sensors

MDPI

Article

Cluster-Based Pairwise Contrastive Loss for Noise-Robust Speech Recognition

Geon Woo Lee [1] and Hong Kook Kim [1,2,3,*]

[1] AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea; geonwoo0801@gist.ac.kr
[2] School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
[3] AunionAI Co., Ltd., Gwangju 61005, Republic of Korea
* Correspondence: hongkook@gist.ac.kr

Abstract: This paper addresses a joint training approach applied to a pipeline comprising speech enhancement (SE) and automatic speech recognition (ASR) models, where an acoustic tokenizer is included in the pipeline to leverage the linguistic information from the ASR model to the SE model. The acoustic tokenizer takes the outputs of the ASR encoder and provides a pseudo-label through K-means clustering. To transfer the linguistic information, represented by pseudo-labels, from the acoustic tokenizer to the SE model, a cluster-based pairwise contrastive (CBPC) loss function is proposed, which is a self-supervised contrastive loss function, and combined with an information noise contrastive estimation (infoNCE) loss function. This combined loss function prevents the SE model from overfitting to outlier samples and represents the pronunciation variability in samples with the same pseudo-label. The effectiveness of the proposed CBPC loss function is evaluated on a noisy LibriSpeech dataset by measuring both the speech quality scores and the word error rate (WER). The experimental results reveal that the proposed joint training approach using the described CBPC loss function achieves a lower WER than the conventional joint training approaches. In addition, it is demonstrated that the speech quality scores of the SE model trained using the proposed training approach are higher than those of the standalone-SE model and SE models trained using conventional joint training approaches. An ablation study is also conducted to investigate the effects of different combinations of loss functions on the speech quality scores and WER. Here, it is revealed that the proposed CBPC loss function combined with infoNCE contributes to a reduced WER and an increase in most of the speech quality scores.

Keywords: joint training; noise-robust speech recognition; speech enhancement; contrastive loss; self-supervised learning; acoustic tokenizer

check for updates

Citation: Lee, G.W.; Kim, H.K. Cluster-Based Pairwise Contrastive Loss for Noise-Robust Speech Recognition. *Sensors* **2024**, *24*, 2573. https://doi.org/10.3390/s24082573

Academic Editor: Hsiao-Chun Wu

Received: 5 March 2024
Revised: 8 April 2024
Accepted: 16 April 2024
Published: 17 April 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The recent developments in neural network architecture and training approaches have facilitated continuous progress, which has manifested in enhanced capabilities in terms of automatic speech recognition (ASR) [1,2]. The current state-of-the-art ASR systems are approaching the levels of human recognition in terms of performance [3] and are ready to be deployed in applications such as voice-based information retrieval, chatbots, and automated transcription systems [4]. Moreover, there is an increasing interest in ASR operating in real-world scenarios for human–robot interactions within industry [5] and dialogue systems for social safety [6]. However, ASR models often experience performance degradation in distant microphone settings or under conditions with a low signal-to-noise ratio (SNR), due to the distortion of the speech signals by real-world ambient noise [7,8].

To improve ASR performance in noisy environments, multi-condition training (MCT) and noise-aware training (NAT) techniques have been studied, using noise as a condition [9,10]. However, unseen noise or unpredicted variations in noise can limit the ASR

performance, even when MCT and NAT techniques are applied. To address this limitation, speech enhancement (SE) models, employed as preprocessors for the ASR model, have been developed to suppress the noise and provide enhanced speech [11–14]. However, these SE models can introduce unintended artifacts into the enhanced speech signal, which can create an additional form of mismatching in ASR, degrading its performance [13,14]. To remedy this mismatching problem, speech signals enhanced by the SE model are added to the training dataset, meaning that the artifacts in the enhanced speech are trained in a multi-condition framework. Nevertheless, the improvements in the ASR performance obtained through the MCT approach are limited, because the artifacts in the enhanced speech remain unpredictable [9].

As an alternative, a pipeline integrating SE and ASR models has been explored in a joint training framework [9,15], where an SE model is used as the front-end of the ASR model. Although jointly training the pipeline leads to a better ASR performance than when using the MCT approach [9], difficulties can occur due to the conflicting gradients between the SE and ASR models, resulting in a convergence issue, which is referred to as a conflicting problem [16,17]. The conflicting gradients originate from the different gradient scales and directions between the SE and ASR models, which is caused by the differences in their neural architectures and loss functions, with different task goals. To solve the conflicting problem, several training approaches have been studied, including those based on asynchronous subregion optimization (ASO) [18,19], gradient surgery [20,21], and knowledge distillation (KD) [22–24].

Among these approaches, KD-based training achieves the best ASR performance by adjusting the gradient scales and directions of both the SE and the ASR models. In other words, the SE model is trained using a loss function that is defined in the middle layer of the ASR model. Therefore, the gradients of the SE model have more positive directions and closer scales than when the SE loss function is defined in the output layer of the ASR model. For example, the output feature vectors from an acoustic model, which is the initial part of the ASR model, are clustered, and the SE model is then trained using the cross-entropy (CE) loss to predict the centroid from the clustering [23]. Instead of directly using ASR, or a part of the ASR model, the loss function for the SE model is designed as the CE loss between the quantized vectors of clean and enhanced speech signals from the Wav2Vec 2.0 pretrained model [24,25]. However, the use of these targets in the CE loss could result in performance degradation, due to overfitting on hard examples [26,27]. To mitigate this problem, metric learning using the supervised contrastive (SupCon) loss [28], which is effective in feature representation and uses pairwise distances, can be employed for image classification [27]. However, applying the SupCon loss requires target labels, whereas the joint training proposed in this paper should be successful without target labels.

