Perturbation-invariant Speech Representation Learning by Online Clustering

by

Heng-Jui Chang

B.S., National Taiwan University (2021)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

Authored by: Heng-Jui Chang
Department of Electrical Engineering and Computer Science
January 10, 2024

Certified by: James R. Glass
Senior Research Scientist
Computer Science and Artificial Intelligence Laboratory
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Perturbation-invariant Speech Representation Learning by Online Clustering

by

Heng-Jui Chang

Submitted to the Department of Electrical Engineering and Computer Science
on January 10, 2024 in partial fulfillment of the requirements for the degree of

Master of Science

ABSTRACT

Despite success across various tasks, self-supervised speech models face significant challenges in enhancing content-related performance with unlabeled data, requiring substantial computational resources. Meanwhile, learning from clustered discrete units has been shown to facilitate accurate phonetic representations. Thus, this thesis investigates speaker and noise-invariant speech representations. First, Speaker-invariant Clustering (Spin) is proposed to extract content representations through online clustering and speaker-invariant cross-view prediction. Second, Robust Spin (R-Spin) is devised to extend Spin to handle more distorted speech signals by leveraging acoustic pieces. Furthermore, this thesis includes a diverse set of evaluation and visualization techniques to quantitatively and qualitatively analyze the perturbation invariability of the proposed methods. This thesis offers approaches to producing perturbation-invariant speech representations and deeply investigates the characteristics of the learned representations, providing insights into these models and cultivating future extension possibilities.

Thesis supervisor: James R. Glass
Title: Senior Research Scientist

Acknowledgments

I would like to express my sincere gratitude to the individuals who have played a pivotal role in the completion of this thesis. Their guidance, support, and encouragement have been invaluable throughout this journey.

To my advisor, Jim Glass, and the members of the Spoken Language Systems group at MIT CSAIL: Jim, your enduring patience and mentorship have been my guiding lights throughout the intricate process of completing this thesis. Your support has been indispensable in ensuring this academic endeavor reaches fruition with clarity and excellence. Moreover, I am grateful to the members of the SLS group for your insightful feedback, which has enriched the quality of my research.

To Prof. Lin-shan Lee at NTU: Your mentorship guided me into the captivating realm of speech processing. Not only did you serve as a beacon in academic research, but your wisdom also extended into life's broader lessons. Your unwavering support and guidance during my formative years laid the foundation for my interest and success in this research area, and I am truly grateful for your mentorship.

To Prof. Hung-yi Lee at NTU: Your guidance has shaped my early research papers. I extend my gratitude to you for providing the opportunity to collaborate on cross-institutional projects, an experience that has enriched my academic journey in ways beyond measure.

To my internship mentor, Andy Chung, at Meta: Your mentorship, coupled with exposure to real-world applications, significantly contributed to the depth and relevance of my research.

To my Uncle Elmer Peng and Aunt Yu-Chien Chang: Heartfelt thanks for your unwavering support and insightful advice on my career path and life choices. Your guidance has been a constant source of wisdom, and I am grateful to have you as role models, shaping my personal and professional journey.

To my family and friends: Each of you has played a unique and indispensable role in my journey. Thank you for being a constant source of inspiration, laughter, and support. I'm especially thankful to my grandmother Chin-Feng Peng, father Shih-Chu Chang, and mother Hsiao-Chien Wang, your boundless love, encouragement, and understanding have been the pillars of strength throughout my life. I love you.

In conclusion, this thesis is not just the result of my efforts but a collective achievement made possible by the support of these remarkable individuals. Thank you all for being an integral part of this academic endeavor.

This work was supported in part by DSTA.

Bibliographic Note

Some of the work presented in this thesis has previously appeared in peer-reviewed publications. Chapter 2 was published at Interspeech 2023 (Chang et al., 2023a). Chapter 3 is currently under review (Chang and Glass, 2023).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Conventional machine learning for speech processing, like automatic speech recognition (ASR), requires large human-transcribed speech corpora to perform well. Commonly used datasets consist of hundreds and thousands of hours of speech recordings. Unfortunately, corrupted audio files and annotation mistakes can jeopardize the training of machine learning models. Thus, obtaining high-quality annotations at this scale is expensive and challenging. To address the issue of collecting labeled speech corpora, researchers have developed approaches that can make the best out of limited data.

Data augmentation is used to generate more diverse data (Park et al., 2019; Ko et al., 2015). Semi-supervised learning first trains an initial model with labeled data and then labels an unlabeled corpus for further training (Xu et al., 2020; Likhomanenko et al., 2020). Unlike semi-supervised learning, **self-supervised learning (SSL)** leverages large-scale unlabeled data to train an encoder to offer good initialization and representations for downstream applications. After pre-training, the model can be fine-tuned with a small amount of labeled data to perform tasks like ASR. SSL pre-trained models have been shown to outperform conventional machine learning approaches in various speech processing tasks (Yang et al.,

2021; Chang et al., 2021b; Tsai et al., 2022; Mohamed et al., 2022).

SSL methods produce pseudo-targets from raw data for the model to learn. Some models learn to predict unknown Mel filterbank features given the partial context of audio utterances (Chung et al., 2019; Liu et al., 2021b,a). Contrastive learning (Oord et al., 2018) serves as a good training objective by encouraging models to distinguish hidden representations from the same or different audio segments (Baevski et al., 2020). Cluster IDs obtained by clustering continuous audio representations can also be used as pseudo-labels (Hsu et al., 2021a). Finally, some methods learn to distill knowledge from the exponential moving average of the model itself (Baevski et al., 2022). Some other approaches combine techniques in a multi-task learning style (Chung et al., 2021).

Many applications of SSL models focus on speech recognition to reduce the need for large-scale transcribed corpora (Hsu et al., 2021a; Baevski et al., 2022; Liu et al., 2023). As a result, extracting content representations has become a crucial aspect of speech SSL research (Tjandra et al., 2021; Chan and Ghosh, 2022; Peyser et al., 2022; Williams, 2022). While a good speech representation encompasses information from multiple aspects, most SSL methods lack explicit speaker disentanglement, making the models fail when perturbation is present in the input signals. A desirable SSL pre-trained speech encoder for content-related applications should be invariant to perturbations like the talker's voice and background noise. Thus, removing unrelated information from speech representations to make speech encoders perturbation-invariant is the goal of this thesis, making it easier for downstream applications to extract content from these models.

Despite the success of SSL for speech processing, the pre-training process requires thousands of GPU hours and leaves huge memory footprints, making training perturbation-invariant models from scratch infeasible. Alternatively, self-supervised fine-tuning (SSFT) methods are proposed to fine-tune pre-trained SSL models with lower computation costs for certain applications (Qian et al., 2022; Zhu et al., 2023). Nevertheless, prior SSFT methods are also costly because of the nature of their training objectives. Hence, due to the

benefits suggested by previous studies (Liu et al., 2023), we incorporate online clustering into the SSFT framework, simultaneously learning linguistic units from speech and reducing perturbation effects through perturbation-invariant training.

1.2 Thesis Contributions

This thesis proposes two self-supervised fine-tuning frameworks with online clustering objectives to produce perturbation-invariant speech representations. In Chapter 2, we introduce the novel speaker-invariant clustering (Spin) framework to make speech models invariant to the talker's voice. This method successfully improves pre-trained SSL models for speech recognition with limited data and resources. Next, Chapter 3 introduces Robust Spin (R-Spin) to enhance Spin with noise-invariant training and advanced learning targets. R-Spin inherits the benefits of Spin but has more freedom to adapt to more diverse acoustic scenarios. In Chapter 4, we offer comprehensive analyses of Spin and R-Spin to understand the mechanisms of these methods, including visualizing hidden representations and measuring the quality of discrete acoustic units. Finally, we conclude this thesis and discuss future work in Chapter 5.

Chapter 2

Speaker-invariant Clustering (Spin)

This chapter introduces speaker-invariant clustering (Spin), a novel self-supervised fine-tuning framework for learning better content representations via online clustering. This method improves pre-trained SSL models to capture phonetic units with minimal training costs.

We provide the background for learning from clustering in Section 2.1. Next, we introduce the Spin SSFT framework in Section 2.2. The experimental results are discussed in Section 2.3. Parts of this chapter were published at Interspeech 2023 (Chang et al., 2023a).

2.1 Background

Before Spin, ContentVec was proposed by Qian et al. (2022) to boost HuBERT (Hsu et al., 2021a), one of the large-scale and powerful SSL models, for capturing better content representations from speech signals. ContentVec is fine-tuned on top of a pre-trained HuBERT model, but the inputs are augmented with speaker perturbation so that the model learns to produce representations invariant to the speaker. ContentVec's learning targets are obtained by converting all LibriSpeech data into the same speaker with a voice conversion model and then applying K-means clustering to the hidden features of HuBERT, given the converted inputs. Although showing promising results, ContentVec suffers from the requirement of a

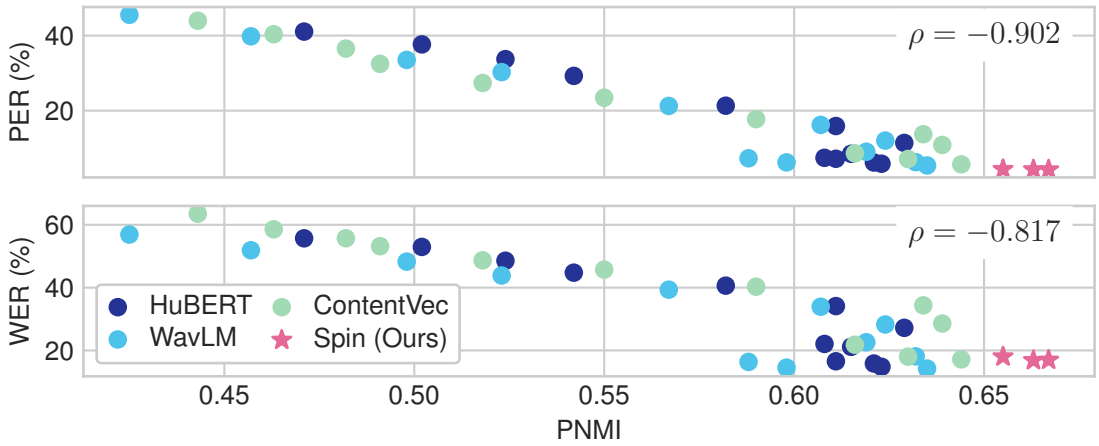


Figure 2.1: Content representation quality (PNMI) vs. phoneme/ word error rates (PER/WER) of SSL model hidden layer representations under a simplified setup in SUPERB Yang et al. (2021). ρ is Spearman's rank correlation coefficient.

voice conversion model and high computational costs (600+ GPU hours). Thus, this prior study inspired us to develop a more efficient algorithm to offer the same performance.

Meanwhile, previous SSL methods utilize clustering continuous representations to capture better linguistic units from speech. For instance, HuBERT (Hsu et al., 2021a) learns from cluster IDs of K-means clustered hidden features from SSL models. Similarly, DinoSR (Liu et al., 2023) incorporates online clustering into the data2vec (Baevski et al., 2022) SSL framework to extract content for ASR. These works both found that learning from discrete units obtained by clustering features yields better ASR performance. Nonetheless, no quantitative analyses were reported to verify whether representations closer to the underlying phonetic content yield better performance for speaker-invariant downstream tasks like ASR.

To verify this assumption, we extract speech representations from each layer of three pre-trained SSL models (HuBERT, WavLM (Chen et al., 2022), and ContentVec (Qian et al., 2022)) and compute two metrics: 1) the phone-normalized mutual information (PNMI; Section 4.5.1) that measures the similarity between phonemes and the discrete units derived by running K-means clustering on the extracted representation ($K = 256$); 2) the phone/word recognition error rate using the extracted features and a lightweight predictor as detailed in Section 2.3.3.

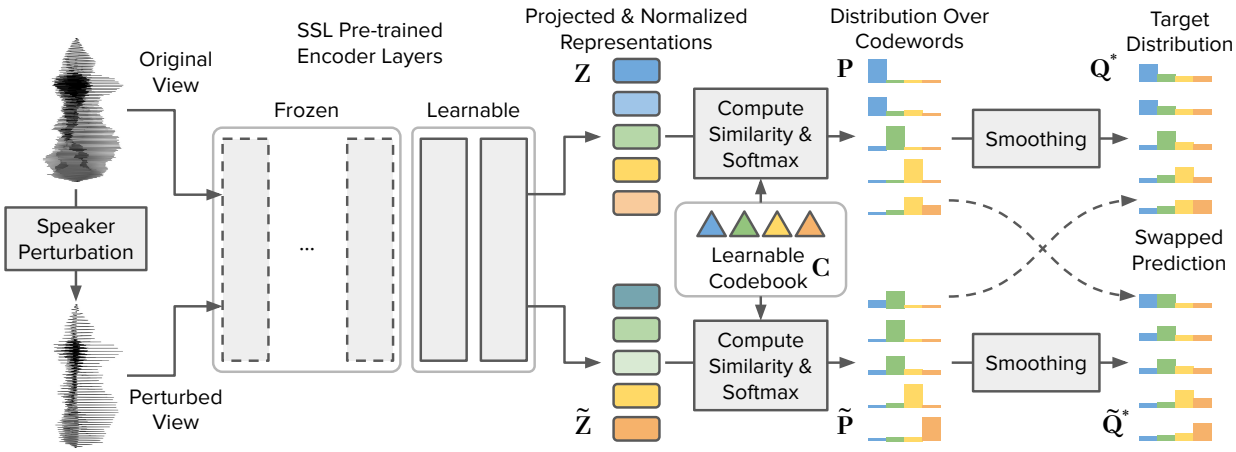


Figure 2.2: The Spin self-supervised fine-tuning framework. A new view is generated with a simple speaker perturbation. A pre-trained speech SSL model extracts representations from both utterances ($\mathbf{Z}$/ $\tilde{\mathbf{Z}}$). Representations are projected, normalized, and quantized with a learnable codebook into probability distributions ($\mathbf{P}$/ $\tilde{\mathbf{P}}$). The distributions are smoothed to enforce full codebook usage ($\mathbf{Q}^*$/ $\tilde{\mathbf{Q}}^*$). Finally, each frame's distribution is used to predict the target distribution produced by the other view ($\mathbf{P} \rightarrow \tilde{\mathbf{Q}}^*$ and $\tilde{\mathbf{P}} \rightarrow \mathbf{Q}^*$).

In Figure 2.1, higher PNMI representations generally offer better recognition results across all models and layers. The Spearman's rank correlation coefficients for PNMI-PER and PNMI-WER verify the strong correlation between the content encoded and downstream performance, leading us to propose an SSFT method that learns from discrete acoustic units to focus on content encoding.

