et al., 2022). Adding a predictive objective through adding masking in a_i, similar to what has been done with other methods described above, led to the popular Data2Vec models (Baevski et al., 2022, 2023).

1.6 Evaluating the Impact on Speech Research

After describing the different methods developed for training these self-supervised encoders, let us delve into the concrete impact it had on speech research. We aim to describe qualitatively and quantitatively this impact, leading to the extreme current popularity of these models, as it is one of the motivations of this work described in Section 0.2. This section exposes the influence it had on the performance of speech models on different tasks first. Afterwards, it details the attempts to establish standard ways for evaluation of these models, before it explores studies where these representations have been used to deepen our understanding of human speech production and processing.

1.6.1 Speech Technology

Most state-of-the-art performances in the speech community now are reached using self-supervised representations instead of hand-crafted spectral inputs. Given the number of recent publications involving self-supervised, it is almost impossible to compile all the tasks and all the new results obtained using SSL. In a (very) limited attempt to highlight a few examples, we collected in Table 1.1 a set of tasks where the SSL representations have allowed to reach new highs. However, we want to highlight that comparing

Task	Metric	Dataset	SSL Rep.	SSL-Perf	HC-Perf
Automatic Speech Recognition	WER (↓)	LibriSpeech-100	Data2Vec Large	3.36	6.1
Speaker Verification	EER (↓)	VoxCeleb1-H	WavLM Base+	2.32	1.90
E2E Intent Classification	Acc (↑)	SLURP	Hubert Large	89.37	86.30
Emotion Recognition	Acc (↑)	IEMOCAP	Hubert Large	79.58	72.7
Speech segmentation	F1-Score (↑)	DIHARD 3	WavLM Large	63.4	52.4
Accent Detection	Acc (↑)	CommonVoice	W2V2-XLSR	97.1	87.9

Tab. 1.1.: Difference in performance between state-of-the-art approaches using hand-crafted or self-supervised representations for a set of speech tasks. (Some tasks are missing) Speech segmentation results are reported in (Lebourdais et al., 2022), accent detection ones in (Zuluaga-Gomez, Ahmed, et al., 2023), emotion recognition ones in (H. Wang et al., 2022) and (J. Wang et al., 2020), intent classification in (H. Huang et al., 2023).“HC-Perf" column shows the highest performance we found on the task using hand-crafted features, generally MFCCs or log-Mel spectrograms.

representation performance based on bibliographical work is complicated, and should be considered with precaution. As also explained in Chapter 3, comparing performances between representations is subject to a set of choices, including but not limited to downstream architecture. These choices are rarely identical in two works coming from different laboratories or institutions. For instance, it is important to note that for some tasks, papers, using or not self-supervision, may use additional datasets or pretrained supervised models. A similar table has been proposed in an extensive review of the domain (Mohamed et al., 2022). This one is an update for the common tasks, and shows other tasks where SSL allowed improvement today.

For Automatic Speech Recognition (ASR), we selected performance trained only on 100h from LibriSpeech and tested on the test-other split. It shows two interesting things, the performance in (somehow) reduced data scenarios, and the robustness to domain shift since *test-other* samples are generally noisier than the ones in the *train-clean-100* split. The first row in the table shows that self-supervised representations allow a 45% drop in Word Error Rates. Systematic, although relatively lower, gains are witnessed on other tasks in the table, with the most impacted ones being speech segmentation and accent detection. It is also useful to note that these performances are reached with diverse self-supervised encoders and that the tasks tackle different, sometimes orthogonal, aspects of the speech signal.

A first rule of thumb concerning the gains from SSL in ASR is the size of the annotated dataset. The less annotated data available for the transcription task, the larger is generally the gap in performance between log-Mel spectrogram approaches and SSL-based ones.

This also explains why, for instance, state-of-the-art English ASR still relies on spectral features, with the latest models trained on over $500k$ hours of transcribed English data (Radford et al., 2023).

However, this is not true for other tasks, as it can be induced from Table 1.1. In that table, the relative performance gain for VoxCeleb1 and its 352 hours of downstream data is higher than the one for IEMOCAP and its 12 little hours of annotated emotion data. Predicting SSL usefulness, quantified here by the performance gains compared to spectral features, needs more insights on the domain shift between downstream and upstream, and the closeness of the downstream task to pretraining objectives. This is discussed partly in Chapters 2 and 3.

1.6.2 Evaluation

As described partly in Section 1.5, a high number of techniques have been proposed in speech self-supervision research. A second multitude, the large number of papers using these representations in different contexts, fostered the need for comprehensive, and standardized benchmarks for speech self-supervised representation, covering the wide range of speech tasks they have been used for.

The SUPERB (Speech processing Universal PERformance Benchmark) benchmark represents the most popular effort for benchmarking SSL models. It has been designed to evaluate the performance of self-supervised learning (SSL) models in a set of speech processing tasks, ranging from very low-level ones such as phoneme recognition to high-level semantic tasks such as intent classification. This is done through fixing a given downstream architecture for every task and learning a model taking as an input the output of the SSL encoder and feeding it to the chosen downstream architecture. By computing a mean score over the considered tasks, leadearboards ranking the models proposed in the community are dressed and regularly updated. More tasks have been proposed in further works, with particular attention to generative tasks (Tsai et al., 2022) and out-of-domain generalization (T.-h. Feng et al., 2023). Chapter 3 discusses the idea of fixing the downstream head per task and its implications.

In the close domain of general audio and music tasks, the HEAR benchmark, close in design to the SUPERB one, has been introduced (Turian et al., 2022). Its set of tasks included a large variety of classification tasks ranging from bio-acoustic ones like environmental sound classification to music genre detection and a few regression tasks

such as gunshot triangularization. A particularity of the HEAR benchmark is that the tasks may have labels at the frame level (called time-stamped tasks in the paper), such as pitch prediction, or at the whole utterance level in a single-label or multi-label classification fashion (such as intent classification or language identification).

Finally, other benchmarks focusing either on specific languages, such as the French LeBenchmark (Evain et al., 2021) or on tasks non-explored in SUPERB, such as prosody-related ones in ProsAudit (de Seyssel et al., 2023) have been proposed.

1.6.3 Impact on Speech Science

While the two first subsections have focused on the performance gains on speech tasks, this one focuses on the impact of SSL on the development of speech science, *i.e.* our understanding of human speech. Various approaches more related to human production and understanding have also been exploiting successfully self-supervised representations. Acoustic-to-articulatory inversion is one of them (Georges et al., 2022) as representations learned on massive datasets help predict articulatory movements that induced speech samples (Maharana et al., 2023). It is also interesting to note that these representations help validate linguistic descriptions of speech. Positive results have been obtained in studies probing self-supervised representations on phonetic and phonemic contents (Wells et al., 2022; Martin et al., 2023).

Understanding the way humans understand and decode incoming speech waves goes also through understanding the cerebral processing of these signals. As they are learned without textual inputs, similarly to babies learning to handle speech, researchers probed the similarity between Wav2Vec 2.0 representations of audio samples and the brain activity of individuals recorded with functional Magnetic Resonance Imaging (fMRI) while they were listening to these audio samples (Millet et al., 2022). The analysis indicates, among other conclusions, that the functional hierarchy of the self-supervised representations aligns with the cortical hierarchy of human speech processing. This has been confirmed in other similar studies (Vaidya et al., 2022).

1.7 Desired Properties of Speech Self-supervised Models

Building on the previous descriptions of the main trends and historical evolutions of self-supervised speech representations, and given the impact in performance it has had during the last years, this section aims to describe a list of desired properties for these models. While we have been discussing ASR performance as a main criterion in the previous sections, these desired characteristics are meant to be orthogonal, at least partly, to it. A few of these properties have been the subject of intensive research work that will be described as a second step for each one of them, along with the limitations and standing challenges to overcome. These properties are partly chased in the core work of this manuscript, especially in Chapter 4. Let us give first a list of criteria:

- **Efficiency**: Gains in performance have been lately linked with increasing sizes of models leading to expensive trainings and inferences. This makes deploying self-supervision-based models in production costly and sometimes intractable, especially for on-device inference settings. Training efficient self-supervised models, in terms of computations, is one of the main challenges towards the democratization and the wide adoption of these models.

- **Robustness**: The main use of self-supervised representations is on tasks where labeled downstream data is scarce. This mainly concerns tasks in specific speech linguistic and acoustic conditions, that may not be present in the pretraining set. A classic example of these conditions is low-resource languages. Self-supervised models, even trained only on English data, have been successfully used on other languages not sharing common roots. Even for English data, gaps between pre-training and fine-tuning may be considerable, when encountering specific acoustic conditions such as air-traffic communications (Juan et al., 2020) or child speech (Jain et al., 2023). Thus, one main desired property is robustness to domain shifts or better out-of-domain generalization, *i.e.* achieving high performance on the largest set of conditions. The representation that is learned should be useful, in terms of final downstream performance gains compared to classic non-SSL features, even facing substantial distributional shifts with limited downstream annotation.

- **Task-coverage**: In the first experiments with modern speech self-supervised models, speech recognition was the main addressed task. For instance, the ground-breaking Wav2Vec (Schneider et al., 2019; Baevski, Zhou, et al., 2020) series did not include any non-ASR downstream experiments in the paper. Other research groups, building

on the release of the models, tried these representations on other tasks such as emotion recognition and speaker verification. The gain in results, especially in low-data scenarios, fostered further experiments on almost all the previously considered tasks and made self-supervision benchmarks include non-ASR tasks in their list of evaluations. In the first chapter, we condition choices in the self-supervised pretraining on the final downstream task of interest, as we believe, and show, that task-oriented choices can improve the final performances. However, we also believe that task-coverage is an intrinsic quality for these models for two reasons. First, while it looks more like a “collateral gain" than a real-intended feature, models developed with ASR in mind were very useful for other tasks, showing that task-agnostic (or maybe should we say high-task-covering) models are possible. Second, with the increase of these models in terms of number of parameters and size of the training datasets, it seems intractable to train large models for every task of interest. We will discuss two points concerning this property, the ongoing research in developing non-ASR-oriented models, and the coverage of the most popular self-supervised representations.

- **Open-source and Reproducibility:** The last years have seen tremendous efforts in the speech community towards sharing code, recipes, data, and ultimately pretrained models and weights. This is also true for the self-supervision community. Open-source allows the community to use and build upon the proposed models. Sharing the code and the data also allows the reproduction of the experiments. In practice, the amount of computations needed has limited the reproduction attempts.

- **Disentanglement and Interpretability:** As there is not yet an agreement in the literature for a mathematical definition of disentangled representations, we will, instead, give a qualitative one. Disentanglement in speech representation learning refers to the process of extracting and representing distinct and independent factors or attributes of speech in a way that separates them from each other. In other words, specifically for speech, it involves learning a representation of speech where different aspects of the speech signal, such as linguistic content, speaker identity, emotional tone, and background noise, are disentangled or separated from one another, making them more easily manipulable. Furthermore, disentanglement implies that changes in one factor do not significantly affect other factors in the representation. This separation and manipulation ability has a strong link with the ability to interpret the obtained representations. Various definitions of interpretability, sometimes with definitions of explainability, have been given in the

corresponding research works. While useful in various speech-related contexts where machine decisions need to be trusted (Ramanarayanan et al., 2022), this part has been mainly overlooked in speech self-supervision.

1.7.1 Robustness and Generalization

Various studies have shown that performing the self-supervised pretraining on unlabeled datasets with the same conditions as the target downstream ones, in terms of language or acoustic and recording conditions improves the final performance (Evain et al., 2021; Hsu, Sriram, et al., 2021). However, all the conditions, either acoustic or linguistic, can hardly be covered in a single pretraining. We give two arguments for this claim. First, new datasets in low-resource languages appear regularly, enriching the limited-size available corpora for those. Second, with language and usages evolving quickly, a full coverage of accents, linguistic practices, and even acoustic conditions may quickly become obsolete.

Generalization to Unseen Languages

A few models have tried to cover a maximum number of languages during the pretraining. An example is the XLSR series (Conneau et al., 2023), in its two versions, with 53 and 436 thousands of hours of speech data in 128 languages, with model sizes ranging from 300M to 2B parameters. Even, in these extreme cases, languages not present in the pretraining may not see improved performance after downstream training. For instance, our experiments on Tunisian speech recognition have shown that WavLM was a better backbone model in that case, even if WavLM has only been trained on English data (Abdallah et al., 2023).

Recently, ML-SUPERB, a benchmark targeting explicitly the performance of the SSL models on non-English datasets including low-resource languages such as Mixtec has been proposed (Shi et al., 2023). It shows, for instance, that multilingualism in the pretraining data improves the final performance, but that it is not the only factor in this performance. For instance, it is surprising to see that the performance of XLSR-53, trained on 53 languages is 3 WER points worse than the one of Hubert Large trained only on English audio data.

Finally, another linguistic shift concerns accents and variations within already seen languages in pretraining. Famous examples of lines of work include African-American English

(Riviere et al., 2021) or air traffic communications (Zuluaga-Gomez, Prasad, et al., 2023). We note, in these studies, that despite linguistic or accent-related shifts, self-supervised representations perform systematically better than hand-crafted representations.

Generalization to New Acoustic Conditions

A few works have attempted to enforce noise-invariance of speech self-supervised representations through noise injection during pretraining (Gat et al., 2022; H. Wang et al., 2022). Why it generally leads to gains in performance, it is heavy in terms of computation as enforcing new invariances requires a full pretraining, and the representations may still fail facing alterations and distortions that were not applied during pretraining. A more efficient way to deal with unseen acoustic conditions is domain adaptation after the pretraining. Two techniques were privileged in the literature.

First, continual pretraining allows the use of target-domain-related unlabeled audio, without fully retraining the self-supervised models (Kessler et al., 2021). It allows one to adapt the representations to the new domain without forgetting what has been learned on the massive datasets it is generally trained on. Second, domain adversarial fine-tuning has also been a popular choice (K. P. Huang et al., 2022b). It consists in making the latent representations of audio samples coming from different domains undistinguishable through a penalty loss associated with the success of a classifier to detect the domain of the samples.

We also propose in this thesis, in Chapter 4, a method allowing for better adaptation in case of reduced available annotated downstream data, using automatic data augmentation for acoustic conditions cloning.

1.7.2 Task-Coverage

Although the first popular self-supervised representations were meant for speech recognition, comprehensive benchmarks, such as SUPERB and SUPERB-SG (Tsai et al., 2022), have shown that popular self-supervised models allow substantial gains in performance compared to Mel spectrograms on a very wide range of speech tasks. While the former has focused on discriminative tasks such as speech, speaker, or emotion recognition, the latter explored more generative tasks, in the sense here of tasks with audio as the output, such as speech enhancement or voice conversion.

However, an important observation of these two benchmarks is the variance of the relative gains. Gains in synthesis tasks are largely more limited than those for discriminative tasks. A clustering of the tasks, performed based on the performance of the considered models, shows this difference with recognition tasks all in the same clusters, and synthesis ones separated into one-element clusters. The steady link between speech SSL and ASR performance is also highlighted, with gains in performance on ASR being among the highest compared to hand-crafted features.

There are many tracks, explored and yet to explore, for the improvement of the task coverage of SSL methods. One of them is pretraining data selection, as current models are mainly trained on clean read utterances of speech from studio recordings, with LibriSpeech (Panayotov et al., 2015) and LibriLight (Kahn et al., 2020) as popular pretraining dataset choices. A second track for this is to make pretraining losses less ASR-oriented. In this context, speaker recognition has received much attention in the design of appropriate losses and model designs with ideas ranging from enforcing non-speaker related invariances (Stafylakis et al., 2019) to unsupervised speaker pseudo-labeling (Danwei & Li, 2021).

Unfortunately, the high costs of large self-supervised pretrainings, combined with the surprising performance of already pretrained alternatives, even if they are ASR-oriented, have been hindering research in developing models specifically tailored for other speech tasks. We also harness this opportunity to highlight works that have been trying to explain this “surprising" performance. Concerning speaker recognition, for instance, a recent work (S. Chen, Wu, Wang, et al., 2022) has shown that the masked prediction objective was behind the main improvements for the speaker-related downstream, while the impact of careful pretext-labeling was negligible. Other works have shown that the learned representations are highly correlated to articulatory trajectories (Cho et al., 2023), showing that these models learn a physical grounding of speech production, which explains why it also covers non-phonetic aspects of speech.

1.7.3 Computational Efficiency

The Wav2Vec2 models represented a shift in terms of scale for self-supervised representations. They set what stayed, for a few years, the two main formats and sizes of popular self-supervised encoders, with a “Base" model comprising 12 layers of transformers handling vectors of dimension $d = 768$, and a “Large" one comprising 24 layers of

transformers with $d = 1024$, for a total number of parameters, reaching around $90M$ and $300M$ respectively for the Base and Large versions. These sizes and structures have been similar for a long series of further approaches such as HuBERT, WavLM, Data2Vec, and Wav2Seq (F. Wu et al., 2023).

Following similar trends in other modalities, the performance obtained using “Large" models has been significantly higher than the one with the “Base" counterparts,[2] especially for ASR. But this comes at the cost of expensive trainings and inferences, the latter limiting the deployment of self-supervision-based models in production settings. The trend has not stopped at the two sizes described in the previous paragraph, with the latest foundational models surpassing the billion of parameters (Y. Zhang et al., 2023).

More with Less

A few works have tried to reduce the inference costs of popular models. One source of inefficiency during inference is the convolutional front-ends. Studies have shown that they involve high memory consumption and that they can be replaced with more efficient learned or non-learned front-ends (Lin et al., 2022; Parcollet et al., 2023)(even though the last option contradicts the feature-learning trend). Pruning has also been explored during fine-tuning of large self-supervised models. Fu et al. (2022) have shown that learning binary masks over the weights of the models allows a reduction of the inference computations without significant WER increase.

Knowledge-Distillation Attempts

If the performance drops due to training smaller self-supervised encoders are excessive, distillation is a popular alternative. Neural network distillation is a knowledge transfer technique in machine learning where a large, complex model (teacher) is used to train a smaller model (student) by transferring its knowledge. The goal is to distill the generalization capabilities of the teacher into a more compact student model, making it computationally efficient for deployment while maintaining or even improving performance. When distilling self-supervised models, important choices concern the dimensions of the student model, should the model be shallower (*i.e.* less layers) or thinner (*i.e.* shorter inputs), the distillation loss, the distilling dataset... DistilHuBERT is a popular

[2] It is important to note that, except WavLM with the released WavLM+ version, the “Base" models are also trained on much smaller datasets than their “Large" counterparts.

attempt to distill the HuBERT Base version, reducing the number of transformer layers from 12 to 3 using multi-level distillation losses.

LightHuBERT (Rui et al., 2022) introduced a configurable distilled version, through the Once-for-All approach. Once-for-All (Cai et al., 2020) networks are a family of neural architectures designed to accommodate diverse computational requirements by training a single, versatile model. Through a mixture of training and pruning, different sub-networks can be derived from the OFA model, allowing for efficient deployment across various resource constraints. Focusing on paralinguistic tasks (non-including speech recognition for instance), TRILLSSON (Shor & Venugopalan, 2022) proposed distilled models reaching reasonable performance compared to teachers with models bearing 22M parameters only. While reaching reasonable performance on the other discriminative tasks, distilled models are yet to bridge completely the gap in automatic speech recognition with their teacher models.

1.7.4 Open-source and Reproducibility

Speech research, due to its closeness to lucrative industrial applications, has historically been quite a closed research domain with non-shared recipe secrets, compared to other computer science fields. Partly explainable by the higher importance of data and computing compared to algorithms, and with the impulse given by popular toolkits such as Kaldi (Povey et al., 2011) and more recently SpeechBrain (Ravanelli et al., 2021) and ESPNet (Watanabe et al., 2018), the last decade has seen a large leaning towards sharing algorithms, resources and pretrained models.

Given the high cost of their training, the open release with commercial rights, of Wav2Vec 2.0, HuBERT and further models from Meta AI and Microsoft, was one of the main reasons behind their fast adoption and popularity. The pretraining data has also been publicly shared, whether it was the LibriLight dataset or the GigaSpeech (G. Chen et al., 2021) one used for WavLM. Through the Fairseq library (Ott et al., 2019), Meta researchers shared all the code behind the development of the released models and allowed partial replication and attempts of training smaller versions or versions trained on different speech data.

The release of the weights of EncoDec, a universal audio codec allowing the representation of audio samples as sequences of integers corresponding to embedding identities, led to impressive second-parties models for music, urban sound, and speech synthesis (Kreuk

et al., 2022). However, the latest large Google models such as Best-RQ (Chiu et al., 2022), SoundStream (Zeghidour, Luebs, et al., 2021), or Universal Speech Model (USM) (Y. Zhang et al., 2023), while claiming state-of-the-art results in the corresponding papers, have not been released. Code or data needed for replication are also concealed.

Recently, academic reproduction of large-scale training of popular self-supervised (and even supervised) has been attempted mainly by the Language Technologies Institute Lab at Carnegie Mellon University (W. Chen et al., 2023). Reproducing these models allows for getting a deeper understanding of the factors behind their success, and enables exploration of various changes in the architectures, the optimization, or the training sets. Two examples are particularly noteworthy: first, the reproduction of HuBERT Large (W. Chen et al., 2023), with changes in the dataset leading to the frame clustering. Second, in a more classic supervised setting, the attempt for a reproduction of Whisper (Y. Peng et al., 2023; Radford et al., 2023) is even more challenging, given that the training dataset of the original model has not been revealed.

1.7.5 Disentanglement

The goal of disentanglement in speech representation learning is to create a compact and meaningful representation of speech that captures the underlying structure of the audio signal without entangling different factors. This disentangled representation can be beneficial for various applications, such as automatic speech recognition (ASR), speaker verification, and speech synthesis, as it allows for better control and understanding of the individual components of speech (Pierre et al., 2022).

Probing and Analysis of Self-supervised Representations

There has been first, a decent literature on analyzing the content of the representations that are learned, and how these represent phonetic, semantic, prosodic, and speaker-related information. It is related to disentanglement, as studies have shown that different factors are located in different layers of the self-supervised encoders. For instance, probing using linear probes, mutual information estimation, or canonical correlation analysis (Pasad et al., 2021) tend to show that non-phonetic and non-semantic information such as acoustic and speaker-related hints are lost in the further transformers layers of

W2V2. Experiments on CPC features have also shown that these representations separate languages, genders, and phonetic classes (de Seyssel et al., 2022).

Unsupervised Disentanglement

Unsupervised disentanglement is the process of learning disentangled representations without using human-annotated labels of the causing factors one aims to separate in the learned representations. As discussed in (Locatello et al., 2020), it is theoretically impossible if not relying on human priors advocating a few loss or architectural choices.

As an example of these approaches, ContentVec (Qian et al., 2022), one of the most serious attempts to further disentangle the representations, reproduced HuBERT training but with more speaker-information-free objectives. They use, first, prior knowledge in designing alteration and augmentations that should only change the speaker-identity (such as pitch-shifting), and make the representations invariant to those alterations with a contrastive loss (T. Chen et al., 2020). Second, they use a pretrained voice-conversion module (trained in part with speaker identities, thus making the approach not fully unsupervised) to generate speaker-independent acoustic clusters used as pretext labels.

Unuspervised disentanglement is also sometimes naturally emergent. In a recent work, combining the self-supervised discrete representations with pitch information and speaker embeddings has allowed compelling speech resynthesis and voice-conversion results (Polyak et al., 2021). This hints that these discrete labels are mainly rich in terms of non-speaker and non-prosodic information.

1.8 Conclusion

This section introduced, through a historical sweep and a broad scan of speech representations and self-supervision approaches, the key concepts needed to understand the context of the following works, and appreciate their contributions to community efforts. It also describes succinctly the impact these models have had on speech technology and science and the efforts deployed to evaluate them. It ends with a description of five desired properties for self-supervised models. These properties have played a significant role in shaping the works described in this document:

- In Chapter 2, we present methods for a careful design of downstream-task-oriented pretext-tasks, in a step towards higher **task-coverage** for speech self-supervised representations.
- In Chapter 3, we show how the current main benchmarks are advantaging large SSL encoders leading to **non-efficient** full pipelines.
- Finally, in Chapter 4, two approaches explicitly target domain adaptation and out-of-domain generalization in an attempt to reinforce the **robustness** of self-supervision-based models. The third approach, presented in Section 4.3 aims for **efficient** inferences using the fine-tuning downstream phase to reduce the computations through input or model shrinking.

All the code and data used for these approaches have been released, with pointers gathered in Section 5.2. Most easy-to-use approaches have been added within the SpeechBrain library for the community to exploit and build upon. A notable example is the MP3S benchmark (for Multi-Probe Speech Self-Supervision),[3] shared for the community and open for contributions adding probes, models, or downstream tasks. All this is coherent with our open-sourcedness and reproducibility pledges detailed in the previous sections. Now the stage is set, let us go into the details of the proposed work.

[3] https://github.com/speechbrain/benchmarks/tree/main/benchmarks/MP3S

2

Pretext-task Selection for Speech Self-supervised Speech Representation Learning

L'instinct dicte le devoir et l'intelligence fournit des prétextes pour l'éluder..

- Marcel Proust (Le temps retrouvé)

We begin our study with attempts to motivate the definition of the pretext-task learned towards maximizing the performance on a set of considered speech downstream tasks. We chose to condition the pretext-task definition on a given downstream task of interest, for instance, speech or emotion recognition. This choice is argued and its implications are commented in the following. Precisely, we will define a scoring function that will allow us to rank and optimize the pretext-task defined in pretext-labeling and contrastive learning approaches.

Three main objectives will structure this chapter. After a finer introduction to the specific topic of pretext-task selection, Section 2.3 will define a way to score a pretext-task utility towards solving a downstream task, and show an evaluation of this score in the case of individual pretext-tasks. Building on this function, we show then in Section 2.4 how we can extend the approach to multi-pretext-task selection and weighting, validating this approach on three downstream tasks, and in different configurations. The results will be discussed in further sections, and we will produce an extensive analysis of the robustness of the approach and its computational gains. Finally, we show, in Section 2.9 how these ideas can also be used in contrastive learning settings, where the pretext-task is mainly defined by the data augmentations applied to get different versions of the input audio.

The work presented in this chapter has been the subject of the three following scientific publications:

- **Zaiem, S.**, Parcollet, T., Essid, S. (2021). Conditional independence for pretext task selection in Self-supervised speech representation learning. *Proc. Interspeech 2021,* 2851-2855, doi: 10.21437/Interspeech.2021-1027
- **Zaiem, S.**, Parcollet, T., Essid, S., Heba, A. (2022). "Pretext Tasks Selection for Multitask Self-Supervised Audio Representation Learning," in *IEEE Journal of Selected Topics in Signal Processing*, 2022, doi: 10.1109/JSTSP.2022.3195430. Impact Factor : 7.695
- **Zaiem, S.,** Parcollet, T., Essid, S. (2022). Automatic Data Augmentation Selection and Parametrization in Contrastive Self-Supervised Speech Representation Learning. *Proc. Interspeech 2022*, 669-673, doi: 10.21437/Interspeech.2022-10191

2.1 Introduction

In Chapter 1, we have presented a possible zoology of speech self-supervised models. The numerous existing SSL approaches are mainly characterized by the nature of the pretext tasks they solve. More precisely, these pretext tasks may be defined through the choice of pretext labels, hereafter referred to as *pretext-task labels*. The automatic extraction of multiple pretext-task labels for SSL (*i.e.* from the data itself) is common in many application domains, such as computer vision (Doersch & Zisserman, 2017), music processing (Hung et al., 2019; H.-H. Wu et al., 2021b), speech processing (Pascual et al., 2019a; Shukla et al., 2020). Learning representations through solving multiple pretext-task labels is commonly referred to as *multitask self-supervised learning*.

As demonstrated by (Pascual et al., 2019a; H.-H. Wu et al., 2021b), 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 pretext-tasks labels which *"intuitively"* correspond to these tasks. Unfortunately, there is no clear understanding of the pretext-task label interactions that may occur during a joint optimization process, and therefore, no common practice describing a group selection strategy for pretext-task 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 and there is therefore no evidence that the obtained model is even the best one. This empirical approach can rapidly become intractable with modern SSL architectures which may contain billions of parameters trained on thousands of hours of speech, not to mention the carbon footprint

of such pretext-task label searches. For instance, the self-supervised training of a single state-of-the-art large Wav2vec 2.0 model (Baevski, Zhou, et al., 2020) on $53.2k$ hours of speech currently requires 128 GPUs for 5.2 days.

This chapter aims to provide a clear, efficient, and theoretically motivated procedure for pretext-task label group selection and weighting based on 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 compute devoted to pretext-task label search. Hence, the contributions of this chapter are fivefold:

1. Introduce a theoretically motivated and computationally efficient method for the selection of *groups* of pretext-task label among a set of candidates and with respect to the considered downstream tasks (Sections 2.2 and 2.4).
2. Validate empirically the proposed approach with a first SSL model relying on different sets of pretext-task labels, corresponding to the ones obtained for three considered speech tasks. (Sections 2.5).
3. Extend our method to state-of-the-art architectures such as wav2vec 2.0 to enhance its performance and expose the scaling capabilities of our solution (Section 2.5.6).
4. Perform a thorough study of the robustness and generalization potential of this technique to various changes including type of data, pretraining and finetuning datasets, and pretext-task candidates with an application on instrument classification.
5. Show how the presented method can be extended to automatic view creation policies in contrastive learning settings (Section 2.9).

2.1.1 Background

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 (Arora et al., 2019; J. D. Lee et al., 2020; C. Wei et al., 2020). On the one hand, contrastive learning (Oord et al., 2018; T. Chen et al., 2020; Xiao et al., 2021a) has been advocated both theoretically and empirically

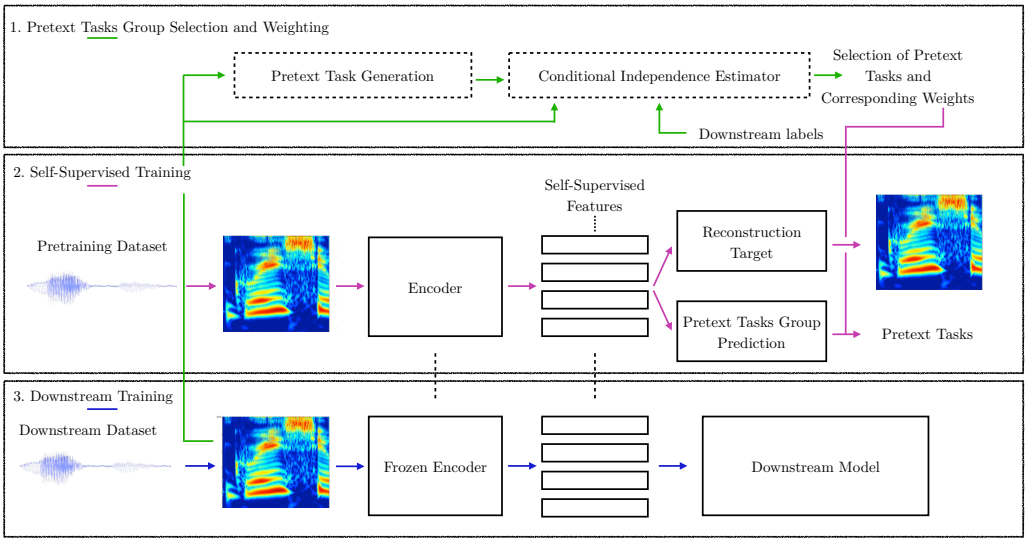

Fig. 2.1.: Illustration of the training pipeline. The three steps are depicted: 1. Selecting the group of pretext-task labels and their corresponding weights; 2. SSL training with the selected pretext task; 3. Training on the downstream task with the pretrained SSL model.

to achieve a balance in the mutual information between alternative representations of the data, keeping just enough shared information to retain the class-related content (**tian**; Bachman et al., 2019; Tschannen et al., 2020). In a recent work (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 et al., 2020). However, all these works have been focusing solely on computer vision or text-related applications, and none of them addressed the multi-task 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 (Z. Chen et al., 2015; Zamir et al., 2018; Dwivedi & Roig, 2019; Shafey et al., 2019). More specifically, combining pretext tasks with SSL has been mainly explored in computer vision and speech (Pascual et al., 2019b; Ravanelli et al., 2020a). Pretext tasks such as Jigsaw (Doersch et al., 2016), colorization and rotation (Gidaris et al., 2018a) have been combined

successfully to improve downstream performance (D. Kim et al., 2018; Shin'ya Yamaguchi et al., n.d.). The two closest works to our line of research are from Lee et al. (J. D. Lee et al., 2020) and Doersch et al. (Doersch & Zisserman, 2017). The former shows that a theoretical link can be established between the conditional independence of the data points and their pretext-task value given the downstream label, 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, while succeeding in presenting a proof-of-concept in multitask speech SSL training, the studies do not offer practical and theoretical solutions to select groups of pretext-task 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 & Elisseeff, 2003). This led to multiple solutions to the feature group selection problem, including LASSO-based techniques (Yuan & Lin, 2006), or multiple kernel formulations (Sonnenburg et al., 2006; Rakotomamonjy et al., 2007). Another type of popular solutions came from the research on submodularity, leading to information-theoretically motivated group selections (K. Wei et al., 2015; Iyer et al., 2022). This has been tried specifically on speech to avoid domain mismatch harming the final downstream performance (Doulaty et al., 2015). Especially on speech, However, these works do not involve any self-supervision, and links between feature selection, self-supervision design, and pretext-task selection are yet to be proved. In the experiments section (Section 2.5), we will consider these lines of work as concurrent baselines.

With this method, we aim at shortening the process of designing SSL models by giving insights on how to select suitable pretext tasks towards solving a given downstream one. We decided to experiment primarily with audio data due to the lack of literature on this domain for multitask SSL, and for the various pretext-task labels available, which are based on decades of signal processing research, before extending to music data. The whole pipeline starting from the acoustic feature extraction to the downstream task scoring follows three major steps summarized in Figure 2.1. First, for every downstream task, our method produces a pretext task selection and weighting. Then, an SSL model is trained, before being used as a feature extractor front-end to one or many downstream tasks.

2.2 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 pretext-task label values and the downstream data points given the downstream labels. Hereafter, we explain the theoretical motivations and describe the computation steps.

2.2.1 Problem Definition and Intuition

Let X, Y and Z be, respectively, the downstream data points, their downstream labels and their pretext-task 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 an automatically computed signal feature, such as the fundamental frequency.

As stated in Section 2.1, Lee et al. (J. D. Lee et al., 2020) linked the utility of a pretext task defined by the prediction of a pretext-task 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 pretext-task 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, which indicates a potential estimator for the pretext task utility.

These results rely on two assumptions that are not upheld in the remainder of this chapter. First, the modeling functions are expected to be linear. Given the complexity of the considered downstream tasks, such as speech and speaker recognition, limiting ourselves to linear modeling would lead to very limited downstream performances. Second, we will estimate the conditional independence using a kernelized independence test, while the quantity involved in the proven bounds is $\epsilon_{CI}^2 = \mathbb{E}[||\mathbb{E}[Z|X] - \mathbb{E}_Y[\mathbb{E}[Z|Y]|X]||^2]$. Computing this quantity is unpractical, especially with varying length speech samples while the method we chose to go with has been successfully tested on sequential data (Gretton et al., 2007).

Intuition. What is the intuition behind the use of conditional independence as a pretext task utility estimator? 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 colorization as the pretext task. In this case, this pretext task would be suited to the downstream one if the final classification label can help imply the colors. For instance, if there are only two classes "Blue skies" and "Yellow deserts", then colorization is an interesting pretext task, as knowing the final label helps a lot for the pretext task, independently of the image. Similarly, if all the classes share the same color palette, colorization may not be an interesting task. (In this toy example, we are ignoring the edge detection aspect of colorization, and only focusing on the color choice part. Obviously, the former aspect plays a role in the success of the colorization pretext task)

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 (Schuller et al., 2007; Loweimi et al., 2015; Serizel et al., 2017; X. Wang et al., 2019; Liu et al., 2020a), 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 of evaluating representation quality intrinsically, *i.e.* independently from the choice of a particular downstream task. A few metrics and tests (Carlin et al., 2011; Schatz et al., 2013; 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 et al., 2020). Finally, recent experiments adapting the self-supervised representation to the speaker identification task have shown substantial improvements compared to task-agnostic representations (S. Chen et al., 2021), validating our intuition that downstream task oriented SSL is an interesting trend towards better downstream performances. We could also mention research in semi-supervised learning that managed to reach results comparable to the best SSL models through leveraging unlabeled data (Hwang et al., 2021; Manohar et al., 2021).

The main issue with CI is the difficulty of computing an estimate of how much two variables are independent given a third one on realistic data (Shah & Peters, 2018). We will start by proposing a simple way to get an estimation of the conditional independence and validate it on individual pretext task selection.

2.2.2 Conditional Independence Estimate Computation

This section details the computation of the conditional independence estimate that is used as a measure of pretext-task label utility. Let $X = \{x_i\}_{i\in\{0,\dots,M\}}$ with x_i being data samples (*e.g.*, Mel-band spectrogram for audio and speech, every x_i here being the Mel-band spectrogram of a given speech segment). Every sample x_i has a corresponding downstream label y_i and an automatically generated pretext-task 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 pretext-task label values.

