R-Spin: Efficient Speaker and Noise-invariant Representation Learning with Acoustic Pieces

Anonymous ACL submission

Abstract

This paper 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 speaker-invariant clustering (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. This paper provides detailed analyses to show how discrete units contribute to speech encoder training and improving robustness in diverse acoustic environments.

1 Introduction

Self-supervised learning (SSL) for encoder pre-training has emerged as a foundational element in speech processing, outperforming conventional approaches across various applications (Mohamed et al., 2022; Liu et al., 2022). Given the substantial cost associated with human annotation of speech data, SSL methods leverage unlabeled audio data to pre-train encoders, generating good representations for downstream tasks like automatic speech recognition (ASR) and speaker identification (Yang et al., 2021; Tsai et al., 2022). The application of SSL models has notably concentrated on ASR, aiming to mitigate the dependence on large transcribed corpora (Hsu et al., 2021a; Baevski et al., 2022; Liu et al., 2023). Thus, 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). Prior studies have devised objective functions to disentangle content from speech, fostering the ability of SSL models to generate speaker-invariant representations through self-supervised fine-tuning (SSFT). In SSFT, unlabeled data is utilized to refine a pre-trained SSL model for specific applications. Qian et al. (2022) propose ContentVec by disentangling speaker and content information, demonstrating promising results. However, ContentVec suffers from the requirement of a voice conversion model and substantial computational costs exceeding 600 GPU hours. Alternatively, Chang et al. (2023) propose Speaker-invariant Clustering (Spin) to produce content representations with minimal fine-tuning resources. Nonetheless, Spin is constrained to fine-tuning only the top layers, thereby lacking the flexibility to adapt to diverse acoustic domains.

Parallel to modeling content information in speech, numerous studies are dedicated to investigating the robustness of speech SSL. 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) proposes HuBERT-MGR via domain adversarial training to render the fine-tuned HuBERT model (Hsu et al., 2021a) 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.

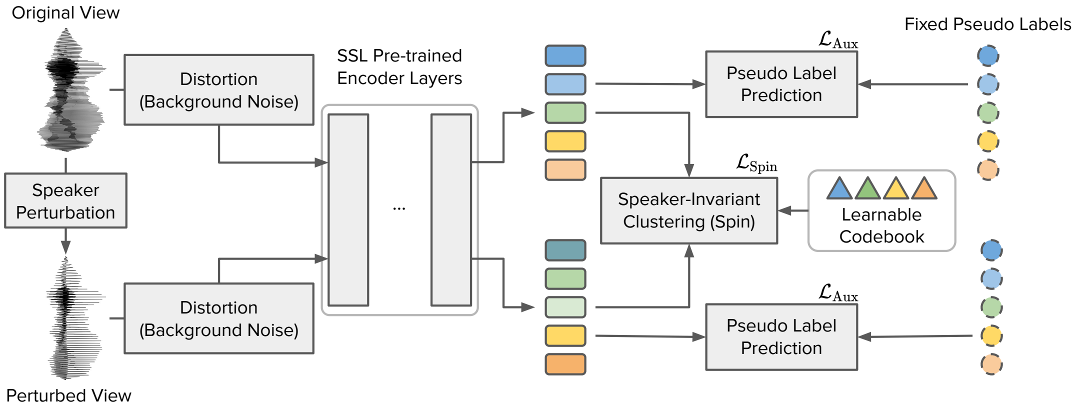

Original View
Speaker Perturbation
Perturbed View
Distortion (Background Noise)
Distortion (Background Noise)
SSL Pre-trained Encoder Layers
...
$\mathcal{L}_{\text{Aux}}$
Pseudo Label Prediction
$\mathcal{L}_{\text{Spin}}$
Speaker-Invariant Clustering (Spin)
Learnable Codebook
$\mathcal{L}_{\text{Aux}}$
Pseudo Label Prediction
Fixed Pseudo Labels

Figure 1: The proposed R-Spin self-supervised fine-tuning framework. The input utterance is perturbed into a different voice and distorted with random noise. 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) (Chang et al., 2023) objective ($\mathcal{L}_{\text{Spin}}$) and frame-wise pseudo-label prediction loss ($\mathcal{L}_{\text{Aux}}$).

To effectively acquire high-quality content and robust representations for real-world applications, this paper extends Spin with noise-invariant training and acoustic piece pseudo-label learning, coined Robust Spin (R-Spin). During training, two utterances of the same content yet subjected to distinct distortions are fed into a speech SSL encoder. The encoder outputs are then frame-wise vector-quantized with a learnable codebook via online clustering, as in Spin. The model is trained to predict swapped cluster ID between the two utterances. To prevent model collapse, an additional pseudo-label prediction loss is introduced. These pseudo-labels are generated by learning acoustic pieces (Ren et al., 2022a) on top of the discrete units produced by a pre-trained Spin model, offering better training targets that closely align with phonemes and characters. Within this framework, the speech encoder learns speaker and noise-invariant representations, benefiting robustness and content extraction simultaneously. The contributions are summarized as follows:

1. We integrate predicting acoustic pieces into Spin, enabling fine-tuning all parameters without collapsing, which allows processing more complex speech recordings.
2. R-Spin inherits the benefit of low training costs from Spin, requiring 12X less computation than prior arts.
3. With noise-invariant training, R-Spin outperforms other SSFT approaches in distorted speech and phoneme recognition tasks like the CHiME-4 challenge (Vincent et al., 2017).
4. We inspect the hidden representations of speech SSL models to quantify the speaker and noise invariability.
5. We offer in-depth analyses of discrete acoustic units to understand how these units help speech encoder training.

2 Method

2.1 Overview

The proposed R-Spin framework is shown in Fig. 1. R-Spin is based on Speaker-invariant Clustering (Spin) (Chang et al., 2023), a self-supervised fine-tuning method with online clustering and swapped prediction for capturing content representations (Sec. 2.2). We introduce noise-invariant training by perturbing inputs to improve robustness (Sec. 2.3). Moreover, an auxiliary pseudo-label prediction loss enables fine-tuning the entire model without collapsing (Sec. 2.4). Acoustic Piece is incorporated with the auxiliary loss to improve performance further (Sec. 2.5).

2.2 Speaker-invariant Clustering

Spin is an efficient SSFT method for improving content representations inspired by Swapping Assignments between Views (SwAV) (Caron et al., 2020). We briefly introduce Spin and suggest readers refer to the original paper for further details.