Therefore, this paper proposes the cluster-based pairwise contrastive (CBPC) loss, which is a self-supervised version of the SupCon loss, to train a pipeline comprising SE and ASR models in order to achieve an improvement in the ASR performance. First, the ASR model is trained using a training dataset and then frozen, as it will be used to transfer the linguistic information to the SE model. Subsequently, the output vectors of the ASR encoder are clustered through K-means clustering for the transfer process, where the cluster indices are referred to as pseudo-labels in this paper. Finally, the proposed CBPC loss function using the pseudo-labels is applied to the SE model training. The contributions of this paper can be summarized as follows:

- The CBPC loss function is proposed to leverage the linguistic information for the SE model by extending the SupCon loss to a self-supervised version. Replacing the CE loss, the proposed CBPC loss is used to train the pipeline with pseudo-labels. Accordingly, the proposed CBPC loss contributes to preventing the SE model from overfitting to the outlier samples in each cluster, resulting in an improved ASR performance compared to that of the CE loss.
- To further improve the ASR performance, the proposed CBPC loss is combined with the information noise contrastive estimation (infoNCE) loss [29] to train the SE model

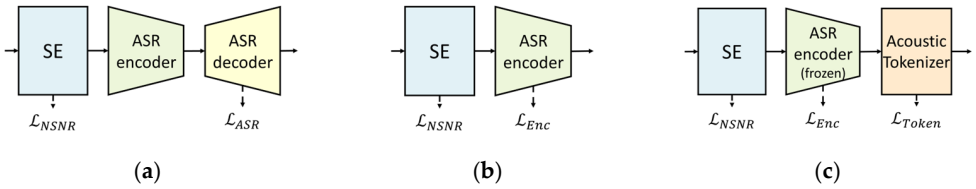
to represent the intra-cluster pronunciation variability. This is because the proposed CBPC loss function focuses on increasing the inter-cluster representation ability. Therefore, the combined loss also contributes to retaining the contextual information among the utterances with the same pseudo-label.

- An ablation study is conducted to examine the contributions of different combinations of loss functions to the SE and ASR performance.

The remainder of this paper is organized as follows: Section 2 presents a brief review of the methodologies of the joint training approaches applied to a pipeline comprising SE and ASR models. Section 3 proposes the CBPC loss function to train the SE model in the pipeline for an improved ASR performance. Subsequently, Section 4 explains the experimental setup and evaluation metrics. Then, Section 5 evaluates the performance of the SE and ASR models trained by the proposed loss function on the noisy LibriSpeech dataset by measuring both the speech quality scores and the word error rate (WER). In addition, the performance of the SE and ASR models trained using the proposed training approach is compared with those of models trained using conventional joint training approaches. Moreover, an ablation study is conducted to discuss the SE and ASR performances according to the different combinations of loss functions applied in the proposed training approach. Finally, Section 6 concludes the paper.

2. Pipeline Comprising SE and ASR for Noise-Robust ASR

A conventional pipeline comprising SE and ASR models for joint training is illustrated in Figure 1a [19–22]. Conventional joint training approaches combine all possible losses, such as the negative SNR (NSNR) loss ($\mathcal{L}_{SE}$) and ASR loss ($\mathcal{L}_{ASR}$), and they jointly or asynchronously train the pipeline [19–22]. However, the SE and ASR models have the following different goals: the prediction of clean speech and word sequences, respectively. Thus, there exist conflicting gradients, due to the different gradient directions of the two losses. Figure 1b shows a pipeline for the training of the SE model using a loss function calculated from the middle layer of the ASR model, i.e., the ASR encoder [24,25]. Compared with the pipeline depicted in Figure 1a, the gradients of the loss function are closer to those of the NSNR loss in the pipeline shown in Figure 1b.



(**a**) (**b**) (**c**)

Figure 1. Block diagrams of a pipeline comprising speech enhancement (SE) and automatic speech recognition (ASR) models: (**a**) a joint training approach using the information on the ASR decoder, (**b**) a joint training approach using the information on the middle layer (ASR encoder) of the ASR model, and (**c**) the proposed joint training approach using an acoustic tokenizer.

In addition, instead of directly using the middle layer of the ASR model, a specifically designed layer can be added to represent the outputs of the ASR model to train the SE model with less conflicting gradients. In this paper, an acoustic tokenizer is designed to leverage the linguistic information derived from the ASR encoder and transfer this information to the SE model [23]. Figure 1c illustrates this pipeline for the training of the SE model, which is achieved by concatenating an acoustic tokenizer to the ASR encoder. In contrast to the ASR encoder shown in Figure 1b, this acoustic tokenizer serves as a surrogate model capable of extracting linguistic information at frame-wise granularity.

To train the acoustic tokenizer, the output vector of the ASR encoder is used as an input feature for K-means clustering. Subsequently, the cluster indices are utilized as pseudo-labels to calculate the proposed CBPC loss. In fact, three different losses are computed, as follows: the NSNR loss, $\mathcal{L}_{NSNR}$; the ASR encoder loss, $\mathcal{L}_{Enc}$; and the acoustic tokenizer loss, $\mathcal{L}_{Tokenizer}$. Finally, the SE model is trained through backpropagation using these losses.

In this paper, the deep complex convolution neural network (DCCRN)-based SE model and conformer (encoder)–transducer (decoder)-based ASR model are employed. For a fair comparison, the architecture and hyperparameters of these models are set identically to those in [30,31], respectively.

3. Proposed Cluster-Based Pairwise Contrastive Loss Function for Joint Training

This section explains the training procedure of the SE model from the ASR encoder combined with the acoustic tokenizer. To distillate the linguistic information from the ASR encoder to the SE model, the conformer–transducer-based ASR model is first trained and then fixed. Subsequently, the acoustic tokenizer is trained by the proposed CBPC loss function using clean speech signals from the training dataset used for the ASR model training. Next, the SE model is trained using a set of clean utterances and their noisy version by applying the three losses described in Figure 1c. Next, the main components of the pipeline (the acoustic tokenizer and the loss functions) are described in detail.

3.1. Acoustic Tokenizer

Figure 2 depicts the training procedure of the acoustic tokenizer using clean speech utterances from the training dataset, where the ASR encoder is frozen, as mentioned previously. Given a dataset, $\boldsymbol{s} = \{\boldsymbol{s}_n\}_{n=1,\cdots,N}$, composed of clean speech utterances with a mini-batch size N, each utterance is sampled at 16 kHz and segmented into consecutive frames of 25 ms in length, with an overlap length of 16 ms, resulting in $\boldsymbol{s} = \{\boldsymbol{s}_{n,t}\}_{n=1,\cdots,N,t=1,\cdots,T}$. Here, $D_f = 400$ and T is the total number of frames in $\boldsymbol{s}$. Then, $\boldsymbol{s}$ is input into the ASR encoder, $Enc(\cdot)$, yielding the output sequence $\boldsymbol{v} = Enc(\boldsymbol{s}) = \{\boldsymbol{v}_{n,m}\}_{n=1,\cdots,N,m=1,\cdots,M}$, where $\boldsymbol{v}_{n,m}\left(\in \mathbb{R}^{D_e}\right)$ is the m-th latent vector with a dimension of $D_e(= 144)$. To speed up the training and inference, $Enc(\cdot)$ employs subsampling layers to reduce the frame rate by a factor of 4, meaning that $M = \lfloor T/4 \rfloor$.