For independence testing, kernel-based Hilbert Schmidt Independence Criterion (HSIC) (Gretton et al., 2007) is used 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 pretext-task 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 pretext-task labels, respectively. Finally, we perform the independence test for every split group using K_c and L_c and aggregate the estimates with a weighted mean taking into account the number of samples per downstream class. Thus, for two speech samples x_i and x_j, holding two pretext-task label values z_i and z_j, the coefficients of the similarity matrices K_c and L_c are computed as follows:

$$K_{ij} = K(x_i, x_j) = \cos(GD(x_i), GD(x_j)). \tag{2.1}$$

$$L_{ij} = RBF(z_i, z_j), \tag{2.2}$$

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

$$\cos(x, x') = \frac{trace(x^T x')}{||x||.||x'||}. \tag{2.3}$$

$$RBF(z, z') = \exp(-\frac{||z - z'||^2}{2\sigma^2}), \tag{2.4}$$

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 are restricted to the points with c as a downstream label. For each downstream class c, and as in (Gretton et al., 2007), with n_c being the number of points of class c, the HSIC value is given by:

$$HSIC_c(X, Z) = \frac{1}{n_c^2} trace(K_c H_c L_c H_c), \tag{2.5}$$

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 label c, and 1_{n_c} a vector of ones of size $n_c \times 1$. The Gaussian Downsampling method, introduced in (Holzenberger et al., 2018), is a technique used to extract a fixed number of equidistant samples from a time series, and in our case speech samples. More details about it are available in Appendix A.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 pretext-task label should be for self-supervision before fine-tuning on the downstream class. The final value for a given pretext-task label and a downstream task is expressed as:

$$HSIC(X, Z|Y) = \frac{1}{M} \sum_{c \in \mathcal{C}} HSIC_c(X, Z) \times n_c. \tag{2.6}$$

with M being the number of points in the whole dataset.

2.3 Validation on Individual Selection

This section validates individual pretext task selection pretraining the encoder on the English Common Voice dataset and using the learned representations for two downstream tasks; Automatic Speech Recongition using TIMIT, and Speaker Verification using VoxCeleb1.

SSL pretraining. The train set of the English Common Voice dataset (version 5.1) (Ardila et al., 2020) is used for SSL pretraining (700 hours). 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. We remove from Common Voice the sentences lasting more than 10 seconds, as they often contain long silence parts due to open microphones.

Downstream evaluation datasets. TIMIT (Garofolo et al., 1992) is considered for the speech recognition task. It is composed of a standard 462-speakers training set, a 50-speakers development set, and a core test set of 192 sentences for a total of 5 hours of clean speech. For the CI estimation, and to get discrete labels to split on, we cut the sentences at the phone level, using the official transcripts. VoxCeleb1 (Nagrani et al., 2017) is used for the speaker verification task. The training set contains $148,642$ utterances from $1,251$ different speakers. The conditional independence is computed at the phone level for ASR and utterance level for speaker recognition making the assumption that phone segments are entirely independent samples

Pretext-task labels and architecture details. Based on previous work conclusions (Ravanelli et al., 2020a; Jiang et al., 2021), apart from the pretext-task label to be tested, our self-supervised model learns to reconstruct the input Mel spectrograms, and to compute 40-dimensioned MFCC feature vectors. These targets are kept to avoid information loss harming heavily downstream performances. Inspired by the PASE model (Pascual et al., 2019b; Ravanelli et al., 2020a), the model consists of an encoder followed by small predictors limited in capacity (more details on the architectures in Section 2.5.4).

Initial results. Figure 2.2 summarizes the results of the experiment for all the considered pretext-task labels, reporting the CI estimates and the downstream performance for each of the two tasks. It shows the evolution of the conditional independence estimator and the PER and EER, respectively on TIMIT and VoxCeleb. In both figures, the two curves seem to follow the same trajectories showing a monotonic relationship.

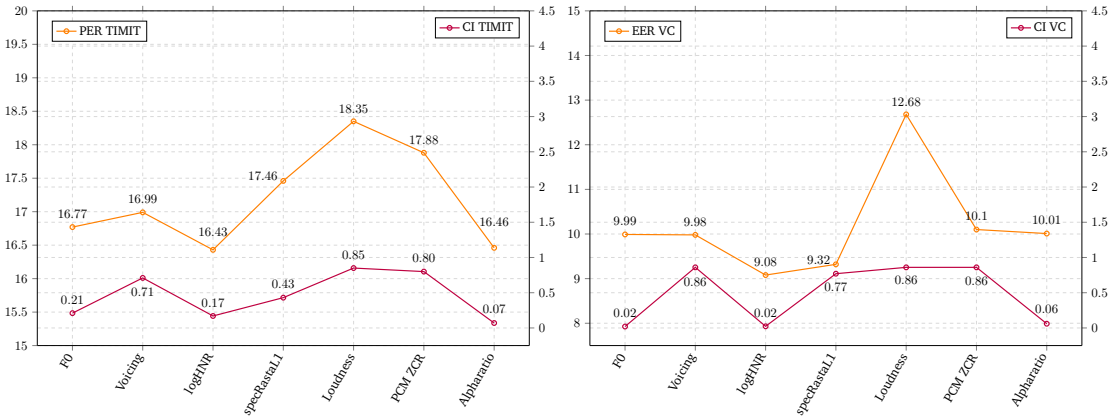

Fig. 2.2.: Left : Phone Error Rate and CI estimate values on TIMIT for every considered pretext-task label — Right: Equal Error Rate and CI estimate values on VoxCeleb for every considered pretext-task label. Error rates appear on the left y axis. We can observe the monotonic relation between the estimator and the downstream errors, particularly for TIMIT.

According to theoretical insights(J. D. Lee et al., 2020), the lower the CI estimate is, the more independent is the pretext-task label from the samples given the labels and the lower should the downstream error be. Hence, we are looking for a monotonic relationship between CI estimates and downstream errors. We consider two classic assessors of monotony: Spearman Correlation and Kendall τ. While Pearson correlation measures the linear correlation between the values, Spearman correlation is a Pearson Correlation on the ranks of the values. Kendall τ considers all the pairs of pretext-task labels and checks whether their order in the CI estimate is the same for the error rate (*i.e* the pair is concordant). The more concordant pairs there are, the higher Kendall τ is.

Spearman correlations reach 0.48 for speaker recognition and a high **0.93** on TIMIT for ASR, while Kendall τ is respectively 0.41 and **0.81** for the two tasks. The correlations between CI and the downstream error are logically positive confirming the work of Lee et al. (J. D. Lee et al., 2020).

2.4 Pretext-task Group Selection and Weighting

While we now are able to predict the utility of every considered pretext task individually, 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 pretext-task

labels and weigh their respective losses to increase or decrease their importance in the self-supervised representation. Such an optimization of the latent representation is expected to provide better downstream performance. Our method minimizes the conditional dependence between the resulting group pretext task, entailing the prediction of a selected pretext-task label group and the downstream samples given the downstream labels.

Given a set of k possible pretext-task labels $(Z_i)_{i\in[k]}$ (this notation means for i an integer between 1 and k), we seek to estimate a set of parameters $(\lambda_i)_{i\in[k]}$ weighting the loss of every pretext-task label Z_i during the pre-training phase. Hence, we define a grouping pretext-task label Z_λ as an orthogonal concatenation of $(Z_i)_{i\in[k]}$ weighted with $(\lambda_i)_{i\in[k]}$: $Z_\lambda = (\lambda_1 Z_1, ..., \lambda_k Z_k)$. These weights $(\lambda_i)_{i\in[k]}$ will be used during the pretraining to scale the loss of every corresponding pretext task

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

$$\begin{aligned}[L_c]_{i,j} &= RBF((Z_\lambda)_i, (Z_\lambda)_j), \\ &= \exp(\frac{-1}{2\sigma^2}\sum_{h=1}^{k}\lambda_h||z_{h,i} - z_{h,j}||_2^2),\end{aligned} \tag{2.7}$$

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

2.4.1 Constraints on the Weights

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

First, the parameters $(\lambda_i)_{i\in[k]}$ must be non-negative, 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, but we are not considering this case here.

Second, the value of the weights must be high enough. Indeed, the presented method for estimating conditional independence assumes that the considered pretext-task 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 $(\lambda_i)_{i\in[k]} = 0$, the utility estimator would be zero and thus the lowest (*i.e.* the best), but it would break the assumption of non-independence between Z and X, and would nullify the participation of the pretext tasks to the training loss. Furthermore, the $HSIC$ value decreases with positive decreasing values of $(\lambda_i)_{i\in[k]}$ and we thus need to ensure that the sum of the weights is significantly higher than zero to ensure that the pretext tasks are significantly considered in the pretraining loss.

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 $(\lambda_i)_{i\in[k]}$ is fixed to one and the problem is summarized as follows:

$$\begin{aligned} \min_{\lambda\in\mathbb{R}^k} \quad & HSIC(Z_\lambda, X, Y), s.t. Z_\lambda = (\lambda_1 Z_1, ..., \lambda_k Z_k), \\ & \lambda_i \geq 0, \forall i \in [k], \sum_i \lambda_i = 1. \end{aligned} \tag{2.8}$$

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 $(W_i)_{i\in[k]}$. Softmax enforces the conditions while being differentiable and the problem becomes:

$$\begin{aligned} \min_{W\in\mathbb{R}^k} \quad & HSIC(Z_\lambda, X, Y), s.t \lambda = Softmax(W), \\ & Z_\lambda = (\lambda_1 Z_1, ..., \lambda_k Z_k). \end{aligned} \tag{2.9}$$

2.4.2 Weights Sparsity

Another trait that is desirable for the weighting vector is sparsity. If a few pretext-task 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 pretext-task 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 & Lin, 2006). To ensure the sparsity of the output

weighting vector, while maintaining the desired property of differentiability, we choose the sparsemax function (Martins & Astudillo, 2016) to replace softmax in equation (2.9).

2.5 Experimental Setup

This section details the experiments validating the introduced group selection and weighting strategy, showing its addition to state-of-the-art predictive coding approaches, and testing its robustness. It, first, describes the selection and weighting processes (Section 2.5.1), the SSL models (Section 2.5.2), the downstream tasks (Section 2.5.3), the obtained results (Section 2.6). Then, it shows how the technique can improve wav2vec2.0 (Baevski, Zhou, et al., 2020) embeddings (Section 2.5.6).

2.5.1 Group Selection and Weighting

In the first attempt, the same pretext tasks presented in Table 2.1 are used for the group selection and weighting experiments. According to Figure 2.1 (*step 1*), we group these pretext-task labels based on their weights, *i.e.* by optimizing Eq. (2.9) to obtain the different 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 pretext-task labels together without any weighting (*i.e.* $\lambda = 1$ for all pretext-task labels) as proposed for PASE (Pascual et al., 2019b). 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 pretext-task label selection is thus compared to two well-established baselines:

Feature	**Description**
Loudness	Intensity & approx. loudness
F0	Fundamental Frequency
Voicing	Voicing Decision
Alpha Ratio (Sundberg & Nordenberg, 2006)	Ratio of spectrum intensity % 1 000 Hz
Zero Crossing Rate	Zero crossing number per frame
RastaSpec L1Norm	L1 Norm of Rasta Spectrum (Hermansky et al., 1992)
log HNR (Murphy & Akande, 2005)	log of Harmonicity to Noise Ratio

Tab. 2.1.: Candidate speech pretext-task labels and descriptions.

Models	LibriSpeech *(WER % ↓)*		VoxCeleb1 *(EER % ↓)*	IEMOCAP *(Acc % ↑)*
	No LM	*LM*		
PASE+ (Ravanelli et al., 2020b)	25.11	16.62	11.61	57.86
vq-wav2vec (Baevski, Zhou, et al., 2020)	17.71	12.80	10.38	58.24
Selections				
All	21.98 ± 0.36	11.70 ± 0.27	11.90± 0.32	56.4 ± 1.3
MRMR	18.94 ± 0.34	10.36 ± 0.26	10.56 ± 0.31	59.6 ±1.29
RFE	20.02 ± 0.34	11.42 ± 0.27	11.91 ± 0.33	55.8 ± 1.3
Softmax	**13.17± 0.28**	**8.00 ± 0.23**	9.24 ± 0.29	60.6 ± 1.27
Sparsemax	17.18 ± 0.32	10.41 ± 0.26	**8.63 ± 0.27**	**60.8 ± 1.28**

Tab. 2.2.: Results observed with the proposed selection strategies on the two considered downstream tasks. Word Error Rate (WER) Equal Error Rate (EER), and Accuracy (Acc) are expressed in percentage and used for LibriSpeech 100 hours, VoxCeleb1 and IEMOCAP respectively. ASR results are given with and without Language Modeling (LM). All SSL models contain $16.3M$ neural parameters.

The Maximum Relevance Minimum Redundancy (MRMR) technique (H. Peng et al., 2005) used as a baseline in this experiment relies on the Conditional Independence based estimator. It is 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 a group of $p = 4$ pretext-task labels $(Z)_{i\in[p]}$ maximizing :

$$Score_{MRMR}(Z) = \frac{-1}{p}\sum_{i\in[p]} HSIC(X, Z_i|Y) - \frac{1}{\binom{p}{2}}\sum_{i<j} I(Z_i, Z_j). \tag{2.10}$$

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 pretext-task labels as features, and the least important feature is eliminated. The process is repeated until only 4 pretext-task labels are kept. We use the scikit-learn implementation with the C-Support Vector Classification as the classifier providing the feature importance values with the default scikit-learn hyperparameters.

2.5.2 Self-supervised Training

In the second step of Figure 2.1, the SSL model learns to predict the selected pretext-task labels. For every one of those, the loss is multiplied by the corresponding assigned weight. As for individual pretext-task testing, 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 weighting vector $(\lambda_i)_{i\in[k]}$, the self-supervised loss is defined as:

$$L_{SSL} = MSE_{mel} + MSE_{mfcc} + \sum_{i=1}^{k} \lambda_i \ell_1(Z_i). \tag{2.11}$$

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 pretext-task 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. The encoder and worker architectures are those described in Section 2.5.4.

The SSL model is learned on the training set of the English Common Voice dataset (version 5.1; 700 hours). 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 shown in table 2.4.

2.5.3 Downstream Tasks

Our proposed pretext-task label selection strategy is compared with the two baselines on three different downstream tasks leading to different groups of pretext-task labels: automatic speech recognition (ASR, with LibriSpeech 100 hours) speaker recognition (SR, with VoxCeleb 1), and emotion recognition (ER with IEMOCAP). Datasets and downstream architectures are inspired by the SUPERB benchmark (S.-w. Yang et al., 2021) for self-supervised learning representations and are carefully described in Section 2.5.5.

Prior to downstream training, the SSL model is 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.

2.5.4 Training and Architectures

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 PASE-like 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 predicts 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, an 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).

2.5.5 Downstream Settings: SUPERB

SUPERB (S. Yang et al., 2021) is a recent benchmark for self-supervised representations of speech data. It fixes for every downstream task an architecture to learn the function that maps the representations to the labels. We use the settings of 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.

Speaker recognition details. VoxCeleb1 (Nagrani et al., 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 https://github.com/salah-zaiem/Multitask-pretext-task-selection). A standard XVector model (Snyder et al., 2018a) is trained following the available VoxCeleb SpeechBrain recipe. The extracted speaker embeddings are tested on the enroll and test splits using cosine as a similarity metric. Performance is reported in terms of equal error rate (EER). While architecture details are given in Section 2.5.4, it is worth noticing that the whole pipeline is fully integrated into Speechbrain and can thus easily be extended.

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). For ASR, the decoder is a vanilla 2-layer 1024-unit BLSTM fed with our self-supervised representations and optimized by CTC loss (Graves, 2012) on characters. The decoding process is based on beam-search with and without shallow fusion using the LibriSpeech official 4-gram language model. Performance is expressed in word error rate.

Emotion Recognition. IEMOCAP (Busso, Bulut, Lee, Kazemzadeh, 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 and 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).

2.5.6 Extending Wav2vec 2.0 to Multitask Self-supervision

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 et al., 2018) based techniques like wav2vec 2.0 (Baevski, Zhou, et al., 2020), on the other hand, currently trust most of the leader-boards for speech-related tasks. Recently, Sadhu et al. (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 in a second experiment to:

$$L_{SSL} = L_{W2V} + \sum_{i=1}^{k} \lambda_i \ell_1(Z_i). \tag{2.12}$$

We use the standard *BASE* wav2vec 2.0 first described in (Baevski, Zhou, et al., 2020) as an 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, Zhou, et al., 2020), and results are strictly equivalent. The entire recipe alongside the large set of hyperparameters needed to properly train a wav2vec 2.0 model is released under our repository and will be made available within SpeechBrain afterwards.

We follow the SUPERB benchmark conventions (S. Yang et al., 2021) both at the data and downstream architecture levels. Hence, and conversely, to the previous experiments, the ASR system solely optimizes 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 pretext-task labels from the training, which is one of our objectives. Another experiment is led with a "naive" pretext-task selection where a constant weight of 0.5 is used across all signal-based pretext-tasks. 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.

2.6 Experimental Results

This section details the main experiments validating the proposed approach on speech data. Table 2.2 shows the results of the group selection methods on the three considered downstream tasks, while Table 2.3 shows the impact of adding a careful selection of pretext tasks to Wav2vec 2.0 training loss. The exact weights obtained with our technique, either with the Softmax or Sparsemax function, and their influence on the conditional independence estimator are shown in Appendix A.2.

2.6.1 Group Selection Results

Baselines detailed in Section 2.5.1 are respectively referred to as "*All*", "*RFE*" and "*MRMR*". First, it is clear from the results reported in Table 2.2 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 pretext-task 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 harmful for the final representation. This motivates the need for a better pretext-task label selection strategy such as the one introduced in this chapter, 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, Zhou, 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. For IEMOCAP, *Softmax* and *Sparsemax* weighting still perform the best among all methods. To investigate how strongly improvements are correlated to the task, we took the best learned model for LibriSpeech (i.e. softmax weighting) and fine-tuned it on VoxCeleb1 and IEMOCAP. It reaches an EER of 10.55% and an accuracy of 59.9% respectively. While it performs better than the baselines, the difference between these results and the best-performing selections shows that the weightings are indeed task-related.

2.6.2 Wav2Vec 2.0 Extension Results

Results reported in Table 2.3 show that our approach improves the performance over the standard wav2vec 2.0 framework for every considered downstream task. While adding pretext tasks naively improves the final performance, the difference in performance between the naive selection and the sparsemax weighting shows the benefit of our method in getting the best downstream performance. Unsurprisingly this difference is

Selections	LibriSpeech *(WER % ↓)*		VoxCeleb1 *(EER % ↓)*		IEMOCAP *(Acc % ↑)*	
	Fr.	Fine.	Fr.	Fine.	Fr.	Fine.
wav2vec 2.0 *BASE*	17.93 ± 0.33	10.21 ± 0.25	7.20 ± 0.26	5.35 ± 0.22	56.6 ± 1.2	**74.0 ± 1.16**
wav2vec 2.0 *BASE* + Naive selection	17.23 ±0.32	10.10 ± 0.25	6.80 ± 0.25	**5.05 ± 0.21**	57.4 ± 1.3	73.7 ± 1.16
wav2vec 2.0 *BASE* + Sparsemax	**16.70 ± 0.31**	**9.18 ± 0.24**	**6.57 ± 0.25**	5.30 ± 0.22	**59.5 ± 1.29**	**74.0 ± 1.16**

Tab. 2.3.: Results observed training 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 downstream performance on all three considered tasks. All models contain $100M$ neural parameters.

Selections	LibriSpeech *(WER % ↓)*	
	Fr.	Fine.
wav2vec 2.0 *BASE*	9.88	6.33
wav2vec 2.0 *BASE* + multitask SSL	**9.5**	**6.01**

Tab. 2.4.: Results observed retraining the Wav2Vec2 model with and without weighted pretext tasks using the sparsemax method, on LibriSpeech 960. “Fr." and “Fine." also respectively refer to Frozen and Finetuned settings. Adding selected pretext tasks still improves the downstream performance. All models contain $100M$ neural parameters.

small (though statistically significant in all but one case), as the Wav2Vec 2.0 BASE is already powerful and the additional workers are anyway useful. Here, it is worth noting that the difference in performance compared to the literature mostly comes from the pre-training conditions. For instance, Wav2Vec 2.0 is commonly pre-trained with larger models on LibriSpeech to achieve lower WER on this dataset.

2.7 Robustness Analysis

This section explores the robustness of the method to changes in the pretraining dataset, in the audio data type and in the set of considered pretext tasks.

2.7.1 Pretraining Dataset Robustness

It is common in the speech SSL literature to train on LibriSpeech 960 before fine-tuning on LibriSpeech100. As explained before, we believe that this introduces a bias due to the closeness of pretraining and fine-tuning data. Studies have shown, for instance, that adding the downstream training dataset to the pretraining set of wav2vec 2.0 leads to better downstream word error rates (Wei-Ning et al., 2021). To verify this, we train our best multitask *BASE* wav2vec 2.0 architecture with the best-performing pretext tasks and their weights on LibriSpeech 960. The model follows the exact same training procedure as for Table 2.3. We fine-tune the models on LibriSpeech 100 exactly as it has been done with the other models. Table 2.4 shows the results. Two observations deserve to be noted. First, in this case, also, adding a selected set of pretext tasks improves the final downstream performance in the frozen and finetuned cases. Second, as expected, the results obtained after training on Librispeech960 are better than with CommonVoice, reaching the lowest 6.01% with the fine-tuned version compared to 9.18% in table 2.3.

2.7.2 Task and Pretext-task Change

To further validate the proposed technique and test its robustness to task and data change, the following section will detail experiments led on multi-task self-supervised learning for musical instrument recognition.

Task change: Instrument Recognition In a first phase, the same pretext-tasks are kept, and the weights are computed in a similar way. However, to be closer to the downstream task, we use the AudioSet "Musical Instrument" partition for the SSL training instead of CommonVoice. The partition contains 57052 files for a total duration of 155 hours. To compute the SSL training weights, the Medley-solos-DB instrument classification dataset is used. Two reasons motivate this choice. First, the music excerpts used come from real instrument recordings as opposed to synthesized audio files from MIDI annotations. Second, every file corresponds to a single instrument played solo thus facilitating the CI estimation. We further test the representations learned in a multi-instrument setting with the OpenMIC-2018 dataset. This tests the robustness of our approach when generalizing to a different downstream dataset of a slightly different task. Hence, we start by computing the pretext tasks weights corresponding to Medley-solos-DB and train the encoder using these weights. Then, it is important to note that

the same encoder will be used for the two downstream tasks, *i.e.* Medley-solos-DB and OpenMIC-2018.

Adding new pretext-task candidates In a second time, we study the impact of adding other pretext tasks to the pool of candidates. To investigate this, we select three additional new candidates: mean spectral centroid, mean spectral kurtosis, and Hammarberg Index. After adding these features, the sparsemax weighting is computed as in 2.9. It is interesting to note first, that two of the three features are not selected for pretraining (even when added individually), thus not changing the weighted selection. Mean spectral centroid is the only feature selected for pretraining, lowering the weights of the other selected tasks. We will refer to this experiment as “Sparsemax+" in the results table (Table 2.5).

Are MFCCs essential? A final change is considered. Following the works of Ravanelli et al. (2020a) and their ablation studies, all the experiments considered MFCCs as one of the workers with a fixed unit weight. Furthermore, when studying ablations, MFCC shows the highest contribution. While the Mel spectrogram reconstruction is needed to avoid any information loss, the MFCC worker can be weighted as well or replaced with other common time-frequency-based representations. To explore this choice and its impact, we select four candidates including MFCC, with SpectralFlatnessPerBand (SFPB) (Herre et al., 2001), Octave band signal intensity (OBSI) (Essid, 2005), and Chroma. These features are computed using the Yaafe toolkit (Mathieu et al., 2010).

The kernel used for these features is the same used for the speech samples, *i.e.* gaussian downsampling followed by the Frobenius inner product. As in the previous paragraph, we compute a Sparsemax-based selection on these four candidates along with the initial best selection of weighted pretext-tasks (without the Spectral Centroid addition). The pretraining loss therefore becomes:

$$L_{SSL} = MSE_{mel} + \sum_{i=1}^{l} \mu_i \ell_1(S_i) + \sum_{i=1}^{k} \lambda_i \ell_1(Z_i) \tag{2.13}$$

with $(S_i)_{i\in[l]}$ the spectral representations, and $\sum_{i=1}^{l} \mu_i = 1$. This experiment will be referred to as "Spectral+" in the results.

Downstream datasets and architectures Medley-solos-DB contains 21572 3-second audio clips distributed among 11 classes. OpenMIC-2018 contains $20,000$ musical samples with partial instrument annotation for 20 instruments. Although not every file is labeled for every instrument, each class has at least 500 confirmed positives, and at least $1,500$

Models	Medley-solos *(Acc% ↑)*	OpenMIC-2018 *(mean-F1 ↑)*
PASE+ (Ravanelli et al., 2020a)	None	64.1
Selections		
All	66.2 ± 0.83	62.89
MRMR	62.3 ± 0.85	64.23
RFE	64.6 ± 0.84	62.80
Softmax	**73.5± 0.78**	65.06
Sparsemax	72.6 ± 0.79	**65.39**
Sparsemax+	**76.1± 0.76**	66.0
Spectral+	74.6 ± 0.77	**67.7**

Tab. 2.5.: Results observed with the proposed selection strategies on the two considered downstream instrument recognition tasks. Accuracy on the test set is computed for Medley-solos-DB while the mean F1 Score is shown for OpenMIC. Higher is better for both.

confirmed positives or negatives. We adopt for downstream finetuning, an X-vector-like architecture, similar to the one used for VoxCeleb1 for Medley-solos-DB. For OpenMIC-2018, we use the official baseline technique relying on a random forest classifier for every considered instrument.

The same grouping techniques presented in the previous section are compared here. Results on the two datasets are shown in table 2.5. We highlight the best results with the standard pool of pretext tasks and the best score after the additions. Accuracy on the test set is computed for Medley-solos-DB while the mean F1 Score is shown for OpenMIC following the SSL literature for music classification (H.-H. Wu et al., 2021a). The results follow those on speech processing tasks both for Medley-solos and OpenMIC. This confirms that the method presented generalizes to other types of data, another pretraining dataset, and downstream tasks that are similar to the one used for weights computing. OpenMIC's best selection results are 3 points higher than the selection done in (H.-H. Wu et al., 2021a). Running the experiment on instrument classification with the three additional pretext tasks in the pool of candidates leads to an even better classification reaching 76.1%. This confirms literature findings on the importance of Spectral Centroids in timbre classification. The same model performs better on OpenMIC reaching 66.0 mean F1-score. This suggests that the selection technique is not harmed by adding irrelevant features while adding relevant ones can improve the final results. Finally, replacing the MFCC tasks with a weighted combination of spectral representations achieves a better result than the Sparsemax selection on Medley-solos-DB with a score of 74.6%. It also reaches the best result on OpenMic with 67.7 mean F1-Score.

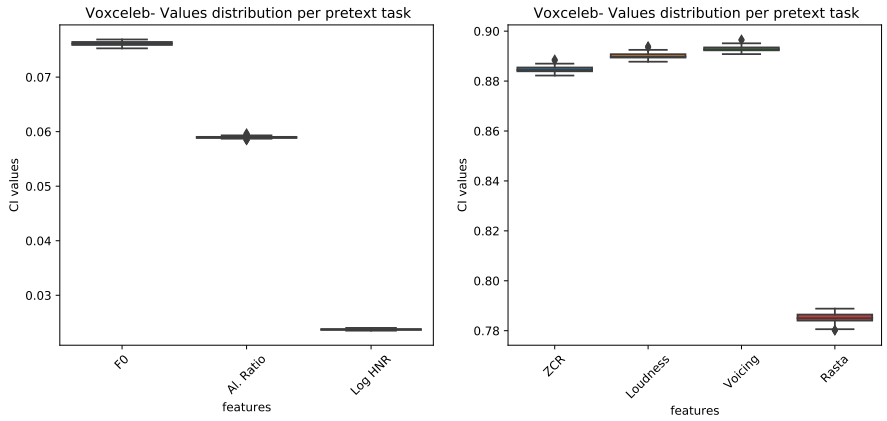

Fig. 2.3.: Boxplots of the CI values for every pretext tasks, when more than 200 speakers are considered. Voicing and Loudness are slightly overlapping, but otherwise, the values are separable. We divide the pretext-tasks in two groups according to their CI values for better visualization of the results.

2.8 Computational Efficiency

After showing the robustness of the approach to different changes including downstream tasks, data types, and reconstruction objectives, this section dives into the efficiency claim of the approach and shows that the score can be computed accurately even with reduced amounts of labeled data.

Efficiency is one of the key motivations of this approach, 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. Two limitations related to the size of the downstream dataset may be faced using our technique. First, very small downstream datasets could not be sufficient for a good estimate of conditional independence. Second, very large downstream datasets may render the CI estimation intractable as the matrices involved get larger. It is important to note here that the computations needed in order to obtain the weights are performed on the downstream dataset, and not on the pretraining unlabeled one. This means that enlarging the unlabeled dataset does not lead to heavier computations.

This section shows experimentally on VoxCeleb1 and Medley-solos-DB that our technique is robust to these two situations. First, we show by taking small subsets of VoxCeleb1 and Medley-solos-DB, that in case of downstream data scarcity, the CI estimations obtained with our method are close to the final estimations, and the ranking of the pretext tasks

is not altered even when we take only 200 speakers among the 1251 in VC. Second, as one of the main motivations of this effort is the reduction of the computation needed to get the best selection of pretext-tasks in self-supervised learning settings, we show that the CI estimation converges quickly with a growing number of speakers considered, and is thus resilient to sampling. Considering one pretext task at a time, we consider subsets of VoxCeleb1 using a growing number of considered speakers ($total = 1251$), and subsets of Medley-solos-DB using an increasing number of samples per considered instrument class. For each of these considered numbers, we run 10 experiments with sampled speakers and music excerpts. We get the CI estimation for every subset and plot the boxplot of the obtained values. Results are shown in Fig. 2.4. We can see that using only 20 speakers exhibits results that are already close to those with 1000 speakers, and the results using 100 audio files per class are close to those with 1500 points per class. Furthermore, we plot the boxplots of CI values obtained using more than 200 speakers to show the separability between the considered features in Fig. 2.3. While values for Voicing and Loudness are slightly overlapping, all the other pretext tasks are already separated and rankable using only 200 random speakers among the whole dataset.

Training the model with a random selection of pretext tasks takes about 30 hours on a single V100 GPU, for the basic model and 40 hours on 24 GPUs for the wav2vec2.0 enhanced one. Finding, through an empirical random search, the best combination of pretext tasks takes a number of experiments that is exponential in the number of pretext tasks considered in the initial set. In contrast, and using as done in this paper, 50 random speakers of VoxCeleb, the whole computation of the optimal weights is performed in a few hours (6 approximatively) when parallelized on 20 CPUs. This runtime is divided into extracting the pretext-task labels, running the Gaussian downsampling of the speech samples (the longest part, as it involves computing the Mel spectrograms before downsampling them), computing the similarity matrices and finally optimizing the HSIC quantity according to the weights. The same durations are observed with LibriSpeech, where the number of points is higher in our experiments, but the speech segments are shorted since we cut them at the word level.

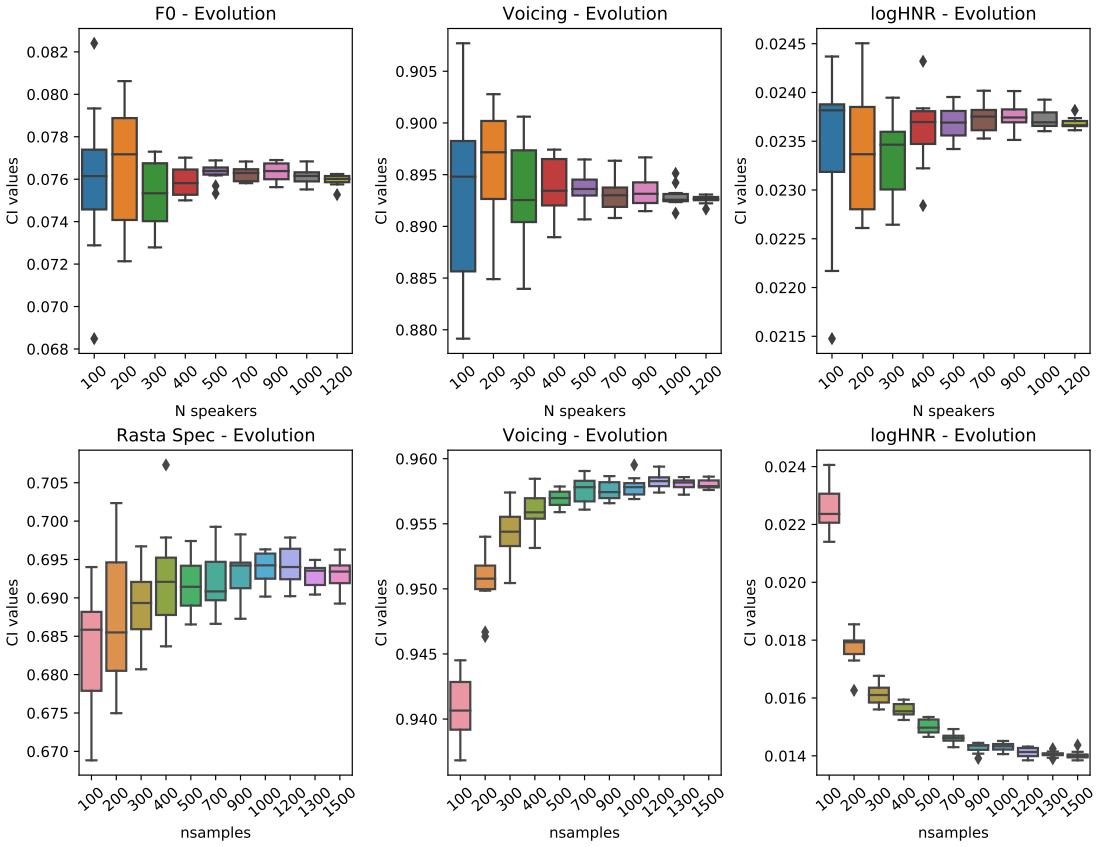

Fig. 2.4.: Evolution of the CI estimation with different numbers of considered speakers for VoxCeleb (First row of plots) and number of samples for Medley (Second row of plots), for three pretext tasks : F0, Voicing and logHNR, Rasta Spech. We can see that the values obtained with 20 speakers and 100 samples per class, while logically exhibiting more variance, are already close to the final values for every pretext task.

2.9 Extension to Constrastive Learning Settings

As described in Section 1.5, pretext-labeling is only one of a set of methods that have been explored to define pretext-tasks leading to learning high-performing speech representations. Another popular family is contrastive learning. Defined in details in Section 1.4.3, it encapsulates successful models such as Wav2Vec 2.0 (Baevski, Zhou, et al., 2020), COLA (Saeed et al., 2021) or Speech SimCLR (Jiang et al., 2021). This section shows that the framework described in this chapter, and validated in individual and multi-task pretext-label selection, can be used in contrastive learning settings.

As we discussed in Section 1.4.3, contrastive learning has been one of the other leading paradigms in speech self-supervised representation learning, especially towards solving paralinguistic classification tasks (Al-Tahan & Mohsenzadeh, 2021; Shor et al., 2022). COLA (COntrastive Learning for Audio) (Saeed et al., 2021) is an audio-adapted version of these models. It consists in learning representations by assigning high similarity to segments extracted from the same audio file and low similarity to segments from different files. The learned representations are then fed to downstream models solving tasks. However, unlike similar approaches in the computer vision literature (T. Chen et al., 2020), COLA does not explore the use of data augmentation to enforce further invariances in the representations. This section explores this use and its variation with the considered downstream task.

In this context, the creation of different versions, often called "views", of a given data point through data augmentation is an essential part of various self-supervised approaches (T. Chen et al., 2020; Grill et al., 2020). On speech data, Kharitonov *and al.* (Kharitonov et al., 2021) have shown that using data augmentation to alter the data during Contrastive Predictive Coding (CPC) (Oord et al., 2018; Rivière et al., 2020) training improves the downstream ASR performance. Two works may be considered as close to the purpose of this paper. First, in image classification settings, adapting the augmentation distribution used in the contrastive pretraining to the downstream classification task has proven effective (Li et al., 2021; Xiao et al., 2021b). This is particularly true when certain differences, to which the representations are trained to be invariant, are crucial for distinguishing the downstream classes. Second, experiments led on contrastive representations (COLA-based) on sound classification show that augmenting the cut segments leads to better results and that the set of best-performing augmentations is downstream task dependent (Emami et al., 2021). Nonetheless, while ablation studies are conducted on the selected augmentations, no prior justification of the choices is developed, making the selection rely on computationally heavy empirical exploration.