2.2 Proposed Method

2.2.1 The Spin Framework

An overview of the proposed Spin framework is illustrated in Figure 2.2. Inspired by Swapping Assignments between Views (SwAV) (Caron et al., 2020) for image representation learning, our idea is to learn speaker-invariant clusters that capture the same content shared between perturbed speech and the original speech.

2.2.2 Speaker Perturbation

To alter speaker identity without changing the spoken content, we adopt an algorithm proposed by Choi et al. (2021) as ContentVec (Qian et al., 2022). The algorithm randomly and uniformly scales formant frequencies and F0, and random equalization is applied. Because voice information resides in the formant frequencies and F0 (Eide and Gish, 1996), and the content is stored in the relative ratio between formant frequencies (Stevens, 1987), this algorithm efficiently alters speakers with little content loss.

2.2.3 Online Clustering

With the speaker-augmented and the original speech pair, we aim to discover the consistent underlying content via speaker-invariant clustering. As illustrated in Figure 2.2, the output of the original view from the encoder is linearly projected and L2-normalized to representation $\mathbf{Z} = [\boldsymbol{z}_1 \dots \boldsymbol{z}_B]^\intercal \in \mathbb{R}^{B \times D}$, where D is the dimension of the representation, and B is the number of frames in a batch. A probability distribution is computed per frame by taking a softmax over the scaled cosine similarity between $\mathbf{Z}$ and a learnable codebook of K codewords $\mathbf{C} = [\boldsymbol{c}_1 \dots \boldsymbol{c}_K]^\intercal \in \mathbb{R}^{K \times D}$ as

$$p\left(k|\boldsymbol{z}_b\right) = \frac{\exp\left(\boldsymbol{z}_b^\intercal \boldsymbol{c}_k / \tau\right)}{\sum_{k'} \exp\left(\boldsymbol{z}_b^\intercal \boldsymbol{c}_{k'} / \tau\right)},$$

for $k \in [K]$, $b \in [B]$,[1] $\|\boldsymbol{c}_k\|_2 = 1$, and $\tau > 0$ is a scaling temperature. We define $q\left(k|\tilde{\boldsymbol{z}}_b\right)$ the distribution over the same codebook using augmented speech. To learn speaker-invariant clusters that capture the unchanged content, distributions over the codebook should ideally be similar regardless of the speaker, i.e., minimizing the cross-entropy $-q\left(k|\tilde{\boldsymbol{z}}_b\right) \log p\left(k|\boldsymbol{z}_b\right)$.

[1] $[N]$ is defined as $\{1, 2, \dots, N\}$.

2.2.4 Codebook Smoothing

In practice, minimizing the aforementioned cross-entropy term leads to a trivial solution where all representations are clustered into a single codeword if q is obtained similarly with p. To address the issue, we smooth the target distribution q to encourage higher utilization of the codewords. Following Asano et al. (2020), q is obtained by

$$\mathbf{Q}^* \in \arg\max_{\mathbf{Q}} \operatorname{Tr}\left(\mathbf{Q}\mathbf{C}\mathbf{Z}^\intercal\right) + \varepsilon H\left(\mathbf{Q}\right), \tag{2.1}$$

where $\mathbf{Q}^* \in [0,1]^{B\times K}$, $q\left(k|\boldsymbol{z}_b\right) = \mathbf{Q}^*_{b,k}$, and $H\left(\mathbf{Q}\right) = -\sum_{ij}\mathbf{Q}_{ij}\log\mathbf{Q}_{ij}$ is the entropy. The optimized variable $\mathbf{Q}$ is constrained so that each row is a probability distribution over the K codewords. When $\varepsilon = 0$, q is a categorical distribution and easily collapses to using only one codeword. When $\varepsilon > 0$, the entropy term smooths the distribution so that all codewords can be utilized more evenly, whereas a higher ε leads to a more uniform distribution. Equation 2.1 can be efficiently solved by the Sinkhorn-Knopp algorithm on GPUs (Cuturi, 2013). Note that no gradient is applied to q.

2.2.5 Speaker-invariant Swapped Prediction

With the smoothed target distribution q, the goal is to perform speaker-invariant swapped prediction by minimizing the cross-entropy loss

$$\mathcal{L}_{\text{Spin}} = -\frac{1}{2B}\sum_b\sum_k\left[q\left(k|\tilde{\boldsymbol{z}}_b\right)\log p\left(k|\boldsymbol{z}_b\right) + q\left(k|\boldsymbol{z}_b\right)\log p\left(k|\tilde{\boldsymbol{z}}_b\right)\right],$$

where the second term emerges from the interchangeability of the role of the augmented and original speech.

This objective encourages the model to produce similar representations at the same position between two different views by learning a codebook encoding speaker-invariant acoustic units. Since learning fewer parameters reduces computation, and top layers encode phonetic

content (Pasad et al., 2021; Chang et al., 2022; Tseng et al., 2022), we propose fine-tuning some of the top layers to balance the tradeoff between downstream performance and training computation. Unlike previous methods, Spin does not require random masking, so all frames are utilized and contribute to updating the network. Spin is limited to pre-trained models because only the positional information is learned if the model is trained from scratch.

2.3 Experiments

2.3.1 Data

Spin is trained with the LibriSpeech train-clean 100 hours subset (Panayotov et al., 2015). We found that training with more data does not improve performance.

2.3.2 Implementation

We apply the Spin method to HuBERT (Hsu et al., 2021a) and WavLM (Chen et al., 2022), with only the last two layers being fine-tuned (7M parameters per layer).[2] We set $D = 256$, $\tau = 0.1$, $\varepsilon = 0.02$, and sweep the codebook sizes $K \in \{128, 256, 512, 1024, 2048\}$. Each view's mini-batch has at most 256 seconds of speech, corresponding to $B = 12.8$k frames. The learning rate is first linearly increased from 0 to 10^{-4} for 2.5k updates, then linearly decreased to 10^{-6} for 2.5k updates. The Sinkhorn-Knopp algorithm iterates three times to compute $\mathbf{Q}^*$ per view. Spin is trained on a single RTX A5000 GPU, taking approximately 45 minutes. We select models that are trained with all 5k updates.

HuBERT and **WavLM** are pre-trained to predict cluster IDs of masked audio frames from clustering MFCC features or hidden representations of pre-trained models. These models serve as baselines for Spin. **data2vec** (Baevski et al., 2022) is trained to masked-predicting hidden representations of the exponential moving average of the model itself. We avoid applying Spin to data2vec because the phonetic content resides at the bottom

[2] Checkpoints: https://github.com/s3prl/s3prl

Table 2.1: SUPERB (Yang et al., 2021) phoneme recognition (PR), automatic speech recognition (ASR), keyword spotting (KS), query-by-example (QbE), intent classification (IC), and slot filling (SF). Metrics include accuracy (Acc%), phoneme error rate (PER%), word error rate (WER%), maximum term weighted value (MTWV), F1 score, and concept error rate (CER%). PT and SSFT denote pre-training and self-supervised fine-tuning. Top-3 best results are underlined. The number of hours of processed speech is computed with Eq. 2.2.

	Training Processed Speech in Hours		Content				Semantic		
			PR	ASR	KS	QbE	IC	SF	
Method	PT	SSFT	PER↓	WER↓	Acc↑	MTWV↑	Acc↑	F1↑	CER↓
wav2vec 2.0♠	640k	0	5.74	6.43	96.23	0.0233	92.35	88.30	24.77
HuBERT♠	506k	0	5.41	6.42	96.30	0.0736	98.34	88.53	25.20
WavLM♠	1439k	0	4.84	6.31	96.79	0.0870	98.63	89.38	22.86
data2vec♠	420k	0	4.69	4.94	96.56	0.0576	97.63	88.59	25.27
ContentVec_{500}♣	506k	76k	4.54♢	5.70	96.40	0.0590	99.10	89.60	23.60
HuBERT + Spin_{256}	506k	356	4.39	6.34	96.53	0.0912	98.34	89.00	24.32
WavLM + Spin_{256}	1439k	356	4.18	5.88	96.20	0.0879	98.52	88.84	24.06

♠ Source: https://superbbenchmark.org/leaderboard (as of March 7, 2023).
♣ Reported in the original ContentVec paper (Qian et al., 2022).
♢ Re-implement for a fair comparison (original: 4.90).

layers (Table 4.1), requiring fine-tuning many more top layers, and thus increasing computation costs. **ContentVec** is a stronger baseline as it is also trained to improve extracting content with speaker disentanglement. ContentVec learns to mask-predict a pre-trained HuBERT hidden representation K-means clusters. Based on the number of clusters in the target, there are two versions: ContentVec_{100} and ContentVec_{500}. These SSL models share a similar architecture: a 7-layer CNN feature extractor followed by a 12-layer transformer encoder (Vaswani et al., 2017), having approximately 95M parameters each. All models are frozen in evaluation tasks, and continuous transformer encoder hidden representations are used unless otherwise specified.

2.3.3 SUPERB Benchmark

This section evaluates Spin on content and semantic tasks in the Speech processing Universal PERformance Benchmark (SUPERB) (Yang et al., 2021). Each task and SSL model uses a set of learnable weights to compute weighted-sum representations across hidden layers of

	ContentVec_{500}	Spin_{128}	Spin_{256}	Spin_{512}	Spin_{1024}	Spin_{2048}
HuBERT	0.00017	0.00000	0.00000	0.00000	0.00000	0.00000
WavLM	0.07486	0.00284	0.00222	0.00300	0.00012	0.00000
ContentVec_{500}		0.25110	0.22849	0.25472	0.04632	0.00837

Figure 2.3: t-test p-values of SUPERB Yang et al. (2021) phoneme recognition error rates. All Spin models here are based on HuBERT. The blue and red cells indicate p-values less and greater than 0.05, respectively.

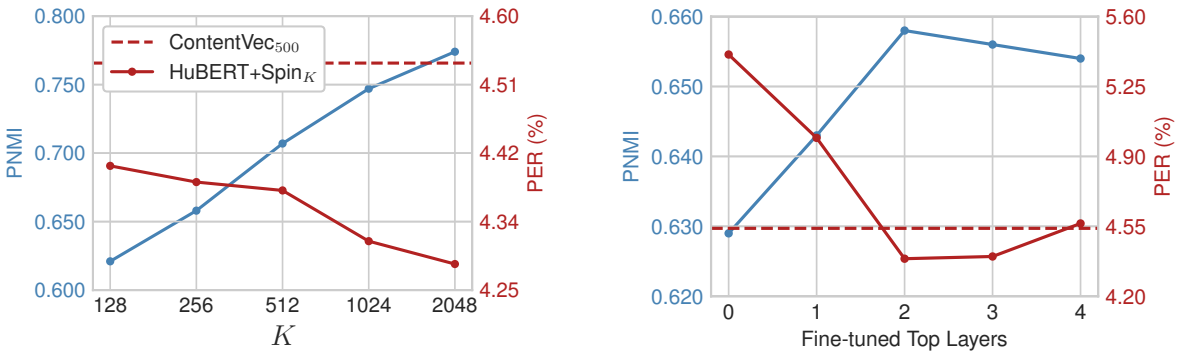
the frozen SSL model. The aggregated features are then fed to a prediction head for supervised training. We report phoneme recognition (PR), automatic speech recognition (ASR), keyword spotting (KS), query-by-example spoken term discovery (QbE), intent classification (IC), and slot filling (SF). We choose $K = 256$ for Spin as it offers the best overall results.

In Table 2.1, Spin benefits learning content representations because HuBERT and WavLM are improved in content-related tasks (PR, ASR, and QbE) while reducing the performance gap with ContentVec. According to the significance test on PR in Figure 2.3, Spin passes a t-test when compared with HuBERT and WavLM. Increasing the codebook size ($K = 1024$ and 2048) outperforms ContentVec with a $p < 0.05$.

In order to quantify machine-independent costs, we examine the hours of processed speech during training, where:

$$\text{processed speech} = \text{training steps} \times \text{effective batch size} \tag{2.2}$$

Based on these data, Spin requires less than 0.5% of the computation of ContentVec in order to outperform it in PR and QbE while offering similar performance in other tasks. Moreover, most models perform similarly in KS and IC, and we found these tasks sensitive to hyperparameters, making them less suitable for comparison. Overall, Spin improves SSL models with a meager budget.



(a) (b)

Figure 2.4: PNMI and PER of HuBERT + Spin with different (a) codebook sizes and (b) fine-tuning layers. Fine-tuning zero layers in (b) denotes the HuBERT baseline. Results in (b) use $K = 256$.

2.3.4 Analysis

This section analyzes the design of the Spin training framework. As plotted in Figure 2.4a, we find that a larger codebook size improves discrete unit quality and PER. Even when K is only 128, Spin outperforms ContentVec, indicating the effectiveness of the proposed method. In Figure 2.4b, Spin surpasses HuBERT and ContentVec when fine-tuning two or three layers. However, fine-tuning the top four layers simultaneously leads to worse performance, and we also observed this phenomenon when fine-tuning more layers. This suggests that providing too much freedom to the model training leads to collapsing because a trivial solution to the Spin objective is to produce representations irrelevant to the inputs. E.g., each frame represents the corresponding position (similar to positional encoding in (Vaswani et al., 2017)). Therefore, this finding implies that this framework cannot be used to process audio recordings from other acoustic domains since prior work has found that the lower layers highly correlate to low-level audio processing like denoising (Chang et al., 2021a; Gong et al., 2023). We address this issue in the next chapter.

2.4 Chapter Summary

This chapter proposes Spin, a self-supervised fine-tuning method that improves content representations motivated by speaker disentanglement and the strong relationship between discrete unit quality and downstream performance. We offer empirical evidence that the proposed method benefits various content-related tasks. Although Spin was only applied to HuBERT and WavLM, Spin provides a new method to further enhance speech representation models after pre-training at a very low cost.

Chapter 3

Robust Spin (R-Spin)

This chapter introduces Robust Spin (R-Spin), a data-efficient self-supervised fine-tuning framework for speaker and noise-invariant speech representations by learning discrete acoustic units with Spin. R-Spin resolves Spin's issues and enhances content representations by learning to predict acoustic pieces. R-Spin offers a 12X reduction in computational resources compared to previous state-of-the-art methods while outperforming them in severely distorted speech scenarios.

In this chapter, we first provide the background of noise-invariant training methods in Section 3.1. We then introduce the proposed R-Spin framework in Section 3.2. Finally, experiments are presented and discussed in Section 3.3.