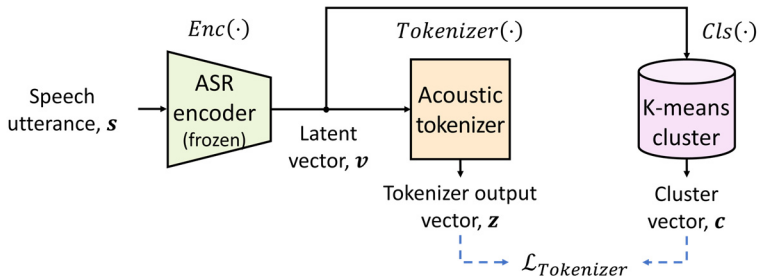


Figure 2. Block diagram of the training procedure of the acoustic tokenizer.

Each $\boldsymbol{v}_{n,m}$ is coded using a K-means clustering algorithm, $Cls(\cdot)$, resulting in a one-hot cluster vector, $\boldsymbol{c}_{n,m} = Cls(\boldsymbol{v}_{n,m}) \in \{0,1\}^{D_c}$, which is referred to as a pseudo-label of $\boldsymbol{v}_{n,m}$. Herein, the number of clusters, D_c, is set to 1.5 k, because the ASR model trained in this work includes the 1 k of linguistic units generated by the unigram language algorithm. In addition to these units, acoustic noise (such as breathing and coughing) is included in the training dataset. To obtain the K-means clusters, the mini-batch K-means algorithm in the scikit-learn [32] package is applied to the pool of $\boldsymbol{v}$, which is obtained from all of the clean speech utterances in the training dataset. Furthermore, the latent vectors corresponding to the silent frames are removed. These silent frames are detected by applying a voice activity detection technique to clean utterances with a 40-dB cutoff amplitude level [33].

Next, $\boldsymbol{v}$ is tokenized into $\boldsymbol{z}$, such as $\boldsymbol{z} = Tokenizer(\boldsymbol{v}) = \{\boldsymbol{z}_{n,m}\}_{n=1,\cdots,N,m=1,\cdots,M} \in \mathbb{R}^{D_c}$, where $Tokenizer(\cdot)$ is constructed via a one-time-distributed layer. To train the tokenizer,

the acoustic tokenizer loss function, $\mathcal{L}_{Tokenizer}(\boldsymbol{z}|\boldsymbol{c})$, is defined using the tokenizer output vectors, $\{\boldsymbol{z}_{n,m}\}$, and cluster vectors, $\{\boldsymbol{c}_{n,m}\}$, as follows:

$$\mathcal{L}_{Tokenizer}(\boldsymbol{z}|\boldsymbol{c}) = -\sum_{n=1}^{N}\sum_{m=1}^{M}\log\left(\frac{\exp(z_{n,m,i}/\tau_a)}{\sum_{j=1}^{D_c}\exp(z_{n,m,j}/\tau_a)}\right) \tag{1}$$

where $z_{n,m,i}$ is the i-th element of $\boldsymbol{z}_{n,m}$ at which $c_{n,m,\,i}$ is 1 and $\tau_a(=0.5)$ denotes the temperature parameter.

3.2. Contrastive Learning for Acoustic Tokenizer

The use of contrastive loss in metric learning facilitates the attraction of positive/negative pairs, and it has demonstrated notable performance improvements over CE loss across various domains [28,34]. The rationale behind this result is that, while the CE loss might be overfitted to hard samples, contrastive loss, which is grounded on the distance between the positive and negative pairs, mitigates the optimization issue associated with specific samples [26,27]. In addition to metric learning, contrastive loss has gained prominence in the realm of self-supervised learning, exhibiting an exemplary performance in the speech domain, such as contrastive predictive coding (CPC) [29] and Wav2vec 2.0 [25].

However, an inherent challenge in feature representation learning through contrastive loss is the potential convergence to a trivial constant solution [35,36]. To address this issue, the spread loss [37] leverages the supervised contrastive (SupCon) loss [28] with the information noise contrastive estimation (infoNCE) loss, which incorporates a regularization term to prevent the representation from collapsing to a singular point [29]. In essence, while the SupCon loss encourages attraction within the same class, the infoNCE loss induces repulsion, effectively resolving the collapsed representation dilemma and ensuring successful feature representation. However, to apply the SupCon loss in this joint training approach, target labels are required, because the SupCon loss is designed in a supervised learning framework.

Therefore, this paper proposes CBPC loss, which is a self-supervised version of the SupCon loss, by using a clustering technique. The training procedure of the acoustic tokenizer using the proposed CBPC loss for noise-robust ASR is illustrated in Figure 3. First, a latent vector, $\boldsymbol{v}_{n,m}$, at the n-th mini-batch and m-th frame, is clustered into $\boldsymbol{c}_{n,m}$, which is then used as a pseudo-label for $\boldsymbol{v}_{n,m}$, as described in Section 3.1. Next, $\boldsymbol{v}_{n,m}$ is tokenized into $\boldsymbol{z}_{n,m}$, and a set of the positive pairs for $\boldsymbol{z}_{n,m}$ is defined as a set with the same pseudo-label, defined as $P(\boldsymbol{z}_{n,m}|\boldsymbol{c}) = \{\boldsymbol{z}_{n,l} \mid Cls(\boldsymbol{v}_{n,l}) = \boldsymbol{c}_{n,m},\ l = 1,\ldots,M\}$. Otherwise, a set of negative pairs for $\boldsymbol{z}_{n,m}$ is defined as $N(\boldsymbol{z}_{n,m}|\boldsymbol{c}) = \{\boldsymbol{z}_{n,l} \mid Cls(\boldsymbol{v}_{n,l}) \neq \boldsymbol{c}_{n,m},\ l = 1,\ldots,M\}$. Then, the proposed CBPC loss function for the acoustic tokenizer conditioned by $\boldsymbol{c}$, $\mathcal{L}_{CBPC}(\boldsymbol{z}|\boldsymbol{c})$, is defined as follows:

$$\mathcal{L}_{CBPC}(\boldsymbol{z}|\boldsymbol{c}) = \sum_{n=1}^{N}\sum_{m=1}^{M}\frac{-1}{|P(\boldsymbol{z}_{n,m}|\boldsymbol{c})|}\sum_{\boldsymbol{z}^{+}_{n,m}\in P(\boldsymbol{z}_{n,m}|\boldsymbol{c})}\log\frac{\exp(\boldsymbol{z}_{n,m}\cdot\boldsymbol{z}^{+}_{n,m}/\tau_c)}{\sum_{\boldsymbol{z}_{k,l}\in\{P(\boldsymbol{z}_{n,m}|\boldsymbol{c}),N(\boldsymbol{z}_{n,m}|\boldsymbol{c})\setminus\{\boldsymbol{z}^{+}_{n,m}\}\}}\exp(\boldsymbol{z}_{n,m}\cdot\boldsymbol{z}_{k,l}/\tau_c)} \tag{2}$$

where $\tau_c(=0.5)$ denotes the temperature of the proposed CBPC loss function and $|P(\boldsymbol{z}_{n,m}|\boldsymbol{c})|$ is its cardinality. As shown in Equation (2), the CBPC loss function aims to maximize the distance between the clusters.

By only applying the CBPC loss in Equation (2), the tokenizer output vectors, $\boldsymbol{z}$, can be overly drawn toward the centroid, which can result in the loss of contextual information [26,27]. Such a phenomenon could subsequently result in a degraded ASR performance. To remedy this issue, the infoNCE loss [29] is incorporated here to ensure repulsion within the intra-cluster, where all of the samples in the same cluster, except itself, are treated as negative samples. Specifically, the infoNCE loss function can be defined as follows [29]:

$$\mathcal{L}_{infoNCE}(\boldsymbol{z}|\boldsymbol{c}) = -\sum_{n=1}^{N}\sum_{m=1}^{M}\log\frac{\exp(\boldsymbol{z}_{n,m}\cdot\boldsymbol{z}_{n,m}/\tau_c)}{\sum_{\boldsymbol{z}_{k,l}\in P(\boldsymbol{z}_{n,m}|\boldsymbol{c})}\exp(\boldsymbol{z}_{n,m}\cdot\boldsymbol{z}_{k,l}/\tau_c)}. \tag{3}$$

Subsequently, the final contrastive acoustic tokenizer loss function used to train the acoustic tokenizer combines the acoustic tokenizer loss in Equation (1) and CBPC loss in Equation (2) with the infoNCE loss in Equation (3), which is defined as follows:

$$\mathcal{L}_{Con-Tokenizer}(\boldsymbol{z}|\boldsymbol{c}) = \theta \cdot \mathcal{L}_{Tokenizer}(\boldsymbol{z}|\boldsymbol{c}) + (1-\theta)\Big(\delta \cdot \mathcal{L}_{CBPC}(\boldsymbol{z}|\boldsymbol{c}) + (1-\delta)\cdot\mathcal{L}_{infoNCE}(\boldsymbol{z}|\boldsymbol{c})\Big) \quad (4)$$

where θ controls the weighting between the acoustic tokenizer and the contrastive losses and δ gives different weights to the proposed CBPC loss and infoNCE loss. The weights, θ and δ, in Equation (4), are determined according to the following procedure: First, θ is fixed at 0.5 and δ is varied in steps of 0.1 from 0.1 to 1.0. The acoustic tokenizer is trained at each step using Equation (4), where the validation dataset in the LibriSpeech dataset is used. Then, the classification accuracy of the trained acoustic tokenizer is calculated by comparing $\boldsymbol{z}_{n,m}$ and $\boldsymbol{c}_{n,m}$, and the δ with the highest accuracy is selected. This process is repeated by varying θ with a fixed δ to select the best value of θ. As a result, θ and δ are set to 0.7 and 0.9, respectively, and the model parameters of the acoustic tokenizer trained with these weights are fixed to train the SE model.

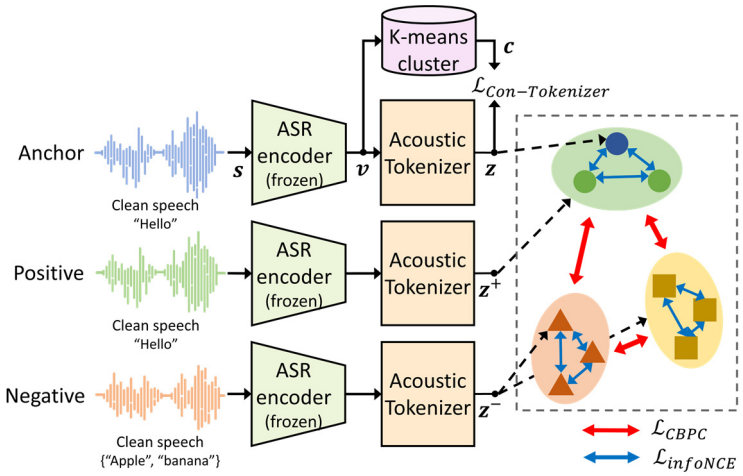


Figure 3. Training procedure of the acoustic tokenizer using the proposed cluster-based pairwise contrastive (CBPC) loss function.

3.3. SE Model Training

Figure 4 displays the training procedure of the SE model using three different loss functions, where the ASR encoder and the acoustic tokenizer are fixed, as mentioned previously. To train the SE model using the information on the ASR encoder through contrastive learning, noisy utterances are generated by mixing a noise signal, $\boldsymbol{d}$, with $\boldsymbol{s}$, such that $\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{d} = \{\boldsymbol{x}_{n,t}\}_{n=1,\cdots,N,t=1,\cdots,T}$. As shown in the figure, $\boldsymbol{x}$ is passed into the SE model, which is randomly initialized, to predict the estimated clean utterances, $\widetilde{\boldsymbol{s}}$. The clean and estimated clean utterances are then input into the ASR encoder to obtain the following two sequences of latent vectors: $\boldsymbol{v} = Enc(\boldsymbol{s})$ and $\widetilde{\boldsymbol{v}} = Enc\left(\widetilde{\boldsymbol{s}}\right)$. Then, the latent vectors are further encoded using the tokenizer, such as $\boldsymbol{z} = Tokenizer(\boldsymbol{v})$ and $\widetilde{\boldsymbol{z}} = Tokenizer\left(\widetilde{\boldsymbol{v}}\right)$. Simultaneously, $\boldsymbol{v}$ is clustered as $\boldsymbol{c}_{n,m} = Cls(\boldsymbol{v}_{n,m})$, as mentioned in Section 3.1.