Finally, a few works have attempted to define how views should be created in contrastive learning settings (Arora et al., 2019; Tian et al., 2020), and thus which and how augmentations should be used. However, and to the best of our knowledge, there is no attempt to theoretically motivate data augmentation in self-supervised settings on speech or audio data. This work will rely on the COLA approach as it is one of the closest to vanilla contrastive learning, and it did not explore the use of data augmentation on speech. It is, nonetheless, perfectly transferable to other contrastive approaches. If we

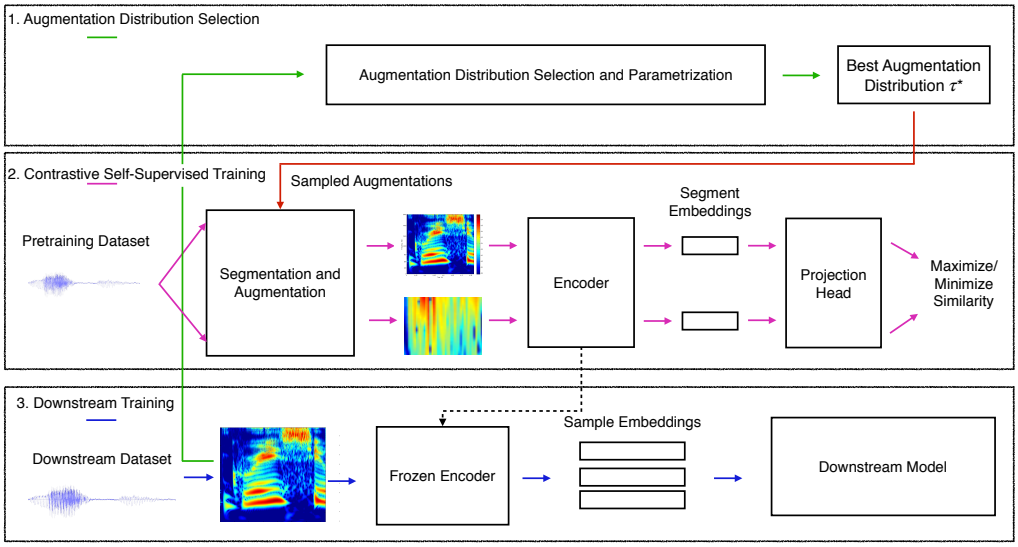

Fig. 2.5.: The three steps of the validation process. (a) select the best augmentation distribution. (b) contrastive pretraining altering the input points with the selected augmentation. (c) use the learned speech representations as input for downstream finetuning.

were to rely only on empirical testing, evaluating a single augmentation distribution would require two full trainings, the self-supervised one and the downstream one. In the specific case of this paper, a single pretraining takes 2 days on a V100 GPU. The method we present prevents this, allowing for an efficient selection of an appropriate data augmentation distribution. The contributions of this section are thus twofold :

1. To highlight the impact of data augmentation on contrastive self-supervised speech representation learning.

2. To propose a method that selects a distribution on the choice of augmentations and their parametrization according to the downstream task of interest, validated on two different downstream tasks. The selected augmentations are qualitatively linked to the recording conditions.

Figure 4.1 presents an overview of the led experiments, summarizing the three steps conducted for every downstream task. First, an augmentation distribution is selected (Section 2.9.1). Second, representations are learned through contrastive pre-training using the selected augmentation distribution (Section 2.9.2). Finally, the learned rep-

resentations are fed to the downstream model to solve the considered task (Section 2.9.2).

2.9.1 Selecting the Augmentation Distribution

This section details the method developed to find a data augmentation distribution for the contrastive learning part, suitable to the final downstream task of interest. It starts by detailing the theoretical motivations behind the method, before delving into the technical details of the implementation.

Theoretical Motivation

In this section, we extend the findings developed for pretext-label selection, described in the previous sections, to the contrastive learning settings through the following steps. The key step consists in considering that in the contrastive learning setting, the pretext task of assigning high similarity to segments originating from the same file can be seen as the prediction, given a random augmented segment, of the original file it was generated from. If a model is able to predict this ID, then it can maximize the similarity of points with the same original ID and minimize the similarity between the other couples. We define an augmentation distribution τ as the set of parameters defining how a chain of augmentations is sampled during training to be applied to the upcoming data points. More precisely, every distribution τ is represented as a vector of $P = 14$ parameters, where every parameter $(\tau(p))_{1 \leq p \leq P}$ is either the probability of applying an augmentation or a boundary for a uniform law from which an augmentation's internal parameter (e.g. room scale) is sampled. With X the speech samples and τ a distribution of augmentations, we define $X' = f(X, \tau)$ with f a function that randomly cuts segments from the speech samples and applies augmentations sampled from τ on them. Given a downstream dataset of samples (X, Y) and an augmentation distribution τ, we can generate N augmented segments per speech sample to get the augmented set of data points X'. To find the optimal augmentation distribution τ^* we resort to minimizing the HSIC quantity with the augmented dataset $X' = f(X, \tau)$ according to:

$$\tau^* = \arg\min_{\tau} HSIC(f(X, \tau), Z|Y) \tag{2.14}$$

with (X, Y) the downstream data points and labels, and Z the pretext labels corresponding here for every augmented view of a speech sample to the ID of the speech sample it originates from.

Implementation

In this section, we chose to limit ourselves to the set of augmentations used in (Kharitonov et al., 2021) for two reasons. First, they have shown effectiveness with the contrastive predictive coding approach improving the final discrimination performances. Second, they are easily implemented within PyTorch using the WavAugment library. Hence, five augmentations are considered: time dropping(Park et al., 2019), pitch shifting (Lent, 1989), reverberation, clipping and band rejection(Park et al., 2019). The first parameters concern the probability of applying each one of the considered augmentations. The second set of parameters are related to those of the chosen augmentations in terms of signal effects; these are described in Table 2.6.

Since the considered augmentations are not differentiable, to minimize the HSIC test described above, we resort to a random search, sampling random distributions and selecting the one with the lowest dependence scoring. It is important to note here, that this phase does not involve any training, and is largely more efficient than thorough testing of the distributions, as a computation takes 3 hours on 20 CPUs. More precisely, for every considered downstream task, we first sample $p = 100$ parametrizations $(\tau_i)_{i\in[1,p]}$. For every parametrization τ_i, we compute the HSIC quantity in Eq.(2.14) following two steps. First, computing the augmented set $X'_i = f(X, \tau_i)$, by computing $N = 20$ views of every speech sample in X. Then, computing $HSIC(X', Z|Y)$ following the technique described in (Zaiem et al., 2021). For every downstream task, the augmentation distribution with the lowest conditional dependence value is selected and will be used during the pretraining to train the encoder that will be exploited as a feature extractor in the downstream training.

2.9.2 Experimental Setup

This section describes the experiments led to validate the proposed approach and the selected augmentation distributions. It starts by describing the details of the contrastive learning phase before reporting the downstream finetuning conditions.

Name	Description	Range
Room scale min	Min room size	[0,30]
Room scale max	Max room size	[30,100]
Band Scaler	Scales the rejected band	[0,1]
Pitch Shift Max	Amplitude of a pitch shift	[150,450]
Pitch Quick pr.	Speeds pitch shifting	[0,1]
Clip Min	Minimal clip factor	[0.3, 0.6]
Clip Max	Maximal clip factor	[0.6, 1]
Timedrop max	Size of a time dropout	[30-150] ms

Tab. 2.6.: Descriptions and ranges of the considered parameters.

Contrastive Learning

As shown in Figure 4.1, during the contrastive pre-training, we start by extracting two random segments from every speech sample of a given batch. These segments are then altered using the considered augmentation distribution before being fed to the encoder. Our pretraining model takes as input the speech samples as 64-Mel band spectrograms. The frame size is $25ms$ and the hop size $10ms$. As in COLA, the encoder is an EfficientNet-B0 (Tan & Le, 2019), a lightweight convolutional neural network. We cut from the input speech samples 1-second long segments that are augmented using the considered augmentation distribution. Fixing the length of the extracted segments allows the use of EfficientNet-B0 even though it has been originally proposed for computer vision, as fixed-length Mel-spectrograms have a 2D structure similar to image inputs. The encoder applies a global time-pooling at its final layer to get a 1280-dimensioned embedding h that represents the whole segment and that will be the one used for downstream finetuning. During the pretraining phase, this embedding is then projected with a dense layer followed by a layer normalization and a hyperbolic tangent activation to a 512-sized vector v. Learning consists of maximizing the similarity of segments originating from the same file while minimizing that of those that do not. As suggested by the final results obtained with COLA, the similarity is computed using the bilinear similarity. More precisely, if g is the function regrouping the encoder and the projection head, x_1 and x_2 two speech segments, and W the bilinear parameters, then the similarity function is $s(x_1, x_2) = g(x_1)^T W g(x_2)$. The input is a batch of size B of distinct speech files that we denote $(x_i)_{i\in[1,B]}$, and a selected augmentation distribution τ from which we can sample at each iteration two augmentation functions A_τ and A'_τ. From each speech sample, two random segments of length 1 second are cut. The first is altered using A_τ while the

second undergoes the A'_τ alteration, leading to two sets $(\tilde{x}_i)_{i\in[1,B]}$ and $(\tilde{x}'_i)_{i\in[1,B]}$. Finally, the loss function for pretraining is the multi-class cross entropy over the bilinear similarity scores, with s the similarity function defined above:

$$\mathcal{L} = -\log \frac{e^{s(\tilde{x}_i,\tilde{x}'_i)}}{e^{s(\tilde{x}_i,\tilde{x}'_i)} + \sum_{j\neq i} e^{s(\tilde{x}_i,\tilde{x}'_j)}}. \tag{2.15}$$

Pretraining dataset. The train set of the English Common Voice dataset (version 8.0) (Ardila et al., 2020) is again used for SSL pretraining. It is important to note that since the COLA embeddings were originally introduced to set non-speech tasks as well, they were trained on AudioSet (Gemmeke et al., 2017), which contains speech and non-speech utterances. Since we will be only working on speech downstream tasks, we selected only speech samples for pretraining. We also use a 1024 batch size. All the models are pre-trained for 100 epochs with ADAM and a 10^{-4} learning rate.

Downstream Fine-tuning

Two downstream tasks are considered: speaker identification and language identification. Two reasons motivate this choice. First, among the list of tasks COLA was applied, we chose the two downstream tasks exhibiting the largest room for improvement. Second, we wanted two tasks that would require different aspects of the considered speech signal, thus maybe requiring different sets of augmentations. A study validating this assumption is provided in Section 2.9.3. VoxCeleb1 (Nagrani et al., 2017) is again used for the speaker recognition task while VoxForge (MacLean, 2018) is used for language identification. 6 European languages are present in the 176,438 samples of the VoxForge dataset, two-tenths are kept for validation and testing.

During the downstream finetuning the projection head is discarded and replaced with a linear classifier directly on top of the encoder, following again the SUPERB conditions. The contrastive encoder is frozen during the finetuning phase as we want it to be used solely as a fixed feature extractor to properly assess the impact of our data augmentation selection on the obtained representation. In the COLA paper, the final class prediction is obtained by averaging the predictions of non-overlapping cut segments of a given test utterance. However, we found it more effective to use the mean over the embeddings of overlapping segments. We proceed in this manner: during training and testing, we cut a 1-s long segment every 200ms, encode every segment separately, and then use the

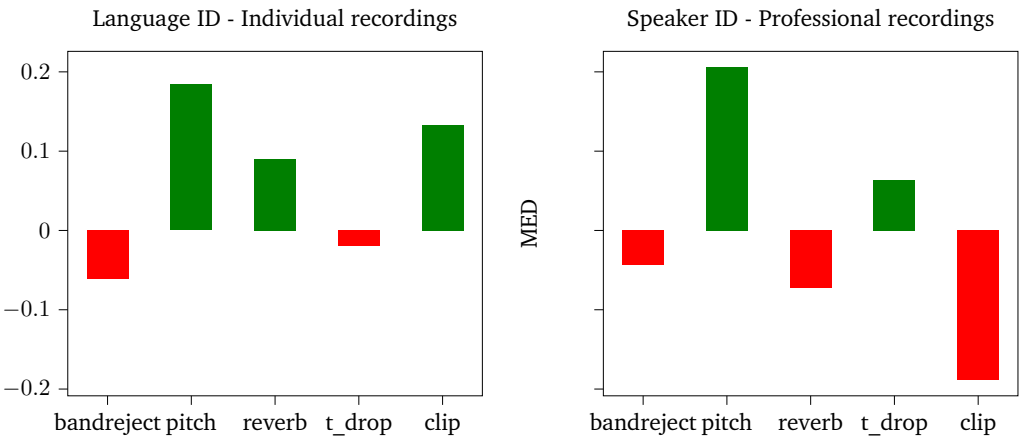

Fig. 2.6.: Difference of the probability of picking an augmentation between the best and worst scoring augmentations, depending on the downstream dataset. Green bars show augmentations that are more likely to get picked for the best scoring distributions for that task. For instance, the far right bars indicate that clipping is an encouraged augmentation on VoxForge, and is discouraged on VoxCeleb1.

Down. Task	**COLA**	**Our Implementations**		
		Without	Basic	Selected
Language ID	71.3	84.9	84.3	**85.2**
Speaker ID	29.9	32.0	45.1	**46.9**

Tab. 2.7.: Results for the two considered downstream tasks. COLA (Saeed et al., 2021) column shows the result of the original paper. "Basic" shows the result with the basic WavAugment recipe. "Selected" shows our approach results.

mean over the encoded representations as a sequence embedding to the classifier. We train on the downstream task for 10 epochs with ADAM with a 10^{-3} learning rate and the additive angular margin loss (Deng et al., 2019) with margin 0.2 and scale 30.

2.9.3 Results and Discussion

Table 2.7 shows the results obtained on the two considered downstream tasks. The “COLA” column shows the results obtained in the original paper. The “Without” column is our implementation of the algorithm without any augmentation during pretraining. “Basic” shows the results reached using the baseline WavAugment augmentation parameters. Finally, the results obtained using the augmentation choice based on the proposed technique can be found in the “Selected” column. The first observation is that the

selected augmentation technique outperforms the baselines on the two considered tasks. For speaker identification, the accuracy obtained with the selected distribution is 46% higher than the non-augmented COLA, and 4% higher than the baseline augmentations. An important point is that in the baseline augmentation setup (i.e. "Basic"), all the augmentations are systematically performed on the input points, thus considerably slowing the pretraining. Indeed, with WavAugment augmentations being CPU-processed, we witnessed that dividing by half the conducted augmentations by lowering their probability, leads to 20% faster trainings.

Discussion

In this part, we will discuss the automatically selected data augmentations, and analyze their dependence on the downstream dataset. We will first study the dependence of the probabilities of applying a given augmentation according to the downstream dataset of interest. Then, we will consider the choice of a few interpretable parameters. This is done through the following procedure: for every downstream task, we start by selecting $k = 10$ best and worst augmentation distributions according to our HSIC scoring. The "Mean Extremal Difference" or "MED" is finally obtained by computing the difference between the two means originating from these two groups *i.e.*, best and worst. More precisely, for an augmentation parameter p:

$$MED(p) = \frac{1}{k}(\sum_{i=0}^{k} \tau_i^{best}(p) - \tau_i^{worst}(p)) \tag{2.16}$$

with τ_i^{best} being the i-th best distribution, τ_i^{worst} being the i-th worst and $\tau(p)$ being the value of parameter p in τ.

Figure 2.6 depicts these values for the probabilities of applying each of the five considered alterations. Green bar means are for positive values, indicating that this augmentation is more likely to be applied in the supposedly best distributions. We observe that clipping and reverberation are more selected for language identification on VoxForge than for speaker identification on VoxCeleb. We think that this is mainly due to the type of recording rather than to the nature of the task. VoxForge samples come from individual contributors who record themselves speaking their native language. The varying recording conditions lead to clipping or heavy reverberation issues, which may be the reason behind the selection of these augmentations in this case. Figure 2.7 shows the mean difference defined above on 3 parameters, which are time-dropping and room-scale boundaries.

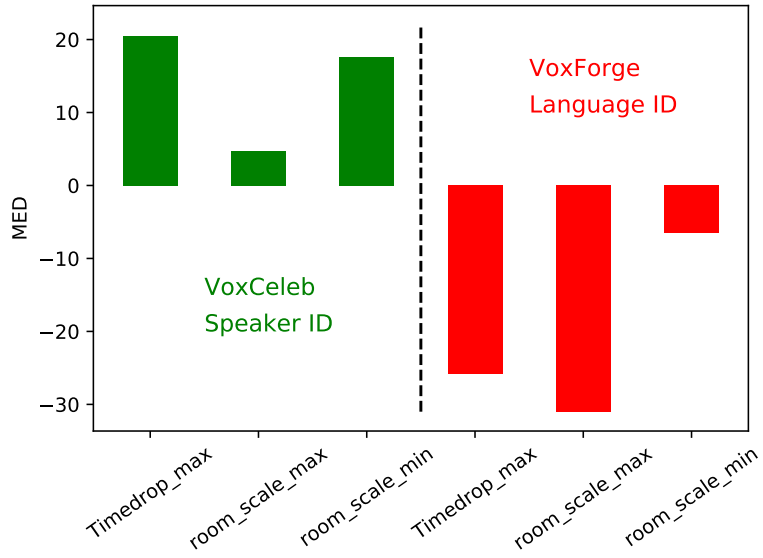

Fig. 2.7.: MED for selected parameters, for every downstream task. Reverb room sizes are coherent with the difference in recording conditions between the two datasets.

Concerning reverberation, it is worth noting that room scales are smaller for VoxForge than for VoxCeleb1, which is once again coherent with the recording conditions, as the first ones are recorded at home, compared to studio conditions. Samples of augmented speech files with various distributions are provided for quantitative comparison by the readers.[1]

2.10 Conclusion

Self-supervised learning of speech representations is a computationally intensive technology, leading to costly trials. Through the methods described in this chapter, we give keys allowing for motivated and optimal choices in two important decisions for speech self-supervised pretraining; pretext-task choice in pretext-label methods and data augmentation in contrastive learning settings. For the former, we presented a method to compute an estimate for conditional independence as a pretext-task utility score. We showed the validity of this score for individual and grouped pretext-task selection. For the latter, we introduced a novel informed method enabling the automatic selection and parametrization of the crucial data augmentation pipeline. Our findings open a

[1] salah-zaiem.github.io/augmentedsamples/

range of possibilities in signal alterations exploration for self-supervision. In this section, these choices are conditioned on a downstream task of interest. This conditioning, while eventually reducing the task coverage of the resulting models, allows to target non-ASR tasks, which has not been thoroughly studied in the literature. However, during the development of these methods, we have been using the community's main evaluation benchmarks and rules to show the performance gains of our approach. This usage, relying on the SUPERB settings described in Section 2.5.5 has raised a few questions and some frustrations around eventual limitations. These limitations are discussed and addressed in the next chapter.

3

Speech Self-Supervised Representations Benchmarking: a Case for Larger Probing Heads

Once again there was the feeling that the ordinary things before one's very eyes were becoming unordinary.

- Tayeb Salih (Season of Migration to the North)

To enable comparisons with competing approaches, the previous section introduced the SUPERB (S.-w. Yang et al., 2021) benchmark and its conditions. This benchmark, along with a few others, emerged as the natural consequence of a few trends:

- With the availability of even larger unlabeled datasets and the variety of introduced paradigms and techniques to train self-supervised representations, a multitude of models have been released.
- The impressive performance gains obtained using self-supervised representations, first on speech recognition tasks, in a second time on a large set of speech technology tasks, made a larger part of the community interested in using this new technology and replacing hand-crafted spectral features in their models and pipelines.
- The size of the most popular models leads naturally to high computational costs of training and inferences with models based on their representations. Consequently, experimenting with these is a costly endeavor both in time and computation. Speech practitioners would therefore need reliable clues and metrics allowing for a motivated choice of representations to use for their tasks.

However, while standardizing the practices for evaluation through comprehensive benchmarks is needed to compare models in the same setting, these standards are a choice of the benchmark developers. This chapter, starting from existing popular benchmarks, explores their robustness precisely to the main choice: downstream decoding architectures. It is divided into two main parts. First, Section 3.3 shows that current benchmarks are very sensitive to the choices made for downstream architectures. Precisely, we highlight that rankings and relative performances of models are much reshuffled when changing these architectures. In a second time, in Section 3.4, we argue that for a set of tasks, the downstream heads currently used should be replaced with more complex alternatives. This is shown through a thorough study of what is expected from evaluation heads and from self-supervision-based pipelines.

The work presented in this chapter has been the subject of the two following scientific publications:

- **Zaiem, S.**, Kemiche, Y., Parcollet, T., Essid, S., & Ravanelli, M. (2023). Speech Self-Supervised Representation Benchmarking: Are We Doing it Right? *in Proc. Interspeech 2023* , Nominated for the Best Student Paper award.
- **Zaiem, S.**, Kemiche, Y., Parcollet, T., Essid, S., & Ravanelli, M. (2023). Speech Self-Supervised Representations Benchmarking: a Case for Larger Probing Heads. *Currently under review in Computer Speech & Language.*

3.1 Introduction

Experimenting with large SSL models is a costly endeavor both in terms of time and computing. The proliferation of approaches for speech SSL (Mohamed et al., 2022) has, therefore, fomented the need for "universal" benchmarks evaluating their performance across multiple downstream tasks. These benchmarks should serve as a means to explore different facets of the speech signal, enabling practitioners to make informed decisions tailored to their specific use cases. Benchmarks also allow the research community to have a common field of comparison for the different proposed SSL techniques and identify areas for improvement. Consequently, there has been a growing proliferation of comprehensive benchmarks in recent years (Evain et al., 2021; S.-w. Yang et al., 2021; T.-h. Feng et al., 2023). These benchmarks offer standardized frameworks for evaluating the effectiveness of speech SSL models and algorithms. They encompass a wide array

of speech applications. Even within a single objective like automatic speech recognition (ASR), they provide various linguistic, acoustic, and prosodic configurations (Tsai et al., 2022).

In prevalent speech SSL benchmarks, the evaluation of self-supervised representations typically involves using downstream decoders that map the frozen representations to the final downstream labels. These downstream probes are generally chosen based on simplicity and limited capacities, such as linear probing for classification tasks or shallow vanilla recurrent neural networks for speech recognition (S.-w. Yang et al., 2021). However, we hypothesize that this benchmarking approach may harm the development of novel SSL technologies in two significant ways. Firstly, the popularity of the main benchmarks, such as SUPERB (S.-w. Yang et al., 2021), has established the considered downstream probes as the standard evaluation setting for any new speech SSL model. The metrics used in these benchmarks also contribute to shape the development of new approaches. Consequently, there may be a tendency to discard models that perform poorly with the selected probes, even if they could potentially excel with other downstream architectures. Secondly, the simplicity of the probes contrasts with the increasing complexity of SSL encoders. Testing with low-capacity probes can lead to an unnecessary transfer of complexity from the probing head, which is intended to be task-specific, to the encoder, which is expected to be more general. This transfer can result in unnecessarily large self-supervised models, leading ultimately to compute-costly inferences. For example, in computer vision, Dubois et al. (Dubois et al., 2022) demonstrated that changing the probe family from linear to multi-layer perceptrons (MLP) leads to different optimal hyperparameter values of SSL models and enables smaller SSL representations.

One potential solution to address these limitations is to explore headless evaluation alternatives that are not tied to specific downstream probes. While a few intrinsic quality assessment metrics for speech embeddings have been proposed (Schatz et al., 2013), their correlation with downstream performances is still uncertain (Algayres et al., 2020). In image classification, Garrido et al. (Garrido et al., 2023) demonstrated a strong correlation between the rank of vision SSL representations and final downstream performance, though the latter performance is obtained using linear probes exclusively. Recognizing these challenges, SUPERB (S.-w. Yang et al., 2021) offers two tracks where researchers can choose their own downstream probes, with or without capacity constraints on the probing architectures. Regrettably, these two tracks have yet to receive any submissions.

This chapter first the dependence of benchmarks on the choice of probing heads. Consequently, given that different probing heads lead to different rankings, we argue that it is important to re-question the current practice followed by prominent benchmarks, where a particular probe is fixed for each task, without a clear justification. In this sense, we provide a more thorough assessment of the benefits of performing the benchmarks with more capacitated probing heads. Precisely, four desired characteristics are assessed: full pipeline performance, inference efficiency, generalization ability and the exploitation of multi-level encoder features. On all these points, our study shows an advantage for higher-capacity probing heads. These ideas and results aim to reshape the way the SSL models are benchmarked, and indirectly, ultimately influence their design towards better rankings in these benchmarks. Hence, the contributions of this work are twofold:

1. We benchmark a set of published state-of-the-art SSL models on various speech tasks, varying the downstream decoders, showing that, except for ASR on Librispeech, the rankings and relative performance are highly impacted by a change in the set of downstream probes (Section 3.2).

2. We provide an extensive study on the impact of selecting higher-capacity decoders on performance, generalization abilities, inference efficiency, and feature-level selection and exploitation (Section 3.4).

3.2 Benchmarking SSL Models: Definition and Protocol

This section formally describes the limitations faced by current speech SSL benchmarks and also details the experimental protocol devised to bring this issue to light.

3.2.1 Problem Definition

Formally, an SSL pipeline consists of two systems: a pre-trained encoder e and a downstream probe f. e is learned through solving a pretext task on unlabeled speech datasets (e.g., Libri-light (Kahn et al., 2020) and LibriSpeech (Panayotov et al., 2015) have been popular choices in the literature), while f is learned for a considered downstream task with its corresponding annotated training dataset. In this framework, the SUPERB benchmark has chosen a probing family $\mathfrak{F}_T$ (*i.e.* a downstream architecture with its hyperparameters, such as an MLP with given number of layers and hidden sizes) for every

considered downstream task T and, for every considered SSL encoder e, it shows a task error rate corresponding to:

$$\min_{f \in \mathfrak{F}_T} E_t(f \circ e); \tag{3.1}$$

with $E_t(f \circ e)$ being the test-set error rate of the SSL pipeline.

However, ideally, as proposed in the *"unconstrained*" track of SUPERB (S. Yang et al., 2021), the shown performance should be:

$$\min_{\mathfrak{F} \in \mathfrak{P}} \min_{f \in \mathfrak{F}} E_t(f \circ e); \tag{3.2}$$

with $\mathfrak{P}$ the set of all probing families. More interestingly, in the *"constrained*" scenario, if we denote by $\mathfrak{C}$ the set of probes that respect a chosen capacity constraint, then the performance of an encoder e could be expressed as follows:

$$\min_{\mathfrak{F} \in \mathfrak{P}} \min_{f \in \mathfrak{F} \cap \mathfrak{C}} E_t(f \circ e). \tag{3.3}$$

Unfortunately, this quantity cannot be computed, as it would require training a model with every known downstream architecture that respects capacity constraints, for each considered encoder and task.

In this study, we aim to investigate whether benchmarking based on the value obtained in Equation (3.1) provides a robust ranking that remains consistent across different probing families. To achieve this, we examine different probing families for each downstream task and analyze whether the rankings and relative differences obtained in the initial experiments remain consistent in the subsequent experiments.

3.2.2 Self-supervised Pretrained Models

For our study, we focused on a subset of state-of-the-art models from the SUPERB benchmark due to their wide adoption within the community. We selected nine SSL models that extract representations directly from the waveform: Wav2vec 2.0 (Baevski, Zhou, et al., 2020), HuBERT (Hsu, Tsai, et al., 2021), WavLM[1] (S. Chen, Wang, et al., 2022), and Data2Vec (Baevski et al., 2022) in both their Base and Large versions. We also included DistilHuBERT (Chang et al., 2022), which is a distilled version of Hubert

[1]We used the Base+ version of WavLM, trained on $94k$ hours of speech data

Base with four times fewer transformer layers. These models share the same frame rate, generating representations of dimension D every 20 ms of audio signal. $D = 1,024$ for the “Large" versions and $D = 768$ for “Base" ones and DistilHuBERT.

These models share similar Transformer-based architectures, but their pretraining pretext tasks vary. Wav2vec2.0 is trained using contrastive predictive coding (CPC), aiming to maximize mutual information between contextual features and predicted future samples. HuBERT and WavLM learn to map unlabeled audio to sequences of pseudo-labels generated through clustering previously generated representations. WavLM introduces training distortions to HuBERT enabling noise-invariant representations. Data2Vec, inspired by teacher-student approaches, employs a masked input view to predict latent representations of the unmasked input data, utilizing a self-distillation setup. We obtained all the pre-trained checkpoints from their respective HuggingFace (HF) official cards (Wolf et al., 2020), except for Wav2vec2.0 Large, for which we used the Fairseq (Ott et al., 2019) checkpoint since the HF version underperformed compared to the results reported in SUPERB.

3.2.3 Downstream Tasks and Datasets

Speech SSL benchmarks attempt to assess universal speech representations by offering a diverse array of tasks that examine various facets of the speech signal. In line with this approach, we introduce seven tasks that cover phonetic, speaker-identity, emotional, and semantic dimensions.

Speech Recognition Tasks. Four speech recognition tasks are considered. For the first one, LibriSpeech (Panayotov et al., 2015) *train-clean-100/dev-clean* subsets are used for training and validation while *test-clean* and *test-other* are kept for testing. The Buckeye dataset (Pitt et al., 2005) is considered as a second ASR task, allowing for testing the ability of the models with fewer labeled data and in a more complex spontaneous setting of English speech. The training, validation, and test splits used in our Buckeye experiments are available in the companion repository with the training set containing approximately 9.5 hours of audio and the test set 1.5 hour. For these two English ASR tasks, we present two sets of results based on the use or not of a language model (LM) during the decoding process. In the experiments labeled “Without LM," we employ greedy decoding. Conversely, the “With LM" experiments utilize the official LibriSpeech 4-gram

language model combined with shallow fusion to the acoustic model. Since low-resource languages are one of the main applications of SSL methods, two low-resource language tasks, extracted from the CommonVoice 11.0 (Ardila et al., 2020) release, are considered: Welsh (*Cymraeg*) and Basque (*Euskera*). To ease reproducibility, we use the splits provided in the CommonVoice release: the Basque train set is 15.8-hour long, with 56 different speakers, while test and dev splits are 10.5 and 9.8-hour long. For Welsh, train, dev and test, splits are respectively, 11, 7.9, and 8 hours with 32 different speakers in the training set. The Word Error Rate (WER) serves as the error metric for all ASR tasks. In all these experiments, the probe is trained using the Connectionist Temporal Classification (CTC) loss at the character level.

Automatic Speaker Verification (ASV). The ASV task consists of a binary classification procedure aimed at determining whether speakers in a pair of utterances are the same. Similar to the SUPERB benchmark, we utilize the VoxCeleb1 train and test splits for this task (Nagrani et al., 2017). It is worthwhile to note that the testing set may include speakers who were not present in the training set. The evaluation metric employed for ASV is the Equal Error Rate (EER).

Emotion Recognition (ER). For ER, we utilize again the IEMOCAP dataset (Busso, Bulut, Lee, Kazemzadeh, et al., 2008). The reported performance represents the mean of 10 runs conducted through cross-validation on 10 folds, where each fold leaves out the data of one speaker for testing purposes.

Intent Classification (IC). While the SUPERB benchmark evaluates the semantic content of SSL representations using the Speech Commands (SC) (Warden, 2018), we employ the more challenging SLURP dataset (Bastianelli et al., 2020) for Intent Classification , as error rates with SC are extremely low. The SLURP collection consists of approximately $72,000$ audio recordings that capture user interactions with a home assistant in single-turn scenarios. The IC task involves classifying each utterance into one of the 18 predefined scenarios, such as "calendar", "email", and "alarm". Classification accuracy serves as the metric for both emotion recognition and intent classification tasks.

Task/Probing Head	First Set	Second Set
LibriSpeech ASR	BiLSTM	Conformer (Gulati et al., 2020)
Buckeye ASR	BiLSTM	ContextNet (Han et al., 2020)
CV Low-Resource ASR	BiLSTM	Linear
Automatic Speaker Verification	X-Vectors (Snyder et al., 2018b)	ECAPA-TDNN (Desplanques et al., 2020)
Emotion Recognition	Time-Pooling + Linear	ECAPA-TDNN (Desplanques et al., 2020)
Intent Classification	Time-Pooling + Linear	BiLSTM + Linear (Lugosch et al., 2019)

Tab. 3.1.: Probes selected for the downstream trainings. More details can be found in the companion repository.

3.2.4 Downstream Probes

This section offers a high-level description of the downstream probes employed in the study. For comprehensive replication of the experiments, detailed information regarding hyperparameters and architectural specifications can be found in the code repository.

Global settings. During the downstream training, the weights of the SSL encoder are kept frozen, learning solely the weights of the downstream decoder. Similarly to SUPERB, we observed that the last-layer representation may not always be optimal. Consequently, we, first, store the representations from all hidden layers of the pre-trained model. These hidden states are then weighted and summed to create the representation forwarded to the decoder. The weights are trained during the downstream process. In order to ensure the validity of our experimental setting, we first reproduced the downstream architectures used in SUPERB during the initial set of experiments. Then, we modified the probes by introducing simpler or more complex alternatives inspired by the relevant literature for each task.

Speech recognition tasks. In the initial set of experiments, aimed at replicating the SUPERB conditions, a vanilla 2-layer Bidirectional LSTM (BiLSTM) with $1,024$ units is utilized. This BiLSTM is followed by a linear layer that maps the latent representations to characters. For the second set of downstream architectures, we employ an encoder-decoder Conformer architecture (Gulati et al., 2020) for the LibriSpeech task. The downstream architecture consists of 12 encoder layers, 4 decoder layers, and 4 attention heads. For the Buckeye task, we employ the convolutional-based ContextNet architecture (Han et al., 2020) with unit strides to maintain the frame rate of the SSL models. In the case of Welsh and Basque from CommonVoice, a two-layer dense neural network

is employed to map each frame representation to the probabilities of the corresponding characters. Additionally, experiments using ContextNet with LibriSpeech are also conducted. The performance of ContextNet and Conformer architectures, which are close to the state-of-the-art on LibriSpeech, motivated their selection as downstream probes. Different probes are selected for ASR tasks to show that eventual variations in performance are not linked to a unique couple of probes.

Automatic speaker verification. In the first experiment, we use the X-vector architecture (Snyder et al., 2018b) with the AM-Softmax loss (F. Wang et al., 2018) for training speaker embeddings. Verification is performed using the cosine similarity backend. In the second experiment, we employ the ECAPA-TDNN neural network (Desplanques et al., 2020), which integrates time-delay neural networks and parallel attention mechanisms to capture temporal dependencies and achieve state-of-the-art results in speaker verification (Desplanques et al., 2020).

Classification tasks. Similar to SUPERB, in the initial set of experiments, we employ linear probing for the classification tasks, namely intent classification and emotion recognition. The representations are first averaged along the time axis and then passed through a linear classification layer. For the second downstream architecture, inspired by state-of-the-art approaches (Y. Wang et al., 2021), we opt for ECAPA-TDNN for emotion recognition. As for intent classification, we follow published work (Lugosch et al., 2019) and utilize two layers of BiLSTM with a hidden size of $1,024$, followed by a linear classifier. This approach allows for considering the order of frame representations, in contrast to using time-pooled features. While the cited works (Lugosch et al., 2019; Desplanques et al., 2020; Y. Wang et al., 2021) employ these architectures on top of handcrafted features (generally log-mel spectrograms), we show in the following that they are still relevant when fed with self-supervised representations. Table 3.1 provides a summary of the probing heads selected for our experiments.

	Models /Tasks	SSL Params.	LibriSpeech train-100 ASR				Buckeye ASR		Welsh	Basque	ASV	ER	IC
	Evaluation Metrics		WER ↓				WER ↓		WER ↓	WER ↓	EER ↓	Acc. ↑	Acc. ↑
	First downstream architectures		LSTM				LSTM		LSTM	LSTM	Xvectors	Pool + Lin.	Pool + Lin.
			Clean	**Other**	**Clean LM**	**Other LM**	**w/o LM**	**with LM**	**Welsh**	**Basque**	**ASV**	**ER**	**IC**
1	DistilHuBERT	23.5M	13.99	34.91	9.96	28.26	35.59	28.29	53.20	46.78	9.10	65.0	46.6
2	Wav2vec 2.0 Base	95M	6.23	14.93	4.86	11.97	24.87	19.48	54.45	51.21	5.29	66.4	59.0
3	Wav2vec 2.0 Large	317.4M	3.72	9.25	3.13	7.48	**20.72**	16.11	45.42	**37.98**	5.69	69.3	66.0
4	HuBERT Base	94.7M	6.24	15.03	5.03	12.31	45.53	26.51	52.92	46.91	4.50	67.5	53.8
5	HuBERT Large	316.6M	3.57	8.12	2.90	6.59	51.30	33.10	51.21	46.15	5.20	71.3	69.9
6	WavLM Base+	94.7M	5.96	14.33	4.84	11.72	42.21	24.41	51.31	46.40	3.74	67.1	57.9
7	WavLM Large	316.6M	3.48	7.37	2.87	5.96	27.31	**14.27**	48.92	41.89	**2.98**	**75.3**	**78.8**
8	Data2vec Base	93.8M	5.30	13.79	4.03	10.97	37.26	30.50	54.00	46.37	5.43	63.0	56.9
9	Data2vec Large	314.3M	**3.10**	**6.50**	**2.58**	**5.38**	22.63	18.63	**44.32**	38.23	4.89	64.1	69.8
	Probe size and inference metrics												
	Downstream Parameters Base		39.9M				39.9M		40.3M	40.3M	7.0M	13.8k	3.1k
	Downstream Parameters Large		42M				42M		42.4M	42.4M	7.7M	18.4k	4.1k
	Second downstream architectures		Conformer				ContextNet		Lin.	Lin	ECAPA	ECAPA	LSTM + Lin.
			Clean	**Other**	**Clean LM**	**Other LM**	**w/o LM**	**with LM**	**Welsh**	**Basque**	**ASV**	**ER**	**IC**
1	DistilHuBERT	23.5M	14.97	36.51	11.54	31.41	58.56	43.61	58.36	53.42	2.85	72.4	74.9
2	Wav2vec 2.0 Base	95M	6.91	15.39	5.09	12.29	30.04	23.04	54.90	42.19	2.82	73.2	77.7
3	Wav2vec 2.0 Large	317.4M	4.32	9.25	3.58	7.03	23.92	18.68	51.66	46.62	3.17	68.4	79.0
4	HuBERT Base	94.7M	6.88	15.68	5.23	12.63	30.44	23.11	43.94	42.83	2.40	**78.2**	79.4
5	HuBERT Large	316.6M	3.96	8.60	**3.10**	6.88	39.39	31.57	41.51	50.22	3.84	71.5	80.1
6	WavLM Base+	94.7M	6.55	14.93	4.98	11.80	27.73	21.69	53.49	43.68	**1.76**	72.6	81.2
7	WavLM Large	316.6M	4.08	8.10	3.13	**6.31**	**15.61**	**12.10**	56.73	**35.49**	1.77	77.4	**85.8**
8	Data2vec Base	93.8M	5.85	14.32	4.53	12.52	40.53	33.45	45.73	45.51	3.75	72.0	73.4
9	Data2vec Large	314.3M	**3.43**	**6.82**	3.27	6.58	25.26	21.50	**41.01**	39.81	2.67	71.3	79.9
	Probe size and inference metrics												
	Downstream Parameters Base		11.2M				32.4M		1.9M	1.9M	9.2M	7.3M	42M
	Downstream Parameters Large		11.2M				32.5M		2.3M	2.3M	9.8M	7.9M	44.1M

Tab. 3.2.: SSL benchmarking results for all tasks and downstream architectures. The number of parameters of the SSL encoder and the probes is shown in the “Params" rows and columns. Upper part corresponds to the results obtained using the first set of probing heads while the bottom part shows these obtained with the second set. Probing heads are compiled in Table 3.1.