For each utterance in a mini-batch, the F0 frequency and the relative ratio between formant frequencies are randomly perturbed to mimic the same sentence spoken by a different speaker (Choi et al., 2021; Qian et al., 2022). The original and perturbed views are fed into a transformer encoder (Vaswani et al., 2017) initialized with an SSL model like HuBERT (Hsu et al., 2021a). The output representations $\mathbf{H} = [\boldsymbol{h}_1 \dots \boldsymbol{h}_B]^{\mathsf{T}}$ of the original view are linearly projected and L2-normalized to $\mathbf{Z} = [\boldsymbol{z}_1 \dots \boldsymbol{z}_B]^\intercal$, where B is the number of frames in a batch. We then use the representations to compute a probability distribution over a learnable codebook as $p\left(\cdot|\boldsymbol{z}_b\right)$. We perform the same operations to the perturbed utterance, resulting in another distribution $q\left(\cdot|\tilde{\boldsymbol{z}}_b\right)$, where $\tilde{\cdot}$ denotes features from the speaker-perturbed view. Next, q is smoothed by solving an optimal transport problem to enforce full codebook usage. Finally, the model is trained to perform swapped predictions between views by minimizing the cross-entropy loss

$$\begin{aligned}\mathcal{L}_{\text{Spin}} = &-\frac{1}{2B}\sum_{b\in[B]}\sum_{k\in[K]} q\left(k|\tilde{\boldsymbol{z}}_b\right)\log p\left(k|\boldsymbol{z}_b\right) \\ &-\frac{1}{2B}\sum_{b\in[B]}\sum_{k\in[K]} q\left(k|\boldsymbol{z}_b\right)\log p\left(k|\tilde{\boldsymbol{z}}_b\right),\end{aligned} \tag{1}$$

where K is the size of the learnable codebook, and the second term emerges from the interchangeability of the role of the perturbed and original speech.[1] Under this SSFT framework, the fine-tuned model produces speaker-invariant representations, making the content of speech signals more accessible to downstream applications.

2.3 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.

2.4 Auxiliary Pseudo-label Prediction Loss

As noted by Chang et al. (2023), 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., 2021; 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, having a loss function of

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

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 Eqs. 1 and 2, the overall loss function is

$$\mathcal{L} = \mathcal{L}_{\text{Spin}} + \lambda\mathcal{L}_{\text{Aux}}, \tag{3}$$

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.

2.5 Acoustic Piece

This section introduces acoustic pieces (Ren et al., 2022b) 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. AP captures 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 AP as the target of $\mathcal{L}_{\text{Aux}}$ to extract better content representations.

Following Ren et al. (2022b), 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 Eq. 2, expecting to encourage the fine-tuned SSL model to encode better phoneme and character representations.

[1] $[N] = \{1, 2, \dots, N\}$ for any positive integer N.

3 Experiments

3.1 Data

The 960 hours of unlabeled English speech in LibriSpeech is used for R-Spin training (Panayotov et al., 2015).[2] Audio distortions are generated with torch-audiomentations.[3] Following 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.[4] 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 (Appendix A.5).

3.2 Implementation

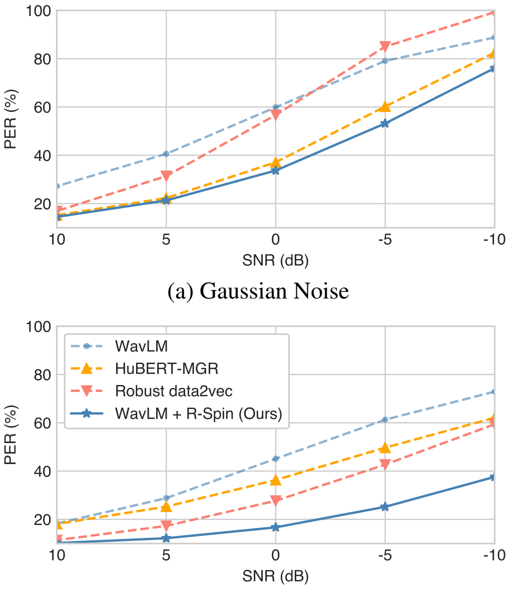
The SSFT experiments are mostly based on WavLM (Chen et al., 2022) because WavLM is pre-trained with a denoising objective, offering a good initialization. HuBERT (Hsu et al., 2021a) is also considered to demonstrate R-Spin's generalizability to SSL models trained with clean speech. We follow the implementations by Chang et al. (2023), which uses PyTorch (Paszke et al., 2019), PyTorch-Lightning (Falcon, 2019), and torchaudio (Yang et al., 2022).[5] The acoustic pieces are generated by learning BPE tokens on top of a HuBERT + Spin$_{2048}$ model (Appendix A.2). Further details can be found in Appendix A.3. We employ GitHub Copilot and ChatGPT for implementation and writing assistance.

3.3 Notations

We denote X + Spin$_K$ as an SSL model X fine-tuned with Spin with a codebook size of K. In X + 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 R-Spin$_{32,\text{ AP40k}}$.

3.4 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-tuned (Appendix A.5).[6] 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 (Sec. 3.6), directly related to the resources used.



(a) Gaussian Noise

(b) MUSAN Noise

Figure 2: Phoneme error rates (PER) under different noise types and SNRs. R-Spin$_{32,\text{ AP40k}}$ is used here.

As shown in the middle columns of Table 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 apart. Hence, we reduce the batch size to train a low-budget version of the Robust data2vec (Appendix A.4). 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 Fig. 2. Note that these models are trained with MUSAN noise except for WavLM, and only HuBERT-MGR uses Gaussian

[2] Released under CC BY 4.0

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

[4] MUSAN: CC BY 4.0 / CHiME-4: CC BY-NC-SA 2.0

[5] https://github.com/vectominist/spin

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

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., 2023)	0.4k	8.4	70.8	47.8	18.4	71.3	62.0
WavLM + Spin$_{2048}$ (Chang et al., 2023)	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 + R-Spin$_{32,\text{ AP40k}}$	8.2k	8.3	36.4	18.2	16.3	34.3	34.1
WavLM + 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.

Table 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 Chang et al. (2023).

noise. The proposed method achieves the overall lowest PERs. HuBERT-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.

3.5 Noisy Speech Recognition

This section assesses R-Spin with a noisy ASR task. We adopt the SUPERB ASR setup with the CHiME-4 corpus (Vincent et al., 2017) to evaluate the models in more realistic noisy recordings (Appendix A.6). The results in the right columns of Table 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.

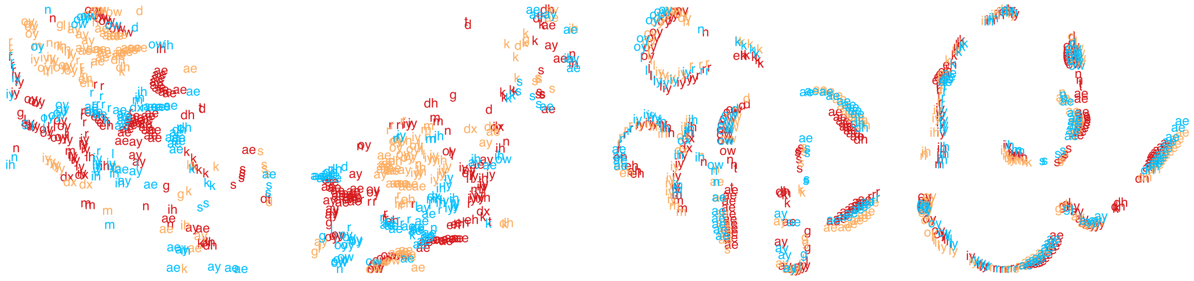
3.6 Data Efficiency

The objective behind developing R-Spin is to enhance speech SSL models with minimal resources, including reducing training data to improve data efficiency. Following Chang et al. (2023), 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 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. A complete comparison of the training costs can be found in Appendix C.