There are three loss functions in this training approach. The speech quality loss, $\mathcal{L}_{NSNR}(\cdot,\cdot)$, is first computed for a given pair of clean and noisy utterances, $\boldsymbol{s}$ and $\widetilde{\boldsymbol{s}}$, which is defined as follows:

$$\mathcal{L}_{NSNR}\left(\boldsymbol{s},\widetilde{\boldsymbol{s}}\right) = -\frac{1}{N}10\log_{10}\left(\frac{\|\boldsymbol{s}\|^2}{\left\|\boldsymbol{s}-\widetilde{\boldsymbol{s}}\right\|^2}\right). \quad (5)$$

The second loss function is the ASR encoder loss, $\mathcal{L}_{Enc}(\cdot,\cdot)$, which is defined as the L_2-norm between two latent vector sequences, $\boldsymbol{v}$ and $\widetilde{\boldsymbol{v}}$, from $\boldsymbol{s}$ and $\widetilde{\boldsymbol{s}}$, as follows:

$$\mathcal{L}_{Enc}\left(\boldsymbol{v},\,\widetilde{\boldsymbol{v}}\right)=\frac{1}{N}\left\|\boldsymbol{v}-\widetilde{\boldsymbol{v}}\right\|^2. \tag{6}$$

Finally, the contrastive acoustic tokenizer loss function is computed using the tokenizer output vectors of $\widetilde{\boldsymbol{s}}$ and $\widetilde{\boldsymbol{z}}$, as well as the cluster vectors, $\boldsymbol{c}$, with positive/negative pairs in the tokenizer output vector domain, $P(\boldsymbol{z}_{n,m}|\boldsymbol{c})$ and $N(\boldsymbol{z}_{n,m}|\boldsymbol{c})$, of $\boldsymbol{s}$. This loss function should be conditioned by $\boldsymbol{z}$ and $\boldsymbol{c}$, as shown in the following equation:

$$\begin{aligned}&\mathcal{L}_{Con-Tokenizer}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{z},\boldsymbol{c}\right)\\&=\theta\cdot\mathcal{L}_{Tokenizer}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{c}\right)+(1-\theta)\left(\delta\cdot\mathcal{L}_{CBPC}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{z},\boldsymbol{c}\right)+(1-\delta)\cdot\mathcal{L}_{infoNCE}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{z},\boldsymbol{c}\right)\right)\end{aligned} \tag{7}$$

where θ and δ are set identically to those in Equation (4). In Equation (7), $\mathcal{L}_{Tokenizer}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{c}\right)$ is a noisy version of Equation (1), rewritten as follows:

$$\mathcal{L}_{Tokenizer}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{c}\right)=-\sum_{n=1}^{N}\sum_{m=1}^{M}\log\left(\frac{\exp\left(\widetilde{\boldsymbol{z}}_{n,m,i}/\tau_a\right)}{\sum_{j=1}^{D_c}\exp\left(\widetilde{\boldsymbol{z}}_{n,m,j}/\tau_a\right)}\right). \tag{8}$$

In addition,

$$\mathcal{L}_{CBPC}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{z},\boldsymbol{c}\right)=\sum_{n=1}^{N}\sum_{m=1}^{M}\frac{-1}{|P(\boldsymbol{z}_{n,m}|\boldsymbol{c})|}\sum_{\boldsymbol{z}^{+}_{n,m}\in P(\boldsymbol{z}_{n,m}|\boldsymbol{c})}\log\frac{\exp\left(\widetilde{\boldsymbol{z}}_{n,m}\cdot\boldsymbol{z}^{+}_{n,m}/\tau_c\right)}{\sum_{\boldsymbol{z}_{k,l}\in\{P(\boldsymbol{z}_{n,m}|\boldsymbol{c}),N(\boldsymbol{z}_{n,m}|\boldsymbol{c})\setminus\{\boldsymbol{z}^{+}_{n,m}\}\}}\exp\left(\widetilde{\boldsymbol{z}}_{n,m}\cdot\boldsymbol{z}_{k,l}/\tau_c\right)} \tag{9}$$

and

$$\mathcal{L}_{infoNCE}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{z},\boldsymbol{c}\right)=-\sum_{n=1}^{N}\sum_{m=1}^{M}\log\frac{\exp\left(\widetilde{\boldsymbol{z}}_{n,m}\cdot\boldsymbol{z}_{n,m}/\tau_c\right)}{\sum_{\boldsymbol{z}_{k,l}\in P(\boldsymbol{z}_{n,m}|\boldsymbol{c})}\exp\left(\widetilde{\boldsymbol{z}}_{m,n}\cdot\boldsymbol{z}_{k,l}/\tau_c\right)}. \tag{10}$$

Finally, the joint loss function for SE training is obtained by combining all of the losses in Equations (5)–(7), denoted as follows:

$$\mathcal{L}=\alpha\cdot\mathcal{L}_{NSNR}\left(\boldsymbol{s},\widetilde{\boldsymbol{s}}\right)+\beta\cdot\mathcal{L}_{Enc}\left(\boldsymbol{v},\widetilde{\boldsymbol{v}}\right)+\gamma\cdot\mathcal{L}_{Con-Tokenizer}\left(\widetilde{\boldsymbol{z}}\middle|\boldsymbol{z},\boldsymbol{c}\right). \tag{11}$$

where α, β, and γ are the weights of the NSNR, ASR encoder loss, and tokenizer loss, respectively. The three weights are determined by following the procedure described in [22]. Consequently, α, β, and γ are set as 0.3, 0.7, and 1.0, respectively.