3.3 Benchmarking Results and Discussion

Table 3.2 (Horizontal) presents the comprehensive benchmarking results for the different SSL models. The upper and lower sections of the table display the performance achieved by the first and second sets of downstream architectures, respectively. Additionally, the number of neural parameters is reported for both the SSL encoder and downstream decoders. For the latter, only two values are provided per task (*i.e.*,“Base" or “Large") as this number only depends on the dimension of the encoder output representations ($D = 1024$ for “Large" and $D = 768$ for “Base"). In the initial set of experiments, we replicated the SUPERB benchmark conditions for two tasks: LibriSpeech and VoxCeleb1. Notably, our results exhibited a Pearson correlation of 0.99 and 0.97, respectively, with the corresponding results on the SUPERB leaderboard. This high correlation validates our successful replication of the benchmark settings.

To study the impact of a decoder change on the final performances, we compute, for every task, the Pearson and Spearman correlations between the performance metrics obtained with the first downstream architectures and those obtained with the second ones, and collect them in Table 3.3. The Pearson correlation evaluates the linear relationship between the two sets of metrics, while the Spearman one assesses the strength and direction of their monotonic relationship. Correlation metrics close to 1 imply proportional performances and similar rankings between the SSL models used with different probes, making the benchmark robust to the considered downstream change. Correlation metrics close to zero indicate no correlation between the results of the two sets of experiments.

All the models tested demonstrate competitive performances on every downstream task and with every related decoding architecture. With the notable exception of LibriSpeech, all the downstream tasks error metrics vary substantially with changing probes. The mean performance of the SSL candidates with the first and second downstream decoders is presented in the last three columns of Table 3.3. Notably, we observe a significant sensitivity to the choice of decoder as replacing the SUPERB decoder results in relative improvements of up to 46.5% for ASV and 27.3% for IC. This demonstrates the substantial impact that the decoder selection has on the performance of the SSL models. Furthermore, the Spearman and Pearson correlation values computed between the performances with the first and second set of downstream probes are low, despite being positive. This suggests significant variations in relative performances and rankings when comparing the results obtained with the two different downstream decoders. For instance, the Spearman correlation coefficients for ER and IC are only 0.34 and 0.66, respectively. It is noteworthy

Task	Pearson	Spearman	Mean DS1	Mean DS2	Diff (%)	FBANKS DS1	FBANKS DS2
LibriSpeech 1-2	0.99	0.97	5.8	6.48	-11.7	22.56	8.91
Librispeech 1-3	0.99	0.98	5.8	7.03	-21.2	22.56	43.12
Buckeye ASR	0.42	0.56	34.16	32.39	5.2	54.17	78.90
Welsh	0.59	0.62	50.64	74.52	-47.2	99.62	> 100
Basque	0.19	0.15	44.66	69.47	-55.6	> 100	> 100
ASV	0.47	0.75	5.2	2.78	46.5	9.28	3.41
ER	0.22	0.34	67.66	73	7.9	48.51	65.7
IC	0.75	0.66	62.1	79.04	27.3	12.6	42.3

Tab. 3.3.: Correlations (Pearson and Spearman) between the performances achieved with the first and second downstream probes are given for each task. The number in the column name indicates whether the results correspond to the first or second set of probing heads, and “DS" stands for “Downstream". “Mean " columns show the mean performance across all the considered SSL encoders. The “Diff" column presents the relative difference in mean performance between the two architectures. The “FBANKS " columns show the performance on every task with Mel spectrograms as input representations. The difference between “Mean DS" and “FBANKS DS" outlines the performance gain in % from using SSL representations instead of handcrafted ones.

that while the assessment of LibriSpeech performance appears to be robust to decoder changes, this does not hold true for other ASR tasks. In the case of the spontaneous English Buckeye corpus, there is a Spearman correlation of 0.56 and a Pearson correlation of 0.42, while the Basque task exhibits correlations, Pearson and Spearman, of only 0.19 and 0.15. The Buckeye ASR scenario is particular as changing the decoder from BiLSTM to ContextNet leads to improved results for some models and detrimental effects for others. Specifically, the best-performing model, WavLM Large using the second decoder, ranks only fourth when evaluated with the SUPERB settings.

However, we noticed a contrasting pattern in the rankings and performance of the considered SSL encoders on the ASR task using LibriSpeech *train-clean-100*, as shown in Table 3.3. Unlike the other downstream tasks, the rankings and performance only exhibit minor variations when the downstream decoder is changed. To validate this observation, we conducted additional experiments using a third downstream decoder, ContextNet, specifically for this task. The results of this supplementary experiment are presented in Table 3.4, and the correlation values between performances with the first probe and the ContextNet are shown in the second row of Table 3.3. Similarly, no significant differences were observed in the ranking of the SSL candidates. For instance, in all three setups without LM decoding, DistilHuBERT consistently exhibits the lowest performance among the candidates. Furthermore, “Large" versions of the considered candidates consistently outperform their “Base" counterparts on this task, independently of the used

Tasks \Models	SSL Params	Clean	Other	Clean LM	Other LM
DistilHuBERT	23.5M	20.52	43.27	10.44	29.17
Wav2vec 2.0 Base	95M	7.24	15.66	4.73	11.21
Wav2vec 2.0 Large	317.4M	4.35	8.68	03.03	6.86
HuBERT Base	94.7M	7.31	16.00	4.60	11.11
HuBERT Large	316.6M	4.04	8.63	2.98	6.45
WavLM Base+	94.7M	6.73	15.33	4.52	10.84
WavLM Large	316.6M	4.09	8.43	2.94	6.15
Data2vec Base	93.8M	5.46	13.34	3.76	10.04
Data2vec Large	314.3M	**3.50**	**6.94**	**2.56**	**5.36**
Probe size and inference metrics					
Downstream Parameters Base				32.4M	
Downstream Parameters Large				32.5M	

Tab. 3.4.: Word Error Rate (WER %) results of LibriSpeech experiments on the two considered test splits with Contextnet as a third downstream probe. “DS" stands for Downstream.

probing head. Table 3.3 further confirms these findings, revealing high Spearman and Pearson correlations exceeding 0.97 for LibriSpeech, while the highest correlation value observed for other tasks is only 0.75. This discrepancy indicates that the SSL encoders might be biased towards the LibriSpeech ASR task, which is not unexpected given its prominent role as a benchmark dataset and its consistent inclusion in the pretraining process datasets. These results lead us to the conclusion that current SSL benchmarking is highly dependent on the choice of the downstream probes, with the notable exception of LibriSpeech ASR.

3.4 On Limited-capacity Probing Heads

The first section has shown that the rankings and relative performances of the benchmarked self-supervised systems are heavily impacted by a change in the downstream probing heads. The question that naturally arises is whether the common choice of probing heads is justified enough to discourage evaluating with other alternatives. The proposed downstream probes in the prominent SUPERB benchmark were selected based mainly on a simplicity criterion. Choosing simple probing heads is generally justified by the fact that it allows for evaluating only the quality of the pre-trained representations and not the downstream probes learning abilities. In this section, we will show that choosing limited-capacity decoders is not optimal. To prove it, and based on the previous

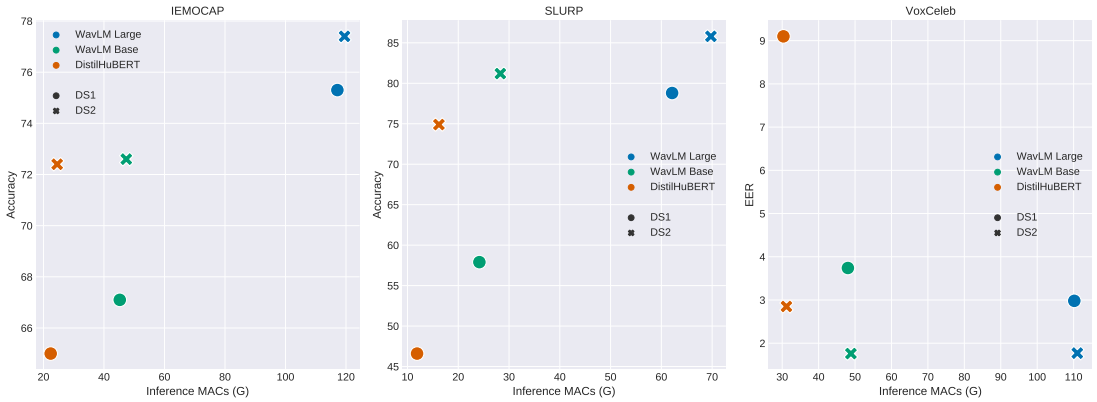

Fig. 3.1.: Performance vs mean total inference cost metrics (in G-MACs) depending on the probing heads used for three models and three different downstream tasks. On all tasks, second downstream probes, larger in capacity, allow smaller SSL models to bridge the gap with bigger ones in term of accuracy with limited additional inference costs. $DS(i)$ for $i \in 1, 2$ corresponds to the results obtained with the $i-th$ set of downstream probes.

experiments and further ones, we will show that larger probing heads: 1) lead to better performance; 2) reduce the error rate gaps between large and smaller SSL encoders, potentially leading to lower inference times; 3) enable the exploitation of multi-level features within the encoders; and 4) do not harm the generalization abilities of the full pipeline.

3.4.1 Performance and Inference Costs

This subsection elaborates two conclusions from the presented results and further computations of inference metrics. First, on most tasks, larger capacity decoders improve significantly the performance, allowing an optimal use of the pretrained representations. Second, larger-capacity probes enable smaller SSL encoders to bridge the performance gap with larger ones, eventually leading to faster inferences.

Concerning performance, Table 3.3 shows that except for the Buckeye ASR task, the mean performance is better with the probes with larger capacities, mainly for Speaker Verification and Intent Classification with respectively 46.5% and 27.3% relative performance improvements (for ASR tasks, the first probe, two layers of BiLSTM, is the largest probe in terms of number of parameters as shown in Table 3.2). Decoders with more capacity seem naturally able to better exploit the benchmarked representations. For

instance, time-pooling the frame-level representations before emotion or intent classification prevents the model from learning to use local or time-ordered signal clues, while it is possible with ECAPA-TDNN or a layer of BiLSTM in the probing head. To know whether the performance increase is imputable to the representations or the probes, we compute the performance of the downstream probes using Mel-scaled spectrograms as the input representation. The spectrograms' extraction is done similarly to the one provided as baseline in the SUPERB benchmark (Tsai et al., 2022). The results are shown in the last two columns of Table 3.3. We can see, first, that the mean performance is significantly better using learned representations than hand-crafted Mel spectrograms, especially for ASR where the final WER is over 100 in three cases. For intent classification, the accuracy using SSL representations, is in average $5x$ better with the first probe and twice higher with the second probe. Moreover, apart for VoxCeleb, where two models perform worse than spectrograms with the second probe, all the representations benchmarked lead to better performances with all probes on all considered tasks. This shows that the lower error rates reached using larger decoders still depend on the quality of the input representations and that the levels of performance reached allow for an informed ranking of those.

Additionally, the findings presented in Table 3.2 shed light on an unexpected outcome when employing low-capacity decoders. With the first set of downstream architectures, the "Large" versions of SSL models consistently outperform their "Base" counterparts. However, this pattern does not hold true with higher-capacity decoders in the second set of probes. For example, the best performances in ASV and ER are achieved using WavLM Base+ and HuBERT Base, respectively. In the context of intent classification, changing the downstream decoder from linear to BiLSTMs results in a significant reduction in the mean absolute difference between the "Base" and "Large" versions' performance, decreasing from 14.23 to 3.28. Again, for emotion recognition, although all four "Large" versions outperform their "Base" counterparts with linear probing, increasing the capacity of the probing head reverses this order for all models except WavLM. Additionally, in the case of ASV, DistilHuBERT achieves better results with an ECAPA decoder than the best-performing model (WavLM Large) with an x-vector-based head, despite having more than 13 times fewer parameters. These findings suggest that using excessively small-capacity heads advantage larger SSL encoders and may have been leading to inflated model sizes.

Since the number of parameters does not present a full picture of the computations involved, the THOP library[2] is used to compute the number of Multiply–Accumulate operations (MACs) implied by the learned models. We compute exactly the mean number of MACs involved in inference (self-supervised feature extraction and downstream decoding) for every sample in the test set. Figure 3.1 shows the number of inference MACs for three models of different sizes and three considered downstream tasks: emotion recognition, intent classification, and speaker verification. For a fair comparison, we select the large models that perform the best on the considered task with the first downstream probe, along with its “Base" counterpart and DistilHuBERT as an even smaller competitor. First, on all three tasks, and for every model, the reached performance is systematically better with bigger decoders. Furthermore, the smallest encoder "DistilHuBERT", while bearing 13 times less parameters than “Large" encoders, reaches a performance with the second decoder that is comparable to the best “Large" model with the first smaller downstream probe. Visually, for every considered model, the x-axis translation between the “DS1" (circle-shaped) and “DS2" points (cross-shaped) shows the MACs quantity increase induced by a bigger decoder head. While the BiLSTM-based decoder is visible on SLURP, the ECAPA-TDNN-based one seems negligible in the two other tasks compared to the self-supervision-based feature extraction costs. The three figures depict clearly both the high performance impact of a small boost in the decoder capacity and its low impact on the total computations needed for inference because of the large cost of feature extraction.

3.4.2 Multi-level Feature Exploitation

The layer-wise content of speech self-supervised representations has been extensively probed throughout the literature (Pasad et al., 2021). These studies generally assess the content with linear probes or with Canonical Correlation Analysis (Pasad et al., 2023). This subsection studies the impact of changing the probing head on the learned weighting of the layers of the models. It concludes that larger probing heads lead to a better exploitation of multi-level features in the considered self-supervised encoders.

As stated in section 3.2.4, during fine-tuning, and in order to cover all the considered downstream tasks, a weighting of the SSL models' layers is learned jointly to the probing heads parameters. With L the number of layers, 1 for the output of the convolutional

[2] github.com/Lyken17/pytorch-OpCounter

front-end and $N-1$ transformer layers in the SSL encoders (3 in total for DistilHuBERT, 13 for “Base" models and 25 for “Large" ones), $(P_i)_{i\in\{1,..,L\}}$ is a learned vector and $W = Softmax(P)$ is the layer weighting vector. Let $(R_i)_{i\in\{1,..,L\}}$ represent, for a given SSL encoder, the N matrices of intermediate embeddings of shape $[T, D]$ with T the number of time frames (50 per second), and D the dimension of the encoder learned representations. Then the input representation decoded by the probing head is:

$$R_{input} = \sum_{i=1}^{L} W_i R_i. \tag{3.4}$$

Figure 3.2 depicts the values (learned during every downstream training) of these weights for the four “Base" models considered in this chapter. The top part shows the learned weights with the first downstream probing heads, and the bottom part shows the second ones. First, it is very interesting to observe that the values of the learned weights seem to depend heavily on the SSL encoder pretraining task. While Data2Vec and Wav2Vec2.0 based, respectively, on masked language modeling and contrastive learning of quantized representations, display different weighting, HuBERT and WavLM, that have similar pretraining tasks, have very similar learned weighting for all the considered tasks, and with the two sets of downstream probes.

Second, it is important to note that the values of the learned weights are heavily impacted by changes in the considered probing head. This is especially the case for non-ASR tasks, and specifically for emotion recognition and intent classification. For these two tasks, with all the self-supervised encoders, only layers above the 9-th are selected with the linear probing approach. However, larger-capacity probes seem to be able to exploit low-level features.

For IEMOCAP, when using the first probing head, *i.e.* time-pooling followed by a linear classifier, the model relies on features from only one high-level layer (the last one for instance, for HuBERT and WavLM). On the contrary, probing with the ECAPA-TDNN—the second probing head considered here—spreads the weights across the different layers. In some cases, the last layers are barely weighted: Data2Vec, for instance, mainly uses the two first ones as shown in the first plot of the third row in Figure 3.2. This tends to indicate that the emotion recognition systems built using the linear probe may be exploiting linguistic content, while the second probe exploits mainly low-level emotion-related features. Concerning intent classification with the SLURP dataset, for HuBERT and WavLM, the main weight moves from the last layer to around the ninth one, while

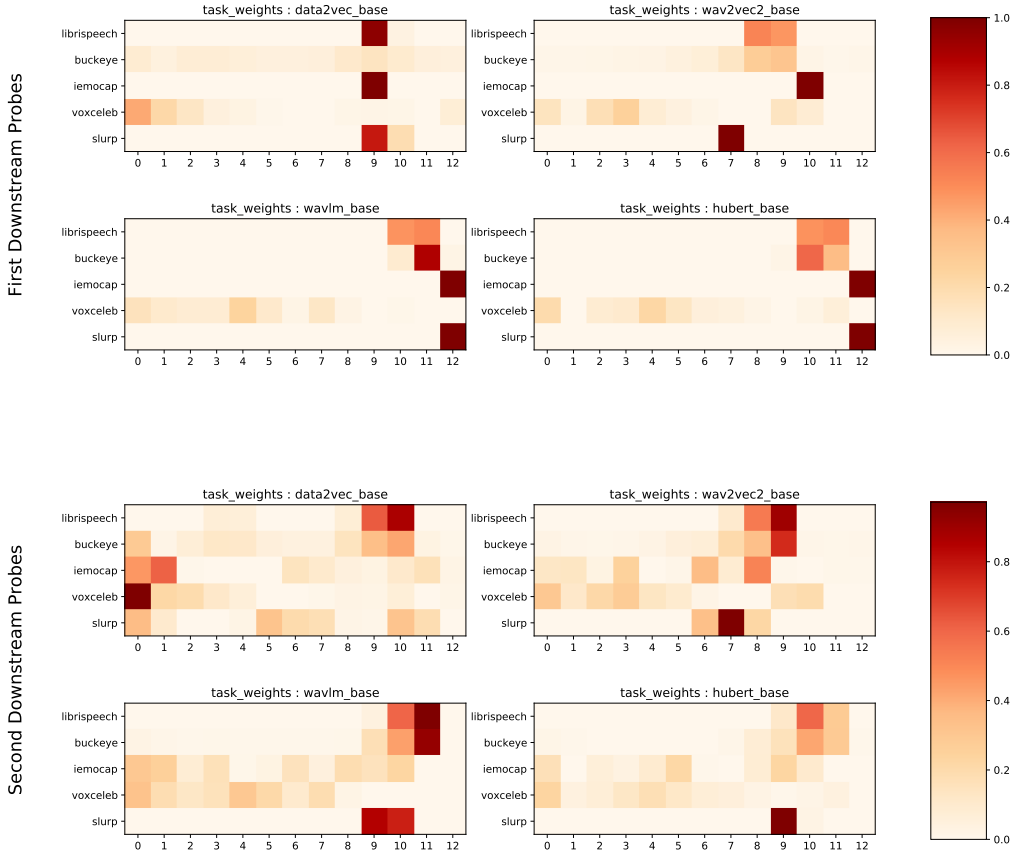

Fig. 3.2.: Values of the layer weights learned during fine-tuning for all “Base" models on the considered tasks. The values on every row sum to 1. The weights obtained with the second downstream probes (bottom part of the figure) are shifted to lower-level layers compared to the first probes ones (top part).

for Data2Vec, the LSTM-based decoder starts using multi-level features, including the first layer, *i.e.* the output of the convolutional front-end. We cannot easily draw a similar conclusion for ASR, where the high-level features are generally the closest to the phonetical content and thus to the nature of the ASR task and seem to be naturally preferred by both the considered decoders. Finally, the VoxCeleb speaker recognition is always selecting low-level features, this is coherent with the layer-wise content probing literature (Pasad et al., 2021), showing the loss of speaker information in high-level features of ASR-oriented self-supervised models.

Building on these observations, we argue that larger-capacity decoders enable the exploitation of multi-level features. In the case of intent classification and emotion recognition, this seems natural given that the first probes, time-pooling followed by a linear classifier, could only exploit features allowing for linearly separable downstream classes. This multi-level extraction may be behind the substantial increase in performance for both intent classification and emotion recognition.

We test this conjecture for emotion recognition with another experiment where one downstream probe is learned using fixed weights obtained with the other one. These results are reported in Table 3.5. Precisely, in this experiment, we fix the weights during the downstream training, with the ones obtained either during the first or second probing. In our set of experiments, for every SSL encoder e, we learn the parameters of a downstream probing head DS and a set of weights for the layers representations W. In Table 3.5, for every SSL encoder e, every column $DS(i)/W(j)$ with $i, j \in 1, 2$, shows the accuracy after decoding with probing head $DS(i)$ but with fixed weights $W(j)$ corresponding to the ones learned initially with $DS(j)$. The results show that, while the larger capacity probing head still performs better than the low capacity ones with their considered weightings, a reasonable part of the performance increase is imputable to the change in the level of features used. With the same ECAPA-TDNN decoder, using multi-level features improves the performance from 68.6 to 73.3 mean accuracy on the 4 SSL encoders considered in this experiment. Another interesting observation is that the first downstream head, time-pooling followed with a linear decoder, is not able to better exploit multi-level features, with very similar performances between the two weightings.

We conclude that probing with larger capacity decoders should be preferred if there is a need to exploit multi-level features, as this allows for increased performance. We will show in the next section that it also has an impact on generalization on out-of-domain samples.

SSL Model / Head/ Weights	DS1/W1	DS1/W2	DS2/W1	DS2/W2
Data2Vec Base	63.0	63.0	62.6	**72.1**
Data2Vec Large	64.0	63.9	67.9	**71.3**
WavLM Base	67.8	67.9	71.6	**72.5**
WavLM Large	75.3	75.3	72.2	**77.6**

Tab. 3.5.: Results of experiments on emotion recognition with fixed layer-weights. The result in column $DS(i)/W(j)$ is the one obtained learning the downstream head of the $i-th$ set with fixed weights corresponding to the ones learned originally with the $j-th$ probing head. The difference between column 3 and 4 shows that the exploitation of multi-level features plays a role in the better performance of DS2.

3.4.3 Generalization Abilities

A major argument for using low-capacity decoders is that they may allow for better generalization. Indeed, the pre-trained representations are learned on massive amounts of data, with a potential higher data heterogeneity, while the decoding head is learned on small annotated datasets with an expected overfitting hazard. Furthermore, multiple studies have examined and shown the generalization robustness of self-supervised representations (T.-h. Feng et al., 2023), which emphasizes, even more, the need to keep this asset. This section aims to show that the models learned with larger capacity decoders are able to generalize as well and even better than their smaller-decoders counterparts, by showing that the performance gains obtained with larger decoders transfer to Out-Of-Domain (OOD) testing samples. Within this scope, we consider the final models obtained with different capacity decoding heads on the considered tasks and test their accuracies on OOD samples, coming from other datasets but having similar downstream classes. This actually enables direct zero-shot generalization performance assessment. Two reasons make two tasks, emotion recognition and speaker verification, relevant for these experiments. First, for both these two tasks, a larger-capacity probing head leads to significantly lower error rates, and we want to test how much this gain is resilient to OOD testing. Second, zero-shot testing requires OOD samples sharing the same labels as the training in-domain set. For ER, several other datasets share, at least partly, the same labels as IEMOCAP (Cao et al., 2014). While speaker verification models trained with VoxCeleb (Nagrani et al., 2017) output a binary label indicating whether two samples come from the same speaker or not, and thus can be tested on any other ASV benchmark, including OOD non-English utterances.

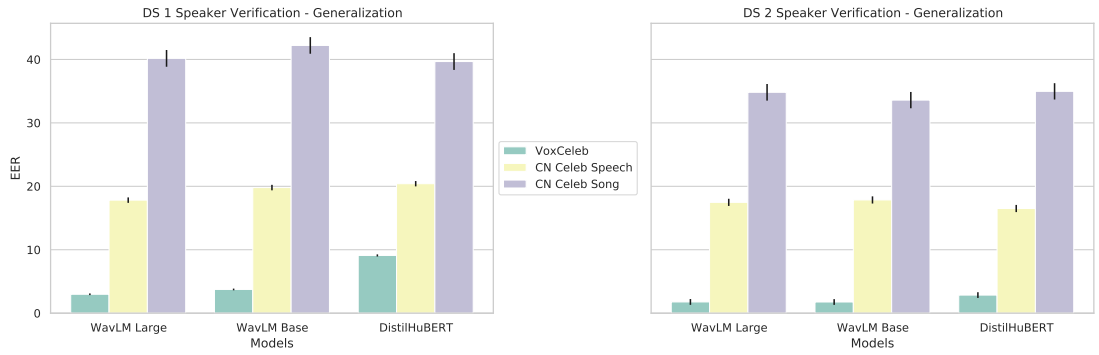

Fig. 3.3.: Generalization performances for automatic speaker verification. CN-Celeb Speech and CN-Celeb Song performances are provided in a zero-shot generalization setting and are not included in the training set. Random performance is at 50 EER, and is not shown for better visualization. Larger probing heads, here ECAPA-TDNN, shown in the right plot, generalize better to out-of-distribution testing samples.

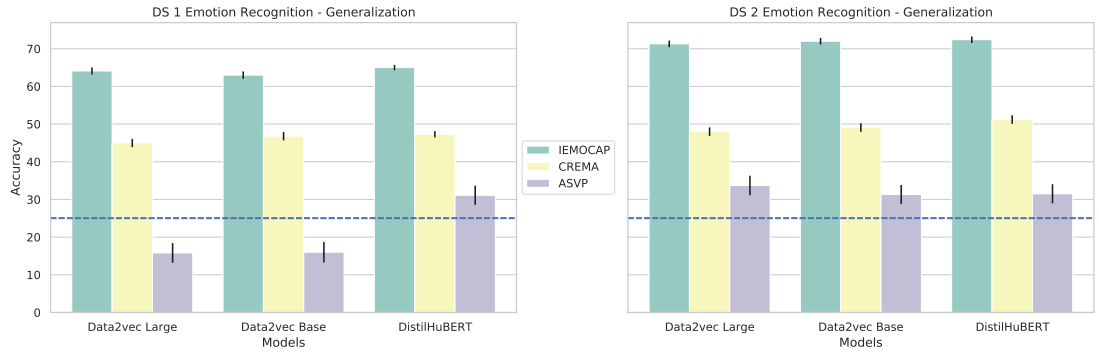

Fig. 3.4.: Generalization performances for emotion recognition. CREMA-D and ASVP-ESD performance is tested in a zero-shot setting. The dashed blue line represents the random accuracy level. Larger probing heads, here ECAPA-TDNN, shown in the right plot, generalize better to out-of-distribution testing samples.

Emotion recognition. To test the generalization abilities of models learned with different decoders, and after training with IEMOCAP as described in Section 4.3.1, we test the models in a zero-shot fashion, without further fine-tuning, on two datasets: CREMA-D (Cao et al., 2014) and ASVP-ESD (Dejoli et al., 2022). CREMA-D is a data set of $7,442$ original clips from 91 English-speaking actors reading sentences using one of six different emotions (*Anger*, *Disgust*, *Fear*, *Happiness*, *Neutrality*, and *Sadness*). ASVP-ESD is a multi-authentic emotional corpus sourced from movies, Youtube channels, and real-life human interactions in natural settings, without any language limitations. The corpus comprises $5,146$ samples, with 60% consisting of non-speech emotional sounds and 40% comprising

speech utterances. For both datasets, only speech elements with labels overlapping with the four IEMOCAP ones (*Angry, Happy, Neutral, Sad*) are considered. For these two corpora, the testing sets are of reduced sizes. So to increase the significance of the reported results, and since the train sets are not used for training, all the splits (train and test) are used for testing. For ASVP-ESD, and to further enforce OOD testing, English samples are removed.

Automatic speaker verification. For speaker verification, the generalization abilities of the models learned on VoxCeleb1, are tested for two out-of-domain scenarios, also in a zero-shot transfer setting. For this, The CN-Celeb dataset (Y. Fan et al., 2019), a comprehensive collection of speaker recognition data, is used. It encompasses over 130,000 utterances from 1,000 Chinese celebrities, spanning 11 diverse genres (interviews, movies, songs...). To further highlight generalization ability, we divide CN-Celeb testing couples into ones that include one singing voice element, and once with only spoken utterances, leading to two generalization testing sets: “CN Celeb Speech" and “CN Celeb Song". The second split is even more challenging in our case, as no singing voice is included in VoxCeleb.

Discussion. Figures 3.3 and 3.4 show the results of these experiments for models built on certain considered SSL encoders. We can note, first, the expected considerable performance loss on the OOD samples, and especially the loss when changing the ER language with ASVP-ESD or testing on singing voice speaker verification with “CN Celeb Song". For both tasks, as stated in previous sections, in domain performance, *.i.e* performance on the test sets of the downstream training datasets, obtained with the second set of larger probing heads are higher than those with SUPERB limited-capacity probes. The two figures further show that this performance gap stands for zero-shot generalization. Concerning emotion recognition, the mean accuracy on the three considered models reaches 49.43 and 32.17 respectively on CREMA-D and ASVP-SED with the ECAPA-TDNN probing head compared with 46.37 and 20.97 with the time-pooling followed with a linear decoder. For speaker verification, enhancing the probing head drives the Equal Error Rate on the “CN Celeb Speech" from 19.34 to 17.27, while it goes from 40.68 to 34.46 on “CN Celeb Song". In subsection 3.4.2, we hypothesized that ER models with the first downstream probes may be using linguistic information since only high-level layers were used. The big drop in performance on ASVP-ESD of Data2Vec “Base" and “Large" models goes in that direction. Changing the language of the inputs leads to catastrophic performance drops. This is not the case for DistilHuBERT as the model only contains three layers. These experiments show that the gain in performance

is not only relevant to in-domain data, but models built on top of frozen SSL encoders reach better out-of-domain zero-shot accuracies with larger-capacity probing heads.

3.5 Conclusion

It is crucial to improve the way the speech community currently benchmarks widely used self-supervised representations. This is important, first because better benchmarks allow SSL users to select properly the models they need for their downstream tasks of interest. Second, it offers the SSL model developers insightful evaluations shaping the training process and decisions. In this chapter, we have shown, by varying the downstream architectures, that the ranking and relative performances of popular self-supervised models heavily depend on the choice of the probing heads. While the community has previously chosen to evaluate the learned representations with limited-capacity decoders, we have revealed, as an additional contribution, that larger-capacity decoders should be preferred in various scenarios. This is motivated by better performances, a reduced performance gap between “Base" and “Large" encoders leading to high (performance/inference costs) ratios, better multi-level feature exploitation, and better out-of-distribution generalization. We hope this diagnosis will support the community in designing new benchmarking approaches and encourage submissions to the SUPERB “Constrained" track described in the introduction or propose new probing heads in the dedicated benchmark section within the SpeechBrain Library.

4 Generalization and Efficiency Using Self-supervised Encoders

Par-delà ce village, d'autres villages, par-delà cette abbaye, d'autres abbayes, par-delà cette forteresse, d'autres forteresses. Et dans chacun de ces châteaux d'idées, de ces masures d'opinions superposés aux masures de bois et aux châteaux de pierre, la vie emmure les fous et ouvre un pertuis aux sages.

MARGUERITE YOURCENAR
L'oeuvre au Noir

In the previously presented evaluations of self-supervised representations, degrees of freedom are limited to avoid noise factors during benchmarking. The encoder weights are frozen and only a weighting of the internal layers is allowed. This amounts to treating the self-supervised encoders as mere feature extractors. With sizes of the encoders ranging from hundreds of millions to a few billion parameters, this approach seems inefficient when a precise downstream task is targeted. This is why in practice, the weights of the encoders are generally fine-tuned during downstream training allowing for lighter downstream heads. As an example, when the encoders are frozen, a simple dense layer cannot be used as a downstream head for speech recognition, as it would not have the context information needed (this explains why the SUPERB benchmark employs a recurrent network for this task). When fine-tuning the weights of the encoder, the transformer layers allow for context to be considered, and thus a simple linear head is enough to perform a given ASR task.

To answer the main interrogations of this thesis, it seems necessary to derive best practices for how self-supervised models are commonly used and not stick to evaluation and

benchmarking settings. Fine-tuning the encoders allows for different interventions and degrees of freedom. This chapter explores a number of them with two objectives in mind: generalization and efficiency. The first objective is tackled with two approaches. First, in Section 4.1, we show how the conditional-independence estimate developed in Chapter 2 can also be used during downstream fine-tuning to clone the acoustic conditions of low-resource target domains. Second, in Section 4.2, through using continual-learning-based approaches during the fine-tuning, we highlight the link between forgetting the pretraining task and out-of-domain generalization. Finally, with a benchmark of different fine-tuning strategies aimed for faster inferences, we show in Section 4.3 that the network specialization induced during the fine-tuning phase allows for shrinking encoders and inputs, leading to faster inferences.

The work presented in this chapter has been the subject of the two following scientific publications:

- **Zaiem, S.**, Parcollet, T., & Essid, S. (2023). Automatic Data Augmentation for Domain Adapted Fine-Tuning of Self-Supervised Speech Representations. in *Proc. Interspeech 2023*.

- **Zaiem, S.**, Algayres, R., Parcollet, T., Essid, S., & Ravanelli, M. (2023). Fine-tuning Strategies for Faster Inference using Speech Self-Supervised Models: A Comparative Study. *ICASSP 2023*-IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

4.1 Acoustic Cloning for Domain Adaptation of Self-supervised Representations

Despite its popularity, self-supervised learning has been shown to suffer from domain mismatch where the fine-tuning samples from the target domain are vastly different from the pretraining ones (Hsu, Sriram, et al., 2021; Riviere et al., 2021). While progress has been made in achieving near-optimal performance on clean datasets such as LibriSpeech, spontaneous speech datasets and non-professionally recorded ones still exhibit lower performance, as displayed in recent speech SSL benchmarks (Evain et al., 2021; Tsai et al., 2022).

To mitigate the performance drop caused by domain mismatch, various domain adaptation techniques have been explored, particularly in transfer learning settings (Olvera et al., 2022). In the self-supervised context, adversarial approaches have been applied during the unsupervised pretraining and tested on speech recognition (Tanaka et al., 2022; Lodagala et al., 2023), emotion recognition (Latif et al., 2022) and speaker recognition (S. Chen et al., 2021). Along with domain adversarial paradigms, Huang *et al.* (2022a) investigated continual learning methods during pretraining. Distinctly, our method does not aim at aligning latent representations but rather transforms the audio waveforms of a neutral dataset to match the acoustic conditions of the target domain using data augmentations, rendering this dataset better suited to the final task in an initial fine-tuning stage.

Thus, we envisage the option of augmenting a supposedly neutral dataset and using it for the first fine-tuning step. The augmentations to be applied and their parameters are chosen in order to optimize the similarity in terms of recording conditions between the modified and the target dataset and hence the final performance. Our method presents three main advantages. First, it enables the use of large and clean available annotated datasets, enhancing the textual diversity of the training corpus. Second, it does not require new pretraining as it directly fine-tunes available SSL models. Finally, it allows an efficient data augmentation exploration, as the selection and parametrization is automatic and does not involve any neural network training. It is, thus, largely more efficient than thorough testing, as scoring 200 augmentation policies takes 3 hours on 10 CPUs, while complete testing of one augmentations distribution necessitates around 20 hours of GPU computations.

In this section, we propose a new method for supervised domain adaptation consisting in applying appropriate signal distortions to a clean labeled dataset used for an initial fine-tuning step. The method is validated with an oracle-simulated experiment and an application with naturally noisy datasets.

Figure 4.1 presents an overview of the method, summarizing the three steps conducted for every considered target dataset. First, and given the labeled target dataset, an augmentation distribution is automatically selected (Section 4.1.1). Second, a first fine-tuning of the self-supervised representation is done, using the neutral dataset distorted with the augmentations selected in the first step. Finally, a second fine-tuning on the small target domain dataset is done leading to the final model that will be evaluated using the target test set (Section 4.1.3).