3.1 Background

In the previous chapter, we focused on capturing linguistic content information in speech signals with the Spin model. However, in addition to modeling content information in speech, numerous studies are dedicated to investigating the robustness of speech SSL representations. While current methods perform well on clean speech datasets, they are vulnerable to out-of-domain data like distorted audio signals (Hsu et al., 2021b). To mitigate this vulnerability, researchers have proposed noise-invariant training techniques. Huang et al. (2022a)

propose HuBERT-MGR via domain adversarial training to render the fine-tuned HuBERT model invariant to domain shifts. WavLM (Chen et al., 2022) integrates denoising with the HuBERT pre-training framework, achieving state-of-the-art performance in many speech processing downstream tasks. Similarly, Zhu et al. (2023) propose Robust data2vec, introducing perturbations to the input to predict the exponential moving average teacher model's representations. In deHuBERT (Ng et al., 2023), the Barlow Twins loss (Zbontar et al., 2021) is applied to encourage representation invariability to input perturbations. Although many methods have shown success in noisy speech recognition (Wang et al., 2022; Zhu et al., 2022; Huang et al., 2022b; Hu et al., 2023), to our knowledge, none have concurrently addressed the disentanglement of speaker and noise while enhancing content information. Furthermore, these approaches exhibit inefficiency, often requiring high computation costs and iterating large corpora over numerous epochs. Due to the need for efficiently obtaining good content and robust representations in real-world applications, we extend Spin with noise-invariant training and acoustic piece pseudo-label learning.

3.2 Proposed Method

3.2.1 The R-Spin Framework

The proposed R-Spin framework is shown in Figure 3.1. R-Spin is based on Spin, which is described in Section 2.2. We introduce noise-invariant training by perturbing inputs (Section 3.2.2) to improve robustness. Moreover, an auxiliary pseudo-label prediction loss (Section 3.2.3) enables fine-tuning the entire model without collapsing. Acoustic Pieces (Section 3.2.4) are incorporated with the auxiliary loss to improve performance further.

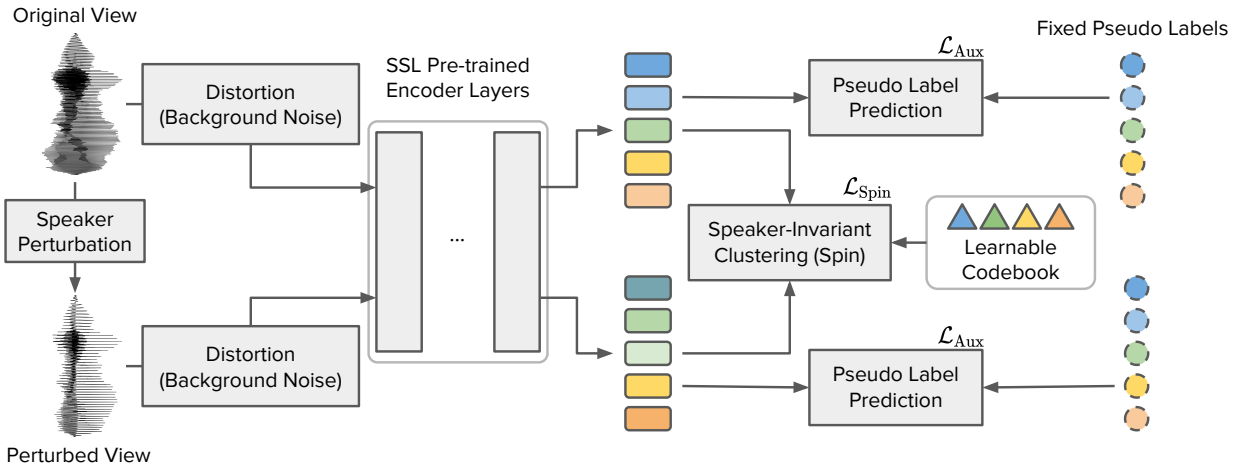


Figure 3.1: The proposed R-Spin self-supervised fine-tuning framework. The input utterance is perturbed into a different voice and distorted with random noise or reverberation. Both the original and perturbed views are fed into an encoder initialized with an SSL pre-trained model. The model is optimized with Speaker-invariant Clustering (Spin) objective ($\mathcal{L}_{\text{Spin}}$) and frame-wise pseudo-label prediction loss ($\mathcal{L}_{\text{Aux}}$).

3.2.2 Noise-invariant Training

To improve the robustness of SSL models, we introduce noise-invariant training by introducing audio distortions after speaker perturbation. After adding background noises to the input signals, the utterances are processed with the Spin SSFT framework. We anticipate the model will acquire the capacity to concurrently eliminate both noise and speaker-related information, thereby enabling the trained model to generate robust content representations.

3.2.3 Auxiliary Pseudo-label Prediction Loss

As mentioned in Chapter 2, Spin is constrained to fine-tuning solely the uppermost layers of pre-trained SSL encoders. Otherwise, the model converges towards a trivial solution, yielding outputs irrelevant to the corresponding inputs. This limitation may not be overly problematic when the application domain closely aligns with the pre-training data. However, given that the lower layers are associated with low-level signal processing like denoising (Chang et al., 2021a; Gong et al., 2023), subjecting these layers to fine-tuning is imperative. This

adjustment is particularly beneficial in enhancing the model's robustness to out-of-domain data. Consequently, we propose a pseudo-label prediction loss to prevent models from collapsing.

The pseudo-label prediction task is a frame-wise classification problem similar to HuBERT (Hsu et al., 2021a). The loss function is

$$\mathcal{L}_{\text{Aux}} = -\frac{1}{2B} \sum_{b \in [B]} \left[\log p\left(y_b | \boldsymbol{h}_b\right) + \log p\left(y_b \left| \tilde{\boldsymbol{h}}_b \right.\right) \right], \tag{3.1}$$

where y_b is the pseudo-label at frame b. The probability distributions are computed by projecting $\boldsymbol{h}$ with a fully connected layer followed by a softmax. The choice of pseudo-labels is flexible, including K-means clusters of acoustic features and codewords produced by Spin. With this loss, the fine-tuned models are expected to preserve content even when all layers are fine-tuned. Combining Equation 2.2.5 and 3.1, the overall loss function is

$$\mathcal{L} = \mathcal{L}_{\text{Spin}} + \lambda \mathcal{L}_{\text{Aux}}, \tag{3.2}$$

where $\lambda > 0$ is a hyper-parameter. $\mathcal{L}_{\text{Aux}}$ has learning targets independent of the model, regularizing and stabilizing the training process. Meanwhile, $\mathcal{L}_{\text{Spin}}$ optimizes on varying labels from a codebook, offering flexibility to improve upon the pseudo-labels in $\mathcal{L}_{\text{Aux}}$. Therefore, the combined loss function is expected to enhance pre-trained speech SSL encoders and mitigate Spin's limitations.

3.2.4 Acoustic Pieces

This section introduces acoustic pieces (APs) (Ren et al., 2022) to $\mathcal{L}_{\text{Aux}}$ to further improve R-Spin. APs are learned by applying byte-pair encoding (BPE) (Sennrich et al., 2016) to discrete acoustic units like K-means clusters of HuBERT representations. APs capture high-level units close to phonemes and characters, useful for pre-training (Wu et al., 2023) and

generation (Shen et al., 2023). Hence, we propose to set APs as the target of $\mathcal{L}_{\text{Aux}}$ to extract better content representations.

Following (Ren et al., 2022), we first merge identical consecutive units in time for each utterance. The BPE algorithm is then applied to the reduced sequences to learn acoustic pieces. Next, we encode the entire training corpus into APs and duplicate the encoded units to the original utterance length. The encoded corpus is then used as the pseudo-labels for Equation 3.1, which is expected to encourage the fine-tuned SSL model to encode better phoneme and character representations.

3.3 Experiments

3.3.1 Data

The 960 hours of unlabeled English speech in LibriSpeech is used for R-Spin training (Panayotov et al., 2015). Audio distortions are generated with torch-audiomentations.[1] Following Robust data2vec (Zhu et al., 2023), background noises are sampled from MUSAN (Snyder et al., 2015) and CHiME-4 (Vincent et al., 2017) corpora, covering music, speech, and outdoor noise. The signal-to-noise (SNR) ratios are uniformly sampled from $[-10, 10]$ during training. We add distortions to each utterance during evaluation, including Gaussian noise, MUSAN noise, and reverberation. The noise and perturbation data sources are listed as follows.

1. **Gaussian Noise**: Gaussian noise is generated with a PyTorch Library.
2. **Background Noise**: Background noise is sampled from the MUSAN dataset.
3. **Reverberation**: We filter waveforms with real and simulated room impulse responses in the RIRS (Ko et al., 2017). The scores for the real and simulated reverberation are averaged.

[1] https://github.com/asteroid-team/torch-audiomentations

3.3.2 Implementation

The experiments are mostly based on WavLM (Chen et al., 2022) because WavLM is pre-trained with a denoising objective, offering a good initialization. HuBERT (Hsu et al., 2021a) is also considered to demonstrate R-Spin's generalizability to SSL models trained with clean speech. The acoustic pieces are generated by learning BPE tokens on top of a HuBERT + Spin_{2048} model.

Speech SSL Models

HuBERT-MGR (Huang et al., 2022a) continues the HuBERT pre-training process with noisy speech and an auxiliary domain adversarial training objective to enhance robustness. HuBERT-MGR is trained with a mix of clean and distorted speech, where the distortions include MUSAN background noise, Gaussian noise, and reverberation. **Robust data2vec** (Zhu et al., 2023) fine-tunes a pre-trained data2vec model. Unlike data2vec, the inputs to the student model include background noise so that the model learns denoising. An additional contrastive learning objective is incorporated to enhance robustness. The pre-trained model weights are obtained from the s3prl toolkit.[2]

Spin Training

Since R-Spin is trained with 960 hours of data speech in LibriSpeech, the pseudo labels for $\mathcal{L}_{\text{Aux}}$ should be generated for all those data with Spin. To avoid generating unseen data with Spin, we train another Spin_{2048} model with the same data (originally 100 hours Section 2.3.2). Each mini-batch before data perturbation has 2,560 seconds of speech, equivalent to 32k frames. The learning rate first linearly increases from 10^{-6} to 10^{-4} in the first 2.5k updates, then linearly decreases to 10^{-6} in the last 7.5k updates. The implementation of the Spin loss follows (Caron et al., 2020).[3] This model takes four hours of training time on four RTX

[2]https://github.com/s3prl/s3prl/tree/main/s3prl/upstream
[3]https://github.com/facebookresearch/swav

A6000 GPUs. Models trained with all 10k updates are used for generating pseudo labels. In total, roughly 7.1k hours of unlabeled speech data are processed.

R-Spin Training

Each SSL model used in this paper has a 7-layer CNN feature extractor and a 12-layer transformer encoder, having roughly 95M parameters in total. Each mini-batch before data perturbation has 384 seconds of speech, equivalent to 19.2k frames in each view of the R-Spin framework. The learning rate first linearly increases from 10^{-6} to 10^{-4} in the first 5k updates, then linearly decreases to 10^{-6} in the last 5k updates. λ in Equation 3.2 is set to 5. Each R-Spin SSFT training takes less than eight hours on two RTX A6000 GPUs. Models trained with all 10k updates are used for evaluation. For the R-Spin training, 1.1k hours of unlabeled speech data are processed. Combined with the Spin training, 8.2k hours of data are used.

Low-budget Robust data2vec

We follow the implementation of (Zhu et al., 2023) with fairseq (Ott et al., 2019).[4] We changed the unlabeled training data from CHiME-4 to the LibriSpeech 960 hours corpus for a fair comparison with our method. Because we observed that a long training schedule is necessary for Robust data2vec converge, the number of updates is the same as the original implementation (100k). Meanwhile, the mini-batch size is reduced from 63 to 6.25 minutes so that the amount of speech data processed is the same as R-Spin. The rest of the hyperparameters remain the same since we found the original ones are sufficiently good. As shown in Table 3.1, the low-budget Robust data2vec model has a significant performance degradation compared with the fully-trained version, implying the necessity to train this model with a large batch size. In contrast, R-Spin achieves superior results under the same budget, indicating that our approach is more efficient.

[4] https://github.com/zqs01/data2vecnoisy

Table 3.1: Phoneme recognition on LibriSpeech and ASR on CHiME-4 test sets. Gaussian noise, MUSAN background noise, and reverberation (Reverb) are respectively added to simulate noisy conditions, where the SNRs are fixed to 0dB. The calculation of the number of hours of processed speech during SSFT follows Eq. 2.2.

Method	SSFT Processed Speech (hours)	LibriSpeech test-other Phoneme Recognition (PER↓)				CHiME-4 ASR (WER↓)	
		Clean	Gaussian†	MUSAN	Reverb†	Real	Sim
No SSFT Baselines							
HuBERT (Hsu et al., 2021a)	0	10.7	74.5	50.2	23.2	72.7	63.1
WavLM (Chen et al., 2022)	0	10.3	59.9	45.1	19.4	52.4	46.4
SSFT Baselines							
HuBERT + Spin_{2048} (Chang et al., 2023a)	0.4k	8.4	70.8	47.8	18.4	71.3	62.0
WavLM + Spin_{2048} (Chang et al., 2023a)	0.4k	**8.2**	59.2	41.2	16.7	52.1	46.6
Robust data2vec (Low-budget)	10.4k	38.8	68.2	52.9	53.7	80.9	78.2
Proposed							
HuBERT + $\text{R-Spin}_{32,\ \text{AP40k}}$	8.2k	8.3	36.4	18.2	16.3	34.3	34.1
WavLM + $\text{R-Spin}_{32,\ \text{AP40k}}$	8.2k	**8.2**	**33.7**	**16.7**	**14.9**	**26.4**	**26.6**
High-budget SSFT Toplines							
ContentVec_{500} (Qian et al., 2022)	76k	8.7	71.4	47.2	16.8	61.4	55.1
HuBERT-MGR (Huang et al., 2022a)	78k	9.5	37.1	36.3	18.3	49.7	44.3
Robust data2vec (Zhu et al., 2023)	105k	6.5	56.7	27.7	19.2	17.5	20.1
Supervised Toplines							
Whisper Base (Radford et al., 2022)	–	–	–	–	–	17.9	23.3
Whisper Small (Radford et al., 2022)	–	–	–	–	–	10.8	14.3

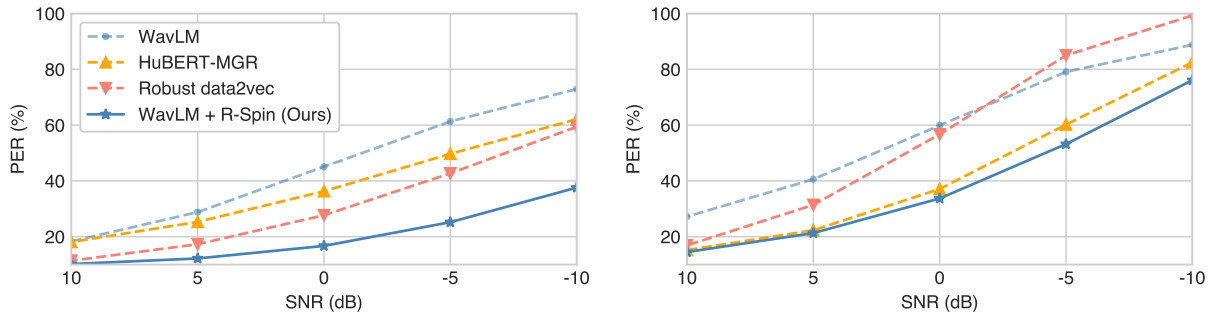
†Unseen perturbation types for R-Spin and Robust data2vec.