3.7 Representation Invariability

3.7.1 Speaker Invariability

This section explores the robustness of SSL models regarding representation invariability by examining the characteristics of representations under diverse



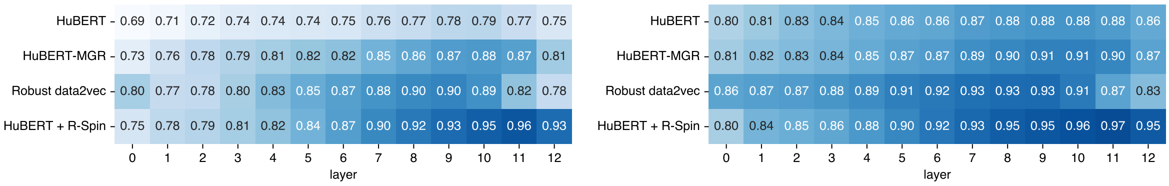
(a) HuBERT
CNN

(b) HuBERT + R-Spin
CNN

(c) HuBERT
Layer 9

(d) HuBERT + R-Spin
Layer 12

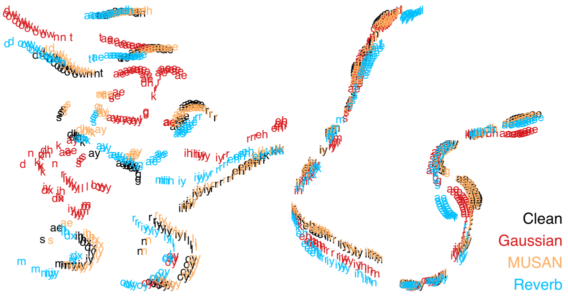
Figure 3: t-SNE (Van der Maaten and Hinton, 2008) visualization of the CNN and the layer with the lowest speaker identification rate given the same clean utterance spoken by three different speakers from TIMIT (Garofolo, 1993). Each color represents a speaker, while each label visualizes a frame representation 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) Gaussian Noise (SNR = 0dB)

(b) Reverberation (real room impulse response)

Figure 4: 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) HuBERT
Layer 9

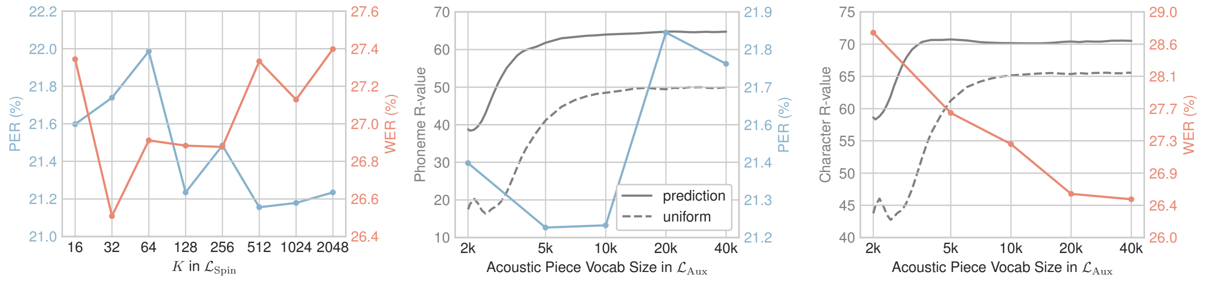
(b) HuBERT + R-Spin
Layer 12

Figure 5: t-SNE visualization of hidden representations of the same audio utterance in Fig. 3 with different distortions indicated by colors, where SNR = 0dB.

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 Fig. 10. In Figs. 3a and 3b, there is a discernible clustering of frames uttered by the same speaker, suggesting that lower layers retain more speaker-specific information. Conversely, Figs. 3c and 3d 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 Fig. 3d. Besides, a comparative analysis between Figs. 3c and 3d 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 and the layer-wise speaker identification results can be found in Appendices B.3 and D.

3.7.2 Noise Invariability

Here, we examine the response of continuous representations to input distortions. Similar to Sec. 3.7.1, we employ t-SNE visualization to explore hidden representations under different distortions. As shown in Fig. 5, the R-Spin representations exhibit a more pronounced overlap than those subjected to HuBERT, suggesting that R-Spin demonstrates greater robustness to noise. Fig. 5b 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



(a)

(b)

(c)

Figure 6: 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.

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. Fig. 4 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 Sec. 2.4, which associates lower layers with fundamental signal processing functions. Overall, the analysis underscores the notable noise invariability offered by R-Spin.

3.8 Importance of Discrete Units

This section analyzes the impact of discrete acoustic units on the downstream performance and the relation between APs, phonemes, and characters.

3.8.1 Codebook and Acoustic Pieces Size

We inspect the importance of the codebook size in Spin. As highlighted by Chang et al. (2023), the codebook size positively correlates with phoneme recognition. A similar trend can be found in Fig. 6a 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 Figs. 6c and 6b, while such improvements are not in phoneme recognition. To analyze this phenomenon, we investigate the phoneme and character segmentation capabilities using discrete units.

3.8.2 Phoneme and Character Segmentation

Here, we segment speech with acoustic pieces and show the R-values in Figs. 6b and 6c. R-value is a robust metric for evaluating word or phoneme segmentation (Räsänen et al., 2009). 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 (Lugosch et al., 2019; McAuliffe et al., 2017).[7] 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 Figs. 6b and 6c, 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. A full comparison of unsupervised phoneme segmentation can be found in Appendix B.4.

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

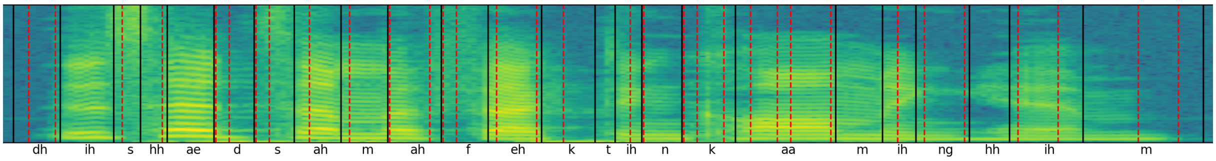


Figure 7: An example of phoneme alignment of an utterance "This had some effect in calming him." from LibriSpeech dev-clean. The black lines indicate the force-aligned boundaries, while the red dashed lines are the predicted boundaries with AP40k.

	CHiME-4	
Method	Real	Sim
Spin_{2048} (Chang et al., 2023)	52.1	46.6
R-$\text{Spin}_{32,\ \text{AP40k}}$ (Proposed)	**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

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

Table 2: CHiME-4 ASR results for ablation studies based on fine-tuned WavLM models.

Furthermore, we show an example of segmenting an utterance with acoustic pieces of 40k vocabularies in Fig. 7. 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. To summarize, APs effectively learn discrete acoustic units and benefit ASR performance.

3.9 Ablation Studies

Under the same CHiME-4 ASR setup in Sec. 3.5, we conduct ablation studies to analyze the design of the proposed methods. As shown in Table 2, by removing the proposed auxiliary loss, the WERs increase significantly, showing that $\mathcal{L}_{\text{Aux}}$ helps ASR performance (Sec. 3.8) and allows fine-tuning the entire model. Second, the absence of the Spin loss resulted in a roughly 5% increase in WERs, indicating the necessity of this loss for achieving perturbation-invariant representations. Speaker perturbation also plays an important role in offering good content representations according to the degraded WERs. Moreover, the fine-tuned model exhibited suboptimal performance when trained without noise, emphasizing the importance of noise-invariant training for improving robustness. The above findings collectively verify the necessity of the proposed approaches.