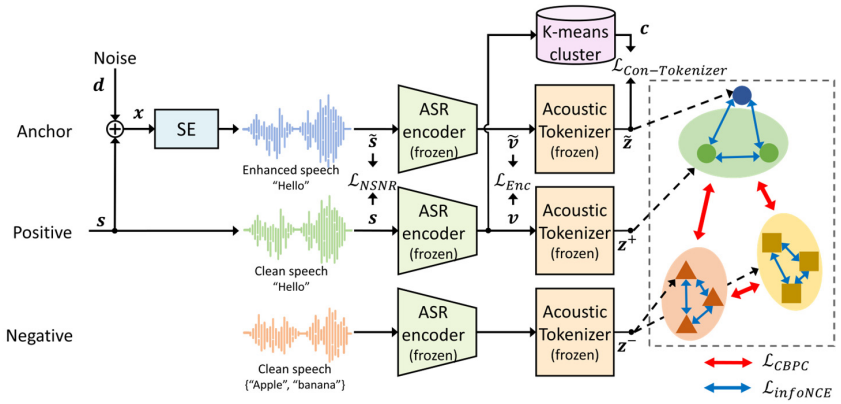


Figure 4. Block diagrams of the proposed joint training approach for the SE model using a negative signal-to-noise ratio (NSNR), an ASR encoder, and the contrastive acoustic tokenizer loss function.

4. Experimental Setup

In this section, the performance of the proposed training approach was evaluated for noise-robust ASR, and it was then compared with the performance of MCT and conventional joint training approaches, including asynchronous subregion optimization (ASO)-based joint optimization [18], gradients-surgery (Grad)-based joint optimization [19], and acoustic tokenizer (Token)-based joint optimization [22]. Here, the ASO-based joint optimization approach was first used to train a pipeline with the SE and ASR encoder losses, and it was then further trained with the combination of the SE and ASR losses. Meanwhile, the Grad-based joint optimization approach was used to train a pipeline with projection errors from the SE gradients to the ASR ones. The Token-based joint optimization approach was implemented similarly to the proposed training approach, but it differed in terms of the loss functions. In particular, the Token-based approach used the cross-entropy loss to train the SE model in the pipeline, while the proposed training approach used contrastive losses such as the CBPC and infoNCE losses. The ASR and SE performance was measured on a simulated noisy dataset mixed with the LibriSpeech [38] and deep noise suppression (DNS) challenge datasets [39].

4.1. Dataset

A total of 281,241 clean speech utterances spoken by 2338 speakers were excerpted from train-clean-100, train-clean-360, and train-other-500 in the LibriSpeech dataset to obtain the clean speech training dataset. Here, the total length of all of the utterances was 960 h, and the average length per utterance was measured as 12.3 s. To simulate various acoustic noise conditions, a noisy training dataset was constructed using the noise dataset released from the DNS dataset. This noise dataset was collected from AudioSet, Freesound, and the Diverse Environments Multi-Channel Acoustic Noise Dataset (DEMAND), which included approximately 150 different noise types. The noisy speech utterances were obtained by mixing each clean speech utterance with noise data that were randomly selected from the noise dataset. To simulate different noise levels, the mixing ratio between the clean speech and noise was controlled, ensuring that the SNR ranged from -5 to 5 dB.

To validate the model training, the dev-clean and dev-other datasets were used as validation datasets. The dev-clean and dev-other datasets were composed of 2703 and 2864 clean speech utterances, respectively. To create the noisy version of the validation dataset, DNS noise was randomly added to each of the clean utterances under an SNR in the range of -5 to 5 dB. Next, to evaluate the ASR and SE performance of the various training approaches, including the proposed training approach, the test-clean and test-other datasets in the LibriSpeech dataset were used, which contained 2620 and 2939 clean speech utterances, respectively. Similar to the validation dataset described above, the noisy version of the evaluation dataset was obtained by adding DNS noise to each of the clean utterances.

4.2. Hyperparameters

4.2.1. Model Architecture

The architecture and hyperparameters of the DCCRN-based SE model were set identically to those used in [30]. In other words, the input feature was a complex spectrum obtained by applying a 512-point short-time Fourier transform to each noisy speech frame with a frame size of 25 ms and a frame hop size of 16 ms. The number of complex convolutional blocks for both the encoder and decoder was set to six, and these six convolutional blocks had varying numbers of channels, such as [32, 64, 128, 128, 256, 256], with a kernel size of 5×2 and a stride size of 2×1.

The architecture and hyperparameters of the conformer–transducer-based ASR model were also set identically to those of the conformer(s) described in [31]. As an input feature of the ASR model, an 80-dimensional log-mel spectrum was extracted. The ASR encoder was composed of 16 conformer blocks, and each conformer block extracted a latent vector with a dimension of 144 (D_e). As a target feature, the linguistic units for transcribing the

target texts consisted of a special token and 1k linguistic units generated by the unigram language model algorithm [40].

4.2.2. Training Details and Implementation

The ASR and SE models were trained using the noisy LibriSpeech training dataset, while the tokenizer was trained using the clean LibriSpeech training dataset. In this study, the Adam optimizer was applied to all of the model training approaches. To adjust the learning rate, the warmup learning rate scheduler technique with 40,000 warmup steps was applied to train the conformer–transducer-based ASR model, while a plateau learning rate scheduler with patience of 5 and a factor of 0.5 was utilized for the acoustic tokenizer and SE model training. In particular, the SpecAugment technique was employed for the ASR model training. All of the experiments were implemented in Python 3.8.10 using TensorFlow 2.11.0 [41] conducted on an Intel(R) Xeon(R) Gold 6226R workstation using four Nvidia RTX 3090s.

5. Performance Evaluation and Discussion

5.1. Results and Discussion of ASR Performance

The ASR performance of each of the training approaches was evaluated by measuring the WER on both the validation and the test datasets. The WER of the ASR model trained using the proposed training approach was then compared to those of six different approaches, as follows: (1) an ASR model trained via the MCT using the clean and noisy training datasets (denoted as MCT-noisy); (2) an SE model trained on the clean and noisy speech training datasets, wherein the enhanced signal was subsequently fed into the MCT-noisy ASR model (denoted as MCT-noisy + standalone-SE); (3) an ASR model trained by the MCT using the clean, noisy, and enhanced data from the standalone-SE datasets (denoted as MCT-all); (4) a combination of the SE and ASR models trained by conventional joint optimization (denoted as Joint-Straight) [9]; (5) a pipeline trained by ASO-based joint optimization (denoted as Joint-ASO) [18]; (6) a pipeline trained by Grad-based joint optimization (denoted as Joint-Grad) [20]; and (7) a pipeline trained by Token-based joint optimization (denoted as Joint-Token) [22].