Notations

We denote X + Spin_K as an SSL model X fine-tuned with Spin with a codebook size of K. In X + $\text{R-Spin}_{K_1, K_2}$, K_1 and K_2 are respectively the codebook size of $\mathcal{L}_{\text{Spin}}$ and the number of classes of pseudo labels for $\mathcal{L}_{\text{Aux}}$. If the pseudo labels are acoustic pieces, "AP" is added to K_2. Unless specified otherwise R-Spin denotes $\text{R-Spin}_{32,\ \text{AP40k}}$.

3.3.3 Noisy Phoneme Recognition

We compare the phoneme recognition performance of SSL and SSFT methods under noisy conditions. The training setup is similar to the SUPERB phoneme recognition task (Yang et al., 2021), where the SSL models are frozen and only a lightweight prediction head is fine-



(a) MUSAN Noise

(b) Gaussian Noise

Figure 3.2: Phoneme error rates (PER) under different noise types and SNRs. R-Spin$_{32,\ \text{AP40k}}$ is used here.

tuned.[5] The LibriSpeech train-clean-100 and the test-other subsets are used as the training and evaluation datasets, respectively. We apply distortions only to testing data to obtain phoneme error rates (PER). We divide results into two categories by the amount of speech processed during SSFT, directly related to the resources used.

As shown in the middle columns of Table 3.1, R-Spin improves SSL models in all conditions, surpassing both low and high-budget methods. WavLM + R-Spin has the best overall PERs because WavLM is pre-trained with a denoising task, showing that model initialization contributes largely to the recognition performance after SSFT. Gaussian noise and reverberation conditions are unseen during R-Spin training. Still, the proposed method improves performance on these tasks, indicating that noise-invariant training generalizes to some out-of-domain perturbations. Furthermore, comparing Robust data2vec with R-Spin is unfair since the training costs are 12 times greater. Hence, we reduce the batch size to train a low-budget version of the Robust data2vec. The noticeable performance drop in the low-budget model implies Robust data2vec requires high computation resources, but our approach still offers competitive results with even fewer training data.

For a more detailed comparison, we plot PERs under different SNRs in Figure 3.2. Note that these models are trained with MUSAN noise except for WavLM, and only HuBERT-MGR uses Gaussian noise. The proposed method achieves the overall lowest PERs. HuBERT-

[5]https://github.com/s3prl/s3prl

MGR excels in Gaussian noise, but R-Spin slightly outperforms this model even though this type of noise is unseen. Still, our method offers improvements even when the SNR is high. Overall, the proposed R-Spin improves capturing content under severe distortions with minimal effort.

3.3.4 Noisy Speech Recognition

This section evaluates R-Spin with a noisy ASR task. We follow the ASR task of SUPERB, but the prediction heads (two-layer BLSTM) are trained with the clean portion of the CHiME-4 speech corpus obtained from the WSJ0 corpus (Paul and Baker, 1992), consisting of 14 hours of clean English speech. The number of training updates is 100k (originally 200k). The trained ASR models are evaluated on the 1-channel track of the CHiME-4 challenge. We report the averaged WERs over each subset (real and simulated data). We apply Whisper normalization to all ASR results for a fair comparison with the Whisper toplines.[6]

The results in the right columns of Table 3.1 reveal that R-Spin surpasses low-budget baseline models. While R-Spin demonstrates commendable performance on CHiME-4, this method falls short compared to Robust data2vec, which benefits from training with a substantially higher budget. Furthermore, we set Whisper Base and Small as toplines due to their robustness demonstrated through large-scale weakly-supervised learning (Radford et al., 2022). R-Spin successfully mitigates the performance gap between WavLM and the Whisper toplines by over 60%. Combining phoneme and speech recognition findings, we conclude that R-Spin effectively enhances pre-trained SSL models in capturing robust content representations.

3.3.5 Data-efficiency

The objective behind developing R-Spin is to enhance speech SSL models with minimal resources, including reducing training data to improve data efficiency. Following Table 2.1,

[6]https://github.com/openai/whisper

Table 3.2: SSL and SSFT costs of models with 95M parameters. The "Init" column shows the pre-trained models used for initialization. △ denotes models in this paper, which will be made publicly available in the near future. Note that the duplicated input utterances by data augmentation are not included when calculating the hours of speech processed. The number of GPU hours required for training is roughly estimated so that the true values might differ slightly. The availability of the models listed is updated in November 2023. Unknown data are left blank.

Model	Init	Updates	Batch Size (minutes)	Processed Speech (hours)	#GPUs	GPU Hours	Open Model
SSL (Clean Speech)							
wav2vec 2.0 (Baevski et al., 2020)	–	400k	96	640k	64	2458	✓
HuBERT (Hsu et al., 2021a)	–	250k + 400k	47	505k	32	1976	✓
WavLM (Chen et al., 2022)	–	250k + 400k	187	1439k	32		✓
data2vec (Baevski et al., 2022)	–	400k	63	420k	16		✓
DinoSR (Liu et al., 2023)	–	400k	63	420k	16	2880	✗
SSL (Noisy Speech)							
wav2vec-Switch (Wang et al., 2022)	–	400k	96	640k	32		✗
SPIRAL (Huang et al., 2022b)	–	200k	100	333k	16	499	✓
SSFT							
ContentVec (Qian et al., 2022)	HuBERT	100k	46	76k	36	684	✓
HuBERT-MGR (Huang et al., 2022a)	HuBERT	400k	12	78k	8	768	✓
Robust data2vec (Zhu et al., 2023)	data2vec	100k	63	105k	16		✓
deHuBERT (Ng et al., 2023)	HuBERT	250k					✗
Spin_{2048} (Chang et al., 2023a)	HuBERT	5k	43	0.4k	1	1	✓
This Paper							
Robust data2vec (low budget)	data2vec	100k	6.3	10.4k	2	44	△
Spin_{2048} (for AP40k)	HuBERT	10k	43	7.1k	2	8	△
$\text{R-Spin}_{32,\ \text{AP40k}}$	HuBERT	10k	6.4	1.1k	2	16	△

an analysis of the duration of speech data processed during training is undertaken to quantify the computational expenses associated with each method. As depicted in the second column of Table 3.1, these values are derived by multiplying the number of training updates and the effective batch size for each update. Compared with the high-budget SSFT methods, R-Spin requires significantly lower training costs, concurrently exhibiting superior performance across diverse conditions. The costs of self-supervised pre-training and fine-tuning of various models are shown in Table 3.2.

Table 3.3: CHiME-4 ASR results for ablation studies based on fine-tuned WavLM models.

Method	CHiME-4 Real	CHiME-4 Sim
Spin_{2048}	52.1	46.6
$\text{R-Spin}_{32,\ \text{AP40k}}$	26.4	26.6
no $\mathcal{L}_{\text{Aux}}$	47.8	45.6
no $\mathcal{L}_{\text{Spin}}$	31.9	32.4
no speaker perturbation	28.3	28.0
no additive noise	49.4	46.8
Pseudo Label for $\mathcal{L}_{\text{Aux}}$		
Spin_{2048} codebook♠	28.3	29.1
MFCC (K-means 512)	46.9	45.4
MFCC (K-means 2048)	48.5	45.5
HuBERT L9 (K-means 512)♠	28.8	29.1
HuBERT L9 (K-means 2048)♠	28.2	28.4
Hyperparameters		
$\lambda = 1$	26.3	27.7
$\lambda = 0.5$	26.6	27.3
Layer to Apply $\mathcal{L}_{\text{Aux}}$		
Layer 11	28.1	28.8
Layer 10	34.7	33.8
Layer to Apply $\mathcal{L}_{\text{Spin}}$		
Layer 11	27.2	27.9
Layer 10	27.0	27.8
Fine-tuned Layers		
Top 10 Layers	29.7	30.0
Top 6 Layers	39.4	37.5
Dataset		
LibriSpeech 100h	27.2	27.6
LibriSpeech 360h	26.6	27.6
Noise SNR Range		
0 – 20dB	29.0	28.6

♠Pairwise t-tests between these results all have $p > 0.05$. Also, $p < 0.05$ when they are compared with $\text{R-Spin}_{32,\ \text{AP40k}}$.

3.3.6 Ablation Studies

Under the same CHiME-4 ASR setup in Section 3.3.4, we conduct ablation studies to analyze the design of the proposed methods.

As shown in Table 3.3, by removing the proposed auxiliary loss (no $\mathcal{L}_{\text{Aux}}$), the WERs increase significantly, showing that $\mathcal{L}_{\text{Aux}}$ not only helps ASR performance but allows fine-tuning the entire model without collapsing. Second, WERs increase by about 5% without the Spin loss (no $\mathcal{L}_{\text{Spin}}$), implying that this loss is essential for achieving perturbation-invariant representations. Speaker perturbation also plays an important role in offering good content representations according to the degraded WERs (no speaker perturbation). Moreover, the fine-tuned model performs poorly without the additive noise during training, demonstrating the loss of robustness without the noise-invariant training (no additive noise). The above results verified the necessity of the proposed approaches.

Pseudo Label for $\mathcal{L}_{\text{Aux}}$

We investigate the effect of choosing different pseudo labels for $\mathcal{L}_{\text{Aux}}$. First, acoustic pieces are essential to the R-Spin training since learning from the original Spin model's 2048 codeword labels increases WERs by over 2%. Next, we replace the pseudo labels with the more commonly used K-means clustered representations (Hsu et al., 2021a). Clustered MFCC features degrade R-Spin the most, no matter the number of clusters used, corroborating the findings by Hsu et al. (2021a). In contrast, clustered HuBERT representations from layer 9 (L9) have similar results compared with Spin_{2048}, and t-test results imply the differences between applying these pseudo labels are statistically insignificant. This suggests that using clustered discrete units from a speech SSL model is an alternative solution if a pre-trained Spin model is unavailable.

Hyperparameter

To examine the impact of the auxiliary loss, we change the value of λ in Equation 3.2. As shown in the bottom left part of Table 3.3, the differences of ASR WERs between different λ's are negligible. We can conclude that combining $\mathcal{L}_{\text{Spin}}$ and $\mathcal{L}_{\text{Aux}}$ is necessary, and the ratio between the two objectives is robust.

Layer to Apply $\mathcal{L}_{\text{Aux}}$

In the R-Spin design, $\mathcal{L}_{\text{Aux}}$ is applied to the last layer. We next apply $\mathcal{L}_{\text{Aux}}$ to other hidden layers to verify that our approach leads to the best overall result. When we move the auxiliary loss $\mathcal{L}_{\text{Aux}}$ to lower layers, the performance degrades significantly, showing that this loss should regularize the entire model. Otherwise, the Spin loss still makes the representations collapse.

Layer to Apply $\mathcal{L}_{\text{Spin}}$

Similar to the previous experiments, we apply $\mathcal{L}_{\text{Spin}}$ to lower layers to find the optimal design. When we move the Spin objective function to lower layers, the ASR performance

also degrades slightly. With the results of $\mathcal{L}_{\text{Aux}}$, we conclude that a relatively good strategy for applying the two proposed loss functions is adding both to the top layer.

Fine-tuned Layers

R-Spin is designed to fine-tune SSL models entirely, but Spin allows fine-tuning the top two layers. Hence, we reduce the number of fine-tuned layers to compare R-Spin with Spin. The results indicate that by fine-tuning only the top layers, the model cannot adapt to noisy scenarios. Thus, R-Spin is beneficial since we can now fine-tune the entire model in contrast to Spin.

Data

We further change the data for R-Spin SSFT to reveal the impact of training corpora on the performance. We found that WERs degrade slightly (LibriSpeech 960 $\rightarrow$ 360 $\rightarrow$ 100 hours) when the training corpus size is reduced. The ASR performance degrades prominently by increasing the SNRs of the background noise for the noise-invariant training. Hence, the choice of noise data and SNRs has a greater impact on the downstream performance than the choice of the clean speech corpus.

3.4 Chapter Summary

This chapter introduces R-Spin, a self-supervised fine-tuning method with speaker and noise-invariant clustering for robust content representations. Results demonstrate R-Spin's effectiveness and generalizability to diverse acoustic scenarios under limited computation budgets. The ablation studies support the necessity of the R-spin design.

Chapter 4

Representation Analysis

This chapter comprehensively analyzes the learned representations and discrete acoustic units in the proposed Spin and R-Spin frameworks to offer insights and further understand the properties of SSL models.

We first provide the background of prior methods for analyzing speech SSL models in Section 4.1. Second, we inspect the speaker identification capabilities of different models to reveal their speaker invariability. Third, we visualize the hidden representations under various perturbation types in Section 4.3. Fourth, we explore the quality of continuous and discrete representations respectively in Sections 4.4 and Section 4.5. Finally, we discuss the phoneme segmentation capabilities of learned discrete units in Section 4.6.

4.1 Background

Although existing speech SSL models perform well on various downstream tasks, fine-tuning the models or learning additional parameters requires hyperparameter tuning, thereby introducing more uncertainties to the evaluation of these models. Moreover, many previous studies focus on analyzing and utilizing the continuous representations of SSL models (Pasad et al., 2021), but recent works have shown promising results with discrete acoustic units (Chang et al., 2023b). Due to the above two issues, the true capabilities of SSL models

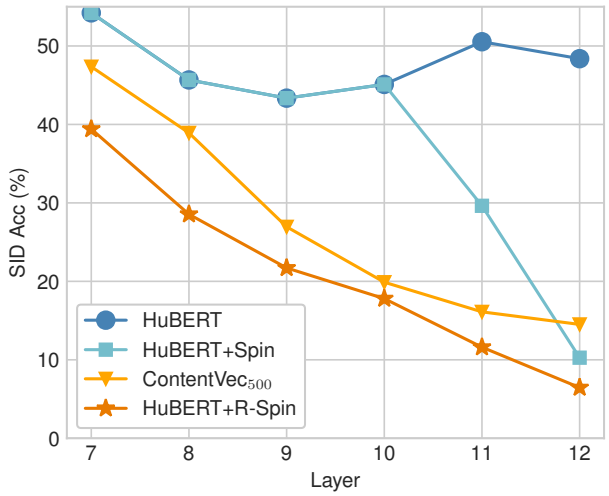


Figure 4.1: Layer-wise speaker identification accuracy.

remain unknown. Thus, we dive into the analysis of both continuous and discrete features in SSL models with numerous techniques.