We investigate the effect of choosing different pseudo-labels for the auxiliary loss. First, acoustic pieces are essential to the R-Spin training since learning from the original Spin model's 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, employing clustered HuBERT representations yields results comparable to Spin_{2048}, and the t-test suggests the disparities between pseudo-labels are statistically insignificant. This suggests that using clustered features from an SSL model is a viable alternative when a pre-trained Spin model is unavailable. For further insights, additional ablation studies are available in Appendix B.1.

4 Conclusion

This paper proposes R-Spin, a self-supervised fine-tuning method with speaker and noise-invariant clustering for robust content representations. Results illustrate the efficacy and broad applicability of R-Spin across various acoustic scenarios, even within constrained computation budgets. The acoustic analyses presented in this study offer insights into the characteristics of discrete units of this nature and strategies for their utilization. Future directions involve scaling to larger models and exploring its application in diverse downstream tasks like robust voice conversion.

Limitations

This paper faces four primary limitations due to constrained computation resources and available data. First, we consider background noises including human speech, music, and natural noises, and evaluate the proposed methods with similar noise types and reverberation, covering many real-world conditions. However, the trained models may encounter challenges in processing more severely distorted audio data, such as air traffic control communications. Second, the dataset employed consists of English utterances spoken by native speakers, predominantly of North American dialects, leaving the performance in other languages and accents unexplored. Third, the experiments are conducted on 95M parameters models, so the scalability of R-Spin remains unknown. 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). These questions can be answered by experimenting with diverse datasets and more computation resources.

Ethics Statement

Our models inherit the biases of SSL models (HuBERT and WavLM) pre-trained on the LibriSpeech corpus. This corpus contains read English audio recordings derived from audiobooks. Limitations arise when confronted with accents and topic domains outside the corpus scope, potentially diminishing the effectiveness of the proposed methods. Thus, the direct application of our models to real-world scenarios may result in increased speech recognition error rates. These errors, if unaddressed, can propagate through downstream applications like natural language processing systems, leading to potential risks for users, such as the misinterpretation of voice commands.

References

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*.

Saurabhchand Bhati, Jesús Villalba, Piotr Żelasko, Laureano Moro-Velazquez, and Najim Dehak. 2021. Segmental contrastive predictive coding for unsupervised word segmentation. *Interspeech*.

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, Alexander H. Liu, and James Glass. 2023. 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. 2021. 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*.

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*.

William A Falcon. 2019. Pytorch lightning. *GitHub*, 3.

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, 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*.

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*.

Shuo Liu, Adria Mallol-Ragolta, Emilia Parada-Cabaleiro, Kun Qian, Xin Jing, Alexander Kathan, Bin Hu, and Bjoern W Schuller. 2022. Audio self-supervised learning: A survey. *Patterns*, 3(12).

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*.

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*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*.

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. 2022a. Speech pre-training with acoustic piece. *Interspeech*.

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

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*.

Luke Strgar and David Harwath. 2022. Phoneme segmentation using self-supervised speech models. In *SLT*.

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*.

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*.

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*.

Yao-Yuan Yang, Moto Hira, Zhaoheng Ni, Artyom Astafurov, Caroline Chen, Christian Puhrsch, David Pollack, Dmitriy Genzel, Donny Greenberg, Edward Z Yang, et al. 2022. Torchaudio: Building blocks for audio and speech processing. In *ICASSP*.

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*.

A Implementation Details

A.1 Speech SSL Models

Each SSL model used in this paper has a 7-layer CNN feature extractor and a 12-layer transformer encoder (Vaswani et al., 2017), having roughly 95M parameters in total. All SSL models are pre-trained with 960 hours of unlabeled speech in the LibriSpeech corpus. **HuBERT** (Hsu et al., 2021a) is pre-trained in two iterations. In the first iteration, the encoder model learns to predict each masked frame's K-means cluster ID of MFCC features. The second iteration model has learning targets obtained by clustering hidden representations of the first iteration model. **WavLM** (Chen et al., 2022) follows the second iteration of HuBERT, but the training process involves a denoising task by adding random noises to the input to increase robustness. **ContentVec** (Qian et al., 2022) 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 of 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. **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.[8]

A.2 Spin

Since R-Spin is trained with 960 hours of 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 HuBERT + Spin$_{2048}$ model with the same data (originally 100 hours in Chang et al. (2023)). Each mini-batch before data perturbation has 2560 seconds of speech, equivalent to 32k frames after downsampling. 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).[9] This model takes four hours of training time on four RTX A6000 GPUs. Models trained with all 10k updates are used to generate pseudo-labels. In total, roughly 7.1k hours of unlabeled speech data are processed. Compared with the model in Chang et al. (2023), a similar performance is achieved on phoneme recognition.

A.3 R-Spin

Each mini-batch before perturbation has 384 seconds of speech, equivalent to 19.2k frames in each view. All input utterances are added with random noise data from MUSAN and CHiME-4 with an SNR between -10 and 10dB. The learning rate first linearly increases from 10^{-6} to 10^{-4} in the first 4k updates, then linearly decreases to 10^{-6} in the last 6k updates. λ in Eq. 3 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 in Appendix A.2, 8.2k hours of data are used during SSFT.

A.4 Low-budget Robust data2vec

We follow the implementation of Zhu et al. (2023) with fairseq (Ott et al., 2019).[10] We changed the training data from CHiME-4 to LibriSpeech for a fair comparison with our method. Because we found 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 similar to R-Spin. The rest of the hyperparameters remain the same since we found the original ones are sufficiently good. As shown in Table 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. Contrarily, R-Spin achieves superior results under the same budget, indicating that our approach is more data-efficient.

[8] https://github.com/s3prl/s3prl/tree/main/s3prl/upstream

[9] https://github.com/facebookresearch/swav

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

A.5 Phoneme Recognition

We follow the setup in SUPERB (Yang et al., 2021), which freezes each SSL model and uses a set of learnable weights to weighted-sum hidden features of all layers. The aggregated frame-wise features are fed into a lightweight linear prediction head to perform downstream tasks. Only the prediction head and the weighted-sum mechanism are fine-tuned with clean and labeled speech data to reveal the capabilities of SSL models. The LibriSpeech train-clean-100 and the test-other subsets are used as the training and evaluation datasets, respectively. The lightweight prediction head is a fully connected layer, projecting hidden representations directly to phoneme labels. Unlike the SUPERB setup, the learning rate is 5×10^{-4} (originally 10^{-2}), and the number of training updates is 30k (originally 100k). Moreover, the noise and perturbation data sources are listed as follows.

1. **Gaussian Noise**: The Gaussian noises are generated with the PyTorch Library.
2. **Background Noise**: The background noises are sampled from the MUSAN dataset.
3. **Reverberation**: We filter waveforms with real and simulated impulse responses in RIRS (Ko et al., 2017).[11] The scores for the real and simulated reverberation are averaged.