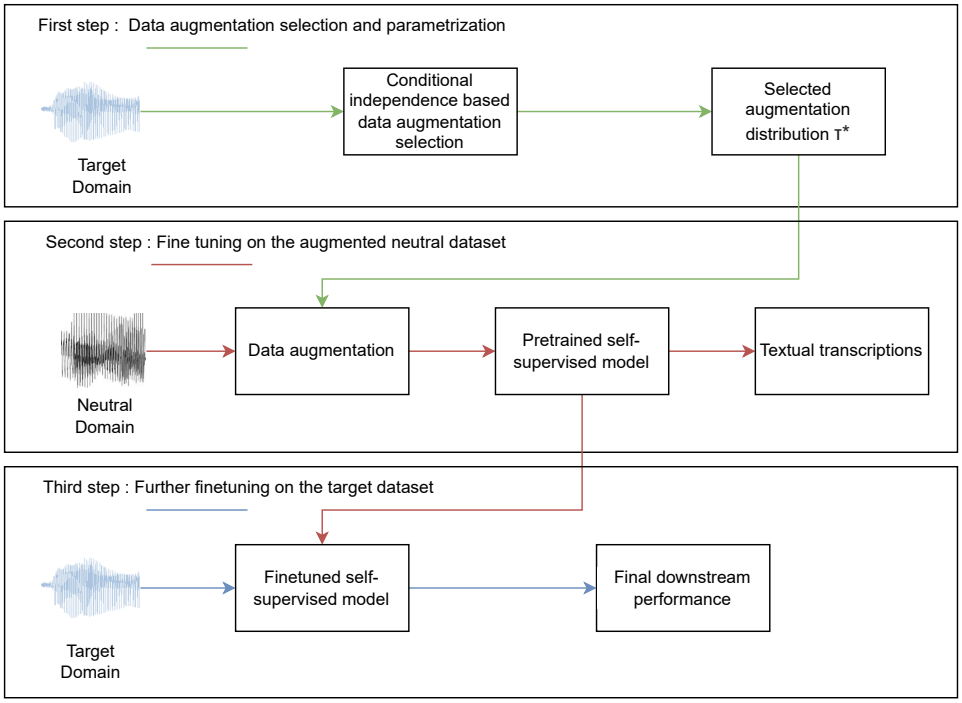

Fig. 4.1.: Summary of the three steps of the method. 1. Starting from the target domain, an augmentation distribution is computed. 2. This distribution is used to distort a neutral dataset for a first fine-tuning. 3. A final fine-tuning is done on the target domain samples.

4.1.1 Selecting the Augmentation Distribution

Given a labeled target speech recognition dataset, our method selects an augmentation distribution that is best suited to its recording conditions. From this distribution, we will sample augmentations to be applied to a larger "clean" dataset which will be used to fine-tune the SSL representations. The goal is to select augmentations bringing the "clean" dataset samples "closer" to those of the target domain, thus leading to better performance on its test sets. This section details the conditional-independence-based method developed to select a data augmentation distribution given the annotated target dataset. It starts by detailing the motivations behind the method, before delving into the technical details of the implementation.

4.1.2 Motivation and Technical Description

Motivation. Inspired by pretext-task selection for speech self-supervised learning, we have shown, in Chapter 2, that conditional independence estimation may be used for automatic data augmentation in contrastive self-supervised learning settings. Furthermore, qualitative analysis, conducted in Section 2.9.3, has indicated that the distortions selected by this technique tend to be close to those of the target downstream dataset.

Let us give an intuition about what happens in these conditional independence computations to understand why it can be useful for domain adaptation as well. Roughly, minimizing the conditional dependence described above maximizes, within the same downstream class, the invariance of distorted samples (*i.e.* views) to the ID of their original speech sample. If a given distortion (for instance, reverberation) is not present in any sample in the original target dataset, randomly applying this distortion would decrease in-class similarity. Inversely, applying augmentations already present in samples in the dataset makes it harder to distinguish their original samples' IDs given the distorted samples and, thus, lowers the conditional dependence estimator. Conditioning on the downstream labels retains the signal clues characterizing the downstream classes since it prevents selecting distortions that are only relevant to one class, as they would reduce in-class similarity in the other classes.

Technical Description. The setting is similar to the one described in Section 2.9. Precisely, let again X and Y be respectively, a set of speech data points and their respective set of downstream labels which are in our case textual transcriptions. With τ an augmentation distribution from which one can sample a chain of augmentations, we compute a distorted dataset $X' = f(X, \tau)$, with f a function that randomly applies augmentations sampled from τ on the speech samples. Specifically, we can generate G augmented versions per speech sample to get the augmented set of data points X', with G a hyperparameter. Every sample x' in X' is a distorted version of a point x in the original dataset X. We will refer to the ID of the original point x as z, defining the Z set. The ID here corresponds to a discrete value indexing the speech segments X. In this context, we have shown that choosing the augmentation distribution τ that minimizes an estimator of the conditional dependence between X and Z given Y leads to the best downstream performance on speaker and language recognition tasks (Zaiem et al., 2021). This section extends this approach in two manners, first applying it for domain adaptation in a supervised setting, and second extending it to the speech recognition task. We use for this the again the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2007).

In summary, to find the optimal augmentation distribution τ^*, we resort to minimizing the HSIC quantity with the augmented dataset $X' = f(X, \tau)$ according to $\tau^* = \arg\min_\tau HSIC(f(X, \tau), Z|Y)$ with $HSIC(X', Z|Y)$ an estimate of the conditional dependence between the distorted speech samples and their original IDs given their downstream textual labels.

Since the considered augmentations are not differentiable according to the considered parameters, we resort again to applying a random search to minimize the HSIC value described above. Thus, we sample random distributions and select the one with the lowest dependence scoring. Specifically, for every considered target dataset, we first sample $p = 200$ distribution parametrizations $(\tau_i)_{i\in[1,p]}$. For every parametrization τ_i, we compute the HSIC quantity following two steps. First, the augmented set $X'_i = f(X, \tau_i)$ is generated by computing $G = 20$ views of every speech sample in X. Then, $HSIC(X'_i, Z|Y)$ is computed following the technique described in Chapter 2. For Y, we consider the 10 classes consisting of the 10 most used words in the dataset and take only the portion of the speech where the word is pronounced, using word-level forced alignment. The augmentation distribution with the lowest HSIC scoring is selected to be applied during fine-tuning.

4.1.3 Experiments

This section describes the experiments led to validate the proposed approach first in a simulated environment, then on real-world distorted datasets.

Shared Experimental Protocol

In all the experiments, the model is composed of two blocks: a pre-trained Wav2Vec2.0 Large model and a downstream decoder. The pre-trained model acts directly on the speech waveform and outputs an embedding of size $1,024$ every 20ms of speech. Two fully connected layers with a hidden size of $1,024$ map each frame vector to one of the considered characters. The whole model is fine-tuned using Connectionist Temporal Classification (CTC) (Graves, 2012) loss. During inference, greedy decoding is applied to the CTC probability outputs without any language-model-based re-scoring following the SpeechBrain recipe (Ravanelli et al., 2021).

Name	Description	Range (Unit)
Low Min	Lowpass minimal frequency cutoff	[100-500] (Hz)
Low Max	Lowpass maximal frequency cutoff	[1000-5000] (Hz)
High Min	Highpass minimal frequency cutoff	[1000,4000] (Hz)
High Max	Highpass maximal frequency cutoff	[4000,6000] (Hz)
Pitch min	Minimal pitch shift	[-6,-2] (semitones)
Pitch max	Maximal pitch shift	[2,6] (semitones)
Min SNR	Minimal SNR for coloured noise	[0,5] (dB)
Max SNR	Maximal SNR for coloured noise	[10,30] (dB)
Min Gain	Minimal gain	[-20,-10] (dB)
Max Gain	Maximal gain	[3,10] (dB)

Tab. 4.1.: Descriptions and parameters' ranges of the selected set of augmentations.

For these experiments, we employ the Torch-Audiomentations library from the Asteroid team (Pariente et al., 2020) as it accelerates the computation of augmentations both during HSIC scoring and training, compared to the WavAugment one used in Chapter 2. From the pool of available augmentations, we selected the ones that have demonstrated efficacy in enhancing recognition performance with the contrastive predictive coding method (Kharitonov et al., 2021). This explains why the set of distortions is slightly different. Hence, seven augmentations are considered: pitch shifting, reverberation, gain (which may reproduce clipping issues), colored noise addition, high and low pass filtering, and polarity inversion. The application of these distortions is controlled with a set of parameters listed in Table 4.1.

Oracle Experiment

Task-specific experimental protocol. In this part, a known distortion distribution is first applied to a clean testing set. The resulting data will be considered as the mismatching target domain (i.e. a simulated one). In a second time, using this generated "noisy" dataset, appropriate augmentations, selected using our conditional independence-based method, are applied to a clean training dataset that will be used for fine-tuning our self-supervised representations. As only the test set is distorted, this simulated experiment only involves one fine-tuning, contrarily to the real-data scenario, where a second fine-tuning stage is held on the target training data, as shown in Figure 4.1. This simulated experiment has two advantages compared to a natural setting. First, it ensures that the distortions in the testing set can be replicated by the set of augmentations considered. Second, since we have access to the augmentation distribution that generated the "distorted" target

LS Split	Baseline	Random	CI Augment	Topline
test-clean	29.86	29.91	**27.20**	26.11
test-other	43.89	42.48	**40.68**	36.92

Tab. 4.2.: Mean WER results on distorted versions of LibriSpeech test splits. While scoring below the topline, our method, named “CI Augment", is significantly better than applying all or random augmentations. “Baseline" corresponds to augmentation-free training.

dataset, it allows estimating the similarity between the augmentation distribution used to create the simulated testing domain and the one obtained with our method, *i.e.* the similarity between the parameters controlling the chain of distortions applied.

In these experiments, $A = 8$ augmentations distributions are sampled and applied on the LibriSpeech *test-clean* and *test-other* splits (Panayotov et al., 2015). For every sampled distribution, these two distorted splits are then considered as the testing datasets. We apply the same augmentation distributions to the *dev-clean* and *dev-other* splits, and use these two sets to compute the optimal augmentations following the method described in the previous section. Finally, we use the computed distribution τ^* with the lower $HSIC$ estimator value as the augmentation for fine-tuning our SSL model on LibriSpeech *train-clean-100* split.

Results. Table 4.2 presents the results obtained on the test splits of LibriSpeech in the oracle experiments, with the column “CI Augment" (the name of the approach, CI standing for Conditional Independence) showing the results of the proposed approach. Each value corresponds to the mean of the values obtained with each of the A target datasets created with the sampled augmentation distributions. The “Topline" corresponds to the result obtained when the training samples are augmented using the same distribution as the one used to generate the distorted testing splits (i.e oracle scenario). Two baselines are considered: the first one referred to as “All" applies all the considered augmentations on the speech samples with their default parameters. Then, “Random" refers to the mean value obtained if applying the $(A - 1)$ randomly sampled augmentations, corresponding to the trainings already performed for the other toplines computation. Our method, while performing worse than the topline, leads to a relative word error rate (WER) improvement of 12.7% compared to the baseline on *test-clean*.

This controlled experiment also enables us to verify if the selected augmentations result in acoustic condition cloning, as suggested in Section 4.1.2. Indeed, the probabilities

of applying a given distortion to each testing set are known. To verify our intuition, for each one of the 8 augmentation distributions applied, we sample 200 other random augmentation distributions and score them using HSIC. For every scored distribution, we consider the vector composed by the seven probabilities of applying the considered distortions. Since these probabilities are known for the target distribution, we can compute an L_2 distance between the vector of probabilities of applying distortions used to create the target dataset, and those of the sampled scored distributions. We observe a Spearman correlation score of 0.51 between the HSIC scores and the distances between vectors of probabilities. Furthermore, the application probabilities of the 10 (top 5%) best scoring distributions are 15% closer to the target ones than those of the 10 worst scoring ones. These results indicate that the selected augmentations, *i.e.* those with low HSIC scoring, create samples closer to the target domain.

Experiments with Naturally Distorted Datasets

In this section, we test and validate the proposed approach on real low-resource “noisy" datasets.

Task-specific experimental protocol. The goal is to adapt a large clean “neutral" labeled dataset to better match the acoustic conditions of a small target dataset. The modified dataset is used during a first fine-tuning of the SSL representation, before further fine-tuning on the target dataset. To ensure a valid evaluation, the target dataset must meet two criteria: first, it should display consistent noisy recording and acoustic conditions. Second, neutral and target datasets should not exhibit different textual settings, *i.e* differences such as spontaneous versus read speech, as our augmentations only address acoustic distortions. The Librispeech *train-clean-100* is used as the clean dataset to be modified. The target datasets, on the other hand, correspond to the largest contributors of the CommonVoice 11.0 English dataset (Ardila et al., 2020). Starting from the ten most prolific contributors, two of them are finally selected after removing elements with heavy accents, and unintelligible or very clean recordings. For these two selected contributors, we partition the recorded samples into the train, validation, and test splits, and only use the training data to compute the augmentation distribution selection. The train splits are 9 and 7 hours long. More details can be found in the repository.

Contributor	Without Augmentations			With Augmentations		
	train-clean-100	Contributor Only	*train-clean-100* + Contributor	All	Random	CI Augment
Contributor 1	102.52	73.0	27.71	27.95	27.33	**24.27**
Contributor 2	96.49	98.92	20.48	20.76	22.23	**16.49**

Tab. 4.3.: Mean WER results on distorted versions of LibriSpeech test-clean and test-other. Our method, named “CI Augment", outperforms the baselines and random augmentations for each one of the two contributors.

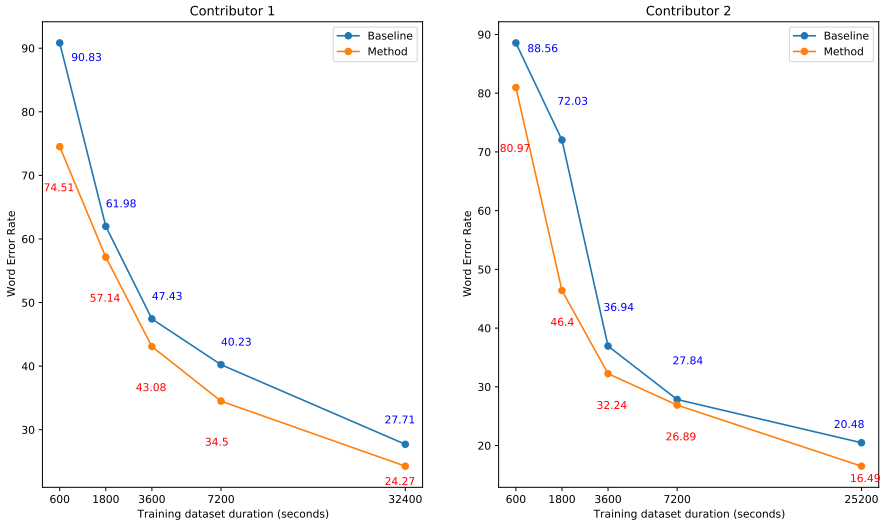

Fig. 4.2.: Effect of selecting augmentations on the performance depending on the quantity of target domain training data for each of the two considered contributors. The x-axis is not linear.

Results. Table 4.3 reports the WERs with or without augmentations during the first fine-tuning on *train-clean-100*. The first vertical part of the table shows the results obtained on the baselines without augmentations. “train-clean-100" corresponds to fine-tuning only on Librispeech *train-clean-100* split non-distorted. “Contributor Only" corresponds to training only on the contributor data. For all other columns, the model is fine-tuned on *train-clean-100* first, with or without augmentations, before further fine-tuning on the contributor data. The “CI Augment" column shows that the augmentations chosen with our conditional-independence-based method lead to better target performance than applying no, all, or random augmentations on the neutral training split. The relative improvement compared to the augmentation-free baseline reaches 12.4% for Contributor 1 and 19.5% for Contributor 2.

Furthermore, we study how this affects the amount of target domain data needed (see Figure 4.2). We start by fine-tuning with the chosen distortions for the “Method" lines and on the clean original LibriSpeech dataset for the “Baseline" lines. Then, the duration of annotated target data used is augmented gradually. For the two contributors, the orange curve representing the evolution of the WER after fine-tuning with the computed distortions is always below the blue curve corresponding to the baseline. The effect is particularly visible with Contributor 1 with a performance 16.6% higher relatively when training with only 2 hours.

4.1.4 Conclusion

Self-supervised representations severely underperform when facing acoustic domain mismatch. We have introduced a method using automatic data augmentation selection to reduce the drop in performance when switching acoustic domains. Experiments led in controlled and natural settings validate our assumption and method and also show that it helps reduce the quantity of annotated data needed in the target domain. However, domain shifts are not limited to acoustic shifts. Linguistic shifts or accent-related ones also represent a large source of failures. The following section will introduce approaches that can improve generalization abilities facing various shifts. They rely on the fact that pretraining sets contain a large variety of settings and that the robustness acquired during that phase should not be lost during the fine-tuning one.

4.2 Less Forgetting for Better Generalization

As said in the introduction, in common usage, the weights of speech self-supervised encoders are fine-tuned during the downstream phase. On the one hand, freezing the self-supervised representations during downstream training makes the SSL backbone a mere feature extractor. In this case, to reach reasonable performance, the downstream head may need to be more complex leading to costly inferences (Zaiem et al., 2023). On the other hand, full fine-tuning of the SSL encoder makes the pretraining “only" a better network initialization. We believe that controlling the fine-tuning trajectory with regard to the pretraining phase will improve the overall generalization ability.

Specifically, we postulate that fine-tuning the whole network weights hurts the generalization abilities of the final obtained model, because the model may “forget" the first task. This is motivated by two reasons. First, models generally learn to solve the self-supervision on massive unlabeled datasets. This large data diversity makes these models robust and explains in large part their generalization abilities (Hendrycks et al., 2019) and should thus be kept after the fine-tuning. Second, research on self-supervision training and probing has shown the closeness of the unsupervised pretraining task to speech recognition (Pasad et al., 2021). For HuBERT (Hsu, Tsai, et al., 2021) for instance, recent works have been exploring performing the ASR task only using the discrete classes used for HuBERT pretraining (Y. Yang et al., 2023).

To prevent this forgetting, and again with the objective of better performances post-finetuning on in-domain and out-of-domain samples in mind, we explore the continual learning literature looking for useful methods for our case. Continual learning (CL), also known as lifelong learning or incremental learning, is a machine learning paradigm that focuses on training models to acquire new knowledge and adapt to changing data over time (Parisi et al., 2019).

Continual learning approaches have been explored lately in the speech recognition research community towards including, within a model scope, new languages (Hou et al., 2022; Libera et al., 2023), new accents (Trinh et al., 2022; Majumdar et al., 2023; Vander Eeckt & Van Hamme, 2023) or new speakers (Diwan et al., 2023) without losing previous abilities. In a close approach, it has been used to further train a Wav2Vec2 model to include new domains where the learned representations can be efficiently used for downstream training. However, these works never explored CL during fine-tuning (J.-H. Lee et al., 2022).

A close line of work is parameter-efficient fine-tuning (PEFT) (Otake et al., 2023). While reducing the number of parameters updated is done mainly for the sake of efficiency in the case of large pretrained models, it also leads to less forgetting through freezing large parts of the network. Those methods are widely adopted in the natural language processing and computer vision communities due to the large size of the models. The speech literature for this is more scarce, with main works on child-directed speech (R. Fan et al., 2022) or emotion recognition (T. Feng & Narayanan, 2023). Specifically, a close effort has tried various adapters for self-supervision-based models on a group of speech tasks with training efficiency as the main target (Z.-C. Chen et al., 2023).

It is important to state a major difference between our objective and the ones in classic continual learning. In the classic setting, the performance on the first tasks matters in the final performance assessment, and, thus, approaches are evaluated on their reduction of forgetting. In our case, while we will use methods inspired by continual learning, the performance on the self-supervised task is not part of our evaluation. The link between forgetting and final performance is a postulate that we will only probe in a second time. Our contributions are two-fold:

1. We explore several continual-learning-based approaches for speech SSL fine-tuning showing substantial performance both on in-domain and out-of-domain testing samples.

2. We highlight the link between the performance gain and the non-forgetting of the self-supervised task by probing the forgetting of the best-performing methods.

In the following, Section 4.3.1 first describes the different explored fine-tuning approaches. Then, Section 4.2.2 gives further implementation details and analyses the obtained results. Finally, Section 4.2.3 unveils the link between forgetting and generalization performance before discussing a few caveats of the described approaches.

4.2.1 Methods

Baselines

The main baseline here is full fine-tuning of the network with the downstream task loss. In this case, the self-supervision part can be seen as a high-performing initialization of the final network. We should expect that after the full fine-tuning, knowledge about the pretraining task and data is mainly lost. A second baseline is more common in the speech self-supervision literature (Baevski, Zhou, et al., 2020). Since masking in the self-supervised training happens generally after the convolutional front-end (Baevski, Zhou, et al., 2020), the first convolutional layers are kept frozen during the fine-tuning. This method could be classified within the freezing-based approaches and is the one used in the Wav2Vec2 paper and similarly for HuBERT (Hsu, Tsai, et al., 2021), WavLM (S. Chen, Wang, et al., 2022) and other common SSL backbones.

Another considered approach, inspired by works diagnosing the link between freezing representations and generalization (Xie et al., 2021), consists in a better initialization of

the downstream head by keeping the encoder frozen during the first steps of downstream training. In a second time, after the initialization of the downstream head, the weights of the whole model are fine-tuned. This method will be called “two-phased" fine-tuning in the following. Finally, a fully frozen baseline is also tested. In this case, and following common practices in frozen SSL benchmarking (S. Yang et al., 2021; Zaiem et al., 2023), the layer input to the downstream head is a weighted sum of the encoder layers, with weights learned during fine-tuning. The frozen alternative is expected to lead to poor results given the reduced downstream heads. Better results may be achieved with more complex heads, such as two layers of BiLSTM as in the SUPERB benchmark, but we chose to keep the setting constant leading to similar inference costs.

Freezing-based

This section presents a group of tested fine-tuning methods. We call them “freezing-based" as they tend to freeze a group or all the weights learned in the pretraining phase. While freezing the encoder completely generally leads to bad performances (Zaiem et al., 2023), various approaches have been exploring partial freezing, or the incorporation of so-called adapters allowing degrees of freedom within the encoder without changing its weights. Three methods are presented and tested in this section. The first one uses adapters within the encoder layers (Majumdar et al., 2023; Vander Eeckt & Van Hamme, 2023). Adapters are lightweight modules intervening after the dense layers that come after self-attention ones. Precisely, instead of feeding to the next encoder layer the output of the feed-forward layer following the attention, this output is passed through the adapter and summed to itself as in residual approaches.

Second, following very successful trends in natural language processing and computer vision, Low-Rank (Hu et al., 2022) (LoRa) fine-tuning is also tested in our setting. It consists in freezing the pretrained model weights and injecting trainable rank decomposition matrices into each layer of the Transformer architecture, reducing the number of trainable parameters for downstream tasks. Precisely, we replace the feed-forward layers after the self-attention mechanism with LoRA layers. The initial matrix $W_0 \in \mathbb{R}^{d*k}$ is replaced with a low-rank decomposition $W_0 + \Delta W$ with $\Delta W = BA$ where $B \in \mathbb{R}^{d*r}$ and $A \in \mathbb{R}^{r*k}$ with r the rank of the low-rank decomposition. Only the LoRA layers are fine-tuned during the downstream training. Ultimately, for inference, $W = W_0 + BA$ has the same shape as the initial feed-forward matrix and, thus, this approach does not lead to more inference computations compared to baselines.

Finally, we explore using Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) during fine-tuning. EWC fine-tuning implies an additional loss, during downstream training, that forces the weights of the final model to be closer to those at the end of the pretraining phase. For every updated parameter, the distance to the initial phase is penalized by the corresponding Fisher information matrix value. The new loss becomes:

$$\mathcal{L}(\theta) = \mathcal{L}_{DS}(\theta) + \sum_i \frac{\lambda}{2} F_i \left(\theta_i - \theta_i^*\right)^2 \tag{4.1}$$

with θ the parameters of the SSL model, $\mathcal{L}_{DS}$ the downstream loss, θ^* the frozen SSL model weights after pretraining, F the Fisher information matrix and λ a weighting hyper-parameter. The Fisher information matrix captures how important a given parameter is for the pretraining task, and thus, the loss above reduces the movement of the most important parameters to the self-supervision task, leading to less forgetting.

Replay-based

We also explore replay methods, often called “experience replay" (Rolnick et al., 2019), during the fine-tuning of self-supervised representations. Replaying the pretraining task explicitly enforces non-forgetting through optimizing for simultaneously low SSL and downstream losses. The fine-tuning loss becomes:

$$\mathcal{L}(\theta) = \mathcal{L}_{DS,X_{DS}}(\theta) + \lambda_R \mathcal{L}_{SSL,X_R}(\theta). \tag{4.2}$$

with $\mathcal{L}_{DS}$ the downstream ASR loss, $\mathcal{L}_{SSL}$, X_{DS} the downstream annotated dataset, X_R the unlabeled replay dataset and λ_R a scaling hyper-parameter. We have also witnessed that SSL episodes should be played less regularly, another parameter we called replay-frequency controls whether a replay episode will be played within the current training step.

Two different models with two different self-supervision losses are considered. First, Data2Vec (Baevski et al., 2022) is trained with a teacher-student loss penalizing the distance between the latent representations of complete audio inputs from a teacher model and a student model final representations of a masked version of the same audio. Second, Wav2Vec2 (Baevski, Zhou, et al., 2020) employs a contrastive predictive loss function that encourages the model to produce similar frame-level contextualized embeddings to locally extracted quantized speech representations. We chose these two

Method	Params Updated	Epoch Duration (s)	More Data	Inference Cost
Frozen	1.9M	190		
CNN Frozen	90.8M	280		
Full FT	95.0M	543		
Replay	95.0M	641	X	
EWC	95.0M	1040		
Lora	0.59M	231		
Adapters	6.6M	240		X

Tab. 4.4.: Summary of the methods tested for fine-tuning. The number of parameters updated varies from the whole SSL network to 45x less. Numbers are shown for fine-tuning Data2Vec Base (Baevski et al., 2022) on one Nvidia V100 GPU on the GigaSpeech dataset “XS" split. EWC and replay lead to slower fine-tunings, because of further loss computations. Replay may need other sources of unlabeled data, while Adapters lead to a slightly increased inference cost.

methods as they are among the best-performing approaches in speech self-supervision while having easy-to-setup training processes.

Compared to other methods, replay has one obvious cost: it may necessitate other data sources if the replay is not done on the fine-tuning data, and the final performance may be very sensitive to this data choice. We have witnessed during our experiments another cost: it implies a large number of hyperparameters and choices compared to other methods. This will be discussed more thoroughly in Section 4.2.3.

Table 4.4 summarizes all the described methods showing the number of updated parameters and the duration of one epoch of fine-tuning Data2Vec Base on one Nvidia V100 GPU on the GigaSpeech dataset “XS" split. Computing additional losses, with the loop over the parameters of the network for EWC, and the SSL computations for replay, has a heavy cost in terms of training duration. For the other methods, updating fewer parameters leads naturally to faster trainings.

4.2.2 Experiments and Results

Datasets

Selected datasets need to cover the two requirements needed for our setting. First, as stated above, this section explores fine-tuning options in low-resource cases, and thus, training sets will be of reduced sizes. Second, to evaluate the link between forgetting and

generalization, out-of-domain testing sets are needed to evaluate OOD generalization. We will evaluate our methods in two languages, English and Danish, the former, because it is the main language in pretraining data, and the latter to test the robustness of eventual gains in another linguistic setting. For the English sets, GigaSpeech (G. Chen et al., 2021) XS subset (10 hours) will be used for training since LibriSpeech (Panayotov et al., 2015) cannot be used as it is in the pretraining sets, prohibiting a proper forgetting estimation. The testing sets include the GigaSpeech test set, LibriSpeech test splits (test-clean and test-other), two datasets of Scottish and Welsh English accents (Demirsahin et al., 2020) and CommonVoice 14.0 English (Ardila et al., 2020) test set. The last three sets can be seen as the OOD testing samples as the two former ones have specific accents, and the latter presents very various accents and noise conditions.

For Danish, we use for training the NST Danish ASR Database.[1] It is very relevant in our case as it consists in read speech samples recorded in very similar conditions. 50 hours of the dataset are selected for training, 5 for validation and 10 for in-domain testing. The CommonVoice 14.0 Danish validation and testing splits are concatenated and used for OOD testing.

Self-supervised Models

As stated in Section 4.3.1, two self-supervised models are considered, Data2Vec Base (Baevski et al., 2022) and Wav2Vec Large XLSR-53 (Conneau et al., 2021). They offer variability in network size (total number of parameters), pretraining dataset diversity and size and finally training loss and methods. The former model is only trained on the grouped LibriSpeech training splits, leading to 960 hours of English read speech. XLSR-53 is trained on a total of 56k hours of speech data covering 53 languages.

Methods Parameters

This section gives the training details for the different methods described in Section 4.3.1.

Replay-based During the replay-based experiments, the pretraining tasks, masked prediction for Data2Vec, and contrastive predictive coding for XLSR are performed along with the ASR downstream one, as described in Section 4.3.1. A replay frequency p_R

[1] nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-55/

controls the number of episodes of replay compared to the downstream ones. During the first epoch of fine-tuning, no replay is done as it has shown to lead to more stable fine-tunings. In the next epochs, $p_R = 0.25$ led to the best results. The hyper-parameters of the replay task, mainly controlling the mask creation, are kept similar to the default ones used for the pretraining. We explained this more in details in Chapter 1 but let us remind that Data2Vec is trained using a teacher-student approach, with the teacher model updated using an exponential moving average (EMA). We follow a similar strategy with the teacher model being updated with the weights of the fine-tuned student model as follows:

$$\theta_T \leftarrow \beta\theta_T + (1-\beta)\theta_S$$

with θ_T the teacher weights, θ_S the student weights (*i.e.* the weights of the fine-tuned encoder) and β a decay weight we fix to $\beta = 0.8$. Finally, replay requires the choice of a replay dataset. In the following, we will call “Auto-replay" experiments where the fine-tuning dataset is also used for the replay episodes. In a second experiment, either for English or Danish fine-tunings, providing a proper replay of the pretraining phase, replay batches will be sampled from LibriSpeech train splits, as they are included in both trainings of Data2vec Base and Wav2Vec2 XLSR. This experiment is called “LS-Replay".

Freezing Based The only hyperparameter for the baselines concerns the length of the freezing phase in the “Two-phased" approach, we fix it to three epochs. We use the LoRaLib toolkit (Hu et al., 2022) to replace the feed-forward layers following the transformer with a LoRa layer with rank r as described in Section 4.3.1. We chose as in (T. Feng & Narayanan, 2023) $r = 16$. For the adapters approach, we follow works in continual learning for speech on the adapter architecture. Each adapter network is composed of the following: (i) a layer-normalization layer, (ii) a downsampling operation to reduce the dimension to d, (iii) the application of a ReLU activation function, (iv) an upsampling operation to restore the original dimension, and (v) the inclusion of a skip connection connecting the input and output of the adapter. Finally, applying Elastic Weight Consolidation requires two choices. First, we fix the hyper-parameter controlling the distance to the original model loss (see Equation (4.1)) to $\lambda = 50$. The second choice concerns the dataset on which the Fisher information matrix is computed. Again, given that it is included in both the pretraining sets, LibriSpeech is chosen. Specifically, the LibriSpeech 10-hour split is selected as it includes samples from the three LibriSpeech training splits, and it is large enough for the expectation computations needed for the Fisher matrix.

Results

Tables 4.5 and 4.6 show the Word Error Rates (WER) obtained in the English and Danish experiments, with Data2Vec for the first table and XLSR in the second table, as the backbone self-supervised representation model. Results for the English training, *i.e.* the training performed on GigaSpeech XS, are shown on the five test sets described in Section 4.2.2, while results for the Danish one are shown on two test sets. The number shown is the mean of three runs with three different random seeds. However, we only selected runs that led to convergence. This point is discussed further in Section 4.2.3.

As expected, the frozen model leads to poor performance. It is the worst-performing approach for both languages either with Data2Vec or XLSR. Even worse, the model is not able to fit with XLSR with frozen features, with this model being notoriously hard to use without fine-tuning. The two classic baselines, freezing the convolutional front-end and the two-phased training, seem to perform better than the full fine-tuning baseline, especially for out-of-domain samples. We can see for instances in Table 4.5 an absolute gain of 3.5% WER and 2.3% on CommonVoice English and Danish with the “Fixed CNN" approach compared to the full fine-tuning approach.

When considering the lower parts of the two tables, presenting the alternative fine-tuning approaches results, we can see that, except for the failing “Adapters" approach, all the methods lead to better performances, both for in-domain and out-of-domain testing cases. This is visible from the numbers in bold in the table, as for every test set, the best performance is systematically obtained from one of the proposed alternatives. For instance, Low-Rank fine-tuning (LoRa), while also being more efficient as shown in Table 4.4, achieves a mean error rate 7.0% lower with Data2Vec and even 14.2% lower with Wav2Vec2 XLSR. “LoRa" and Elastic Weight Consolidation (EWC) are comparable in terms of performance, with the latter involving slower fine-tunings.

The replay-based approaches show two rows, “LS-Replay" and “Auto-Replay", as described in Section 4.2.2, depending on the replay dataset, either LibriSpeech (LS) or the fine-tuning set. In all our settings, with both SSL backbones and on both target languages, replaying LibriSpeech samples leads to lower WERs. “LS-Replay" is the best overall performing approach in two cases, Scottish and Welsh accented samples with Data2Vec Base and all Danish test sets with XLSR. The second case is surprising with the test and train data being in a different language compared to the replay samples.

Method	English Training						Danish Training	
	GS Test	LS test-clean	LS test-other	Scottish	Welsh	CV	NST Test	CV
Baselines								
Frozen	33.38	17	22.81	38.05	33.22	56.12	70.35	83.57
Full FT	26.92	9.83	17.47	26.9	22.32	53.4	13.75	36.57
Fixed CNN	26.67	10.01	16.94	25.52	22.65	49.98	13.8	34.38
Two-Phase	26.67	10.14	17.71	26.28	23.65	49.1	14.63	36.56
Freezing-Based								
LoRa	25.74	**9.27**	**15.73**	25.18	21.88	50.81	**12.89**	**31.13**
EWC	**25.57**	9.4	16.3	**24.97**	21.08	50.11	12.95	31.70
Adapters	30.62	12.81	19.72	35.16	30.8	56.42	45.48	62.43
Replay								
LS-Replay	26.07	9.71	16.34	25.14	**20.37**	**48.35**	12.93	32.36
Auto-Replay	26.25	9.54	17.16	25.8	22.91	50.48	13.14	35.93

Tab. 4.5.: WER Results on different test sets using Data2Vec Base as a backbone SSL model. The English fine-tuning is performed on the GigaSpeech “XS" subset and the Danish one on 50 hours of the NST dataset. We can see that LoRa, EWC, and replay methods outperform the considered baselines on all the testing sets.

The two CommonVoice columns allow us to have a proper look at out-of-domain generalization. CommonVoice is a crowd-collected dataset showing various accents and recording conditions. This explains in part the high WER values in these columns. Compared to the full fine-tuning baseline, different freezing or replay-based approaches, allow a relative gain in performance that can reach 9.4% and 14.8% with Data2Vec Base, respectively for English and Danish. Relative gains even reach 15.7% and 22.5% for Wav2Vec2 XLSR, respectively again for English and Danish.

4.2.3 Analysis and Discussion

Probing the Forgetting

We have shown in the previous section that continual-learning-inspired approaches lead to substantial performance gains both for in-domain and out-of-domain testing samples. To diagnose the link between the performance gains and non-forgetting of the self-supervision part, this section probes the forgetting of the models fine-tuned with the presented approaches. At every epoch of fine-tuning, a checkpoint of the model is saved for further probing. Two quantities are computed. First, we compute the L_2 distance

Method	English Training						Danish Training	
	GS Test	LS test-clean	LS test-other	Scottish	Welsh	CV	NST Test	CV
Baselines								
Frozen	> 100	>100	>100	>100	>100	>100	>100	>100
Full FT	28.85	11.89	24.43	32.35	28.42	60.69	10.99	30.41
Fixed CNN	28.98	12	24.35	33.49	29.42	58.88	10.8	27.87
Two-Phase	27.42	10.97	21.66	30.23	25.08	56.19	11.21	28.94
Freezing-Based								
LoRa	**26.68**	10.73	**19.79**	**28.61**	**24.02**	50.83	10.37	24.7
EWC	27.21	**10.55**	20.14	29.58	27.02	**51.12**	10.35	24.44
Adapters	28.8	12.76	20.3	29.05	26.36	50.61	18.85	33.34
Replay								
LS-Replay	27.54	10.85	20.21	29.15	27.53	53.98	**9.29**	**23.56**
Auto-Replay	28.6	11.53	22.75	31.08	28.52	53.17	11.22	29.48

Tab. 4.6.: WER Results on different test sets using Wav2Vec Large XLSR as a backbone SSL model. The English fine-tuning is performed on the GigaSpeech “XS" subset and the Danish one on 50 hours of the NST dataset.

between the final representations output by the SSL encoder after and before fine-tuning. In the frozen scenario, this quantity would remain equal to zero. Precisely, with e the SSL encoder after fine-tuning, e^* the one before, $(x_i)_{i\in[1,n]}$ a testing set of n speech samples, the similarity loss is defined as :

$$L_{SIM} = \frac{1}{n}\sum_{i=1}^{n} ||e(x_i) - e^*(x_i)||_2$$

The second quantity computed is, given a testing set, the self-supervised task loss obtained with the fine-tuned model. This quantity evaluates the ability of the fine-tuned models to still perform the masked speech modeling task used during pretraining. The probing is performed using Data2Vec Base. While Data2vec, as described in Section 4.3.1, uses a teacher-student approach for training, we will use the same model, *i.e.* the fine-tuned one for teacher and student for probing. It means that, practically, we will be testing the ability of the model to produce similar latent representations with or without masked parts. We think this is crucial for final performance as a model with this ability will be able to perform reasonable transcriptions even if parts of speech are unclear or blurry due to noise conditions or mispronunciations. We compute these two quantities, at every epoch of fine-tuning, on two English test sets, the in-domain test split of GigaSpeech and the out-of-domain test set of the CommonVoice dataset in its 14.0 version. The

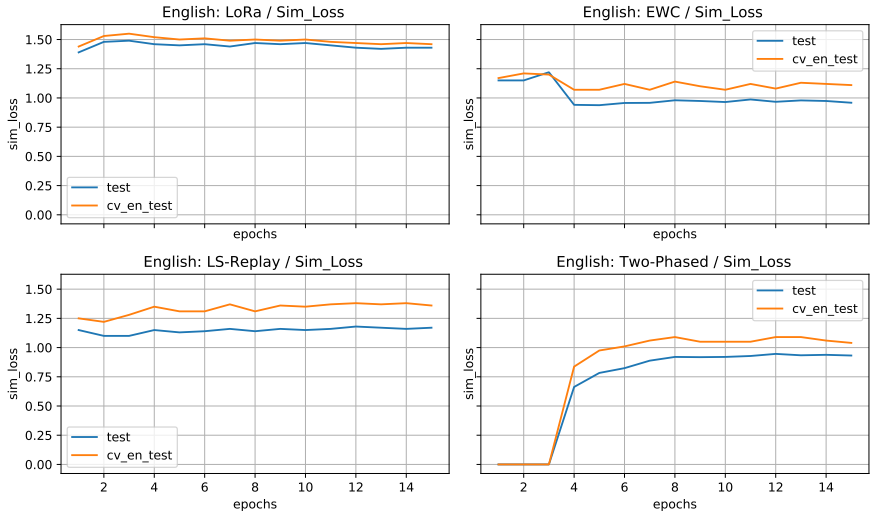

Fig. 4.3.: Evolution of the similarity loss for 4 considered fine-tuning approaches. The best-performing approaches lead to high dissimilarity either for in-domain or out-of-domain testing. There does not seem to be a link between the similarity of the final representations and the final downstream performance.

similarity probing results obtained with four fine-tuning approaches are shown in Figure 4.3. We chose to show the three best-performing approaches, Low-Rank fine-tuning, elastic weight consolidation, and LS-Replay, along with the best baseline, the two-phased fine-tuning. The value of the similarity loss does not seem linked to the final performance with the two-phased approach and EWC bearing close distance values, and LoRa and LS-Replay showing lower similarities.