Pasad et al. (2021) have conducted an extensive study to understand the behavior of wav2vec 2.0. They utilize the Canonical Correlation Analysis (CCA) and mutual information to compute the similarity between continuous speech representations and linguistic units like phonemes and words. Sicherman and Adi analyze the discrete units obtained by SSL models to understand the properties of these units (Sicherman and Adi, 2023). Chang et al. (2023b) report a comprehensive study of the usefulness of discrete speech units via the application of these units to various downstream tasks.

In this chapter, we extensively examine the characteristics of speech SSL representations with several metrics, tasks, and visualization approaches, aiming to reveal the perturbation-invariability and proximity to linguistic units.

4.2 Speaker Identification

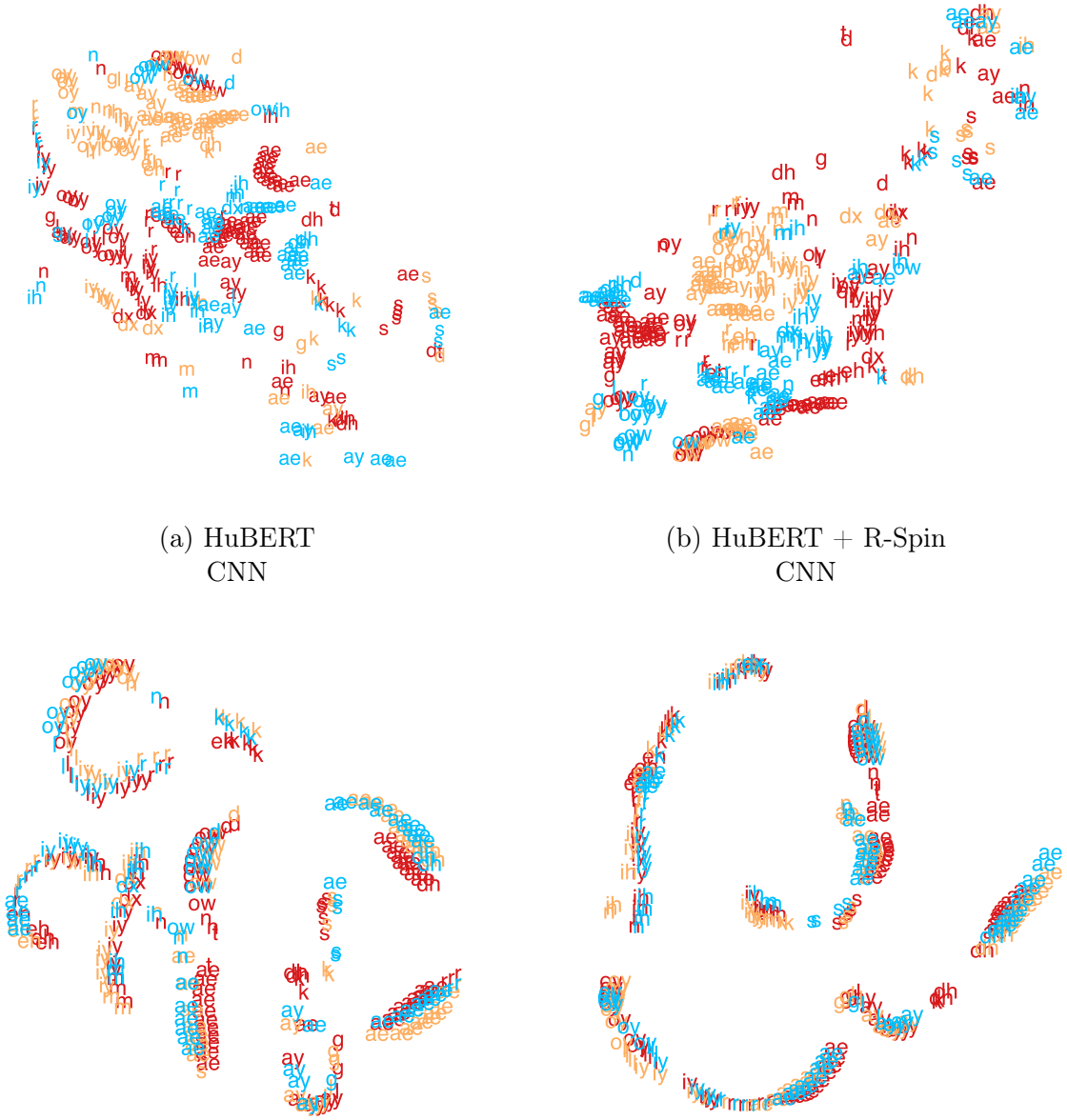
This section inspects each model's invariability to speaker changes by computing the speaker identification (SID) accuracy with different hidden layer representations. The SID task follows SUPERB's setup but with 50k training updates. As shown in Figure 4.1, R-Spin has

significantly lower SID accuracy for the top layers, demonstrating the effect of fine-tuning the whole model with a speaker-invariant objective. Moreover, requiring 9X less training costs, our method produces representations with less speaker information than ContentVec. Therefore, the proposed method outperforms prior speaker-invariant training approaches in removing speaker ID.

4.3 Visualization of Hidden Representations

4.3.1 Speaker Invariability

This section explores the robustness of SSL models regarding representation invariability by examining the characteristics of representations under diverse perturbations. The visualization of these representations articulated by distinct speakers is facilitated by applying t-SNE (Van der Maaten and Hinton, 2008). We show the layer with the lowest speaker identification (SID) rate according to Figure 4.1. In Figures 4.2a and 4.2b, there is a discernible clustering of frames uttered by the same speaker, suggesting that lower layers retain more speaker-specific information. Conversely, Figures 4.2c and 4.2d illustrate that top layer features are grouped according to phonemes rather than speakers. Moreover, the top layer representations are context-dependent, as exemplified by the spatial arrangement of phonemes such as "carry" (`/k/ /eh/ /r/ /iy/`) and the same phoneme `/iy/` in the word "oily" (`/oy/ /l/ /iy/`) in Figure 4.2d. Besides, a comparative analysis between Figures 4.2c and 4.2d reveals that R-Spin features exhibit a more prominent overlap among speakers than HuBERT. As a result, this section substantiates the speaker-invariability of the proposed approach. Detailed visualizations are shown in Figures 4.3 and 4.4.



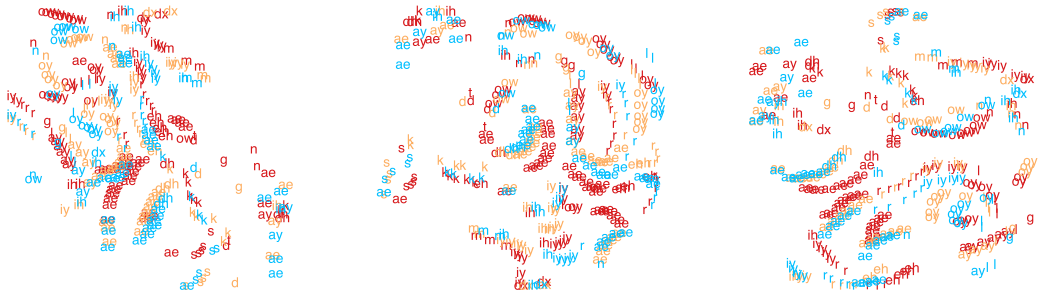
(a) HuBERT CNN

(b) HuBERT + R-Spin CNN

(c) HuBERT Layer 9

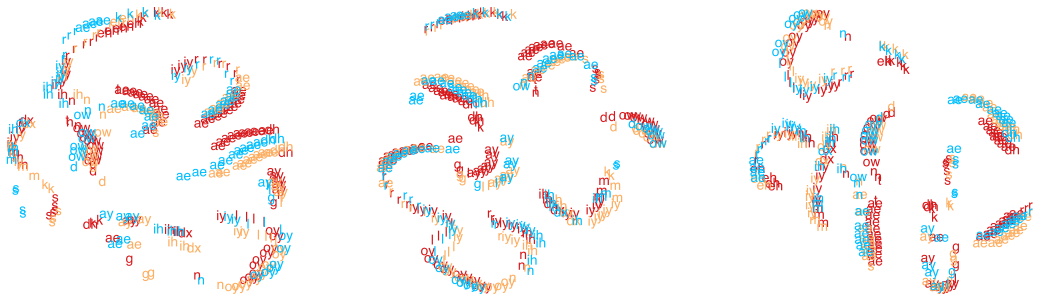
(d) HuBERT + R-Spin Layer 12

Figure 4.2: t-SNE Van der Maaten and Hinton (2008) visualization of the CNN and the layer with the lowest speaker identification rate given the same clean utterance spoken by three different speakers from TIMIT Garofolo (1993). Each color represents a speaker, while each label visualizes a frame representation and the corresponding phoneme label. The transcription is "Don't ask me to carry an oily rag like that." The silence frames are omitted for clarity.

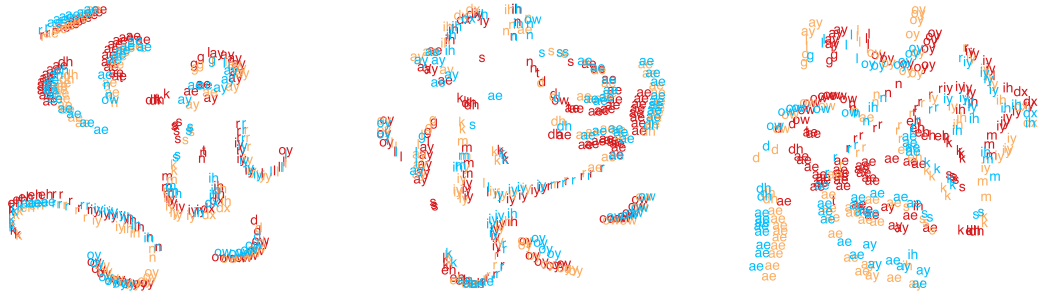


(a) Layer 1 (b) Layer 2 (c) Layer 3

(d) Layer 4 (e) Layer 5 (f) Layer 6

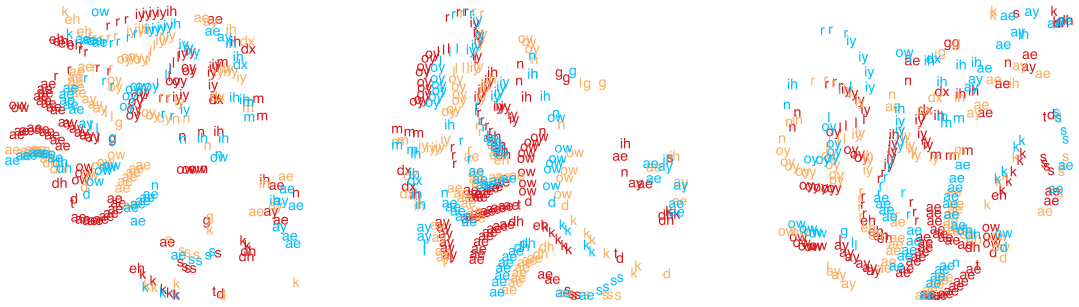


(g) Layer 7 (h) Layer 8 (i) Layer 9



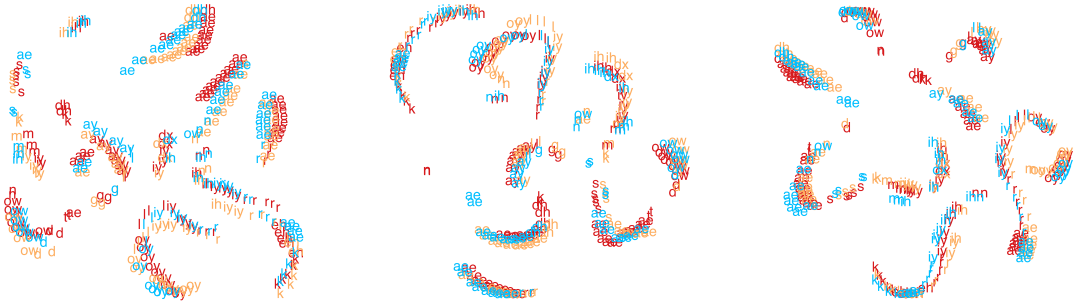
(j) Layer 10 (k) Layer 11 (l) Layer 12

Figure 4.3: t-SNE visualization of HuBERT representations of the same utterance spoken by three speakers (see Fig. 4.2 for details).