A.6 CHiME-4 ASR

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.[12]

A.7 Acoustic Pieces

We implemented the BPE algorithm in Python. The AP vocabulary sizes vs. the actual APs used are shown in Fig. 8. Since some merging operations in BPE replace previous vocabularies with new ones, the actual number of used vocabularies in the encoded LibriSpeech corpus is smaller than the APs. Thus, the number of classes in the linear prediction head for $\mathcal{L}_{\text{Aux}}$ is adjusted accordingly. E.g., the prediction head's output for R-Spin$_{32,\,40k}$ is 19857 instead of 40000.

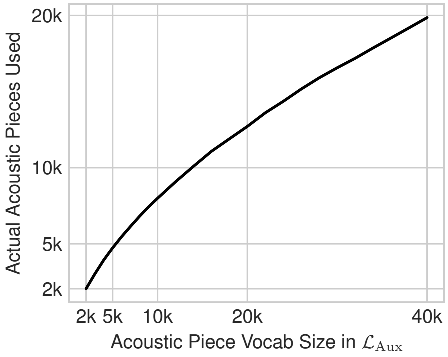


Figure 8: AP size vs. actual vocabularies used.

B Additional Experiments

B.1 Ablation Studies

We report more ablation studies in Table 3 to demonstrate the effectiveness of R-Spin.

B.1.1 Hyperparameters

Here, we change the value of λ in Eq. 3 to analyze the impact of the auxiliary loss. The differences of ASR WERs between different λ's are negligible. Considering the results in Table 2, 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.

B.1.2 Layer to Apply $\mathcal{L}_{\text{Aux}}$

After moving $\mathcal{L}_{\text{Aux}}$ 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.

B.1.3 Layer to Apply $\mathcal{L}_{\text{Spin}}$

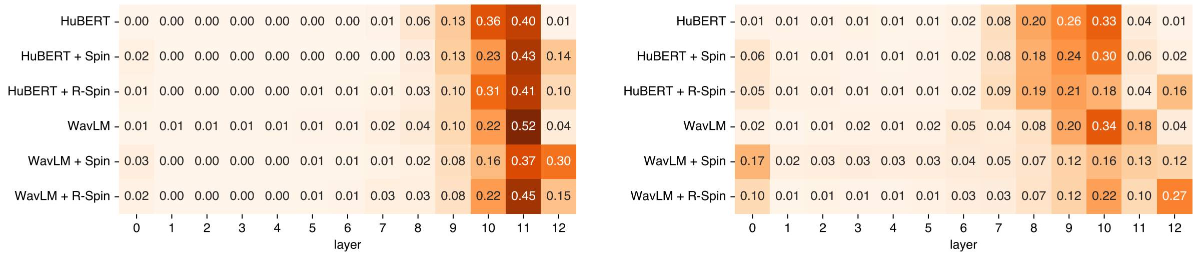
Moving the Spin objective function to lower layers, the ASR performance slightly degrades. With the results in the previous section, we conclude that a good strategy for applying the two proposed loss functions is adding both to the top layer.

B.1.4 Number of Fine-tuned Layers

The results of varying the number of fine-tuned layers indicate that the model cannot adapt to noisy scenarios by fine-tuning only the top layers. Thus, our approach is beneficial since we can now fine-tune the entire model in contrast to Spin.

[11] Released under Apache 2.0

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



(a) Phoneme Recognition

(b) Automatic Speech Recognition

Figure 9: Normalized weights of the weighted sum mechanism in the SUPERB PR and ASR.

	CHiME-4	
Method	Real	Sim
Spin$_{2048}$ (Chang et al., 2023)	52.1	46.6
R-Spin$_{32,\text{ BPE40k}}$	26.4	26.6
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

Table 3: CHiME-4 ASR results for additional ablation studies based on fine-tuned WavLM models.

B.1.5 Data

We found 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.

B.2 Importance of Hidden Representations

To better understand the importance of each layer's representation for phoneme and speech recognition, we plot the weights in the weighted sum mechanism in the models trained for these tasks. The weights are originally a probability distribution over all hidden layers (including the CNN feature extractor). The representations from each layer are then weighted and summed with these weights. However, the scale of the representations differs from layer to layer. Suppose the weight of a layer is very small, but the norm of the corresponding hidden vectors is large. In that case, the representations from that layer still contribute significantly to the downstream prediction head. Consequently, we normalize each weight by multiplying with the averaged L2 norm of the corresponding layer's representations, which can be written as

$$\hat{w}_l = w_l \cdot \mathbb{E}\left[\left\|\boldsymbol{h}^{(l)}\right\|_2\right],$$

where w_l is the unnormalized weight of layer l, $\boldsymbol{h}^{(l)}$ is the hidden representation produced by layer l, and $\mathbb{E}$ is the expectation over all samples from the LibriSpeech dev-clean and dev-other sets (Chang et al., 2022). Next, the new set of weights $\hat{w}_l$ is normalized to sum to one. The results are shown in Fig. 9.

Although the last layer of R-Spin has the least speaker and noise information, the second last layer offers the best phoneme representations. Still, the downstream prediction head for this task focuses more on the last layer of Spin and R-Spin than the models without SSFT, indicating the effectiveness of the perturbation-invariant objectives. In contrast, the best layers for ASR are shifted from the ninth and tenth layers toward the last layer when R-Spin is applied.

B.3 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 Fig. 10, R-Spin has significantly lower SID accuracy for the top layers, demonstrating the effect

Method	Precision↑	Recall↑	F1↑	OS→0	R-val↑
Baseline					
Oracle Uniform	56.49	62.99	59.56	11.50	63.47
Unsupervised					
CPC (Kreuk et al., 2020)	83.89	83.55	83.71		86.02
SCPC (Bhati et al., 2021)	84.63	86.04	85.33		87.44
HuBERT readout (Strgar and Harwath, 2022)	**90.98**	**88.48**	**89.71**		**90.98**
Spin Codebook					
HuBERT + Spin$_{128}$ (Chang et al., 2023)	64.76	87.87	74.56	35.69	64.25
HuBERT + Spin$_{256}$ (Chang et al., 2023)	61.71	90.84	73.49	47.22	56.02
HuBERT + Spin$_{512}$ (Chang et al., 2023)	60.78	95.46	74.27	57.07	49.60
HuBERT + Spin$_{1024}$ (Chang et al., 2023)	59.93	97.95	74.36	63.44	45.11
HuBERT + Spin$_{2048}$ (Chang et al., 2023)	58.58	**99.46**	73.73	69.77	40.26
HuBERT + Spin$_{2048}$ (for AP)	61.31	96.87	**75.09**	58.00	49.34
HuBERT + R-Spin$_{32,\ \text{AP40k}}$	64.73	71.47	67.93	**10.41**	71.05
WavLM + R-Spin$_{16,\ \text{AP40k}}$	60.73	68.22	64.26	12.32	67.36
WavLM + R-Spin$_{32,\ \text{AP40k}}$	**65.12**	73.76	69.17	13.28	**71.33**
WavLM + R-Spin$_{64,\ \text{AP40k}}$	63.02	73.63	67.91	16.83	69.08
WavLM + R-Spin$_{128,\ \text{AP40k}}$	61.44	72.42	66.48	17.88	67.49
WavLM + R-Spin$_{256,\ \text{AP40k}}$	60.80	78.61	68.57	29.28	63.95
WavLM + R-Spin$_{512,\ \text{AP40k}}$	59.66	82.94	69.40	39.01	58.89
WavLM + R-Spin$_{1024,\ \text{AP40k}}$	59.03	89.09	71.01	50.94	52.09
WavLM + R-Spin$_{2048,\ \text{AP40k}}$	58.47	94.19	72.15	61.08	45.67
Acoustic Pieces					
HuBERT + Spin$_{2048}$ AP5k	60.80	**71.64**	**65.77**	17.83	66.92
HuBERT + Spin$_{2048}$ AP10k	61.37	68.51	64.74	11.64	67.97
HuBERT + Spin$_{2048}$ AP20k	61.76	68.74	65.06	11.29	68.34
HuBERT + Spin$_{2048}$ AP40k	**62.10**	68.54	65.16	**10.37**	**68.65**