Table 1 compares the average WERs of the ASR models trained using different training approaches, where the performance evaluation was carried out on four different noisy datasets constructed by mixing noise with the dev-clean, dev-other, test-clean, and test-other datasets. First, the conventional training approaches were compared. As shown in the table, the average WER of the MCT-noisy + standalone-SE model was increased because the standalone-SE model unexpectedly distorted the speech. However, the standalone-SE model improved the speech quality, which will be discussed in the next subsection. Upon adding enhanced speech to the training data, the WERs of MCT-all were marginally lower than those of MCT-noisy for all datasets. This was because the mismatching issue between the training and evaluation was somewhat mitigated.

Table 1. Comparison of the average word error rates (WERs) (%) of the ASR models according to different training approaches on the noisy LibriSpeech dataset.

Training Approach	Dev		Test		Average
	Clean	Other	Clean	Other	
MCT-noisy	22.77	23.18	22.95	23.41	23.08
+standalone-SE	28.94	29.03	29.38	29.14	29.12
MCT-all	22.61	22.74	22.68	22.82	22.71
Joint-Straight	22.39	22.51	22.40	22.64	22.49
Joint-ASO	22.31	22.42	22.37	22.58	22.42
Joint-Grad	20.11	20.89	20.88	20.98	20.72
Joint-Token	19.86	20.40	20.28	20.67	20.30
Proposed	**19.14**	**19.85**	**19.48**	**19.63**	**19.53**

Second, the average WERs of the ASR models were compared according to the different joint training approaches. Note that the training hyperparameters for Joint-Straight, Joint-ASO, and Joint-Grad were set identically to those in the corresponding papers. As shown in the table, Joint-Token exhibited the lowest WERs among all of the joint training approaches. Furthermore, the average WER of the ASR model trained by the joint training approach using the proposed contrastive loss was relatively reduced by 15.39% compared to that of MCT-noisy. Moreover, the proposed joint training approach achieved a relative WER reduction of 3.82%, compared to Joint-Token, which had the lowest WER among the conventional joint training approaches.

5.2. Results and Discussion of SE Performance

The speech quality scores of the various training approaches were compared and examined by measuring the perceptual evaluation of speech quality (PESQ) [42], short-time objective intelligibility (STOI) [43], and the following three mean opinion scores: signal distortion (CSIG), background noise intrusiveness (CBAK), and overall signal quality (COVL) [44]. Table 2 compares the average PESQ, STOI, CSIG, CBAK, and COVL scores of the SE models evaluated on the test-clean dataset, according to the different training approaches.

Table 2. Comparison of the average perceptual evaluation of speech quality (PESQ), short-time objective intelligibility (STOI), and mean opinion scores, such as signal distortion (CSIG), background noise intrusiveness (CBAK), and overall signal quality (COVL), of the SE models, according to the different training approaches on the noisy LibriSpeech dataset.

Training Approach	PESQ	STOI	CSIG	CBAK	COVL
Noisy	1.7256	0.6967	1.8457	1.1615	1.3937
+standalone-SE	2.6512	0.8277	2.9671	2.5482	2.3410
Joint-Straight	2.4872	0.7504	2.8081	2.1950	2.1725
Joint-ASO	2.5871	0.7888	2.8213	2.3119	2.2647
Joint-Grad	2.5531	0.7719	2.8001	2.2964	2.2581
Joint-Token	2.6653	**0.8311**	3.1204	**2.5684**	**2.4509**
Proposed	**2.6802**	**0.8311**	**3.1275**	2.5653	2.4507

As shown in the table, the standalone-SE model significantly improved the speech quality compared to noisy speech. Next, the SE models were excerpted from the pipeline trained using the different training approaches. It is shown in the third to fifth rows of the table that the SE models trained by the Joint-Straight, Joint-ASO, and Joint-Grad approaches achieved better speech quality than that of the noisy speech. However, all of the quality scores were lower than those of the standalone-SE model. The reason for these degraded results was that the conventional training approaches focused on improving the ASR performance rather than the SE performance.

In contrast, the SE model trained by Joint-Token and the proposed training approach achieved higher speech quality scores compared to the other three SE models trained by the conventional joint training approaches. Moreover, they were even better than those of the standalone-SE model. This was because the proposed training approach attempted to enhance the speech for better speech recognition, which resulted in better speech quality. Specifically, the proposed training approach significantly improved the speech intelligibility, as measured by PESQ, compared to Joint-Token, while the other speech quality scores of the proposed training approach were comparable to those of Joint-Token.

5.3. Discussion of Performance Contribution According to Different Losses

This ablation study examined the effectiveness of the proposed training approach according to different combinations of losses on the average WERs, as shown in Table 3.

Note that the ASR model in the first row of the table corresponds to the ASR model trained by Joint-Grad, which showed the lowest WER among all of the conventional approaches, except for Joint-Token, as shown in Table 1. The second to the last rows present the WERs based on the loss functions used in the proposed training approach. The results in the second row present the effects of applying the combination of the SE and ASR encoder losses, $\mathcal{L}_{NSNR}$ and $\mathcal{L}_{Enc}$, on the ASR performance. Unfortunately, this loss combination increased the WER compared to Joint-Grad. In contrast, it is shown in the third row in the table that the tokenizer loss, $\mathcal{L}_{Tokenizer}$, contributed to a marked reduction in the average WER, which corresponded to the Joint-Token training approach. Next, the proposed CBPC loss, $\mathcal{L}_{CBPC}$, was combined with the previous three losses, resulting in lower WERs than those found in the case without combining $\mathcal{L}_{CBPC}$, as shown in the fourth row of the table. Finally, all of the losses were combined to train the SE model. As shown in the last row of the table, this combination provided the lowest WER of all of the different loss combinations. This was because $\mathcal{L}_{CBPC}$ mitigated the overfitting issue on the hard samples across the inter-cluster and $\mathcal{L}_{infoNCE}$ improved the separability of the samples within the intra-cluster.

Table 3. Ablation study on the effectiveness of different loss combinations in the proposed training approach, measured as WER (%) ($\surd$ = applied to the proposed training approach).