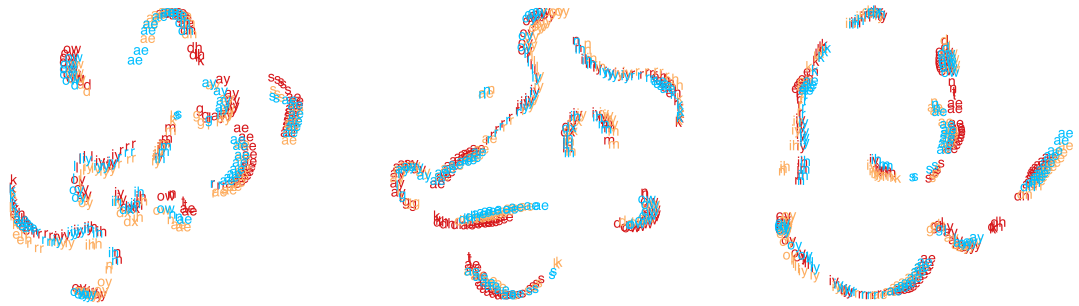


(a) Layer 1 (b) Layer 2 (c) Layer 3

(d) Layer 4 (e) Layer 5 (f) Layer 6



(g) Layer 7 (h) Layer 8 (i) Layer 9



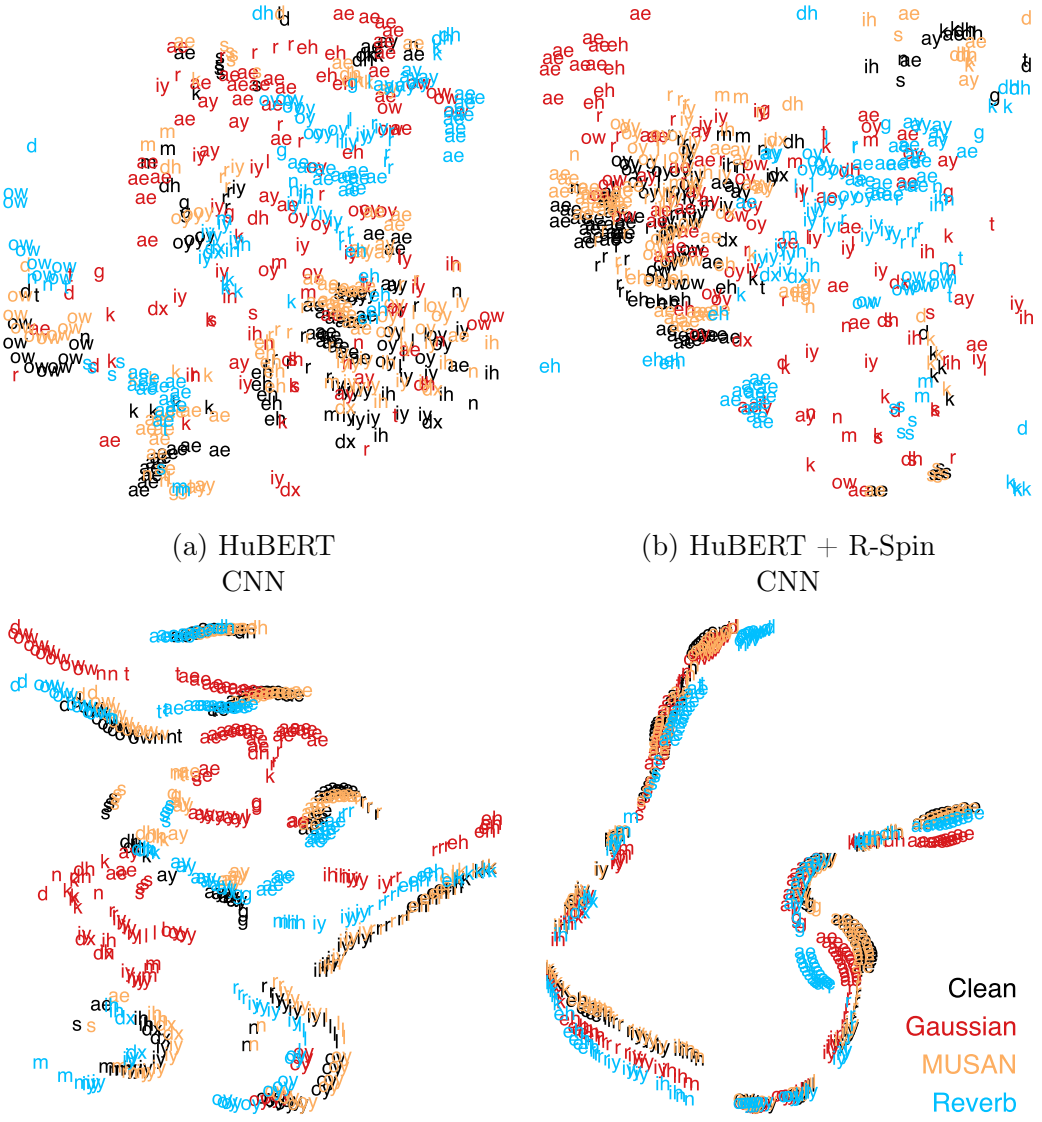
(j) Layer 10 (k) Layer 11 (l) Layer 12

Figure 4.4: t-SNE visualization of HuBERT + R-Spin representations of the same utterance spoken by three speakers (see Fig. 4.2 for details).

4.3.2 Noise Invariability

In this section, we examine the response of continuous representations to input distortions. As in Section 4.3.1, we employ t-SNE visualization to explore hidden representations under different distortions. As shown in Figure 4.5, the R-Spin representations exhibit a more pronounced overlap than those subjected to HuBERT, suggesting that R-Spin demonstrates greater robustness to noise. Figure 4.5d reveals that features exposed to MUSAN background noise exhibit a high degree of overlap with unperturbed features, whereas the other two perturbation types diverge more significantly from clean speech features. This divergence is attributed to Gaussian noise and reverberation being unseen during R-Spin training. Nevertheless, HuBERT + R-Spin yields similar representations under various distortions, resulting in closely located visualized frames.

Subsequently, we compute linear centered kernel alignment (CKA) similarities (Kornblith et al., 2019) of frame-wise features with and without noisy inputs, where a higher similarity indicates a higher invariability to distortions. The evaluation involves the construction of datasets derived from the LibriSpeech dev-clean and dev-other sets, augmented with various distortions. Figure 4.6 illustrates that R-Spin exhibits superior noise invariability for the upper layers than other models, indicating the efficacy of noise-invariant training even if the noise types are unseen. In contrast, Robust data2vec has a greater noise invariability starting from the bottom layers. Lower layers tend to demonstrate lower similarities, suggesting a higher sensitivity to perturbations. This observation aligns with existing research discussed in Section 3.2.3, which associates lower layers with fundamental signal processing functions. Overall, the analysis underscores the notable noise invariability offered by R-Spin. Detailed visualizations are shown in Figures 4.7 and 4.8.



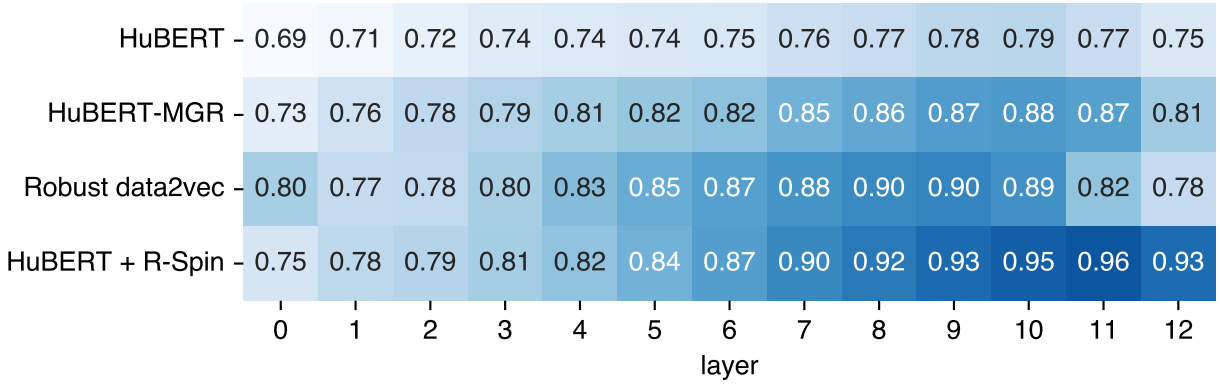
(a) HuBERT CNN

(b) HuBERT + R-Spin CNN

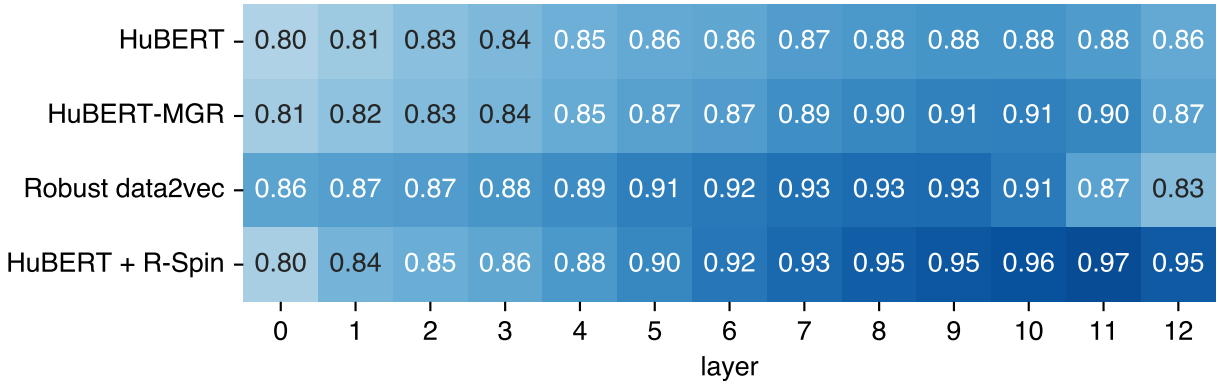
(c) HuBERT Layer 9

(d) HuBERT + R-Spin Layer 12

Figure 4.5: t-SNE visualization of hidden representations of the same audio utterance in Fig. 4.2 with different distortions indicated by colors, where SNR = 0dB.

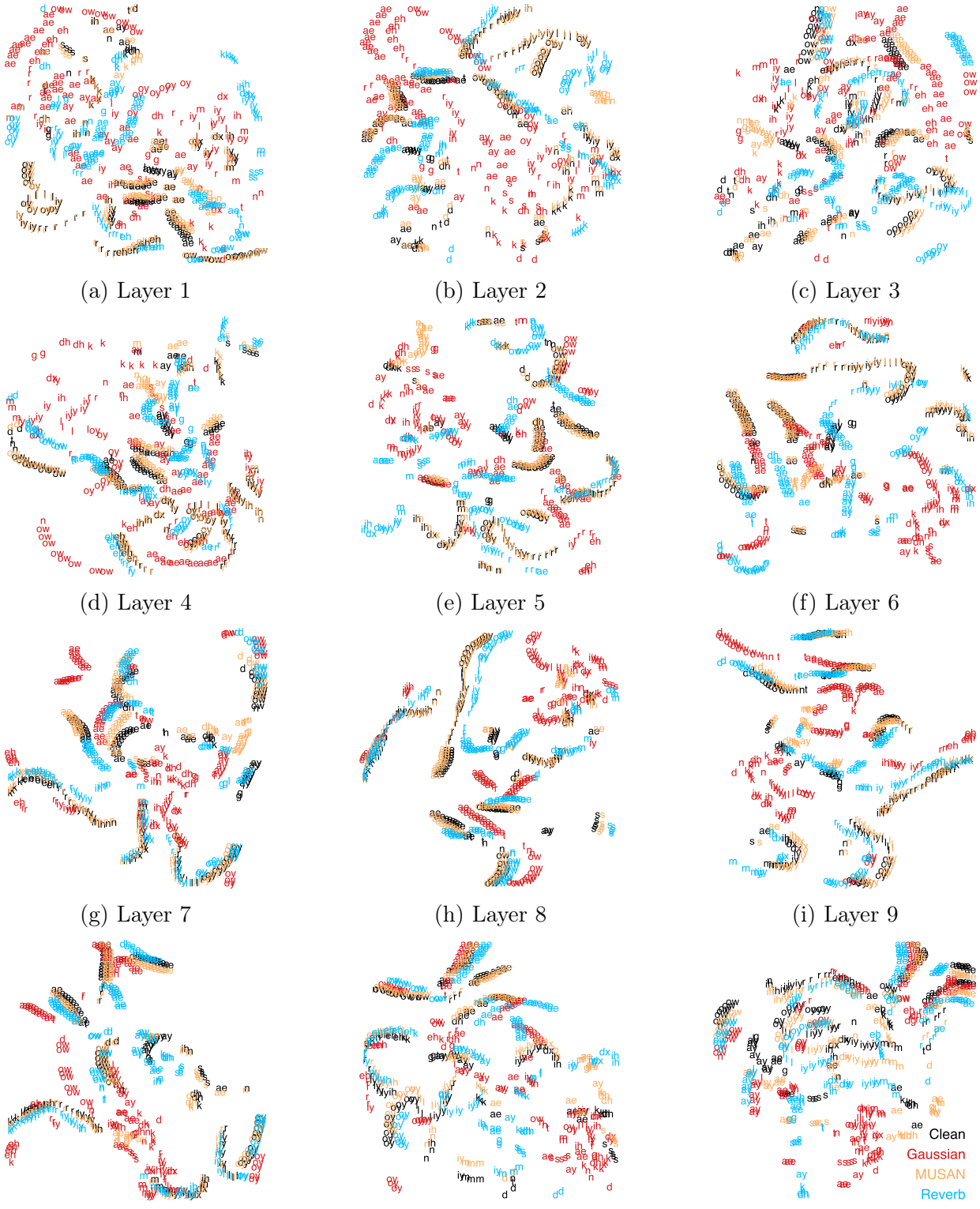


(a) Gaussian Noise (SNR = 0dB)



(b) Reverberation (real room impulse response)

Figure 4.6: Layer-wise perturbation invariability analyses with Linear CKA, where higher values indicate higher invariability to perturbations. The zeroth layer denotes the CNN feature extractor.



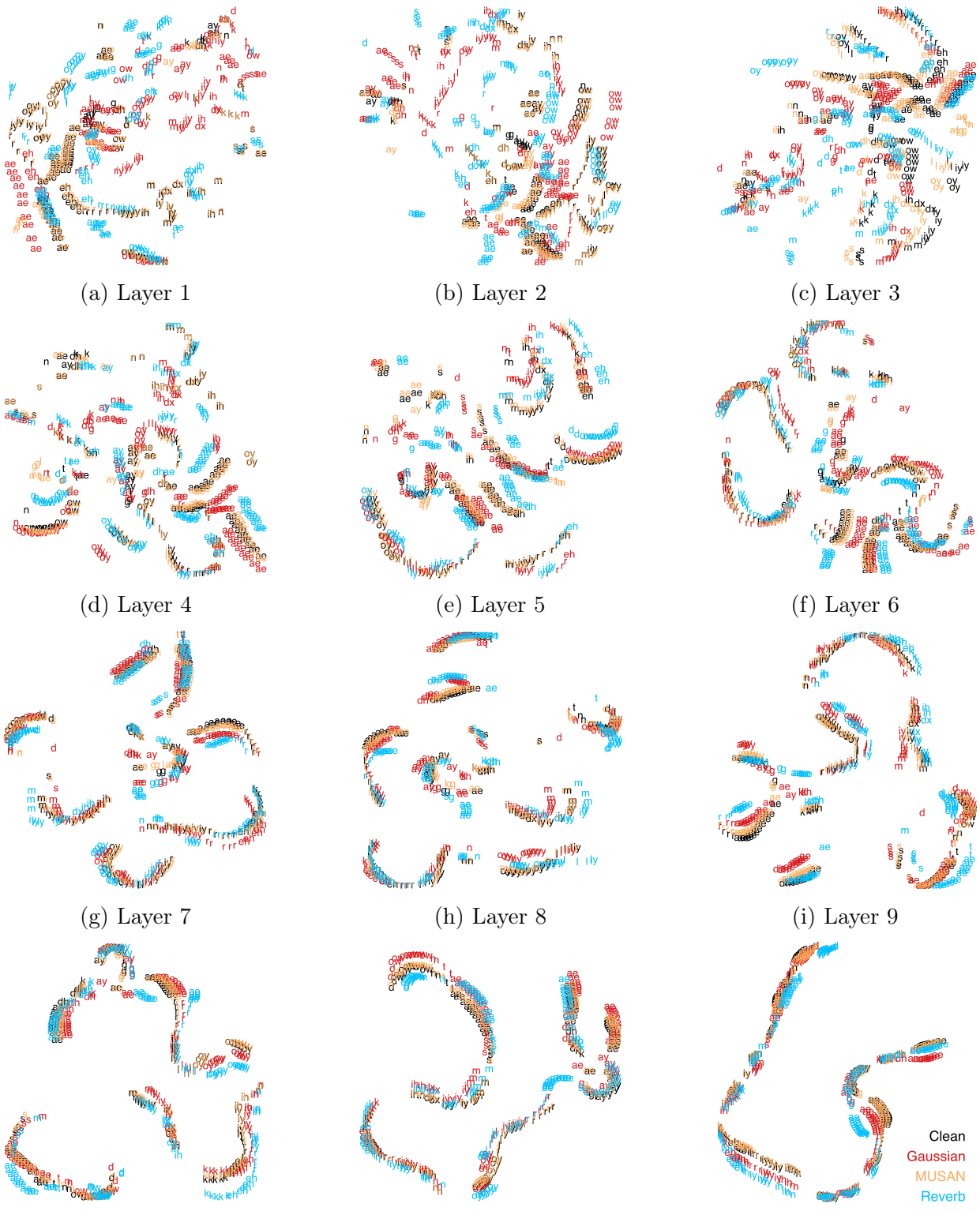
(a) Layer 1 (b) Layer 2 (c) Layer 3

(d) Layer 4 (e) Layer 5 (f) Layer 6

(g) Layer 7 (h) Layer 8 (i) Layer 9

(j) Layer 10 (k) Layer 11 (l) Layer 12

Figure 4.7: t-SNE visualization of HuBERT representations of the same utterance under different distortions (see Fig. 4.5 for details).