Table 4: Unsupervised phoneme segmentation on TIMIT test set. OS and R-val respectively denote the over-segmentation rate and R-value (Räsänen et al., 2009). Oracle uniform is a segmentation method by splitting speech into equal-length segments given the ground truth number of phoneme boundaries. Unknown results are left blank.

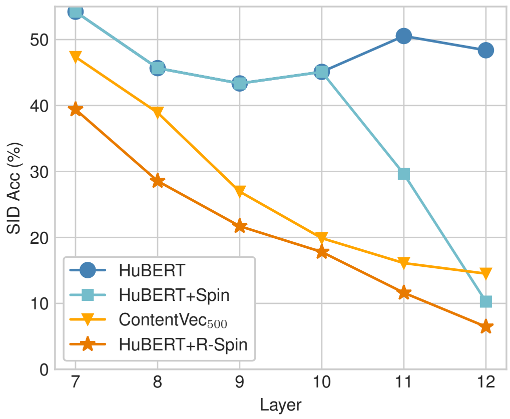


Figure 10: Layer-wise speaker identification accuracy.

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.

B.4 Unsupervised Phoneme Segmentation

This section inspects the phoneme segmentation capability of the proposed methods. As shown in Table 4, segmenting speech with Spin codebook or acoustic pieces is inferior to prior methods specifically designed for phoneme segmentation because no explicit constraints are added to encourage phoneme boundary detection. Still, some R-Spin discrete units like R-Spin$_{32,\ \text{AP40k}}$ surpass the oracle uniform baseline, indicating that the discrete unit boundaries are close to phoneme boundaries.

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., 2023)	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	△
R-Spin$_{32,\ \mathrm{AP40k}}$	HuBERT	10k	6.4	1.1k	2	16	△

Table 5: 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.

Task	Updates	Hours	GPU
(A.2) Spin	10k	4	A6000×2
(A.3) R-Spin	10k	8	A6000×2
(A.4) Robust data2vec	100k	22	A6000×2
(A.5) SUPERB PR	30k	10	2080 Ti
(A.6) SUPERB ASR	100k	20	A5000
(B.3) SUPERB SID	50k	4	A6000

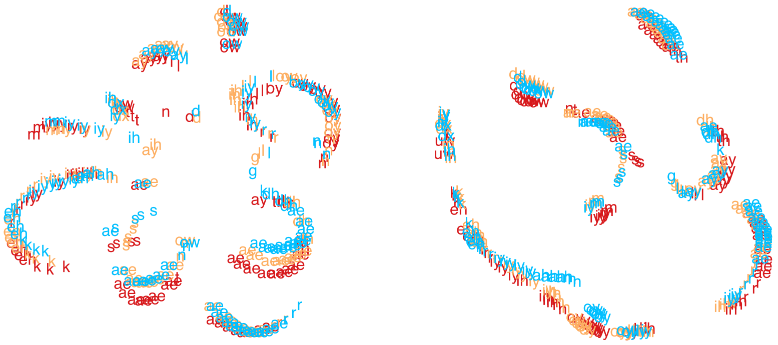
Table 6: Computation resources used in the experiments.

C Computation Resources

The costs of self-supervised pre-training and fine-tuning of various models are shown in Table 5. The required computation resources for each training task in this paper are listed in Table 6. Note that all results in this paper are obtained with a single run.

D t-SNE of Hidden Representations

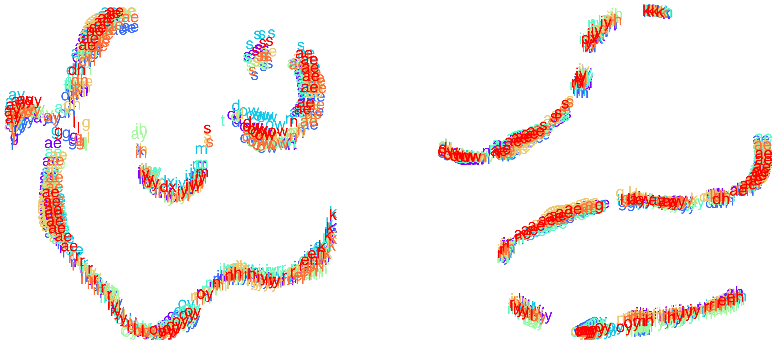
We plot more t-SNE visualization of hidden representations in Figs. 11, 12, 13, 14, 15, and 16 for reference.