However, the conclusion is different when probing the performance on the self-supervision task. Figure 4.4 shows the evolution of the SSL loss for the same methods as in the previous figure. In this case, the two-phased approach is an obvious outlier(intrus) with the loss twice as high as the “LS-Replay" and 6 to 7 times as high as for LoRa and EWC. Forgetting, here defined as losing the ability to perform the pretraining task, is correlated with low performance, and this is true for in-domain and out-of-domain testing samples. This probing experiment seems to confirm the starting postulate and explains the gain in performances witnessed in Section 4.2.2.

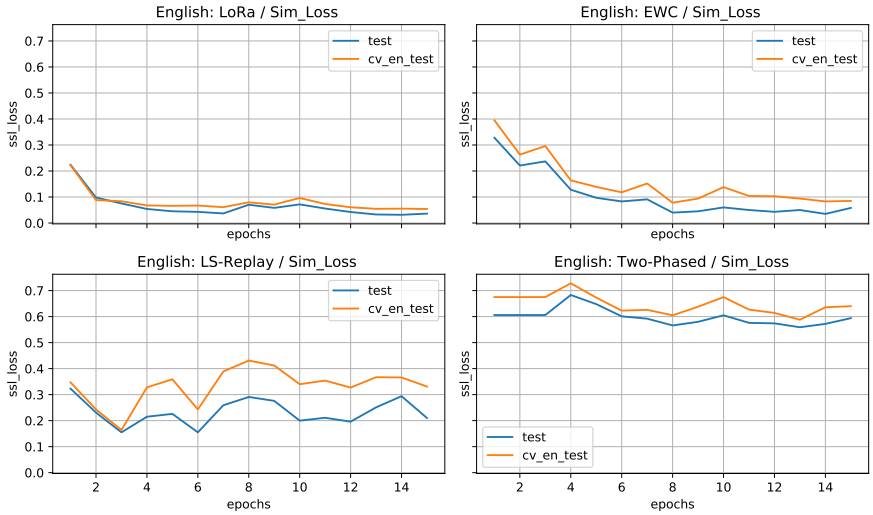

Fig. 4.4.: Evolution of the self-supervision task loss for 4 considered techniques. The best-performing approaches on the ASR task (First row and left plot of second row) seem to be the ones best-performing at the SSL task after multiple epochs of fine-tuning.

Parameters and Stability

This section discusses two caveats to the proposed approaches. The first one concerns the number of hyperparameters introduced by the techniques. As discussed in Sections 4.3.1 and 4.2.2, the presented fine-tuning approaches introduce various hyperparameters and choices that need trials and tuning. For instance, LoRa necessitates a rank for the low-rank matrices replacing the feed-forward layers in the transformer architecture. Also, applying Elastic Weight Consolidation (EWC) requires a weighting hyperparameter named λ in Equation (4.1), and a dataset choice for the computation of the Fisher information matrix. Adapters will involve a modeling architecture choice with all the hyperparameters related to that from downsampling dimensions to the number of layers. Finally, replay-based approaches are among the most hyperparameter-hungry methods described here. Replaying episodes implies a choice of the replay frequency, the replay samples, and the number of training steps before the beginning of the replay. The results shown in the tables 4.5 and 4.6 are those of the best-performing parametrization for every technique, but what is not shown is the influence of these hyperparameters.

We highlight this aspect by reporting the results of a group of experiments related to the tuning of these hyperparameters for the best-performing set of techniques. We consider for low-rank fine-tuning, EWC and “LS-Replay" the most impacting hyper-parameter,

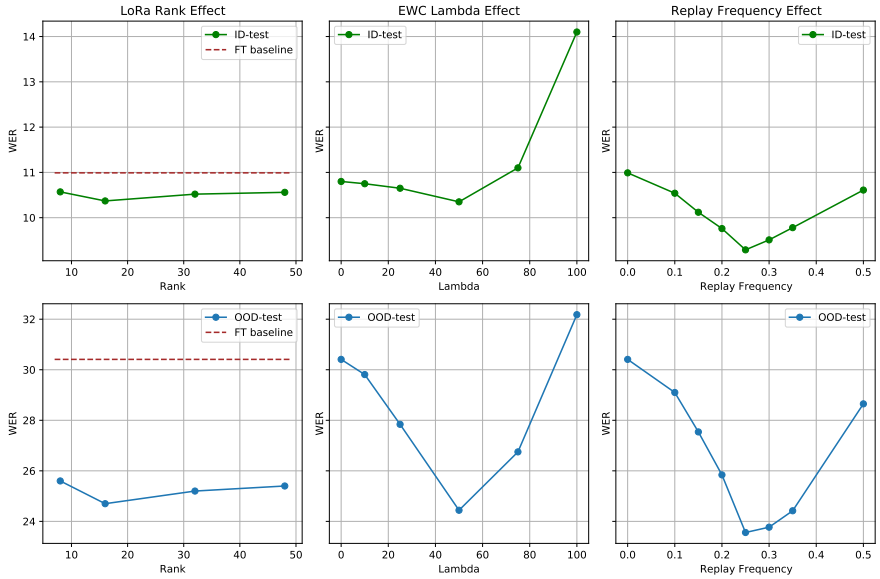

Fig. 4.5.: Effect of different hyper-parameters on the final performance on Danish in-domain (ID) and out-of-domain (OOD) test sets, for three different techniques (LoRa, EWC, and LS-Replay), with XLSR backbone. While LoRa seems quite robust to changes in the main hyperparameter, other approaches require careful tuning. In the second and third columns, the fine-tuning baseline is shown for $x = 0$, while it is shown with a horizontal dashed line for the LoRa plots.

respectively, the rank of the LoRa layers, the λ parameter controlling the weight of the distance penalization in EWC and the frequency of replay episodes during fine-tuning. We show the results with different values of these hyperparameters on the in-domain (NST) and out-of-domain (CommonVoice) testing samples for the Danish training performed with XLSR as the SSL backbone in Figure 4.5. For LoRA, as also observed in (T. Feng & Narayanan, 2023), the final performance is not severely impacted by reasonable changes in the main hyperparameter, the rank of the Lora layer. However, this is not the case for the two other techniques, as shown clearly in the “V" shapes of the plots in the second and third column of Figure 4.5. With inappropriate choices of replay frequencies or lambda values, the word error rates can be as high or even higher than the full fine-tuning baseline. This sensitivity is one main caveat to the use of these approaches as they will require, depending on the task and the technique used, a certain exploration of different values for the introduced parameters.

A second caveat we wish to report concerns the instability introduced by the proposed approaches. In Section 4.2.2, it is stated that the results shown are those of the mean of 3 runs that led to convergence. Successful experiments are those where the training

loss reasonably approaches zero after the considered number of training steps. In our experiments, fine-tuning Data2Vec has shown more stable than XLSR and would lead generally to convergence in all cases. Introducing penalties in losses or reducing the number of parameters updated can lead to trainings being less stable and not converging to an interesting solution.

4.2.4 Conclusion

Self-supervised encoders are generally learned using massive unlabeled datasets during pretraining, leading to robust and generalizing representations. A full fine-tuning for downstream purposes may bias the final model towards the reduced downstream setting and hurt generalization abilities. This is why we have tested continual-learning-inspired fine-tuning approaches for self-supervised-based speech recognition. Results show that Low-Rank fine-tuning, Elastic Weight Consolidation, and Replay allow substantial gains compared to the full fine-tuning baseline. These gains are correlated with less forgetting and, precisely, better performance on the pretraining task after fine-tuning. However, we have seen that this may come at the cost of efficiency (one of our main objectives as described in Chapter 1), either during training or even during inference in the case of added adapters. This suggests an opposite path, trading performance and out-of-distribution abilities with efficiency.

4.3 Fine-tuning Strategies for Faster Inference using Speech Self-Supervised Models

In this section, we will look at forgetting during inference as a double-edged sword. Pretrained encoders are trained on large corpora, allowing for robust representations. Consequently, as we have just shown, freezing, at least partly, these encoders during downstream training reduces forgetting improving out-of-domain performance. But what if out-of-domain generalization mattered less than inference costs? Can we, conversely, enforce structural forgetting policies, focusing on the target downstream domain, and reducing the computations needed during inference? These are questions we will provide an answer for in this section.

As described in Section 1, recent trends in SSL for speech have shown that the improvements in terms of performance are often driven by larger architectures, leading to potentially long inference times (S. Chen, Wang, et al., 2022). For instance, Sanyuan et al. 2022, have shown that switching from WavLM Base to WavLM Large halved the observed word error rate (WER) on a held-out English ASR task. Preserving reasonable inference times while increasing the capacity of the model is of critical interest to maximize the impact of this new technology on real-life products.

As a matter of fact, several approaches have been proposed to shorten inference times using SSL models. Some attempted to distill state-of-the-art models by using shallower or thinner networks (Chang et al., 2022; Rui et al., 2022) or through downsampling the inputs (Y. Lee et al., 2022; H.-J. Chen et al., ICASSP 2023). However, while the downstream performance of distilled student models is comparable to larger teacher models on most speech classification tasks, a large gap is still witnessed for more complex tasks such as ASR (T.-h. Feng et al., 2023). Also, low-bit quantization during pretraining has recently emerged as a successful approach for faster inference times (Yeh et al., 2022). Compared to our proposed methods, these two approaches bear the advantage of leading to generalist models usable for multiple downstream tasks. However, they have two major downsides. First, they necessitate access to the very large pretraining dataset, which may or may not be publicly and commercially available, such as for recent and large-scale state-of-the-art speech recognition models (Radford et al., 2022). Second, even if the pretraining set is available, applying quantization or distillation to these large models remains a particularly challenging and costly task due to the original dataset and model sizes. For instance, academic attempts for distilling SSL models have been solely applied to Base models (*i.e.* less than 100M parameters), and are generally restricted to a thousand hours of speech pretraining data, compared to $94k$ hours for 317M-parameter WavLM Large.

4.3.1 Setting and Methods

This section outlines the global setting for comparing the considered techniques before providing a detailed description of the approaches and their motivations.

Benchmarking Setting

The study is conducted under the two strong yet realistic following conditions. First, we suppose that we do not have access to the pretraining dataset that would enable, for instance, extensive distillation or quantization approaches. Second, we limit ourselves to using only the annotated training data of the target dataset during fine-tuning, eliminating transfer-learning-based approaches. The second condition is relevant when the target domain is rare and specific enough not to take advantage of transfer learning from classic large annotated datasets. This is generally true in two popular cases where using self-supervised models is privileged: low-resourced languages and speech datasets with specific acoustic conditions (Zuluaga-Gomez, Prasad, et al., 2023).

We use the released pre-trained and non-fine-tuned WavLM Large (S. Chen, Wang, et al., 2022) as the SSL model, as it tops speech self-supervised learning benchmarks and exhibits resilience to noisy conditions (T.-h. Feng et al., 2023). In all the experiments of this section, we use the *train-clean-100* split of LibriSpeech (Panayotov et al., 2015) as our training set, the *dev-clean* split for validation and finally the *test-clean* split for testing. Following common practices (Baevski, Zhou, et al., 2020), we freeze the convolutional front-end and only fine-tune the transformers part of WavLM Large consisting of 24 transformers layers. The self-supervised *encoder* outputs a frame vector of dimension $1,024$ for every segment of 320 speech samples which corresponds in the case of a 16-kHz sampling rate to 20 ms of audio signal. Two fully connected layers with a hidden size of $1,024$ map each frame vector to the probabilities of the considered characters. Connectionist Temporal Classification (CTC) (Graves, 2012) loss is used for training.

During inference, the decoding of the probabilities of the characters is completed in two ways: with or without using a Language Model (LM). In the experiments labeled as *Without LM*, greedy decoding is applied, outputting the character with the maximal probability at each step before applying CTC-based reformatting to get the predicted words. In the *With LM* experiments, we use the Librispeech official 4-gram language model, trained using the KenLM library (Heafield, 2011), and decode the sentence using shallow fusion (Toshniwal et al., in SLT 2018) considering the language modeling score of a beam of the most acoustically probable sequences. An n-gram model is chosen over more complex language modeling approaches to reduce word error rates while keeping low inference times. This is done using the PyCTCDecode[2] library with default parameters. To understand the impact of this aspect on the results tables, it is crucial to

[2]https://github.com/kensho-technologies/pyctcdecode

Technique		WER ↓	GPU (s)	CPU (s)	WER-LM ↓	GPU-LM (s)	CPU-LM (s)	MACs (G)
Baseline	**Full Model**	4.09	134	1121	3.31	152	1128	386.53
Layer Drop	**Drop Prob**							
	0.5	11.28	96	721	5.89	156	776	244.19
	0.4	8.32	102	816	4.58	145	844	272.28
	0.3	6.56	109	888	3.84	157	913	300.98
	0.25	5.91	113	932	3.72	148	950	314.24
Layer Removal	**Num. Kept Layers**							
	12	14.39	93	726	8.64	127	739	236.64
	16	8.16	109	852	5.53	131	861	286.60
	20	5.14	117	988	3.62	142	989	336.57
Early Exit : Entropy Threshold	**Mean Exit Layer**							
0.06	13.80	12.08	96	757	9.25	122	765	252.36
0.03	17.61	7.67	116	974	6.55	137	976	326.28
0.025	20.52	6.66	128	1127	5.87	149	1132	364.92
0.01	23.98	6.20	142	1275	5.49	165	1280	386.53
Early Exit : Layer Sim. Threshold	**Mean Exit Layer**							
0.92	15.97	10.23	99	812	8.17	123	819	274.11
0.95	17.18	8.78	104	850	7.35	126	864	291.68
0.965	21.44	6.79	120	1070	5.93	131	1073	358.85
0.98	24.00	6.20	128	1153	5.49	149	1153	386.51
Two Steps EE : Layer Sim. Threshold	**Mean Exit Layer**							
0.96	14.52	21.95	102	866	8.75	180	938	285.68
0.97	21.46	6.17	126	1138	4.34	152	1167	382.00
0.98	23.0	4.54	130	1175	3.87	151	1196	386.54
Downsampling Technique	**Downsampling Factor**							
Convolutional Downsampling	2	4.61	84	582	3.48	98	600	192.97
	3	**5.47**	**69**	**414**	**4.12**	**91**	**436**	**134.86**
	4	21.88	67	335	14.60	106	340	96.11
Averaging Downsampling	2	4.93	80	570	3.66	98	578	192.97
	3	6.01	64	406	4.27	90	422	134.86
	4	26.84	60	326	18.02	115	385	96.11
Signal Downsampling	2	4.85	86	569	3.58	97	575	192.97
	3	5.83	72	427	4.08	89	458	134.86
	4	16.08	63	330	11.10	97	369	96.11
DistilHuBERT	Linear Decoder	30.74	56	240	16.20	130	311	101.74
	BiLSTM Decoder	16.30	95	545	10.57	128	613	161.06

Tab. 4.7.: Word error rates and inference metrics on LibriSpeech *test-clean* split for the considered approaches and various parameters per method. All models are finetuned on LibriSpeech *train-clean-100*. “GPU" and “CPU" indicate the inference times in seconds on GPU and CPU. “-LM" suffixes indicate that the decoding uses a language model. “Drop Prob" is the probability of randomly dropping layers during inference. Early-exiting experiments come with an exiting threshold and a resulting mean exit layer computed over the test set.

note that this decoding is performed on CPU and that the decoding time is prolonged when the model is uncertain about its predictions. Indeed, PyCTCDecode proceeds to prune elements of the beam that are scored too low by the language model compared to the maximal beam score. It leads to a penalty for models with high error rates before the LM addition, as they systematically had longer decoding times. With the stage of the comparison set, we will proceed to the descriptions of the selected candidates.

Layer Dropping and Replacement

With multiple studies on layer-wise probing of self-supervised models showing that phonetic content is divided among the layers of the transformers (Pasad et al., 2021), removing layers has emerged as a possibility for faster inferences. Experiments led mainly on text language models have shown that dropping higher level layers is preferable to avoid heavy performance drops (Sajjad et al., 2023). In a first experiment, we will study the effect of fine-tuning the SSL model after having removed a number of layers. In a second one, given the fact that WavLM has been trained with *layerdrop* (A. Fan et al., 2020), *i.e.* random layer omission during training, we fine-tune it with layer drop probability equal to $q = 0.5$ and study the effect of keeping various layerdrop rates during the testing phase.

Early-exiting

Similarly, early-exiting is a relevant approach to reduce computations during inference (Yoon et al., 2022; Berrebbi et al., 2023). It consists in allowing the model to use an early layer representation and feed it directly to the decoder, saving the computation of further layers. During fine-tuning, starting from the twelfth layer, a specific downstream decoder is learned on top of every layer. During inference, a heuristic metric computed after each layer indicates whether the model should output a decoded sequence using the current layer or go further. Given well-calibrated heuristics, early-exiting should reduce the mean exit layer and, thus, the mean inference time. Furthermore, studies on what has been called “overthinking" (Berrebbi et al., 2023) have shown that SSL models could benefit from early exiting both inference time and performance. Inspired by previous works, two heuristics are tested: the entropy of the decoder outputs, and a measure of similarity between the representations of consecutive layers.

As described in Section 4.3.1, each downstream decoder consists of two linear layers outputting logit probabilities after each time frame of 20 ms. Each layer i of the transformer outputs a representation R_i of size $[L, D]$ with L the number of signal time frames and D the hidden dimension of WavLM Large ($D = 1,024$). Operating on this representation through the decoder D_i, the vectors of logit probabilities L_i are of size $[L, V]$ with V the number of different characters in the dataset (in our case $V = 31$). The entropy H_i of the output of layer i is defined as:

$$H_i = -\frac{1}{NV} \sum_{j=1}^{N} \sum_{k=1}^{V} L_{i,j,k} \log(L_{i,j,k}), \tag{4.3}$$

with $L_{i,j,k}$ being the probability of the character k at time frame j predicted by decoder D_i. During fine-tuning, to learn the weights of every decoder weights, we pass through all the layers of the model and sum the CTC losses over the outputs of all decoders. During inference, after each layer i starting from the twelfth, R_i is decoded and H_i is computed. If H_i is lower than a fixed entropy threshold, we do not go further in the SSL transformer, and decode the logit probabilities into words. Hence, reducing the entropy threshold increases the confidence required for exiting, leading systematically to later exits and thus, higher inference times. The second exiting heuristic is layer representation similarity. For each layer $i \geq 12$, we compute the cosine similarity between R_i and R_{i-1}. Similarly to the first approach, if the similarity is higher than a fixed threshold at layer i, the model exits to D_i and decodes the logits into a word sequence. This second approach is slightly faster as it does not involve computing the decoding into logits at each layer.

Sequence Downsampling

Inspired by works on distilling speech models with smaller sampling rates (Y. Lee et al., 2022; Rui et al., 2022), and given the quadratic memory bottleneck of transformers architectures as a function of input lengths, we assess the capacity of the SSL model, trained on 16-kHz audio inputs, to adapt to lower sampling rates. Given a speech file x consisting in T speech samples $x = (x_i)_{i \in [1,T]}$ and a downsampling factor k, a function f, learned or unlearned depending on the chosen method, downsamples x to $x' = f(x) = (x'_i)_{i \in [1, \lfloor T/k \rfloor]}$, a sequence of size $\lfloor T/k \rfloor$. The downsampled sequence x' is then fed to the SSL feature extractor instead of x. Three methods for downsampling the input sequences are evaluated. The first one is signal decimation (*i.e.* classic signal downsampling). Second, we test a learned downsampling strategy, through a

one-dimensional (1D) convolution layer of kernel size 160 and a stride equal to the downsampling factor, ran on the input waveform. Finally, we test an averaging 1D downsampling with a fixed window of size 16 (*i.e.* a constant convolution).

Each one of the techniques is evaluated with 3 downsampling factors: 2, 3 and 4. For instance, this corresponds for the first approach to downsampling the signals from 16 kHz to, respectively, 8000, 5333, and 4000 Hz. As explained in Section 4.3.1, the SSL model outputs a character every 320 audio samples, which corresponds to 20 ms of audio with a 16-kHz sampling rate. With lower sampling rates, the number of output characters may become lower than the lengths of the corresponding textual sequences. This is why, when dealing with downsampling factors 3 and 4, the size of the decoder output layer is doubled. It is then reshaped to fit the number of considered characters, before being fed to the classifier softmax function. This allows every frame of audio to output two characters instead of one.

4.3.2 Results and Robustness Study

Table 4.7 shows the results obtained with the different techniques. Reported GPU inference times are for a Nvidia Tesla V100 SXM2 32 Go GPU, while CPU inferences are using one Intel Cascade Lake 6248 processor with a 27.5 MB cache and 2.50 GHz clock speed. The inference times and MACs are those for running inference on the whole 5.4 hours of the *test-clean* split.

Layer removals. The results of the layer removal and dropping approaches are displayed in the upper part of Table 4.7. Surprisingly, for a given proportion of layers dropped, keeping the layerdrop performs better than training with the reduced number of layers. For instance, randomly dropping 50% of the layers during the test leads to 11.28 of WER, compared to 14.39 when dropping the last 12 layers during fine-tuning(*i.e.* again 50% of the layers). It suggests that while training systematically with the same layers adapts the models directly to inference conditions, removing the information contained in the last layers of the model harms too severely the performance.

Early-exiting. The middle part of Table 4.7 shows the obtained results using early-exiting with different values of entropy and similarity thresholds. Increasing the entropy threshold (or decreasing the similarity one) leads naturally to earlier exits (lower “Mean Exit Layer" values) and reduced inference times but higher WERs. Results show that using our two considered heuristics to control exiting does not prevent significant performance

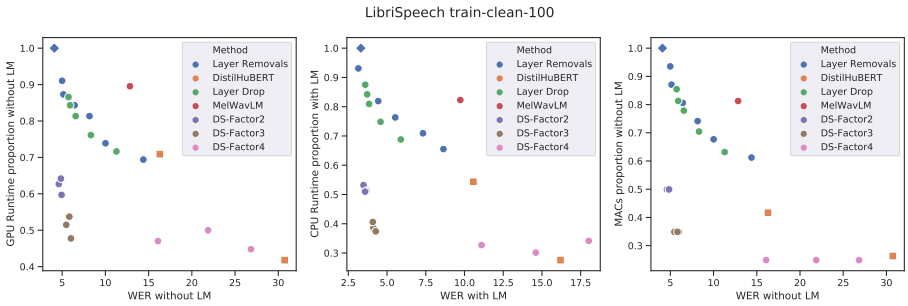

Fig. 4.6.: WER and inference metrics with or without language modeling for the presented techniques fine-tuned on LibriSpeech-100h. The best techniques, characterized by both low Word Error Rates (WERs) and inference times, are Factor2 and Factor3 downsamplings, located in the bottom left of the figures. The full model is indicated by a blue diamond, while DistilHubert baselines are represented by orange squares. Inference time measurements are shown as a proportion of the measure done with the full model.

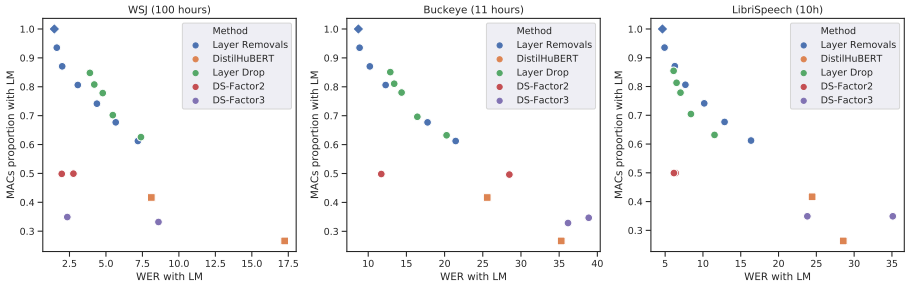

Fig. 4.7.: WER with LM decoding and MACs for the considered methods on WSJ, Buckeye and LibriSpeech-10h sets. While WSJ exhibits results similar to LibriSpeech,reducing the quantity of fine-tuning data causes significant performance drops for the downsampling methods.

drops in the case of earlier exits. We can also observe that even when using the whole network (*i.e.* low entropy cases), this technique leads to lower performances compared to the full model trained without early exiting. We suggest the following explanation: since early exits encourage the model to push the phonetic content required for decoding towards early layers, it undermines the ability of the model to learn hierarchical features, ultimately resulting in poorer performance even when exiting later in the model. To verify this explanation, we propose a two-step approach where the SSL model weights are first fine-tuned without early-exits, before freezing them when learning the early-exit decoders. As shown in Table 4.7, this leads to good performances in case of late-exiting, but with the cost of steeper drops when exiting earlier. This suggests that successful early-exiting should decouple hierarchical feature extraction and decoding preparation. In both cases,

early exiting lags behind layer-removal techniques in terms of ratio inference gains / performance drop.

Downsampling. Results of the downsampling experiments are shown in the last part of Table 4.7. Downsampling by factors 2 and 3 lead to high gains in inference times without significant drops in performance. For instance, compared to running the full model, signal downsampling with a factor 3 using a language model for decoding, leads to 61.34% relative CPU inference time reduction, with an absolute increase of only 0.81 in WER. Downsampling with factor 4, while naturally leading to further gains in inference times, results in intolerable performance costs. The three considered downsampling strategies are very similar in terms of error rates and computational savings, with a slight advantage for convolutional downsampling when sequences lengths are reduced with factors 2 and 3. For comparison with baselines, we add two experiments using DistilHuBERT (Chang et al., 2022). When using the simple linear decoder used in this benchmark, DistilHuBERT shows performances largely below the ones in the original paper. For a fair comparison, we produced an experiment with a BiLSTM decoder. While improving largely the performance, this comes at a high cost in terms of inference times.

Figure 4.6 presents a visual comparison between all the presented methods. Clearly, factor 2 and 3 downsampling techniques are the best performing methods with low WER, jointly with low GPU and CPU inference times. While being the fastest, higher downsampling factors and DistilHuBERT suffer from high error rates.

Robustness to Changes in the Downstream Dataset

Finally, we test the robustness of these conclusions to changes in the characteristics of the downstream dataset. Three datasets are considered. We tested the same methods with a 100-hour subset of the Wall Street Journal (WSJ) dataset (Paul & Baker, 1992).[3] We also test the robustness of the approach to dataset size variation by reducing the fine-tuning dataset to LibriSpeech-10h train set in first experiment and training on a small spontaneous English dataset, the Buckeye corpus (Pitt et al., 2005) containing 11 hours of data, in a final one.

Figure 4.7 shows the performance obtained with the presented methods on the three datasets. While WSJ shows very similar patterns to the first set of experiments, we can

[3]We combined WSJ0 and WSJ1, 70-hour long each, and removed all utterances that contain non-letter symbols in their transcriptions. Then, we extracted a 100-hour random subset of the remaining sentences

see in the case of less fine-tuning data that the downsampling method performance drops significantly. Downsampling with factor 3, while suffering a relative WER augmentation of only 33.7% for Librispeech-100h and 39.1% for WSJ, witnesses a drop of 384.9% with Buckeye and 571.7% with LibriSpeech-10h compared to the full model performance. In contrast, the other methods seem more resilient to reduced data quantity. Despite this, downsampling the sequences by a factor 2 using a learned convolution remains a good option for highly reduced inference times without unacceptable performance drop.

4.3.3 Conclusion

In this section, we explored different methods to reduce speech recognition inference times using large self-supervised models through fine-tuning. The comparison of these methods indicates that sequence downsampling is the best-performing option allowing substantial computation gain with low-performance drops. Experiments led on other downstream datasets show that the size of the downstream dataset is critical to avoid high error rates.

4.4 Chapter Conclusion

In this chapter, following Figure 5 presented in the introduction, we have studied the practical usage of pretrained self-supervised encoders with two main objectives: generalization and efficiency. While self-supervised encoders are generally frozen for standardized benchmarks and evaluations, their weights are generally fine-tuned in practical settings. Unlocking this additional degree of freedom, we studied intelligent fine-tuning strategies for speech recognition applications. After proposing a method to alleviate acoustic shifts during downstream training, we have shown that downstream fine-tuning can take various forms depending on the target, with two examples aiming for robustness and efficiency. In the first one, we have confirmed the link between forgetting and out-of-distribution generalization by using continual-learning-inspired transfer methods. In the second one, through structural pruning and downsampling, we have explored methods trading specialization and faster inferences. This chapter closes the set of works that will be presented in this thesis. In the next one, we will move on to the concluding parts.

5 Conclusion

Life is unfair. It extends and distracts us, then surprises us and changes us, until we are someone else. Was it me, twenty years ago in Alexandria? How does life judge me today on mistakes and faults I committed then? Why would god look back to these earthly days, judging what we have done a long while ago as if we had only one life and we did not change during it?

- Youssef Zeidan (Azazeel)

This chapter concludes this thesis with three main notes. First, in Section 5.1 we summarize this thesis and the main contributions of our work. Second, in Section 5.2, we detail code and data contributions provided to the research community for replication and further investigations. Finally, we discuss in Section 5.3 a few tracks for future works and investigations.

5.1 Summary

In this thesis, we have questioned the different choices that are made in speech self-supervision-based models for a variety of speech tasks. In particular, we focused on questioning common practices from pretext-task definition to downstream fine-tuning, with two main goals in mind: efficiency, both during pretraining exploration and final inferences, and enhancing generalization abilities. As described in the introduction, we believe that learned representations, especially unsupervised ones, will continue replacing hand-crafted ones in future models. And, thus, we hope the light shed on those in the

presented ideas and results will help the community build better speech representations and optimally use them.

In Chapter 1, we provide the pieces of context needed to appreciate the contributions provided in this thesis and the historical progress that makes the tackled questions relevant today. Precisely, this chapter covers speech representation development efforts and their evolution from spectral features towards learned representations, first with supervision and then with unlabeled data. In a second time, it presents an overview of the self-supervision research efforts, first on non-audio modalities, and then focusing on the speech domain.

In Chapter 2, we develop efficient and motivated approaches for the definition of pretext tasks towards better performance on targeted downstream ones. We show that a conditional-independence estimator allows for scoring pretext-tasks either in approaches based on predicting pretext labels or in contrastive settings. On four different tasks, probing phonetic, speaker, emotional, and linguistic content, we show that a careful selection of pretext-tasks and tailored view-creation policies lead to significant improvements in downstream performances.

The evaluation of the proposed models in Chapter 2 relied on fixed downstream heads for every considered task. This led to some frustration around one question: what if these representations were better with other heads? In Chapter 3, we provide an answer to this interrogation. We show that the current evaluation and rankings of self-supervised models are dependent on the choice made for downstream heads. We also show that limited-capacity heads should be avoided as they favor large self-supervised encoders. To reach this conclusion, we have studied downstream head choices over four criteria: final performance, inference efficiency, out-of-domain generalization, and multi-level feature exploitation. On all these points, linear probing has shown worse than more complex probing heads.

During evaluation, self-supervised encoders are frozen. However, it is common in practical settings to fine-tune the parameters of the encoder for the target downstream task. Focusing on speech recognition, Chapter 4 explores different options during fine-tuning toward the goals stated above; efficiency and generalization. The first one is tackled through model or input-shrinking approaches during fine-tuning. We propose two methods for the second goal. Building on elements developed in Chapter 2, we use acoustic conditions cloning to transform samples from a clean dataset to samples closer to the target acoustic domain. We also explore continual-learning-inspired methods to

reduce forgetting during the fine-tuning of self-supervised models. We show that this reduction in forgetting leads to better out-of-domain performances.

The use of unlabeled data and learned representations is still at its dawn and we hope these works bring the field a step closer to the extension of speech technology to new languages, new tasks, and new communities.

5.2 Code and Data Contributions

Apart from ideas and conclusions, a key element in contributing to global research efforts resides in sharing research artifacts, which in our domain mainly means code, weights, and data.

5.2.1 Code

Following our pledge for open-source research developed in Chapter 1, one of the contributions of this thesis has been the sharing of source code related to each of our research studies. This enables anyone to replicate the results discussed in the manuscript. Below, we provide links to the public and open-source GitHub repositories corresponding to each chapter.

- **Chapter 2** github.com/salah-zaiem/Multitask-pretext-task-selection. github.com/salah-zaiem/augmentations.
- **Chapter 3** github.com/speechbrain/benchmarks/tree/main/benchmarks/MP3S. The code is integrated within the “Benchmarks" SpeechBrain sub-library. We call it “MP3S" standing for “Multi-Probe Speech Self-Supervision".
- **Chapter 4** github.com/salah-zaiem/speeding_inferences.

5.2.2 Data

One of the *side-quests* of these three years has been to advance speech technologies in Tunisian Arabic, as I was born and raised in Tunisia. As common with low-resource languages, the main factor impeding the development of speech technology in this case

was the lack of transcribed data. This scarcity is often due to research on the language not being financially profitable. During this Ph.D. journey, I participated in releasing two main corpora for Tunisian Arabic speech processing:

- The TunSwitch Dataset with three subsets.[1] TunSwitch CS comprises code-switched utterances with three mixed languages: Tunisian Arabic (74%), English (13%), and French (13%). These are collected from local radio broadcasts and podcasts. TunSwitch TO is a collection of read speech utterances without code-switching. TunUnlabeled is a large collection of unlabeled speech samples, mainly from television shows. I like to consider it as one of the hardest speech recognition tasks on the market, with three languages involved, and the major one of them being very much low-resource, all in spontaneous multi-speaker settings.
- TARIC-SLU is an extension of the TARIC dataset (Masmoudi et al., 2014) to a spoken language understanding task with 60 slot labels centered around conversations in train stations for ticket buying.

5.3 Future Work

After describing our main contributions, a few limitations ought to be listed. On the one hand, the approaches proposed in Chapter 2 come with the cost of task specialization, as the choices are conditioned on a targeted downstream task. Within our framework, combining multiple downstream tasks is complicated as it requires a dataset with multiple annotations. On the other hand, while we have produced experiments showing the utility of careful selection of pretext-tasks to improve state-of-the-art approaches, we have not included within our scoring framework common predictive or clustering-based pretext-tasks. This is mainly due to the explosion of self-supervised training costs, making competition in training new models hard for academic actors. Finally, the tasks discussed in this document did not include generative ones, in the sense of tasks with audio as an output. This limitation, linked to the nature of current popular speech self-supervised representations is discussed more in detail in Section 5.3.2.

Let us finally discuss tracks for future work, in an attempt to overcome partly the limitations listed above. Part of these tracks have been introduced swiftly in the introduction of this work.

[1] https://zenodo.org/records/8370566

5.3.1 Data Selection for Self-Supervised Learning

In Figure 5, the first box starting from the left shows an important part of self-supervised pretraining: data selection. While scaling trainings with more data has been shown to improve the performance on downstream tasks, different works tend to show that training on carefully selected subsets may reach the same, if not better, performance levels (Grangier et al., 2023). This problem is not limited to unsupervised or self-supervised settings. Lately, supervised models for English ASR have been also trained on extremely large datasets, without exploring proper data selection policies. Another possible set of applications is tasks where training data is generated such as supervised speech enhancement for instance. In that case, data selection may allow one to avoid the generation of training samples that are not within the distribution of the target data.

Research on the topic, especially for speech applications, has been scarce. We give two reasons for this scarcity. First, as said in the introduction, there is a strong trend towards scaling to larger data and we do not seem to have yet reached a ceiling for scale benefits. Second, for a large number of use cases, trivial solutions are good enough. For instance, if you want a representation model for a task in a given language, having a large proportion of that language for pretraining leads to better performance (Evain et al., 2021). We consider that data selection approaches can be divided into two main trends.