(a) Layer 1 (b) Layer 2 (c) Layer 3

(d) Layer 4 (e) Layer 5 (f) Layer 6

(g) Layer 7 (h) Layer 8 (i) Layer 9

(j) Layer 10 (k) Layer 11 (l) Layer 12

Figure 4.8: t-SNE visualization of HuBERT + R-Spin representations of the same utterance under different distortions (see Fig. 4.5 for details).

Table 4.1: ABX error rates (%) on the ZeroSpeech 2021 phonetic dev set (Nguyen et al., 2020). Within and Cross denote within and across speakers. Clean and Other denote clean and other corpus partitions. Only the layer with the lowest average score is reported for each model and is specified in column "Layer".

		Within Speaker		Across Speaker		
Method	Layer	Clean	Other	Clean	Other	Avg
Nguyen et al. (Nguyen et al., 2022)	–	3.26	3.81	4.00	5.91	4.25
Chorowski et al. (Chorowski et al., 2021)	–	2.95	3.54	4.50	7.05	4.51
HuBERT	11	3.07	3.90	3.71	6.19	4.22
WavLM	11	2.73	3.41	3.21	4.95	3.58
data2vec	4	4.03	5.09	4.72	6.97	5.20
DinoSR	5	3.08	3.43	3.42	4.42	3.59
ContentVec_{100}	12	2.98	3.70	3.44	5.17	3.82
ContentVec_{500}	12	3.91	4.37	4.46	5.80	4.64
HuBERT + Spin_{2048}	12	**2.44**	**3.00**	**2.81**	**3.76**	**3.00**
WavLM + Spin_{2048}	12	2.75	3.33	3.24	4.17	3.37
HuBERT-MGR	11	3.38	3.81	3.96	5.48	4.16
Robust data2vec	4	4.18	5.12	4.92	7.24	5.37
HuBERT + $\text{R-Spin}_{32,\ \text{AP40k}}$	12	3.56	3.94	3.95	4.92	4.09
WavLM + $\text{R-Spin}_{32,\ \text{AP40k}}$	12	3.58	3.71	3.87	4.71	3.97
WavLM + $\text{R-Spin}_{2048,\ \text{AP40k}}$	12	3.34	3.53	3.64	4.57	3.77

4.4 Acoustic Unit Discovery

This section inspects linguistic units captured in representations with Zero Resource Speech Benchmark (ZeroSpeech) 2021 (Nguyen et al., 2020). The phonetic task measures how well speech representations distinguish between different phonemes via the ABX discrimination test (Schatz, 2016). We report $K = 2048$ for Spin since it performs the best in this task. As shown in Table 4.1, Spin boosts both models and surpasses the baselines, especially for HuBERT, surpassing prior art and reducing the average ABX error rate by a relative 29%. Although the performance gain for WavLM is minor, error rates of other corpus partitions are reduced, indicating that Spin helps WavLM in a noisier scenario. The results directly demonstrate that Spin improves extracting phonemes. Moreover, we observed that R-Spin does not offer significant improvements in this task because R-Spin models are aimed to

handle more complex audio types, thereby losing some capabilities for clean speech. Still, R-Spin offers better ABX scores compared with other noise-robust SSFT approaches.

4.5 Discrete Unit Quality

This section analyzes discrete acoustic unit quality to reveal the relationship between speech representations and phonemes. To inspect this property, we take discrete units produced by an SSL model like the codeword IDs in the Spin model. For other models that cannot directly extract discrete codes, we apply K-means clustering on the hidden representations of a model and take the cluster IDs as pseudo labels (Hsu et al., 2021a).

4.5.1 Metrics

We adopt three metrics proposed in (Hsu et al., 2021a), where higher values imply better quality.

1. **Cluster Purity** measures the purity of each phoneme's associated discrete units.
2. **Phone Purity** measures the average phoneme purity within one class of discrete units.
3. **Phone-normalized mutual information** (PNMI) measures the uncertainty reduction for the underlying phone when observing the codeword of a frame.

K-means clustering is performed on a random 10-hour subset of the LibriSpeech train-clean-100 split. The discrete units are evaluated on the combination of LibriSpeech dev-clean and dev-other splits. The offline clustering scores are averaged over three runs.

4.5.2 Results

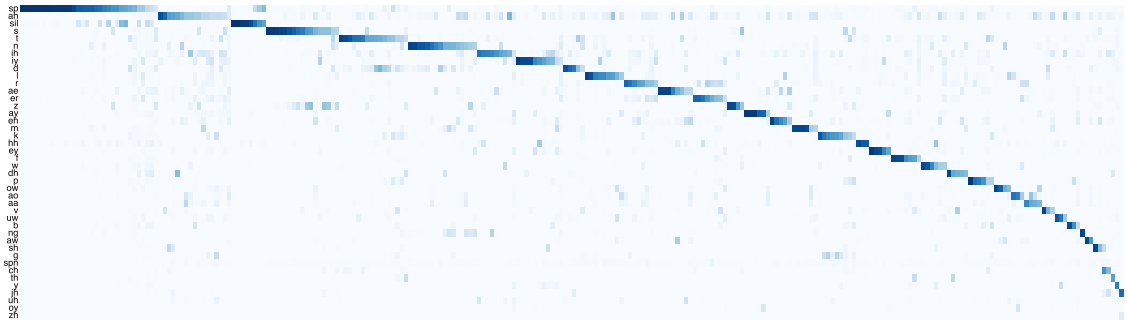
First, we cluster continuous representations into 256 clusters and report the layer with the highest PNMI, as shown in the upper part of Table 4.2. Independent of codebook sizes and

Table 4.2: Discrete unit quality. Only the layer with the highest PNMI is reported for each model and is specified in column "Layer".

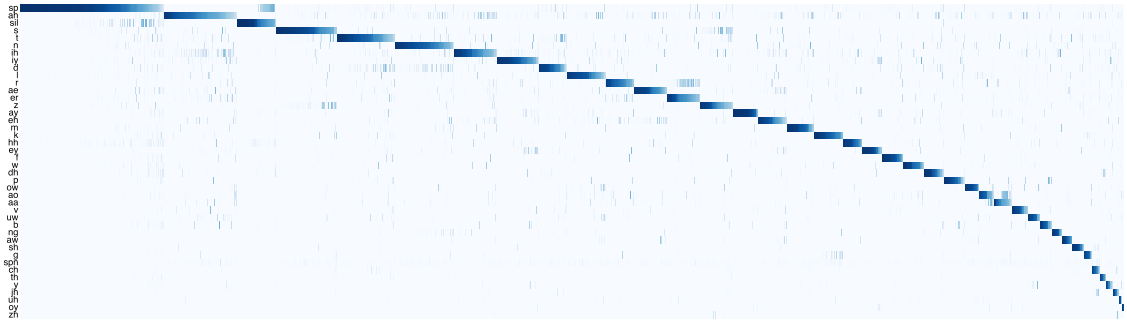
Method	Layer	Active Clusters	Cluster Purity	Phone Purity	PNMI
K-means Clustering ($K = 256$)					
HuBERT	7	256	0.154	0.639	0.630
WavLM	11	256	**0.178**	0.624	0.640
data2vec	4	256	0.173	0.652	0.630
DinoSR	5	256	0.168	0.631	0.616
ContentVec$_{100}$	12	256	0.169	0.650	0.643
ContentVec$_{500}$	8	256	0.154	0.639	0.629
HuBERT + R-Spin$_{256}$	12	256	0.150	0.641	0.655
HuBERT + R-Spin$_{2048}$	12	256	0.151	**0.654**	**0.666**
WavLM + Spin$_{256}$	12	256	0.137	0.644	0.658
WavLM + Spin$_{2048}$	12	256	0.153	0.650	**0.666**
HuBERT + R-Spin$_{32,\ \text{AP40k}}$	12	256	0.152	0.608	0.607
WavLM + R-Spin$_{32,\ \text{AP40k}}$	12	256	0.162	0.612	0.613
WavLM + R-Spin$_{2048,\ \text{AP40k}}$	12	256	0.153	0.627	0.632
Online Clustering (**Codebook**)					
VQ-APC (Chung et al., 2020)	–	98	0.078	0.240	0.189
Co-training APC (Yeh and Tang, 2022)	–	164	0.089	0.308	0.294
DinoSR	–	217	**0.189**	0.582	0.569
HuBERT + Spin$_{256}$	–	256	0.138	0.642	0.658
WavLM + Spin$_{256}$	–	256	0.133	**0.646**	**0.659**
HuBERT + R-Spin$_{256,\ \text{AP40k}}$	–	256	0.135	0.547	0.568
WavLM + R-Spin$_{256,\ \text{AP40k}}$	–	256	0.144	0.620	0.636

pre-trained models, Spin outperforms all baselines in PNMI. Increasing the codebook size in Spin improves all three metrics (Spin$_{256}$ vs. Spin$_{2048}$), indicating that a larger codebook learns more fine-grained phoneme representations.

For online clustering (codebook learning), we compare the codebook in Spin$_{256}$ with VQ-APC (Chung et al., 2020) and Co-training APC (Yeh and Tang, 2022), where the latter two methods leverage codebook learning to improve content modeling. We produce discrete units for Spin by taking arg max over p per frame. In the lower part of Table 4.2, codebooks in Spin achieve high PNMI compared with prior works. Unlike prior methods, because of the constraint in Equation 2.1, all learned codewords are utilized in Spin. Besides, similar to the previous observations in Section 4.4, R-Spin offers worse discrete unit quality, which is also caused by noise-invariant training.



(a) $K = 256$

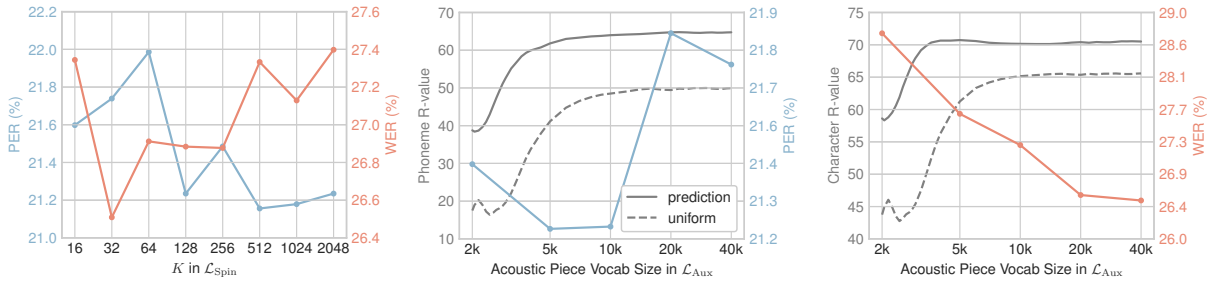


(b) $K = 2048$

Figure 4.9: $P(\text{phone}|\text{code})$ for HuBERT + Spin$_K$. The vertical axes represent the phones sorted from high to low frequencies.

Next, we visualize $P(\text{phone}|\text{code})$ in Figure 4.9 to demonstrate the relation between learned codewords and phonemes. Since the vertical axes are sorted by phoneme occurrence frequency in human speech, the figures show that Spin applies more codewords to represent high-frequency phonemes. Furthermore, because off-diagonal values of $K = 2048$ are lower than those of $K = 256$ (Figure 4.9b vs. 4.9a), a larger codebook helps each code to focus on encoding one phoneme, consistent with phone purity in Table 4.2. Overall, Spin learns good discrete acoustic units and improves continuous representations in SSL models.

Besides, we inspect the importance of the codebook size in Spin. As highlighted in Section 2.3.4, the codebook size positively correlates with phoneme recognition. A similar trend can be found in Figure 4.10a but has an inverted trend for ASR. However, the observed performance discrepancy is less than 1%, suggesting that the impact of codebook size on R-Spin is marginal. In contrast, substantial improvements in ASR are observed with larger acoustic piece vocabularies, as evidenced by Figures 4.10c and 4.10b, while such improvements are



(a) (b) (c)

Figure 4.10: WavLM + R-Spin results with different (a) codebook size K's and (b)(c) AP vocabulary sizes in $\mathcal{L}_{\text{Aux}}$. (b) and (c) depict the phoneme and character segmentation R-values, where the dotted curves are the baselines by segmenting each utterance with equal-length segments given the number of boundaries obtained by the acoustic pieces. The PERs are calculated by averaging over different noise conditions on LibriSpeech test-other. The WERs are the averaged scores of the real and simulated evaluation sets of CHiME-4.

not in phoneme recognition. To analyze this phenomenon, we investigate the phoneme and character segmentation capabilities using discrete units.