(a) HuBERT
Layer 9

(b) HuBERT + R-Spin
Layer 12

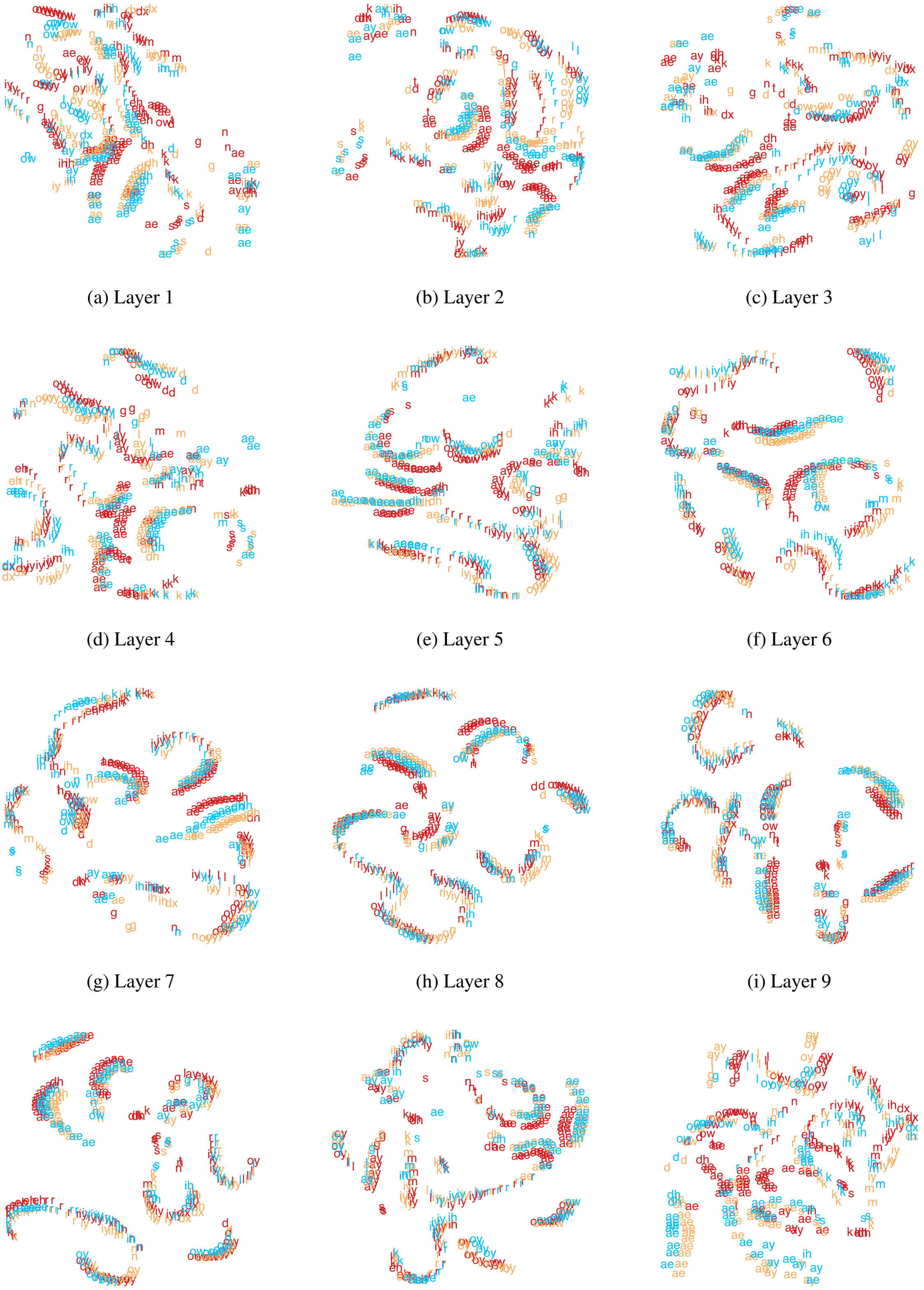
Figure 11: t-SNE visualization of three speakers with different English dialects (see Fig. 3 for details).



(a) HuBERT
Layer 9

(b) HuBERT + R-Spin
Layer 12

Figure 12: t-SNE visualization of eight speakers with different English dialects (see Fig. 3 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

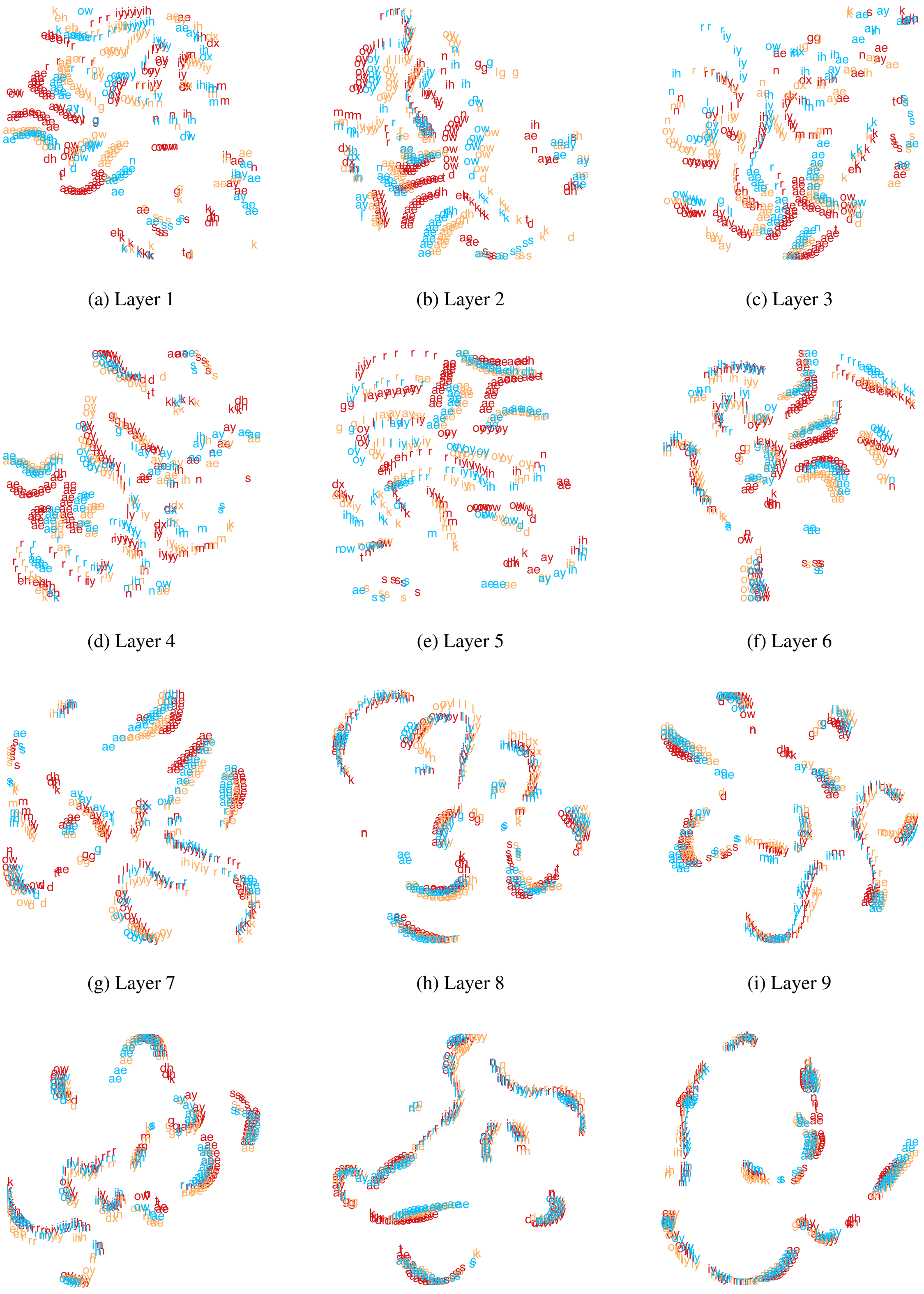
(i) Layer 9

(j) Layer 10

(k) Layer 11

(l) Layer 12

Figure 13: t-SNE visualization of HuBERT representations of the same utterance spoken by three speakers (see Fig. 3 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