Maximizing Training/testing Similarity

One line of work has considered that data selection should maximize the similarity between the training and validation samples. In this context, interesting work has explored data selection for unsupervised pretraining using self-supervised representations (Lu et al., 2022). The idea is, given a target test set, to select the training samples maximizing their similarity to the target distribution, using, pretrained beforehand, self-supervised representations to compute the similarity. However, the quality of the selection here is biased by the data and methods used to train the similarity space. Concerning supervised settings, a recent work has benchmarked different unsupervised approaches for training data selection (Gody & Harwath, 2023). The selection process, being unsupervised, should enable the allocation of supervision/annotation efforts efficiently towards the smallest set leading to reasonable performances. The limitation of these lines of work is that they do not take into account the model that is being learned and the ongoing optimization phase. We call these methods similarity-based approaches for data

selection, while approaches that consider the model parameters and results for selection will be called model-based ones. Let us give an example of those.

Bi-level Optimization

Data selection has also been studied as a bi-level optimization problem using coresets frameworks (Borsos et al., 2020; Grangier et al., 2023). Bilevel optimization refers to the optimization of an outer objective function that involves solving an inner optimization problem. In the context of supervised training, the outer objective typically represents the performance of the model on a dataset, and the inner optimization involves finding the model parameters that minimize a loss function on a subset of the data. Coresets come into play when dealing with large datasets, and their goal is to create a smaller, representative subset of the data (the “coreset") that can be used for training without significantly sacrificing the model’s performance. Inspired by these approaches, we aim to work on model-based data selection.

We believe that data selection, and especially model-based approaches, is a very promising track mainly in two scenarios: data creation such as supervised speech enhancement, and weakly-labeled data in semi-supervised settings. In these settings we can call “Infinite Data Scenarios", it would allow for avoiding backpropagating the loss signal from badly generated training elements.

5.3.2 Discrete Generative Representations

While self-supervised representations are today almost ubiquitous for transcription and classification tasks, their use in “Generative" speech tasks has been limited. By generative, we mean here tasks that output speech from a speech or non-speech input such as text-to-speech (TTS) or speech enhancement. This is mainly due to the information loss occurring within the transformations of the models, losing progressively information about the acoustic context and speaker identity within the contextual network, which are needed for generative approaches. In Section 1.5.2, we introduced succinctly emergent discrete representations of speech, in particular, and audio samples in general. The SoundStream model (Zeghidour, Luebs, et al., 2021) was seminal for this trend as it allowed for compressing speech samples into a sequence of integers representing speech embeddings and reconstructing with high fidelity the input from the sequence of integers.

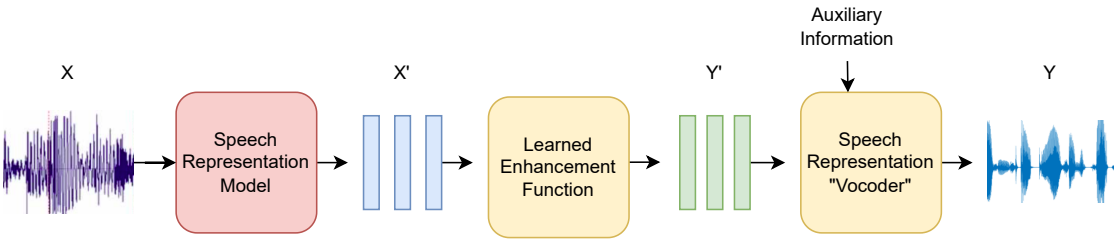

Fig. 5.1.: A speech enhancement pipeline with “Regenerative" speech representations. During evaluation, the central yellow box is learned using downstream supervised data.

While it has been used first for compression, the discrete aspect allowed other applications. In contrast to discriminative SSL, Discrete CoDec-based representations offer a great tool for generative tasks (C. Wang et al., 2023), as they are designed for resynthesis and thus suffer from low information loss. In the next parts, we will discuss the possibilities offered by these representations, their limitations, and tracks for improvement.

Advantages and Possibilities

To discuss advantages, let us quickly introduce the regeneration paradigm where learned or hand-crafted representations can be used as an intermediate step in a generation process. With the example of text-to-speech, the task would be divided into learning a mapping between text and the representation, and another mapping from the representation to the audio target. This approach may not be very surprising or novel to the speech research community. Indeed, in speech synthesis research, spectral features have been used for text-synthesis followed by Vocoder research efforts on how to map back power spectrograms to speech samples (*i.e.* phase recovery).

In a slightly more complex setting, Figure 5.1 shows what this kind of pipelines would resemble for a speech enhancement task. We will call X the input noisy speech samples and Y the target clean audio. $R(X)$ and $R(Y)$ are the two extracted representations of, respectively, X and Y using the self-supervised model we are testing. Within the regeneration paradigm, the function $X->Y$ is broken into the three parts that are represented in the Figure:

- The representation extraction part ($X \rightarrow R(X)$) is learned using one of the self-supervision procedures described in this document.

- The “fitting" part learns to perform the task of interest, speech enhancement here, in the representation space ($R(X) \rightarrow R(Y)$). This function is learned using supervised or parallel data.

- The “regeneration" part learns to map the clean representation $R(Y)$ back to the audio domain Y ($R(Y) \rightarrow Y$). This function can be learned using large unlabeled audio data.

The whole motivation behind this paradigm is that two out of the three parts (namely representation and regeneration) can be learned without supervision and that learning this space may allow for more data efficiency during the learning of the “fitting" part. The discrete representations that emerged lately in the field, allow for high performance on the regeneration part as they are specifically designed for it. Let us see how they were also good for the fitting one.

Instead of the regression task of predicting the next spectrogram frames, audio sequence generation or continuation is modeled as next token generation, and can therefore use the large set of techniques derived in natural language processing research for language modeling. For instance, this is how the audio continuation model proposed in AudioLM (Borsos et al., 2023) achieved state-of-the-art performance in audio and speech generation. Within this paradigm, discrete representations have been a favored choice (Kharitonov et al., 2023). To understand why, let us note that it is common to transform regression problems into classification tasks in the machine-learning community. Rather than using the square loss function for training on the original regression issue, practitioners opt for the cross-entropy loss in a discretized classification setup. This reformulation frequently improves performance, even though the cross-entropy loss doesn’t inherently capture the distance between classes.

Let us cite a few examples of this. Binning, *i.e.* quantizing, the pixel space has led to better generation of images than regression-based approaches (Van Den Oord et al., 2016). In audio beat estimation, training a model to predict a tempo bin has shown better performance than computing the value directly (Böck & Davies, 2020). In speech SSL, this idea can be seen through the quantization of HuBERT (Hsu, Tsai, et al., 2021) internal representations in pretext-labeling approaches described in Section 1.5.

Issues and Limitations

However, when taking a closer look at the audio modeling in AudioLM, we see that SoundStream tokens are not the only ones used to generate continuations. This can be seen, within the paradigm defined in the previous section, as a “fitting" problem while “regeneration" is easy by-design for CoDec-based representations. The proposed turn-around is to condition the continuations, consisting of the acoustic tokens from SoundStream, on what is called in the AudioLM paper "Semantic" tokens. The semantic tokens are quantized versions of more classic self-supervised embeddings, learned with the Best-RQ random projection approach (Chiu et al., 2022). This raises interesting questions :

- Why are current models not able to model acoustic token continuation without auxiliary semantic ones? How can we improve the “fitting" issue with acoustic tokens? How to mitigate possible trade-offs between “fitting" and “regeneration"?

To separate the two types of tokens, the authors look at the results of speech resynthesis and phoneme discrimination. The “Semantic" tokens allow an easy prediction (*i.e.* with limited probing heads) of phoneme identities, while the “Acoustic" ones lead to better resynthesis.

Introducing Hierarchy

The recent SpeechTokenizer (X. Zhang et al., 2023) model introduced the idea of content hierarchy within the layers of quantization of the SoundStream approach. In their work, an additional distillation loss penalizes the distance between the first layer of quantization and pretrained “Semantic" representations (in this case, representations from HuBERT (Hsu, Tsai, et al., 2021)). This hierarchical approach, limited for the moment to the blurry separation between acoustic and semantic, may be expanded towards more disentangled content in the further layers, through supervised disentanglement, using representation trained for speaker or prosody-related content.

Better Quantization

The “semantic" tokens discussed in the previous section are obtained through k-means clustering of the continuous self-supervised representations. This quantization is, first,

independent of the self-supervised learning process, and is not optimized for the fitting or regeneration task. We want to explore in future works, quantization-aware self-supervised learning, and differentiable discretization approaches optimized for generative speech modeling.

5.3.3 Relevance of Self-Supervision versus Scale

The last year has seen a rise in terms of labeled data used for the main tasks in the field such as English ASR (Y. Peng et al., 2023) or speaker verification (Yakovlev et al., 2023), leading to state-of-the-art results. The utility of unsupervised pretraining in these cases, *i.e.* large downstream labeled training data, is yet to be proven. Besides that, supervised ASR training, at scale, has also surprisingly shown to lead to internal representations that can be used for a large variety of other tasks, such as infant cry classification (Charola et al., 2023) or even non-human-related such as audio tagging and background classification (Gong et al., 2023).

These two points highly question the future of self-supervised learning. The first point means that self-supervised representations may only be needed in low-resource scenarios, and are useless starting from a certain supervision scale. The second one indicates that, even in these cases, they are challenged with supervised representations learned on large-scale datasets. Thus, we believe these interrogations call for answers. Let us discuss the second point first.

Difference between large supervision and large self-supervision

Large supervised models learned on publicly available and disclosed training sets, are today commonly used as features for downstream pipelines. In this context, a question that naturally arises is the differences between their representations and self-supervised ones. A fair comparison would require two models, with approximatively the same architectures, trained on the same datasets, one using the labels and one with a self-supervised objective. With these models, are there differences in the tasks they are most useful for? In their generalization abilities? We think there are two reasons which justify possible differences :

- If the training is done with HuBERT-like objectives, which are the closest to ASR settings, the quantization leading to cluster identities brings its share of noise

compared to clean ASR labels. Even in the case of these clusters showing high purities, different clusters may cover the same phoneme or grapheme. The effect of this noise is not necessarily negative. It may allow for learning better internal continuous representations of audio samples as the models would not be trained to radically separate similar phonemes and sequences. Furthermore, those noisy labels may contain information that is not only phoneme related, and thus explain the performance of these models on non-ASR tasks.

- Something that self-supervised models do not learn, while ASR-supervised ones do, is alignment, *i.e.* a mapping between time-frames and the corresponding character or phoneme. Despite the fact that models trained with the CTC or cross-entropy loss do not learn perfect alignments as alignment information is not generally explicitly provided in speech recognition trainings, the timestamps obtained are much better than random (Hannun, 2017) and generally provide a solid alignment baseline.

Targeting tasks with costly/complicated annotation processes

With the most self-supervised approaches trained with ASR-like or ASR-oriented tasks, the differences highlighted in the previous questions may be minimal. However, making self-supervised representations with low-resource settings in mind also means focusing on the tasks where data is scarce and hard to collect. This includes, as an example, reliable emotion recognition data, as the creation of the main datasets typically requires recording professional comedians, playing the different emotions. Training with low resources in mind also implies shifting the main testing downstream tasks from English to low-endowed languages. Even the training sets should be reconsidered, as current models are trained on large clean read audio-book data, a resource that is only available in a limited set of languages. Finding out which methods are robust to noise in the pretraining set, and generalize better to out-of-pretrainining-domain linguistic or acoustic conditions is an important step toward keeping the relevance of self-supervision.

Bibliography

Abdallah, A. A. B., Kabboudi, A., Kanoun, A., & Zaiem, S. (2023). Leveraging data collection and unsupervised learning for code-switched tunisian arabic automatic speech recognition. *arXiv preprint arXiv:2309.11327* (cit. on p. 41).

Algayres, R., Zaiem, M. S., Sagot, B., & Dupoux, E. (2020). Evaluating the reliability of acoustic speech embeddings. *INTERSPEECH 2020 - Annual Conference of the International Speech Communication Association* (cit. on pp. 12, 55, 89).

Al-Tahan, H., & Mohsenzadeh, Y. (2021). Clar: contrastive learning of auditory representations. *International Conference on Artificial Intelligence and Statistics*, 2530–2538 (cit. on p. 76).

Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., & Weber, G. (2020, May). Common voice: a massively-multilingual speech corpus. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the twelfth language resources and evaluation conference* (pp. 4218–4222). European Language Resources Association. (Cit. on pp. 58, 81, 93, 119, 127).

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., & Saunshi, N. (2019). A Theoretical Analysis of Contrastive Unsupervised Representation Learning. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 9904–9923 (cit. on pp. 51, 76)
The number of negative samples has been an important point to take into account in contrastive algorithms. Increasing highly this number may lead to class collisions even if the number of NS is much smaller than the number of classes. If we consider that similar points are elements of the same class, then, making their embeddings closer should help if the downstream task is a classification one on a subset of these classes.

Bachman, P., Hjelm, R. D., & Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, *32* (cit. on p. 52).

Baevski, A., Babu, A., Hsu, W.-N., & Auli, M. (2023). Efficient self-supervised learning with contextualized target representations for vision, speech and language. *International Conference on Machine Learning*, 1416–1429 (cit. on pp. 27, 35).

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., & Auli, M. (2022). Data2vec: a general framework for self-supervised learning in speech, vision and language. *International Conference on Machine Learning*, 1298–1312 (cit. on pp. xxv, 12, 27, 35, 91, 125–127).

Baevski, A., & Mohamed, A. (2020). Effectiveness of self-supervised pre-training for asr. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7694–7698 (cit. on p. 34).

Baevski, A., Schneider, S., & Auli, M. (2019). Vq-wav2vec: self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453* (cit. on p. 30).

Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). Wav2vec 2.0: a framework for self-supervised learning of speech representations. *Proceedings of the 34th International Conference on Neural Information Processing Systems* (cit. on p. 63).

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). Wav2vec 2.0: a framework for self-supervised learning of speech representations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 12449–12460, Vol. 33). Curran Associates, Inc. (Cit. on pp. 8, 13, 32, 33, 39, 51, 62, 66–68, 75, 91, 123, 125, 137).

Bastianelli, E., Vanzo, A., Swietojanski, P., & Rieser, V. (2020, November). SLURP: a spoken language understanding resource package. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 7252–7262). Association for Computational Linguistics. (Cit. on p. 93).

Bell, P., Fainberg, J., Klejch, O., Li, J., Renals, S., & Swietojanski, P. (2020). Adaptation algorithms for neural network-based speech recognition: an overview. *IEEE Open Journal of Signal Processing*, *2*, 33–66 (cit. on p. 22).

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828 (cit. on p. 19).

Berrebbi, D., Yan, B., & Watanabe, S. (2023). Avoid overthinking in self-supervised models for speech recognition. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on p. 139).

Böck, S., & Davies, M. E. (2020). Deconstruct, analyse, reconstruct: how to improve tempo, beat, and downbeat estimation. *ISMIR*, 574–582 (cit. on p. 152).

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., et al. (2023). Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (cit. on pp. 31, 152).

Borsos, Z., Mutny, M., & Krause, A. (2020). Coresets via bilevel optimization for continual learning and streaming. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 14879–14890, Vol. 33). Curran Associates, Inc. (Cit. on p. 150).

Bredin, H. (2017). Tristounet: triplet loss for speaker turn embedding. *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 5430–5434 (cit. on p. 25).

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., & Narayanan, S. S. (2008). Iemocap: interactive emotional dyadic motion capture database. *Language resources and evaluation*, *42*, 335–359 (cit. on p. 93).

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, E. (, Provost, E. M., Kim, S., Chang, J. N., Lee, S., & Narayanan, S. S. (2008). Iemocap: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, *42*, 335–359 (cit. on p. 66).

Cai, H., Gan, C., Wang, T., Zhang, Z., & Han, S. (2020). Once for all: train one network and specialize it for efficient deployment. *International Conference on Learning Representations* (cit. on p. 45).

Cao, H., Cooper, D. G., Keutmann, M. K., Gur, R. C., Nenkova, A., & Verma, R. (2014). Crema-d: crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing*, *5*(4), 377–390 (cit. on pp. 106, 107).

Carlin, M. A., Thomas, S., Jansen, A., & Hermansky, H. (2011). Rapid evaluation of speech representations for spoken term discovery. *Proc. Interspeech 2011*, 821–824 (cit. on p. 55).

Chang, H.-J., Yang, S.-w., & Lee, H.-y. (2022). Distilhubert: speech representation learning by layer-wise distillation of hidden-unit bert. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7087–7091 (cit. on pp. 91, 136, 143).

Charola, M., Kachhi, A., & Patil, H. A. (2023). Whisper Encoder features for Infant Cry Classification. *Proc. INTERSPEECH 2023*, 1773–1777 (cit. on p. 154).

Chen, G., Chai, S., Wang, G.-B., Du, J., Zhang, W.-Q., Weng, C., Su, D., Povey, D., Trmal, J., Zhang, J., Jin, M., Khudanpur, S., Watanabe, S., Zhao, S., Zou, W., Li, X., Yao, X., Wang, Y., You, Z., & Yan, Z. (2021). GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio. *Proc. Interspeech 2021*, 3670–3674 (cit. on pp. 45, 127).

Chen, H.-J., Meng, Y., & Lee, H.-y. (ICASSP 2023). Once-for-all sequence compression for self-supervised speech models. (Cit. on p. 136).

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., & Wei, F. (2022). Wavlm: large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1505–1518 (cit. on pp. 34, 91, 123, 136, 137).

Chen, S., Wu, Y., Wang, C., Chen, Z., Chen, Z., Liu, S., Wu, J., Qian, Y., Wei, F., Li, J., & Yu, X. (2021). Unispeech-sat: universal speech representation learning with speaker aware pre-training. (Cit. on pp. 55, 113).

Chen, S., Wu, Y., Wang, C., Liu, S., Chen, Z., Wang, P., Liu, G., Li, J., Wu, J., Yu, X., & Wei, F. (2022). Why does Self-Supervised Learning for Speech Recognition Benefit Speaker Recognition? *Proc. Interspeech 2022*, 3699–3703 (cit. on p. 43).

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *International conference on machine learning*, 1597–1607 (cit. on pp. 24, 25, 47, 51, 76).

Chen, W., Chang, X., Peng, Y., Ni, Z., Maiti, S., & Watanabe, S. (2023). Reducing Barriers to Self-Supervised Learning: HuBERT Pre-training with Academic Compute. *Proc. INTERSPEECH 2023*, 4404–4408 (cit. on p. 46).

Chen, Z., Watanabe, S., Erdogan, H., & Hershey, J. (2015). Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. *INTERSPEECH* (cit. on p. 52).

Chen, Z.-C., Fu, C.-L., Liu, C.-Y., Li, S.-W. D., & Lee, H.-y. (2023). Exploring efficient-tuning methods in self-supervised speech models. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 1120–1127 (cit. on p. 122).

Chiu, C.-C., Qin, J., Zhang, Y., Yu, J., & Wu, Y. (2022). Self-supervised learning with random-projection quantizer for speech recognition. *International Conference on Machine Learning*, 3915–3924 (cit. on pp. 34, 46, 153).

Cho, C. J., Wu, P., Mohamed, A., & Anumanchipalli, G. K. (2023). Evidence of vocal tract articulation in self-supervised learning of speech. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on p. 43).

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M. (2021). Unsupervised Cross-Lingual Representation Learning for Speech Recognition. *Proc. Interspeech 2021*, 2426–2430 (cit. on p. 127).

Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., Riesa, J., Rivera, C., & Bapna, A. (2023). Fleurs: few-shot learning evaluation of universal representations of speech. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 798–805 (cit. on pp. 34, 41).

Danwei, C., & Li, M. (2021). The dku-dukeece system for the self-supervision speaker verification task of the 2021 voxceleb speaker recognition challenge. (Cit. on p. 43).

Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The helmholtz machine. *Neural computation*, *7*(5), 889–904 (cit. on p. 4).

Dejoli, T. T. L., He, Q., & Xie, W. (2022). Audio,speech and vision processing lab emotional sound database (asvp-esd). (Cit. on p. 107).

Demirsahin, I., Kjartansson, O., Gutkin, A., & Rivera, C. (2020). Open-source Multi-speaker Corpora of the English Accents in the British Isles. *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, 6532–6541 (cit. on p. 127).

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: additive angular margin loss for deep face recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4690–4699 (cit. on p. 82).

de Seyssel, M., Lavechin, M., Adi, Y., Dupoux, E., & Wisniewski, G. (2022). Probing phoneme, language and speaker information in unsupervised speech representations. *arXiv preprint arXiv:2203.16193* (cit. on p. 47).

de Seyssel, M., Lavechin, M., Titeux, H., Thomas, A., Virlet, G., Revilla, A. S., Wisniewski, G., Ludusan, B., & Dupoux, E. (2023). Prosaudit, a prosodic benchmark for self-supervised speech models. *Interspeech* (cit. on p. 38).

Desplanques, B., Thienpondt, J., & Demuynck, K. (2020). ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. *Proc. Interspeech 2020*, 3830–3834 (cit. on pp. 94, 95).

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (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). Association for Computational Linguistics. (Cit. on pp. 12, 24).

Diwan, A., Yeh, C.-F., Hsu, W.-N., Tomasello, P., Choi, E., Harwath, D., & Mohamed, A. (2023). Continual learning for on-device speech recognition using disentangled conformers. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (cit. on p. 122).

Doersch, C., & Zisserman, A. (2017). Multi-task self-supervised visual learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2070–2079 (cit. on pp. 50, 53).

Doersch, C., Gupta, A., & Efros, A. A. (2016). Unsupervised visual representation learning by context prediction (cit. on p. 52).

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87 (cit. on p. 18).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: transformers for image recognition at scale. *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (cit. on p. 26).

Doulaty, M., Saz, O., & Hain, T. (2015). Data-selective transfer learning for multi-domain speech recognition (cit. on p. 53).

Dubois, Y., Ermon, S., Hashimoto, T. B., & Liang, P. S. (2022). Improving self-supervised learning by characterizing idealized representations. *Advances in Neural Information Processing Systems*, *35*, 11279–11296 (cit. on p. 89).

Dumais, S. T. (2004). Latent semantic analysis. *Annual Review of Information Science and Technology (ARIST)*, *38*, 189–230 (cit. on p. 5).

Dunbar, E., Cao, X. N., Benjumea, J., Karadayi, J., Bernard, M., Besacier, L., Anguera, X., & Dupoux, E. (2017). The zero resource speech challenge 2017. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 323–330 (cit. on p. 28).

Dupoux, E. (2018). Cognitive science in the era of artificial intelligence: a roadmap for reverse-engineering the infant language-learner. *Cognition*, *173*, 43–59 (cit. on p. 4).

Dwivedi, K., & 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 (cit. on p. 52).

Elbanna, G., Scheidwasser-Clow, N., Kegler, M., Beckmann, P., El Hajal, K., & Cernak, M. (2022). Byol-s: learning self-supervised speech representations by bootstrapping. *HEAR: Holistic Evaluation of Audio Representations*, 25–47 (cit. on p. 35).

Emami, M., Tran, D., & Koishida, K. (2021). Augmented contrastive self-supervised learning for audio invariant representations. *arXiv preprint arXiv:2112.10950* (cit. on p. 76).

Essid, S. (2005). *Classification automatique des signaux audio-fréquences: reconnaissance des instruments de musique* [Doctoral dissertation, Université Pierre et Marie Curie-Paris VI]. (Cit. on p. 71).

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., & Besacier, L. (2021). Lebenchmark: a reproducible framework for assessing self-supervised representation learning from speech. (Cit. on pp. 38, 41, 88, 112, 149).

Fainberg, J., Klejch, O., Loweimi, E., Bell, P., & Renals, S. (2019). Acoustic model adaptation from raw waveforms with sincnet. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 897–904 (cit. on p. 21).

Fan, A., Grave, E., & Joulin, A. (2020). Reducing transformer depth on demand with structured dropout. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (cit. on p. 139).

Fan, R., Zhu, Y., Wang, J., & Alwan, A. (2022). Towards better domain adaptation for self-supervised models: a case study of child asr. *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1242–1252 (cit. on p. 122).

Fan, Y., Kang, J., Li, L., Li, K., Chen, H., Cheng, S., Zhang, P., Zhou, Z., Cai, Y., & Wang, D. (2019). Cn-celeb: a challenging chinese speaker recognition dataset. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7604–7608 (cit. on p. 108).

Fechner, G. (1966). Elements of psychophysics. vol. i. (cit. on pp. 2, 20).

Feng, T., & Narayanan, S. (2023, June). *Peft-ser: on the use of parameter efficient transfer learning approaches for speech emotion recognition using pre-trained speech models*. (Cit. on pp. 122, 128, 134).

Feng, T.-h., Dong, A., Yeh, C.-F., Yang, S.-w., Lin, T.-Q., Shi, J., Chang, K.-W., Huang, Z., Wu, H., Chang, X., et al. (2023). Superb@ slt 2022: challenge on generalization and efficiency of self-supervised speech representation learning. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 1096–1103 (cit. on pp. 37, 88, 106, 136, 137).

Fu, Y., Zhang, Y., Qian, K., Ye, Z., Yu, Z., Lai, C.-I. J., & Lin, C. (2022). Losses can be blessings: routing self-supervised speech representations towards efficient multilingual and multitask speech processing. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 20902–20920, Vol. 35). Curran Associates, Inc. (Cit. on p. 44).

Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *29*(2), 254–272 (cit. on p. 2).

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., Dahlgren, N., & Zue, V. (1992). Timit acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium* (cit. on p. 58).

Garrido, Q., Balestriero, R., Najman, L., & Lecun, Y. (2023). Rankme: assessing the downstream performance of pretrained self-supervised representations by their rank. *International Conference on Machine Learning*, 10929–10974 (cit. on p. 89).

Gat, I., Kreuk, F., Lee, A., Copet, J., Synnaeve, G., Dupoux, E., & Adi, Y. (2022). On the robustness of self-supervised representations for spoken language modeling. *arXiv preprint arXiv:2209.15483* (cit. on p. 42).

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., & Ritter, M. (2017). Audio set: an ontology and human-labeled dataset for audio events. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 776–780 (cit. on p. 81).

Georges, M.-A., Schwartz, J.-L., & Hueber, T. (2022). Self-supervised speech unit discovery from articulatory and acoustic features using vq-vae. *arXiv preprint arXiv:2206.08790* (cit. on p. 38).

Gidaris, S., Singh, P., & Komodakis, N. (2018a). Unsupervised representation learning by predicting image rotations. *CoRR*, *abs/1803.07728* (cit. on p. 52).

Gidaris, S., Singh, P., & Komodakis, N. (2018b). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (cit. on p. 24).

Gody, R., & Harwath, D. (2023). Unsupervised fine-tuning data selection for asr using self-supervised speech models. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on p. 149).

Gong, Y., Khurana, S., Karlinsky, L., & Glass, J. (2023). Whisper-AT: Noise-Robust Automatic Speech Recognizers are Also Strong General Audio Event Taggers. *Proc. INTERSPEECH 2023*, 2798–2802 (cit. on p. 154).

Good, I. J. (1952). Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, *14*(1), 107–114 (cit. on p. 3).

Grangier, D., Ablin, P., & Hannun, A. (2023). Adaptive training distributions with scalable online bilevel optimization. (Cit. on pp. 149, 150).

Graves, A. (2012). Connectionist temporal classification. In *Supervised sequence labelling with recurrent neural networks* (pp. 61–93). Springer. (Cit. on pp. 3, 66, 116, 137).

Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B., & Smola, A. (2007). A kernel statistical test of independence. *Advances in Neural Information Processing Systems 20: Proceedings of the 2007 Conference, 585-592 (2008)* (cit. on pp. 54, 56, 57, 115).

Grezl, F., & Fousek, P. (2008). Optimizing bottle-neck features for lvcsr. *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 4729–4732 (cit. on p. 28).

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, *33*, 21271–21284 (cit. on pp. 26, 76).

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020). Conformer: Convolution-augmented Transformer for Speech Recognition. *Proc. Interspeech 2020*, 5036–5040 (cit. on p. 94).

Gump, M., Hsu, W.-N., & Glass, J. (2020). Unsupervised Methods for Evaluating Speech Representations (cit. on p. 55).

Guyon, I., & Elisseeff, A. (2003). An introduction of variable and feature selection. *J. Machine Learning Research Special Issue on Variable and Feature Selection*, *3*, 1157–1182 (cit. on p. 53).

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*, 389–422 (cit. on p. 63).

Haider, D., Lostanlen, V., Ehler, M., & Balazs, P. (2023). Energy preservation and stability of random filterbanks. *arXiv preprint arXiv:2309.05855* (cit. on p. 20).

Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., Gulati, A., Pang, R., & Wu, Y. (2020). ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context. *Proc. Interspeech 2020*, 3610–3614 (cit. on p. 94).

Hannun, A. (2017). Sequence modeling with ctc. *Distill*, *2*(11), e8 (cit. on p. 155).

Heafield, K. (2011). KenLM: faster and smaller language model queries. *Proceedings of the Sixth Workshop on Statistical Machine Translation* (cit. on p. 137).

Hendrycks, D., Mazeika, M., Kadavath, S., & Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, *32* (cit. on p. 122).

Hermansky, H., Morgan, N., Bayya, A., & Kohn, P. (1992). Rasta-plp speech analysis technique. *1*, 121–124 vol.1 (cit. on p. 62).

Herre, J., Allamanche, E., & Hellmuth, O. (2001). Robust matching of audio signals using spectral flatness features. *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*, 127–130 (cit. on p. 71).

Hinton, G. E. (2012). A practical guide to training restricted boltzmann machines. In *Neural networks: tricks of the trade: second edition* (pp. 599–619). Springer. (Cit. on p. 4).

Holzenberger, N., Du, M., Karadayi, J., Riad, R., & Dupoux, E. (2018). Learning word embeddings: unsupervised methods for fixed-size representations of variable-length speech segments. *Interspeech 2018* (cit. on pp. 28, 57, 181).

Hou, W., Zhu, H., Wang, Y., Wang, J., Qin, T., Xu, R., & Shinozaki, T. (2022). Exploiting adapters for cross-lingual low-resource speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, *30*, 317–329 (cit. on p. 122).

Hsu, W.-N., Sriram, A., Baevski, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., et al. (2021). Robust wav2vec 2.0: analyzing domain shift in self-supervised pre-training. *arXiv preprint arXiv:2104.01027* (cit. on pp. 41, 112).

Hsu, W.-N., Tsai, Y.-H. H., Bolte, B., Salakhutdinov, R., & Mohamed, A. (2021). HuBERT: how much can a bad teacher benefit asr pre-training? *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6533–6537 (cit. on pp. 8, 34, 91, 122, 123, 152, 153).

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: low-rank adaptation of large language models. *International Conference on Learning Representations* (cit. on pp. 124, 128).

Huang, H., Balam, J., & Ginsburg, B. (2023). Leveraging Pretrained ASR Encoders for Effective and Efficient End-to-End Speech Intent Classification and Slot Filling. *Proc. INTERSPEECH 2023*, 2933–2937 (cit. on pp. xxiii, 36).

Huang, K. P., Fu, Y.-K., Zhang, Y., & Lee, H.-y. (2022a). Improving Distortion Robustness of Self-supervised Speech Processing Tasks with Domain Adaptation. *Proc. Interspeech 2022*, 2193–2197 (cit. on p. 113).

Huang, K. P., Fu, Y.-K., Zhang, Y., & Lee, H.-y. (2022b). Improving distortion robustness of self-supervised speech processing tasks with domain adaptation. *arXiv preprint arXiv:2203.16104* (cit. on p. 42).

Hung, Y.-N., Chen, Y.-A., & Yang, Y.-H. (2019). Multitask learning for frame-level instrument recognition. (Cit. on p. 50).

Hwang, D., Misra, A., Huo, Z., Siddhartha, N., Garg, S., Qiu, D., Sim, K. C., Strohman, T., Beaufays, F., & He, Y. (2021). Large-scale asr domain adaptation using self- and semi-supervised learning. (Cit. on p. 55).

Iyer, R., Khargonkar, N., Bilmes, J., & Asnani, H. (2022). Generalized submodular information measures: theoretical properties, examples, optimization algorithms, and applications. *IEEE Transactions on Information Theory*, *68*(2), 752–781 (cit. on p. 53).

Jain, R., Barcovschi, A., Yiwere, M., Bigioi, D., Corcoran, P., & Cucu, H. (2023). A wav2vec2-based experimental study on self-supervised learning methods to improve child speech recognition. *IEEE Access* (cit. on p. 39).

Jiang, D., Li, W., Cao, M., Zou, W., & Li, X. (2021). Speech SimCLR: Combining Contrastive and Reconstruction Objective for Self-Supervised Speech Representation Learning. *Proc. Interspeech 2021*, 1544–1548 (cit. on pp. 58, 75).

Jiang, D., Li, W., Cao, M., Zhang, R., Zou, W., Han, K., & Li, X. (2020). Speech simclr: combining contrastive and reconstruction objective for self-supervised speech representation learning. (Cit. on p. 32).

Juan, Z.-G., Motlicek, P., Zhan, Q., Braun, R., & Vesely, K. (2020). Automatic speech recognition benchmark for air-traffic communications. *Proceedings of Interspeech 2020*, 2297–2301 (cit. on p. 39).

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., et al. (2020). Libri-light: a benchmark for asr with limited or no supervision. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7669–7673 (cit. on pp. 5, 43, 90).

Kamper, H., Jansen, A., & Goldwater, S. (2015). Fully unsupervised small-vocabulary speech recognition using a segmental bayesian model. *Sixteenth Annual Conference of the International Speech Communication Association* (cit. on p. 28).

Kessler, S., Thomas, B., & Karout, S. (2021). Continual-wav2vec2: an application of continual learning for self-supervised automatic speech recognition. *arXiv preprint arXiv:2107.13530* (cit. on p. 42).

Kharitonov, E., Rivière, M., Synnaeve, G., Wolf, L., Mazaré, P.-E., Douze, M., & Dupoux, E. (2021). Data augmenting contrastive learning of speech representations in the time domain. *2021 IEEE Spoken Language Technology Workshop (SLT)*, 215–222 (cit. on pp. 76, 79, 117).

Kharitonov, E., Vincent, D., Borsos, Z., Marinier, R., Girgin, S., Pietquin, O., Sharifi, M., Tagliasacchi, M., & Zeghidour, N. (2023). Speak, read and prompt: high-fidelity text-to-speech with minimal supervision. *arXiv preprint arXiv:2302.03540* (cit. on p. 152).

Kim, D., Cho, D., Yoo, D., & Kweon, I. S. (2018). Learning image representations by completing damaged jigsaw puzzles. *CoRR, abs/1802.01880* (cit. on p. 53).

Kim, E., Jeon, J.-J., Seo, H., & Kim, H. (2022). Automatic Pronunciation Assessment using Self-Supervised Speech Representation Learning. *Proc. Interspeech 2022*, 1411–1415 (cit. on p. 9).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In Y. Bengio & Y. LeCun (Eds.), *2nd international conference on learning representations, ICLR 2014, banff, ab, canada, april 14-16, 2014, conference track proceedings*. (Cit. on p. 30).

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, *114*(13), 3521–3526 (cit. on p. 125).

Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y., & Adi, Y. (2022). Audiogen: textually guided audio generation. *arXiv preprint arXiv:2209.15352* (cit. on p. 45).

Lakhotia, K., Kharitonov, E., Hsu, W.-N., Adi, Y., Polyak, A., Bolte, B., Nguyen, T.-A., Copet, J., Baevski, A., Mohamed, A., & Dupoux, E. (2021). Generative spoken language modeling from raw audio. (Cit. on p. 55).

Latif, S., Rana, R., Khalifa, S., Jurdak, R., & Schuller, B. (2022). Self Supervised Adversarial Domain Adaptation for Cross-Corpus and Cross-Language Speech Emotion Recognition. *IEEE Transactions on Affective Computing* (cit. on p. 113).

Lavechin, M., Bousbib, R., Bredin, H., Dupoux, E., & Cristia, A. (2020). An open-source voice type classifier for child-centered daylong recordings. *Interspeech* (cit. on p. 5).

Lebourdais, M., Tahon, M., LAURENT, A., & Meignier, S. (2022). Overlapped speech and gender detection with WavLM pre-trained features. *Proc. Interspeech 2022*, 5010–5014 (cit. on pp. xxiii, 36).

Lee, J.-H., Lee, C.-W., Choi, J.-S., Chang, J.-H., Seong, W. K., & Lee, J. (2022). CTRL: Continual Representation Learning to Transfer Information of Pre-trained for WAV2VEC 2.0. *Proc. Interspeech 2022*, 3398–3402 (cit. on p. 122).

Lee, J. D., Lei, Q., Saunshi, N., & Zhuo, J. (2020). Predicting what you already know helps: provable self-supervised learning. (Cit. on pp. 51, 53, 54, 59).

Lee, M., & Chang, J.-H. (2018). Augmenting bottleneck features of deep neural network employing motor state for speech recognition at humanoid robots. *ArXiv*, *abs/1808.08702* (cit. on p. 29).

Lee, Y., Jang, K., Goo, J., Jung, Y., & Kim, H. R. (2022). FitHuBERT: Going Thinner and Deeper for Knowledge Distillation of Speech Self-Supervised Models. *Proc. Interspeech 2022*, 3588–3592 (cit. on pp. 136, 140).

Lent, K. (1989). An efficient method for pitch shifting digitally sampled sounds. *Computer Music Journal*, *13*(4), 65–71 (cit. on p. 79).