4.6 Phoneme Segmentation with Discrete Units

In this section, we segment speech with acoustic pieces and show the R-values in Figures 4.10b and 4.10c. R-value is a robust metric for evaluating word or phoneme segmentation (Räsänen et al., 2009), which is calculated with recall (R) and precision (P):

$$\text{R-value} = 1 - \frac{|r_1| + |r_2|}{2},$$

where

$$\begin{cases} r_1 &= \sqrt{(1-R)^2 + (OS)^2} \\ r_2 &= (-OS + R - 1)/\sqrt{2} \\ OS &= R/P - 1 \end{cases}$$

The predicted boundaries are computed by finding the locations where the adjacent discrete units differ. We perform this task on the force-aligned LibriSpeech dev-clean and dev-other sets, including both phonetic and word level alignments (Lugosch et al., 2019; McAuliffe

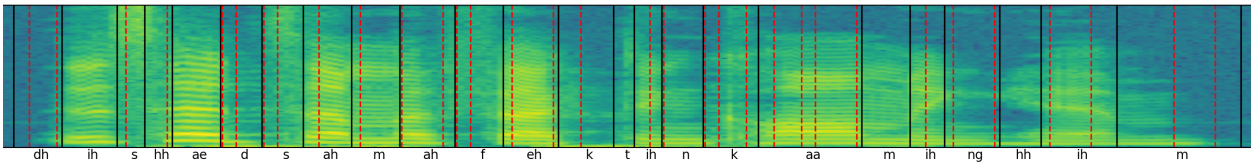


Figure 4.11: An example of phoneme alignment of an utterance "This had some effect in calming him." from LibriSpeech dev-clean. The black lines indicate the force-aligned boundaries, while the red dashed lines are the predicted boundaries with AP40k.

et al., 2017).[1] The character boundaries are obtained by dividing each force-aligned word segment into equal-length segments corresponding to individual characters within the word. More accurate boundaries can be computed with character-based aligners, but we only need a rough estimation of the character segmentation quality.

As depicted in both Figures 4.10b and 4.10c, larger AP vocabulary sizes have superior segmentation, indicating that a greater number of APs contribute to forming units that closely resemble linguistic units. The baseline, which involves uniformly segmenting utterances based on the number of boundaries derived from APs, underscores the non-random nature of AP boundaries. Although the segmentation capability of APs is incomparable with other unsupervised speech segmentation algorithms (Kreuk et al., 2020), they present significantly improved targets for $\mathcal{L}_{\text{Aux}}$, consequently enhancing the accuracy of ASR.

Furthermore, we show an example of segmenting an utterance with acoustic pieces of 40k vocabularies in Figure 4.11. The red dashed stripes visually depict that the boundaries of APs are mostly aligned with phoneme boundaries. Notably, the predicted boundaries occasionally exhibit a slight temporal lag compared to the ground truth, like the initial occurrences of `ah` and `m`. We suspect the 50Hz framerate of HuBERT or the Spin training objective causes this phenomenon since they could reduce time resolution and introduce temporal shifts in representations. Still, the actual cause remains a subject for future investigation.

[1] https://zenodo.org/record/2619474 (CC-BY 4.0)

4.7 Chapter Summary

This chapter comprehensively analyzed the characteristics of perturbation-invariant speech representations and discrete units like acoustic pieces. The analyses demonstrate the perturbation invariability of Spin and R-Spin, indicating the effectiveness of the proposed methods. Finally, the findings offer insights into the properties and applications of continuous and discrete acoustic representations, thereby benefiting future studies.

Chapter 5

Conclusions and Future Work

5.1 Summary of Contributions

This thesis introduces two novel and efficient methods for learning perturbation-invariant speech representations for content-related tasks. First, Speaker-invariant Clustering (Spin) is proposed to remove speaker information and preserve content. Second, building on top of Spin, Robust Spin (R-Spin) mitigates the shortcomings of Spin and extends Spin to process more diverse audio recordings. Both approaches offer significant improvements in phoneme recognition and automatic speech recognition tasks. Furthermore, to understand the characteristics of the perturbation-invariant representations, we conduct a wide range of analyses, including quantitative and qualitative. The analyses offer insights into the properties of continuous and discrete acoustic representations, benefiting future studies and developments.

5.2 Future Work

Beyond this thesis, there are some potential extensions to fully understand and leverage perturbation-invariant speech representation models. First, the dataset employed consists of English utterances spoken by native speakers, predominantly of North American dialects,

leaving the performance in other accents unexplored. Thus, it is suggested that the impact of English accents and dialects on Spin and R-Spin be examined. Second, Spin and R-Spin are both good at learning linguistic units from English speech, so extending these methods to other languages or multilingual situations is worth studying. Third, scaling the proposed frameworks in this thesis to larger scales is crucial for real-world applications since the experiments are conducted on 95M parameters models. Last, to fully comprehend the capabilities of the proposed method, further analyses and extensions to other applications are recommended for future exploration (Sicherman and Adi, 2023).

References

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. *ICLR*.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. data2vec: A general framework for self-supervised learning in speech, vision and language. In *ICML*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS*.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*.

David M Chan and Shalini Ghosh. 2022. Content-context factorized representations for automated speech recognition. *Interspeech*.

Heng-Jui Chang and James Glass. 2023. R-Spin: Efficient Speaker and Noise-invariant Representation Learning with Acoustic Pieces. In *arXiv*.

Heng-Jui Chang, Alexander H. Liu, and James Glass. 2023a. Self-supervised Fine-tuning for Improved Content Representations by Speaker-invariant Clustering. In *Interspeech*.

Heng-Jui Chang, Alexander H Liu, Hung-yi Lee, and Lin-shan Lee. 2021a. End-to-end whispered speech recognition with frequency-weighted approaches and pseudo whisper pre-training. In *SLT*.

Heng-Jui Chang, Shu-wen Yang, and Hung-yi Lee. 2022. DistilHuBERT: Speech representation learning by layer-wise distillation of hidden-unit bert. In *ICASSP*.

Xuankai Chang, Takashi Maekaku, Pengcheng Guo, Jing Shi, Yen-Ju Lu, Aswin Shanmugam Subramanian, Tianzi Wang, Shu wen Yang, Yu Tsao, Hung yi Lee, and Shinji Watanabe. 2021b. An exploration of self-supervised pretrained representations for end-to-end speech recognition. In *ASRU*.

Xuankai Chang, Brian Yan, Kwanghee Choi, Jeeweon Jung, Yichen Lu, Soumi Maiti, Roshan Sharma, Jiatong Shi, Jinchuan Tian, Shinji Watanabe, et al. 2023b. Exploring speech recognition, translation, and understanding with discrete speech units: A comparative study. *arXiv*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE JSTSP*, 16.

Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Lee, Hoon Heo, and Kyogu Lee. 2021. Neural analysis and synthesis: Reconstructing speech from self-supervised representations. *NeurIPS*.

Jan Chorowski, Grzegorz Ciesielski, Jarosław Dzikowski, Adrian Łańcucki, Ricard Marxer, Mateusz Opala, Piotr Pusz, Paweł Rychlikowski, and Michał Stypułkowski. 2021. Information Retrieval for ZeroSpeech 2021: The Submission by University of Wroclaw. In *Interspeech*.

Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. 2019. An unsupervised autoregressive model for speech representation learning. In *Interspeech*.

Yu-An Chung, Hao Tang, and James Glass. 2020. Vector-quantized autoregressive predictive coding. In *Interspeech*.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language

modeling for self-supervised speech pre-training. In *ASRU*.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *NIPS*.

Ellen Eide and Herbert Gish. 1996. A parametric approach to vocal tract length normalization. In *ICASSP*.

John S Garofolo. 1993. Timit acoustic phonetic continuous speech corpus. *LDC*.

Yuan Gong, Sameer Khurana, Leonid Karlinsky, and James Glass. 2023. Whisper-at: Noise-robust automatic speech recognizers are also strong audio event taggers. *Interspeech*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021a. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *TASLP*, 29.

Wei-Ning Hsu, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve, et al. 2021b. Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. *arXiv*.

Yuchen Hu, Chen Chen, Qiushi Zhu, and Eng Siong Chng. 2023. Wav2code: Restore clean speech representations via codebook lookup for noise-robust asr. *arXiv*.

Kuan Po Huang, Yu-Kuan Fu, Yu Zhang, and Hung-yi Lee. 2022a. Improving distortion robustness of self-supervised speech processing tasks with domain adaptation. *Interspeech*.

Wenyong Huang, Zhenhe Zhang, Yu Ting Yeung, Xin Jiang, and Qun Liu. 2022b. Spiral: Self-supervised perturbation-invariant representation learning for speech pre-training. *ICLR*.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Interspeech*.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. 2017. A study on data augmentation of reverberant speech for robust speech recognition. In *ICASSP*.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *ICML*.

Felix Kreuk, Joseph Keshet, and Yossi Adi. 2020. Self-supervised contrastive learning for unsupervised phoneme segmentation. *Interspeech*.

Tatiana Likhomanenko, Qiantong Xu, Jacob Kahn, Gabriel Synnaeve, and Ronan Collobert. 2020. slimipl: Language-model-free iterative pseudo-labeling. *arXiv preprint arXiv:2010.11524*.

Alexander H Liu, Heng-Jui Chang, Michael Auli, Wei-Ning Hsu, and James R Glass. 2023. Dinosr: Self-distillation and online clustering for self-supervised speech representation learning. *NeurIPS*.

Alexander H Liu, Yu-An Chung, and James Glass. 2021a. Non-autoregressive predictive coding for learning speech representations from local dependencies. In *Interspeech*.

Andy T Liu, Shang-Wen Li, and Hung-yi Lee. 2021b. TERA: Self-supervised learning of transformer encoder representation for speech. *TASLP*, 29.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech model pre-training for end-to-end spoken language understanding. *Interspeech*.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al. 2022. Self-supervised speech representation learning: A review. *IEEE JSTSP*.

Dianwen Ng, Ruixi Zhang, Jia Qi Yip, Zhao Yang, Jinjie Ni, Chong Zhang, Yukun Ma, Chongjia Ni, Eng Siong Chng, and Bin Ma. 2023. De'hubert: Disentangling noise in a self-supervised model for robust speech recognition. In *ICASSP*.

Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Evgeny Kharitonov, Alexei Baevski, Ewan Dunbar, and Emmanuel Dupoux. 2020. The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling. *arXiv*.

Tu Anh Nguyen, Benoit Sagot, and Emmanuel Dupoux. 2022. Are discrete units necessary for spoken language modeling? *IEEE JSTSP*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL-HLT: Demonstrations*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP*.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech*.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. 2021. Layer-wise analysis of a self-supervised speech representation model. *ASRU*.

Douglas B. Paul and Janet M. Baker. 1992. The design for the Wall Street Journal-based CSR corpus. In *HLT*.

Cal Peyser, W. Ronny Huang, Andrew Rosenberg, Tara Sainath, Michael Picheny, and Kyunghyun Cho. 2022. Towards disentangled speech representations. *Interspeech*.

Kaizhi Qian, Yang Zhang, Heting Gao, Junrui Ni, Cheng-I Lai, David Cox, Mark Hasegawa-Johnson, and Shiyu Chang. 2022. Contentvec: An improved self-supervised speech representation by disentangling speakers. In *ICML*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv*.

Okko Johannes Räsänen, Unto Kalervo Laine, and Toomas Altosaar. 2009. An improved speech segmentation quality measure: the r-value. In *Interspeech*.

Shuo Ren, Shujie Liu, Yu Wu, Long Zhou, and Furu Wei. 2022. Speech pre-training with acoustic piece. *Interspeech*.

Thomas Schatz. 2016. *ABX-discriminability measures and applications*. Ph.D. thesis, Université Paris 6 (UPMC).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Feiyu Shen, Yiwei Guo, Chenpeng Du, Xie Chen, and Kai Yu. 2023. Acoustic bpe for speech generation with discrete tokens. *arXiv*.

Amitay Sicherman and Yossi Adi. 2023. Analysing discrete self supervised speech representation for spoken language modeling. In *ICASSP*.

David Snyder, Guoguo Chen, and Daniel Povey. 2015. Musan: A music, speech, and noise corpus. *arXiv*.

Kenneth N Stevens. 1987. Relational properties as perceptual correlates of phonetic features. In *International Conference of Phonetic Sciences*.

Andros Tjandra, Ruoming Pang, Yu Zhang, and Shigeki Karita. 2021. Unsupervised learning of disentangled speech content and style representation. *Interspeech*.

Hsiang-Sheng Tsai, Heng-Jui Chang, Wen-Chin Huang, Zili Huang, Kushal Lakhotia, Shu-wen Yang, Shuyan Dong, Andy Liu, Cheng-I Lai, Jiatong Shi, Xuankai Chang, Phil Hall, Hsuan-Jui Chen, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. 2022. SUPERB-SG: Enhanced speech processing universal PERformance benchmark for semantic and generative capabilities. In *ACL*.

Liang-Hsuan Tseng, Yu-Kuan Fu, Heng-Jui Chang, and Hung-yi Lee. 2022. Mandarin-english code-switching speech recognition with self-supervised speech representation models. *AAAI SAS Workshop*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and Ricard Marxer. 2017. An analysis of environment, microphone and data simulation mismatches in robust speech recognition. *Computer Speech & Language*, 46:535–557.

Yiming Wang, Jinyu Li, Heming Wang, Yao Qian, Chengyi Wang, and Yu Wu. 2022. Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition. In *ICASSP*.

Jennifer Williams. 2022. *Learning disentangled speech representations*. Ph.D. thesis, The University of Edinburgh.

Felix Wu, Kwangyoun Kim, Shinji Watanabe, Kyu J Han, Ryan McDonald, Kilian Q Weinberger, and Yoav Artzi. 2023. Wav2seq: Pre-training speech-to-text encoder-decoder models using pseudo languages. In *ICASSP*.

Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert. 2020. Iterative pseudo-labeling for speech recognition. *Interspeech*.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. 2021. SUPERB: Speech processing universal performance benchmark. In *Interspeech*.

Sung-Lin Yeh and Hao Tang. 2022. Autoregressive Co-Training for Learning Discrete Speech Representation. In *Interspeech*.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*.

Qiu-Shi Zhu, Jie Zhang, Zi-Qiang Zhang, Ming-Hui Wu, Xin Fang, and Li-Rong Dai. 2022.

A noise-robust self-supervised pre-training model based speech representation learning for automatic speech recognition. In *ICASSP*.

Qiu-Shi Zhu, Long Zhou, Jie Zhang, Shu-Jie Liu, Yu-Chen Hu, and Li-Rong Dai. 2023. Robust data2vec: Noise-robust speech representation learning for asr by combining regression and improved contrastive learning. In *ICASSP*.