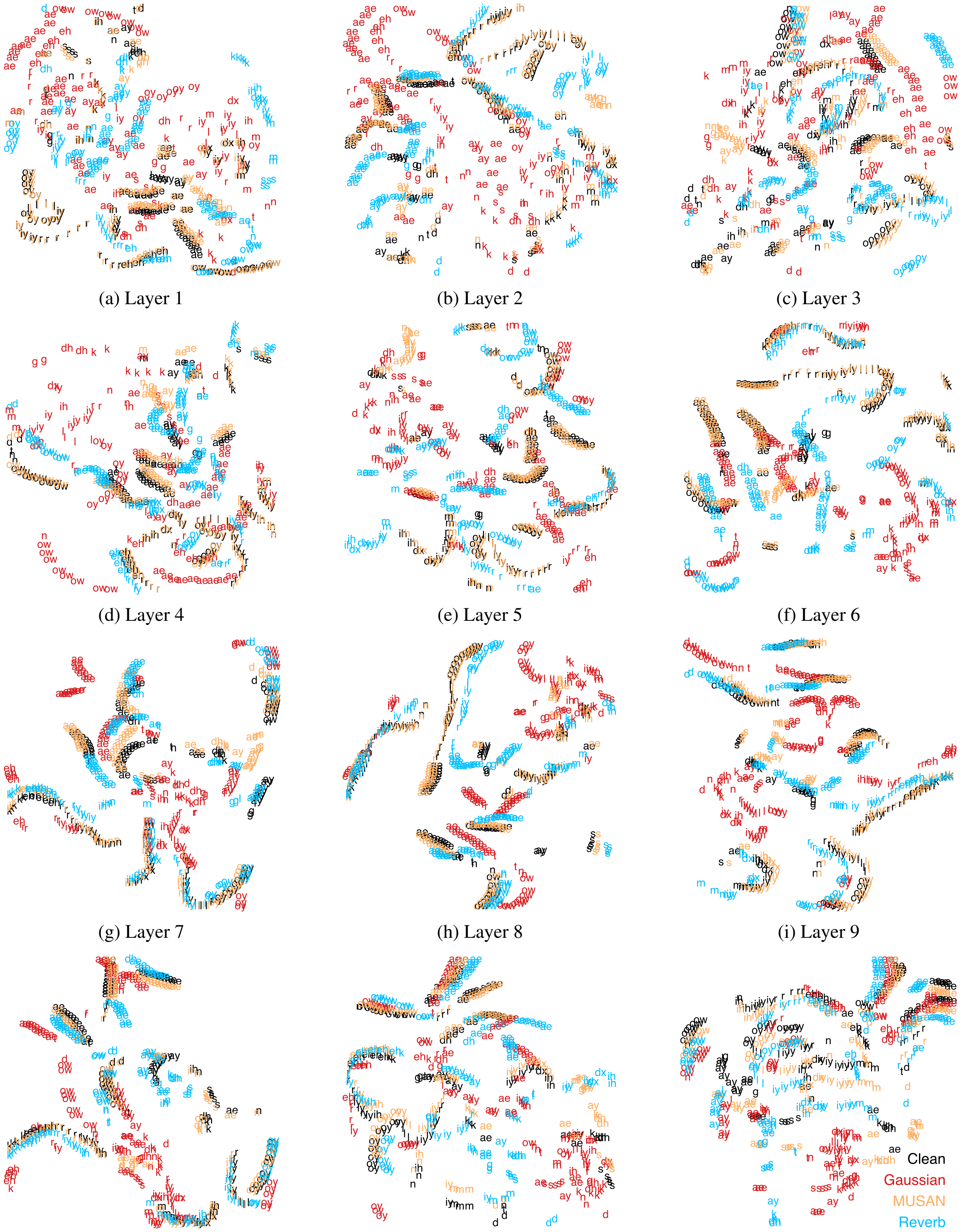
(i) Layer 9

(j) Layer 10

(k) Layer 11

(l) Layer 12

Figure 14: t-SNE visualization of HuBERT + R-Spin representations of the same utterance spoken by three speakers (see Fig. 3 for details).



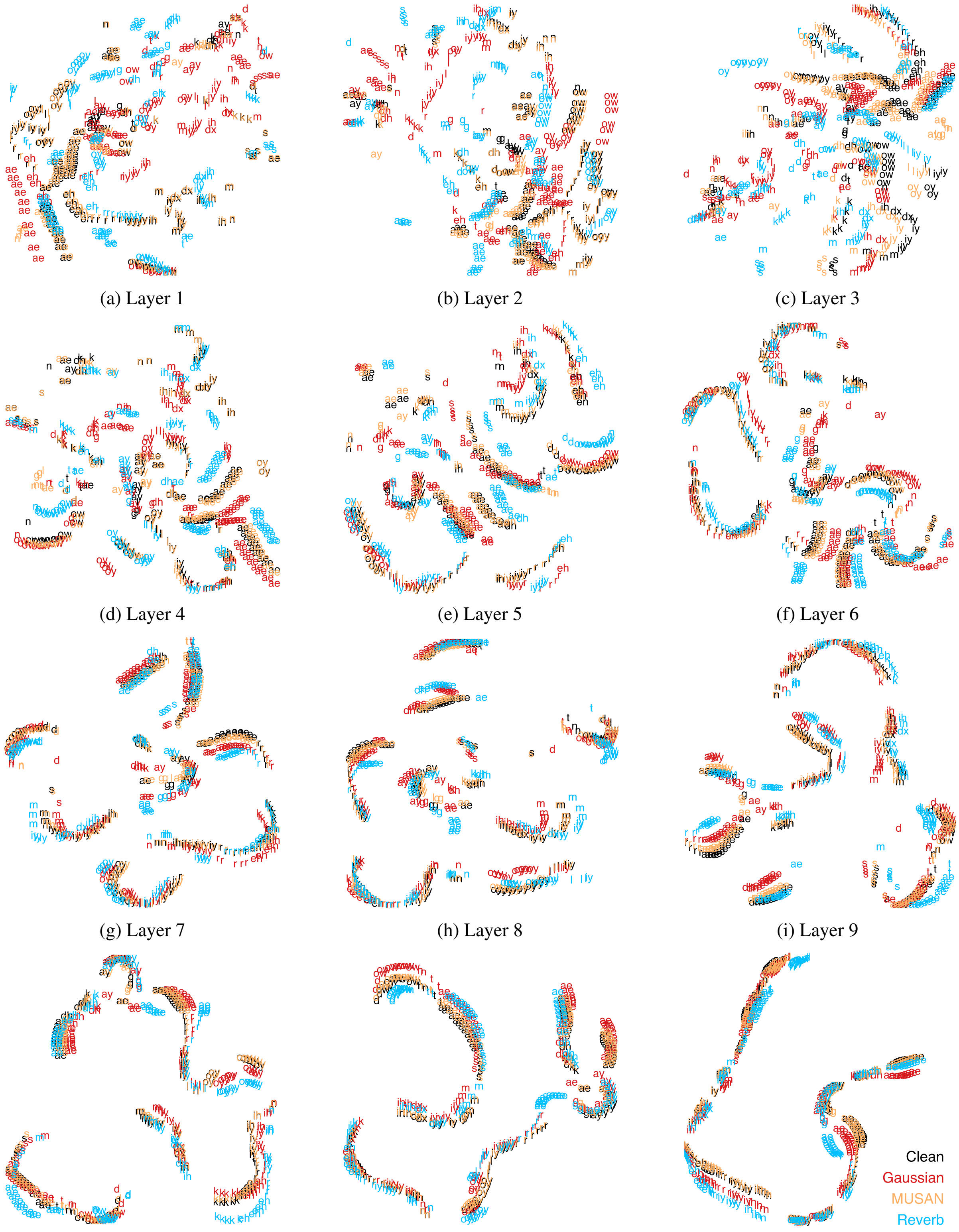
(a) Layer 1 (b) Layer 2 (c) Layer 3

(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 15: t-SNE visualization of HuBERT representations of the same utterance under different distortions (see Fig. 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 16: t-SNE visualization of HuBERT + R-Spin representations of the same utterance under different distortions (see Fig. 5 for details).