Li, Y., Pogodin, R., Sutherland, D. J., & Gretton, A. (2021). Self-supervised learning with kernel dependence maximization. (Cit. on pp. 52, 76).

Libera, L. D., Mousavi, P., Zaiem, S., Subakan, C., & Ravanelli, M. (2023). Cl-masr: a continual learning benchmark for multilingual asr. (Cit. on p. 122).

Lin, T.-Q., Lee, H.-y., & Tang, H. (2022). Melhubert: a simplified hubert on mel spectrogram. *arXiv preprint arXiv:2211.09944* (cit. on p. 44).

Liu, A. T., Li, S.-W., & Lee, H.-y. (2021). Tera: self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 2351–2366 (cit. on p. 31).

Liu, A. T., Yang, S.-w., Chi, P.-H., Hsu, P.-c., & Lee, H.-y. (2020a). Mockingjay: unsupervised speech representation learning with deep bidirectional transformer encoders. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (cit. on p. 55).

Liu, A. T., Yang, S.-w., Chi, P.-H., Hsu, P.-c., & Lee, H.-y. (2020b). Mockingjay: unsupervised speech representation learning with deep bidirectional transformer encoders. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6419–6423 (cit. on p. 31).

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., & Bachem, O. (2020). A sober look at the unsupervised learning of disentangled representations and their evaluation. *The Journal of Machine Learning Research*, *21*(1), 8629–8690 (cit. on p. 47).

Lodagala, V. S., Ghosh, S., & Umesh, S. (2023). Pada: pruning assisted domain adaptation for self-supervised speech representations. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 136–143 (cit. on p. 113).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*, 91–110 (cit. on p. 18).

Loweimi, E., Doulaty, M., Barker, J., & Hain, T. (2015). Long-term statistical feature extraction from speech signal and its application in emotion recognition (cit. on p. 55).

Lu, Z., Wang, Y., Zhang, Y., Han, W., Chen, Z., & Haghani, P. (2022). Unsupervised Data Selection via Discrete Speech Representation for ASR. *Proc. Interspeech 2022*, 3393–3397 (cit. on p. 149).

Lugosch, L., Ravanelli, M., Ignoto, P., Tomar, V. S., & Bengio, Y. (2019). Speech Model Pre-Training for End-to-End Spoken Language Understanding. *Proc. Interspeech 2019*, 814–818 (cit. on pp. 94, 95).

MacLean, K. (2018). Voxforge. *Ken MacLean.[Online]. Available: http://www. voxforge. org/home.[Acedido em 2012]* (cit. on p. 81).

Maharana, S. K., Adidam, K. K., Nandi, S., & Srivastava, A. (2023). Acoustic-to-articulatory inversion for dysarthric speech: are pre-trained self-supervised representations favorable? *arXiv preprint arXiv:2309.01108* (cit. on p. 38).

Majumdar, S., Acharya, S., Lavrukhin, V., & Ginsburg, B. (2023). Damage control during domain adaptation for transducer based automatic speech recognition. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 130–135 (cit. on pp. 122, 124).

Manohar, V., Likhomanenko, T., Xu, Q., Hsu, W.-N., Collobert, R., Saraf, Y., Zweig, G., & Mohamed, A. (2021). Kaizen: continuously improving teacher using exponential moving average for semi-supervised speech recognition. (Cit. on p. 55).

Martin, K., Gauthier, J., Breiss, C., & Levy, R. (2023). Probing Self-supervised Speech Models for Phonetic and Phonemic Information: A Case Study in Aspiration. *Proc. INTERSPEECH 2023*, 251–255 (cit. on p. 38).

Martins, A. F. T., & Astudillo, R. F. (2016). From softmax to sparsemax: a sparse model of attention and multi-label classification. *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 1614–1623 (cit. on p. 62).

Masmoudi, A., Khmekhem, M. E., Esteve, Y., Belguith, L. H., & Habash, N. (2014). A corpus and phonetic dictionary for tunisian arabic speech recognition. *LREC*, 306–310 (cit. on p. 148).

Mathieu, B., Essid, S., Fillon, T., Prado, J., & Richard, G. (2010). Yaafe, an easy to use and efficient audio feature extraction software [http://ismir2010.ismir.net/proceedings/ismir2010-75.pdf]. *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 441–446 (cit. on p. 71).

Mauch, M., & Dixon, S. (2014). Pyin: a fundamental frequency estimator using probabilistic threshold distributions. *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, 659–663 (cit. on p. 34).

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017). Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. *Proc. Interspeech 2017*, 498–502 (cit. on p. 66).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, *26* (cit. on pp. 5, 23).

Millet, J., Caucheteux, C., orhan pierre, p., Boubenec, Y., Gramfort, A., Dunbar, E., Pallier, C., & King, J.-R. (2022). Toward a realistic model of speech processing in the brain with self-supervised learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 33428–33443, Vol. 35). Curran Associates, Inc. (Cit. on p. 38).

Mohamed, A., Lee, H.-y., Borgholt, L., Havtorn, J. D., Edin, J., Igel, C., Kirchhoff, K., Li, S.-W., Livescu, K., Maaløe, L., et al. (2022). Self-supervised speech representation learning: a review. *IEEE Journal of Selected Topics in Signal Processing* (cit. on pp. 29, 36, 88).

Murphy, P., & Akande, O. (2005). Cepstrum-Based Harmonics-to-Noise Ratio Measurement in Voiced Speech. In G. Chollet, A. Esposito, M. Faundez-Zanuy, & M. Marinaro (Eds.), *Nonlinear speech modeling and applications* (pp. 199–218). Springer Berlin Heidelberg. (Cit. on p. 62).

Nagrani, A., Chung, J. S., & Zisserman, A. (2017). Voxceleb: a large-scale speaker identification dataset. *Interspeech 2017* (cit. on pp. 58, 65, 81, 93, 106).

Niizumi, D., Takeuchi, D., Ohishi, Y., Harada, N., & Kashino, K. (2021). Byol for audio: self-supervised learning for general-purpose audio representation. *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8 (cit. on p. 35).

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. *European conference on computer vision*, 69–84 (cit. on p. 24).

Olvera, M., Vincent, E., & Gasso, G. (2022). On the impact of normalization strategies in unsupervised adversarial domain adaptation for acoustic scene classification. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 631–635 (cit. on p. 113).

Oord, A. v. d., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (cit. on pp. 51, 66, 76).

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. (2023). Dinov2: learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (cit. on p. 26).

Otake, S., Kawakami, R., & Inoue, N. (2023). Parameter efficient transfer learning for various speech processing tasks. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on p. 122).

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). Fairseq: a fast, extensible toolkit for sequence modeling. *Proceedings of NAACL-HLT 2019: Demonstrations* (cit. on pp. 45, 92).

Palaz, D., Doss, M. M., & Collobert, R. (2015). Convolutional neural networks-based continuous speech recognition using raw speech signal. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4295–4299 (cit. on p. 21).

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–5210 (cit. on pp. 43, 66, 90, 92, 118, 127, 137).

Parcollet, T., Zhang, S., van Dalen, R., Ramos, A. G. C., & Bhattacharya, S. (2023). On the (in) efficiency of acoustic feature extractors for self-supervised speech representation learning. *Interspeech 2023* (cit. on pp. 21, 44).

Pariente, M., Cornell, S., Cosentino, J., Sivasankaran, S., Tzinis, E., Heitkaemper, J., Olvera, M., Stöter, F.-R., Hu, M., Martín-Doñas, J. M., Ditter, D., Frank, A., Deleforge, A., & Vincent, E. (2020). Asteroid: the PyTorch-based audio source separation toolkit for researchers. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2020-October*, 2637–2641 (cit. on p. 117).

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: a review. *Neural Networks*, *113*, 54–71 (cit. on p. 122).

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. *Proc. Interspeech 2019*, 2613–2617 (cit. on p. 79).

Pasad, A., Chou, J.-C., & Livescu, K. (2021). Layer-wise analysis of a self-supervised speech representation model. *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 914–921 (cit. on pp. 12, 46, 102, 105, 122, 139).

Pasad, A., Shi, B., & Livescu, K. (2023). Comparative layer-wise analysis of self-supervised speech models. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on p. 102).

Pascual, S., Ravanelli, M., Serrà, J., Bonafonte, A., & Bengio, Y. (2019a). Learning Problem-Agnostic Speech Representations from Multiple Self-Supervised Tasks. *Proc. Interspeech 2019*, 161–165 (cit. on pp. 33, 50).

Pascual, S., Ravanelli, M., Serrà, J., Bonafonte, A., & Bengio, Y. (2019b). Learning problem-agnostic speech representations from multiple self-supervised tasks. (Cit. on pp. 12, 52, 58, 62).

Paul, D. B., & Baker, J. M. (1992). The design for the Wall Street Journal-based CSR corpus. *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992* (cit. on p. 143).

Peng, H., Long, F., & 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 (cit. on p. 63).

Peng, Y., Tian, J., Yan, B., Berrebbi, D., Chang, X., Li, X., Shi, J., Arora, S., Chen, W., Sharma, R., Zhang, W., Sudo, Y., Shakeel, M., Jung, J.-w., Maiti, S., & Watanabe, S. (2023). Reproducing whisper-style training using an open-source toolkit and publicly available data. *arXiv preprint arXiv:2309.13876* (cit. on pp. 46, 154).

Pennington, J., Socher, R., & Manning, C. (2014, October). GloVe: global vectors for word representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. (Cit. on p. 23).

Pierre, C., Larcher, A., & Jouvet, D. (2022). Are disentangled representations all you need to build speaker anonymization systems? *Proc. Interspeech 2022*, 2793–2797 (cit. on p. 46).

Pitt, M., Johnson, K., Hume, E., Kiesling, S., & Raymond, W. (2005). The buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability. *Speech Communication*, *45*, 89–95 (cit. on pp. 92, 143).

Polyak, A., Adi, Y., Copet, J., Kharitonov, E., Lakhotia, K., Hsu, W.-N., Mohamed, A., & Dupoux, E. (2021). Speech Resynthesis from Discrete Disentangled Self-Supervised Representations. *Proc. Interspeech 2021*, 3615–3619 (cit. on p. 47).

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The kaldi speech recognition toolkit. *IEEE 2011 workshop on automatic speech recognition and understanding*, (CONF) (cit. on pp. 3, 45).

Qian, K., Zhang, Y., Gao, H., Ni, J., Lai, C.-I., Cox, D., Hasegawa-Johnson, M., & Chang, S. (2022, 17–23 Jul). ContentVec: an improved self-supervised speech representation by disentangling speakers. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, & S. Sabato (Eds.), *Proceedings of the 39th international conference on machine learning* (pp. 18003–18017, Vol. 162). PMLR. (Cit. on p. 47).

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356* (cit. on p. 136).

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. *International Conference on Machine Learning*, 28492–28518 (cit. on pp. 20, 37, 46).

Rakotomamonjy, A., Bach, F., Canu, S., & Grandvalet, Y. (2007). More efficiency in multiple kernel learning. *Proceedings of the 24th International Con- ference on Machine Learning (ICML)*, *227* (cit. on p. 53).

Ramanarayanan, V., Lammert, A. C., Rowe, H. P., Quatieri, T. F., & Green, J. R. (2022). Speech as a biomarker: opportunities, interpretability, and challenges. *Perspectives of the ASHA Special Interest Groups*, *7*(1), 276–283 (cit. on p. 41).

Ravanelli, M., & Bengio, Y. (2018). Speaker recognition from raw waveform with sincnet. *2018 IEEE spoken language technology workshop (SLT)*, 1021–1028 (cit. on pp. 2, 21).

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., & Bengio, Y. (2021). Speechbrain. (Cit. on pp. 45, 65, 116).

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., & Bengio, Y. (2020a). Multi-task self-supervised learning for robust speech recognition. (Cit. on pp. 52, 58, 71, 72).

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., & Bengio, Y. (2020b). Multi-task self-supervised learning for robust speech recognition. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6989–6993 (cit. on pp. 33, 63).

Riviere, M., Copet, J., & Synnaeve, G. (2021). Asr4real: an extended benchmark for speech models. *arXiv preprint arXiv:2110.08583* (cit. on pp. 42, 112).

Rivière, M., Joulin, A., Mazar'e, P.-E., & Dupoux, E. (2020). Unsupervised pretraining transfers well across languages. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7414–7418 (cit. on p. 76).

Robinson, J. D., Chuang, C., Sra, S., & Jegelka, S. (2021). Contrastive learning with hard negative samples. *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (cit. on p. 26).

Rogers, A., Kovaleva, O., & Rumshisky, A. (2021). A primer in bertology: what we know about how bert works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866 (cit. on p. 12).

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., & Wayne, G. (2019). Experience replay for continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. (Cit. on p. 125).

Rui, W., Bai, Q., Ao, J., Zhou, L., Xiong, Z., Wei, Z., Zhang, Y., Ko, T., & Li, H. (2022). LightHuBERT: Lightweight and Configurable Speech Representation Learning with Once-for-All Hidden-Unit BERT. *Proc. Interspeech 2022*, 1686–1690 (cit. on pp. 45, 136, 140).

Sadhu, S., He, D., Huang, C.-W., Mallidi, S. H., Wu, M., Rastrow, A., Stolcke, A., Droppo, J., & Maas, R. (2021). wav2vec-C: A Self-Supervised Model for Speech Representation Learning. *Proc. Interspeech 2021*, 711–715 (cit. on p. 66).

Saeed, A., Grangier, D., & Zeghidour, N. (2021). Contrastive learning of general-purpose audio representations. *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3875–3879 (cit. on pp. xxiv, 32, 75, 76, 82).

Sainath, T. N., Kingsbury, B., & Ramabhadran, B. (2012). Auto-encoder bottleneck features using deep belief networks. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4153–4156 (cit. on pp. 2, 29).

Sajjad, H., Dalvi, F., Durrani, N., & Nakov, P. (2023). On the effect of dropping layers of pre-trained transformer models. *Computer Speech and Language*, *77*, 101429 (cit. on p. 139).

Saunshi, N., Malladi, S., & Arora, S. (2020). A mathematical exploration of why language models help solve downstream tasks. *CoRR*, *abs/2010.03648* (cit. on p. 52).

Schatz, T., Peddinti, V., Bach, F., Jansen, A., Hermansky, H., & Dupoux, E. (2013). Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline. *INTERSPEECH 2013 : 14th Annual Conference of the International Speech Communication Association*, 1–5 (cit. on pp. 55, 89).

Schluter, R., Bezrukov, I., Wagner, H., & Ney, H. (2007). Gammatone features and feature combination for large vocabulary speech recognition. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, *4*, IV–649 (cit. on p. 20).

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). Wav2vec: unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862* (cit. on pp. 12, 21, 39).

Schuller, B., Vlasenko, B., Minguez, R., Rigoll, G., & Wendemuth, A. (2007). Comparing one and two-stage acoustic modeling in the recognition of emotion in speech. *2007 IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, 596–600 (cit. on p. 55).

Sennrich, R., Haddow, B., & Birch, A. (2016, August). Neural machine translation of rare words with subword units. In K. Erk & N. A. Smith (Eds.), *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1715–1725). Association for Computational Linguistics. (Cit. on p. 23).

Serizel, R., Bisot, V., Essid, S., & Richard, G. (2017). Acoustic Features for Environmental Sound Analysis. In T. Virtanen, M. D. Plumbley, & D. Ellis (Eds.), *Computational Analysis of Sound Scenes and Events* (pp. 71–101). Springer International Publishing AG. (Cit. on p. 55).

Shafey, L. E., Soltau, H., & Shafran, I. (2019). Joint speech recognition and speaker diarization via sequence transduction. *CoRR*, *abs/1907.05337* (cit. on p. 52).

Shah, R., & Peters, J. (2018). The hardness of conditional independence testing and the generalised covariance measure. *Annals of Statistics*, *48* (cit. on p. 55).

Shi, J., Berrebbi, D., Chen, W., Hu, E.-P., Huang, W.-P., Chung, H.-L., Chang, X., Li, S.-W., Mohamed, A., Lee, H.-y., & Watanabe, S. (2023). ML-SUPERB: Multilingual Speech Universal PERformance Benchmark. *Proc. INTERSPEECH 2023*, 884–888 (cit. on p. 41).

Shin'ya Yamaguchi, S., Kanai, T., Shioda, S., Takeda, N., & Tokyo, J. (n.d.). *Multiple Pretext-Task for Self-Supervised Learning via Mixing Multiple Image Transformations* (tech. rep.). (Cit. on p. 53).

Shor, J., Jansen, A., Han, W., Park, D., & Zhang, Y. (2022). Universal paralinguistic speech representations using self-supervised conformers. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3169–3173 (cit. on p. 76).

Shor, J., & Venugopalan, S. (2022). TRILLsson: Distilled Universal Paralinguistic Speech Representations. *Proc. Interspeech 2022*, 356–360 (cit. on p. 45).

Shukla, A., Petridis, S., & Pantic, M. (2020). Learning speech representations from raw audio by joint audiovisual self-supervision (cit. on p. 50).

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018a). X-vectors: robust dnn embeddings for speaker recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5329–5333 (cit. on p. 66).

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018b). X-vectors: robust dnn embeddings for speaker recognition. *(ICASSP)*, 5329–5333 (cit. on pp. 94, 95).

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *J. Mach. Learn. Res.*, *7*, 1531–1565 (cit. on p. 53).

Stafylakis, T., Rohdin, J., Plchot, O., Mizera, P., & Burget, L. (2019). Self-supervised speaker embeddings. *arXiv preprint arXiv:1904.03486* (cit. on p. 43).

Stevens, S. S., & Volkmann, J. (1940). The relation of pitch to frequency: a revised scale. *The American Journal of Psychology*, *53*(3), 329–353 (cit. on p. 19).

Sundberg, J., & 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 (cit. on p. 62).

Tan, M., & Le, Q. (2019). Efficientnet: rethinking model scaling for convolutional neural networks. *International conference on machine learning*, 6105–6114 (cit. on p. 80).

Tanaka, T., Masumura, R., Sato, H., Ihori, M., Matsuura, K., Ashihara, T., & Moriya, T. (2022). Domain Adversarial Self-Supervised Speech Representation Learning for Improving Unknown Domain Downstream Tasks. *Proc. Interspeech 2022*, 1066–1070 (cit. on p. 113).

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., & Isola, P. (2020). What makes for good views for contrastive learning? *Advances in neural information processing systems*, *33*, 6827–6839 (cit. on p. 76).

Toshniwal, S., Kannan, A., Chiu, C.-C., Wu, Y., Sainath, T., & Livescu, K. (in SLT 2018). A comparison of techniques for language model integration in encoder-decoder speech recognition, 369–375 (cit. on p. 137).

Trinh, V. A., Ghahremani, P., King, B., Droppo, J., Stolcke, A., & Maas, R. (2022). Reducing Geographic Disparities in Automatic Speech Recognition via Elastic Weight Consolidation. *Proc. Interspeech 2022*, 1298–1302 (cit. on p. 122).

Tsai, H.-S., Chang, H.-J., Huang, W.-C., Huang, Z., Lakhotia, K., Yang, S.-w., Dong, S., Liu, A., Lai, C.-I., Shi, J., Chang, X., Hall, P., Chen, H.-J., Li, S.-W., Watanabe, S., Mohamed, A., & Lee, H.-y. (2022, May). SUPERB-SG: enhanced speech processing universal PERformance benchmark for semantic and generative capabilities. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 8479–8492). Association for Computational Linguistics. (Cit. on pp. 37, 42, 89, 101, 112).

Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., & Lucic, M. On mutual information maximization for representation learning. In: In *8th international conference on learning representations (iclr)*. 2020, April (cit. on p. 52).

Turian, J., Shier, J., Khan, H. R., Raj, B., Schuller, B. W., Steinmetz, C. J., Malloy, C., Tzanetakis, G., Velarde, G., McNally, K., Henry, M., Pinto, N., Noufi, C., Clough, C., Herremans, D., Fonseca, E., Engel, J., Salamon, J., Esling, P., ... Bisk, Y. (2022, June). HEAR: Holistic Evaluation of Audio Representations. In D. Kiela, M. Ciccone, & B. Caputo (Eds.), *Proceedings of the neurips 2021 competitions and demonstrations track* (pp. 125–145, Vol. 176). PMLR. (Cit. on p. 37).

Umesh, S., Cohen, L., & Nelson, D. (1999). Fitting the mel scale. *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, *1*, 217–220 (cit. on p. 20).

Vaidya, A. R., Jain, S., & Huth, A. (2022). Self-supervised models of audio effectively explain human cortical responses to speech. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, & S. Sabato (Eds.), *International conference on machine learning, ICML 2022, 17-23 july 2022, baltimore, maryland, USA* (pp. 21927–21944, Vol. 162). PMLR. (Cit. on p. 38).

Vallabha, G., Mcclelland, J., Pons, F., Werker, J., & Amano, S. (2007). Unsupervised learning of vowel categories from infant-directed speech. *Proceedings of the National Academy of Sciences of the United States of America*, *104*, 13273–8 (cit. on p. 4).

Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, *30* (cit. on pp. 25, 30, 32).

Van Den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *International conference on machine learning*, 1747–1756 (cit. on p. 152).

Vander Eeckt, S., & Van Hamme, H. (2023). Weight averaging: a simple yet effective method to overcome catastrophic forgetting in automatic speech recognition. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on pp. 122, 124).

Wang, C., Chen, S., Wu, Y., Zhang, Z.-H., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., He, L., Zhao, S., & Wei, F. (2023). Neural codec language models are zero-shot text to speech synthesizers. *ArXiv*, *abs/2301.02111* (cit. on p. 151).

Wang, F., Cheng, J., Liu, W., & Liu, H. (2018). Additive margin softmax for face verification. *IEEE Signal Processing Letters* (cit. on p. 95).

Wang, H., Qian, Y., Wang, X., Wang, Y., Wang, C., Liu, S., Yoshioka, T., Li, J., & Wang, D. (2022). Improving noise robustness of contrastive speech representation learning with speech reconstruction. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6062–6066 (cit. on pp. xxiii, 36, 42).

Wang, J., Xue, M., Culhane, R., Diao, E., Ding, J., & Tarokh, V. (2020). Speech emotion recognition with dual-sequence lstm architecture. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6474–6478 (cit. on pp. xxiii, 36).

Wang, X., Yu, F., Wang, R., Darrell, T., & Gonzalez, J. E. (2019). Tafe-net: task-aware feature embeddings for low shot learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 55).

Wang, Y., Boumadane, A., & Heba, A. (2021). A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding. *arXiv preprint arXiv:2111.02735* (cit. on pp. 8, 95).

Warden, P. (2018). Speech commands: a dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209* (cit. on p. 93).

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Enrique Yalta Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., & Ochiai, T. (2018). ESPnet: End-to-End Speech Processing Toolkit. *Proc. Interspeech 2018*, 2207–2211 (cit. on p. 45).

Wei, C., Shen, K., Chen, Y., & Ma, T. (2020). Theoretical analysis of self-training with deep networks on unlabeled data. *CoRR*, *abs/2010.03622* (cit. on p. 51).

Wei, K., Iyer, R., & Bilmes, J. (2015, July). Submodularity in data subset selection and active learning. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (pp. 1954–1963, Vol. 37). PMLR. (Cit. on p. 53).

Wei-Ning, H., Anuroop, S., Alexei, B., Tatiana, L., Qiantong, X., Vineel, P., Jacob, K., Ann, L., Collobert, R., Gabriel, S., & Michael, A. (2021). Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training. *Proc. Interspeech 2021*, 721–725 (cit. on p. 70).

Wells, D., Tang, H., & Richmond, K. (2022, September). Phonetic analysis of self-supervised representations of english speech. In H. Ko & J. Hansen (Eds.), *Proceedings of the annual conference of the international speech communication association, interspeech* (pp. 3583–3587, Vol. 2022-September). ISCA. (Cit. on p. 38).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: state-of-the-art natural language processing. *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45 (cit. on p. 92).

Wu, F., Kim, K., Watanabe, S., Han, K. J., McDonald, R., Weinberger, K. Q., & Artzi, Y. (2023). Wav2seq: pre-training speech-to-text encoder-decoder models using pseudo languages. *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on p. 44).

Wu, H.-H., Kao, C.-C., Tang, Q., Sun, M., McFee, B., Bello, J. P., & Wang, C. (2021a). Multi-task self-supervised pre-training for music classification. (Cit. on p. 72).

Wu, H.-H., Kao, C.-C., Tang, Q., Sun, M., McFee, B., Bello, J. P., & Wang, C. (2021b). Multi-task self-supervised pre-training for music classification. *ICASSP 2021* (cit. on p. 50).

Xiao, T., Wang, X., Efros, A. A., & Darrell, T. (2021a). What should not be contrastive in contrastive learning. (Cit. on p. 51).

Xiao, T., Wang, X., Efros, A. A., & Darrell, T. (2021b). What should not be contrastive in contrastive learning. *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (cit. on p. 76).

Xie, S. M., Ma, T., & Liang, P. (2021, 18–24 Jul). Composed fine-tuning: freezing pre-trained denoising autoencoders for improved generalization. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (pp. 11424–11435, Vol. 139). PMLR. (Cit. on p. 123).

Xu, P., Zhu, X., & Clifton, D. A. (2023). Multimodal learning with transformers: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (cit. on p. 27).

Yakovlev, I., Okhotnikov, A., Torgashov, N., Makarov, R., Voevodin, Y., & Simonchik, K. (2023). VoxTube: a multilingual speaker recognition dataset. *Proc. INTERSPEECH 2023*, 2238–2242 (cit. on p. 154).

Yang, S.-w., 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., Lee, K.-t., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., & Lee, H.-y. (2021). SUPERB: Speech Processing Universal PERformance Benchmark. *Proc. Interspeech 2021*, 1194–1198 (cit. on pp. 64, 87–89).

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., Lee, K.-t., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., & Lee, H.-y. (2021). Superb: speech processing universal performance benchmark. (Cit. on pp. 65, 67, 91, 124).

Yang, Y., Shen, F., Du, C., Ma, Z., Yu, K., Povey, D., & Chen, X. (2023). Towards universal speech discrete tokens: a case study for asr and tts. (Cit. on p. 122).

Yeh, C.-F., Hsu, W.-N., Tomasello, P., & Mohamed, A. (2022). Efficient speech representation learning with low-bit quantization. *arXiv preprint arXiv:2301.00652* (cit. on p. 136).

Yoon, J. W., Woo, B. J., & Kim, N. S. (2022). Hubert-ee: early exiting hubert for efficient speech recognition. *arXiv preprint arXiv:2204.06328* (cit. on p. 139).

Yu, D., & Seltzer, M. L. (2011). Improved bottleneck features using pretrained deep neural networks. *Twelfth annual conference of the international speech communication association* (cit. on p. 28).

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, *68*, 49–67 (cit. on pp. 53, 61).

Zaadnoordijk, L., Besold, T., & Cusack, R. (2022). Lessons from infant learning for unsupervised machine learning. *Nature Machine Intelligence*, *4*, 1–11 (cit. on p. 4).

Zaiem, S., Kemiche, Y., Parcollet, T., Essid, S., & Ravanelli, M. (2023). Speech self-supervised representations benchmarking: a case for larger probing heads. (Cit. on pp. 121, 124).

Zaiem, S., Parcollet, T., & Essid, S. (2021). Conditional Independence for Pretext Task Selection in Self-Supervised Speech Representation Learning. *Proc. Interspeech 2021*, 2851–2855 (cit. on pp. 79, 115).

Zamir, A. R., Sax, A., Shen, W. B., Guibas, L. J., Malik, J., & Savarese, S. (2018). Taskonomy: disentangling task transfer learning. *CoRR*, *abs/1804.08328* (cit. on p. 52).

Zanon Boito, M., Ortega, J., Riguidel, H., Laurent, A., Barrault, L., Bougares, F., Chaabani, F., Nguyen, H., Barbier, F., Gahbiche, S., & Estève, Y. (2022, May). ON-TRAC consortium systems for the IWSLT 2022 dialect and low-resource speech translation tasks. In E. Salesky, M. Federico, & M. Costa-jussà (Eds.), *Proceedings of the 19th international conference on spoken language translation (iwslt 2022)* (pp. 308–318). Association for Computational Linguistics. (Cit. on p. 9).

Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., & Tagliasacchi, M. (2021). Soundstream: an end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *30*, 495–507 (cit. on pp. 31, 46, 150).

Zeghidour, N., Teboul, O., Quitry, F. d. C., & Tagliasacchi, M. (2021). Leaf: a learnable frontend for audio classification. *arXiv preprint arXiv:2101.08596* (cit. on p. 21).

Zeghidour, N., Usunier, N., Synnaeve, G., Collobert, R., & Dupoux, E. (2018). End-to-end speech recognition from the raw waveform. *arXiv preprint arXiv:1806.07098* (cit. on pp. 2, 21).

Zeng, H., Wu, Z., Zhang, J., Yang, C., Zhang, H., Dai, G., & Kong, W. (2019). Eeg emotion classification using an improved sincnet-based deep learning model. *Brain sciences*, *9*(11), 326 (cit. on p. 21).

Zhang, X., Zhang, D., Li, S., Zhou, Y., & Qiu, X. (2023). Speechtokenizer: unified speech tokenizer for speech language models. (Cit. on p. 153).

Zhang, Y., Han, W., Qin, J., Wang, Y., Bapna, A., Chen, Z., Chen, N., Li, B., Axelrod, V., Wang, G., et al. (2023). Google USM: scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037* (cit. on pp. 13, 34, 44, 46).

Zuluaga-Gomez, J., Ahmed, S., Visockas, D., & Subakan, C. (2023). Commonaccent: exploring large acoustic pretrained models for accent classification based on common voice. *arXiv preprint arXiv:2305.18283* (cit. on pp. xxiii, 9, 36).

Zuluaga-Gomez, J., Prasad, A., Nigmatulina, I., Sarfjoo, S. S., Motlicek, P., Kleinert, M., Helmke, H., Ohneiser, O., & Zhan, Q. (2023). How does pre-trained wav2vec 2.0 perform on domain-shifted asr? an extensive benchmark on air traffic control communications. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 205–212 (cit. on pp. 42, 137).

Appendix

A

A.1 More on Gaussian Downsampling

The Gaussian Downsampling method, introduced in (Holzenberger et al., 2018), is a technique used to extract a fixed number of equidistant samples from a time series, and in our case speech samples. It involves applying a Gaussian weight to the samples to capture the acoustic information in a word, which is often concentrated at the boundaries. The downsampling process can be described mathematically as follows:

Let $x_1, x_2, \ldots, x_T$ be a speech segment, consisting of a sequence of 40-dimensional log mel features. Equidistant downsampling samples k vectors at intervals $\frac{T}{k-1}$ with proportional interpolation as needed. The i-th sample, denoted as $\hat{x}_{q_i}$, is computed using the following formula:

$$\hat{x}_{q_i} = x_{\lfloor q_i \rfloor} \cdot (\lceil q_i \rceil - q_i) + x_{\lceil q_i \rceil} \cdot (q_i - \lfloor q_i \rfloor)$$

When q_i is an integer, x_{q_i} is taken as the sample; otherwise, the sample is a weighted sum of its left and right neighbors. The closer the neighbor, the more weight it contributes. The embedding of $x_1, x_2, \ldots, x_T$ is the concatenation of $\hat{x}_{q_1}, \hat{x}_{q_2}, \ldots, \hat{x}_{q_k}$.

To introduce non-equidistant downsampling, we assume the space between two samples follows a linear progression and is symmetric with respect to the center. The degree of non-equidistance is controlled by the hyperparameters k and b. The formula for computing the non-equidistant samples is:

$$\Delta_j = q_{j+1} - q_j = k(j-1) + b \quad \text{and} \quad q_{j+1} - q_j = q_{k-j+1} - q_{k-j} \quad \text{for} \quad j = 1, 2, \ldots, \lceil \frac{k}{2} \rceil$$

To avoid losing information during downsampling, Gaussian weighted interpolation can be applied. The formula for computing the Gaussian weighted samples is:

$$\hat{x}_{q_i} = \frac{\int_{T+0.5}^{0.5} g_i(t) f(t) dt}{Z_i}$$

where $g_i(t)$ is a Gaussian density function centered at q_i with variance σ_i^2, and Z_i is the normalization term.

In summary, the Gaussian Downsampling method involves extracting equidistant or non-equidistant samples from a time series and applying Gaussian weights to capture the acoustic information. This technique can be used to create fixed-size embeddings from variable-length sequences of acoustic frames.

A.2 Interactions between Pretext Labels

To understand the interactions between pretext-task labels, studying the evolution of the CI estimate as a function of the weights shows which pretext-task labels seem interchangeable, which ones are complementary, and which ones seem only harmful to the considered downstream task. Figure A.1 shows the CI estimates for weighted combinations of groups of three pretext-task 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 pretext-task label is selected with a weight equal to one, while its weight is equal to zero on the hypotenuse of the right triangle. Figure A.1 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 area.

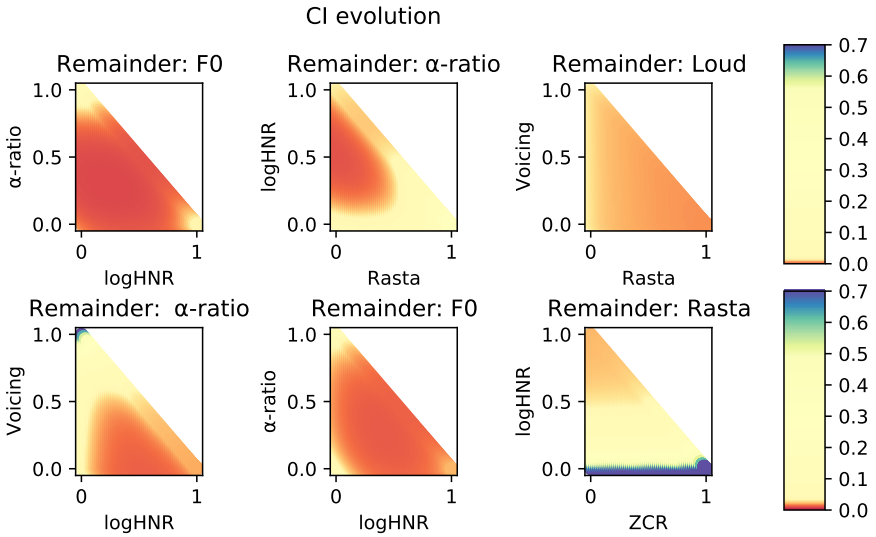

Fig. A.1.: CI-Based utility estimator as a function of the weighting for groups of three pretext-task labels. Top line is for Librispeech, while the bottom one is for VoxCeleb. Three pretext-task labels are presented on every plot, one on the x-axis, one on the y-axis and one that is equal to $1 - x - y$ (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 pretext-task labels. For instance, in the top left corner, the point $(0.5, 0.3)$ corresponds to the CI value of a pretext task weighting logHNR with 0.5, α-ratio with 0.3, and F0 with 0.2.

Selection	α-zero	F0	Loudness	Spec 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	0.45	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
IEMO RFE	0	0	1	1	1	1	0
IEMO MRMR	0	1	0	0	1	1	1
IEMO Spa	0.16	0.22	0	0.14	0.12	0.17	0.19
IEMO Soft	0.29	0.32	0.06	0.24	0.03	0.02	0.03

Tab. A.1.: Weights for every pretext task in every experiment. When the technique only outputs a selection of the pretext tasks, 1 is assigned as a weight for the selected tasks and zero for the non-slected. This table confirms the sparsity induced by the Sparsemax function.

ECOLE
DOCTORALE

Titre: Apprentissage auto-supervisé informé de représentations du signal de parole

Mots clés: traitement de la parole; apprentissage profond; apprentissage auto-supervisé;

Résumé: L'apprentissage des caractéristiques a été un des principaux moteurs des progrès de l'apprentissage automatique. L'apprentissage auto-supervisé est apparu dans ce contexte, permettant le traitement de données non étiquetées en vue d'une meilleure performance sur des tâches faiblement étiquetées.
La première partie de mon travail de doctorat vise à motiver les choix dans les pipelines d'apprentissage auto-supervisé de la parole qui apprennent les représentations non supervisées. Dans cette thèse, je montre d'abord comment une fonction basée sur l'indépendance conditionnelle peut être utilisée pour sélectionner efficacement et de manière optimale des tâches de pré-entraînement adaptées à la meilleure performance sur une tâche cible.
La deuxième partie de mon travail de doctorat étudie l'évaluation et l'utilisation de représentations auto-supervisées pré-entraînées. J'y explore d'abord la robustesse des benchmarks actuels d'auto-supervision de la parole aux changements dans les choix de modélisation en aval. Je propose, ensuite, de nouvelles approches d'entraînement en aval favorisant l'efficacité et la généralisation.

Title: Informed speech self-supervised learning

Keywords: speech processing; deep learning; self-supervised learning

Abstract: Feature learning has been driving machine learning advancement with the recently proposed methods getting progressively rid of handcrafted parts within the transformations from inputs to desired labels. Self-supervised learning has emerged within this context, allowing the processing of unlabeled data towards better performance on low-labeled tasks.
The first part of my doctoral work is aimed towards motivating the choices in the speech self-supervised pipelines learning the unsupervised representations. In this thesis, I first show how conditional-independence-based scoring can be used to efficiently and optimally select pretraining tasks tailored for the best performance on a target task.
The second part of my doctoral work studies the evaluation and usage of pretrained self-supervised representations. I explore, first, the robustness of current speech self-supervision benchmarks to changes in the downstream modeling choices. I propose, second, fine-tuning approaches for better efficicency and generalization.

Institut Polytechnique de Paris
91120 Palaiseau, France