Training Approach	Loss Function					Dev		Test		Average
	$\mathcal{L}_{NSNR}$	$\mathcal{L}_{Enc}$	$\mathcal{L}_{Tokenizer}$	$\mathcal{L}_{CBPC}$	$\mathcal{L}_{infoNCE}$	Clean	Other	Clean	Other	
Joint-Grad						20.11	20.89	20.88	20.98	20.72
Proposed training	$\surd$	$\surd$				22.58	22.78	22.74	23.15	22.81
	$\surd$	$\surd$	$\surd$			19.86	20.40	20.28	20.67	20.30
	$\surd$	$\surd$	$\surd$	$\surd$		19.26	20.04	19.85	20.10	19.81
	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	**19.14**	**19.85**	**19.48**	**19.63**	**19.53**

Table 4 compares the average speech quality scores of the SE models evaluated on the test-clean dataset according to different combinations of losses. As shown in the table, upon the integration of the tokenizer loss, all speech quality scores were improved compared to those of the standalone-SE model, which showed the highest speech quality scores among all of the conventional approaches, as shown in Table 2. However, applying the proposed CBPC loss function, $\mathcal{L}_{CBPC}$, reduced the speech quality scores slightly. Finally, the proposed training approach using the combination of all losses achieved comparable CSIG, CBAK, and COVL scores to Joint-Token, but a higher PESQ score than Joint-Token, which confirms that PESQ is a metric that is related to ASR performance [45].

Table 4. Ablation study on the effectiveness of different loss combinations in the proposed training approach, measured as PESQ, STOI, CSIG, CBAK, and COVL ($\surd$ = applied to the proposed training approach).

Training Approach	Loss Function					PESQ	STOI	CSIG	CBAK	COVL
	$\mathcal{L}_{NSNR}$	$\mathcal{L}_{Enc}$	$\mathcal{L}_{Tokenizer}$	$\mathcal{L}_{CBPC}$	$\mathcal{L}_{infoNCE}$					
Noisy						1.7256	0.6967	1.8547	1.1615	1.3937
standalone-SE						2.6512	0.8277	2.9671	2.5482	2.3410
Proposed training	$\surd$	$\surd$				2.6500	0.8221	2.9595	2.5410	2.3387
	$\surd$	$\surd$	$\surd$			2.6653	**0.8311**	3.1204	**2.5684**	**2.4509**
	$\surd$	$\surd$	$\surd$	$\surd$		2.6638	0.8302	3.1192	2.5651	2.4472
	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	**2.6802**	**0.8311**	**3.1275**	2.5653	2.4507

6. Conclusions

In this paper, the CBPC loss was proposed for noise-robust ASR in a joint training framework. To this end, a pipeline was constructed by using SE and ASR models. In this pipeline, an acoustic tokenizer leveraged the linguistic information from the ASR model to the SE model. The acoustic tokenizer took the outputs of the ASR encoder and provided a pseudo-label through K-means clustering. Then, to mitigate the problem of overfitting on hard samples across the inter-cluster, the proposed CBPC loss function was used to train the acoustic tokenizer. In addition, the infoNCE loss function was combined into the proposed CBPC loss function to improve the intra-cluster separability of the samples.

The WER of the ASR model trained using the proposed training approach was evaluated on a noisy LibriSpeech dataset and compared with those of ASR models trained using conventional training approaches, including MCT, MCT+standalone-SE, and four different joint training approaches. The results revealed that the ASR model trained by the proposed training approach with the CBPC loss function achieved the lowest WER among all of the compared models. In particular, the average WER of the ASR model trained using the proposed training approach was relatively reduced by 15.39% and 3.82% compared to those of the MCT and Joint-Token models, respectively. Next, the speech quality scores of the SE models were compared according to the different training approaches. Consequently, it was also observed that the proposed training approach provided the highest speech quality scores compared to the other approaches.

An ablation study was also conducted to investigate the effects of different combinations of loss functions used in the proposed training approach on the WER and speech quality scores. As a result, the combination of all loss functions, such as the tokenizer loss, CBPC loss, and infoNCE loss, provided the lowest WER and highest speech quality scores, except for CBAK.

In this work, the output vectors from the ASR encoder were clustered, and their cluster indices were used for the target labels of the acoustic tokenizer. As a result, after training the acoustic tokenizer using contrastive loss, there might be some mismatch between the target labels and the outputs of the acoustic tokenizer, as in metric learning [46,47]. In future studies, K-means clustering should be implemented, along with the acoustic tokenizer, to address this mismatch, which is expected to further improve the ASR and SE performance.

Author Contributions: All authors discussed the contents of the manuscript. H.K.K. contributed to the research idea and the framework of this study, and G.W.L. performed the experimental work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute of Information & communications Technology Planning & evaluation (IITP) grant, funded by the Korean government (MSIT) (No. 2019-0-00330, Development of AI Technology for Early Screening of Child/Child Autism Spectrum Disorders based on Cognition of the Psychological Behavior and Response).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were used. LibriSpeech: https://www.openslr.org/12 (accessed on 3 March 2024). Deep noise suppression challenge: https://github.com/microsoft/DNS-Challenge/tree/interspeech2020/master (accessed on 3 March 2024).

Conflicts of Interest: H.K.K. is the founder of AunionAI Co., Ltd., and owns stocks of the company. G.W.L. declares no conflict of interest.

References

1. Wang, D.; Chen, J. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1702–1726. [CrossRef] [PubMed]
2. Nossier, S.A.; Wall, J.; Moniri, M.; Glackin, C.; Cannings, N. An experimental analysis of deep learning architectures for supervised speech enhancement. *Electronics* **2021**, *10*, 17. [CrossRef]

3. Radford, A.; Kim, J.W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. Robust speech recognition via large-scale weak supervision. In Proceedings of the International Conference on Machine Learning (ICML), Baltimore, MD, USA, 17–23 July 2023; pp. 28492–28518. Available online: https://proceedings.mlr.press/v202/radford23a/radford23a.pdf (accessed on 5 March 2024).
4. Caldarini, G.; Jaf, S.; McGarry, K. A literature survey of recent advances in chatbots. *Information* **2022**, *13*, 41. [CrossRef]
5. Bingol, M.C.; Aydogmus, O. Performing predefined tasks using the human–robot interaction on speech recognition for an industrial robot. *Eng. Appl. Artif. Intell.* **2020**, *95*, 103903. [CrossRef]
6. Iio, T.; Yoshikawa, Y.; Chiba, M.; Asami, T.; Isoda, Y.; Ishiguro, H. Twin-robot dialogue system with robustness against speech recognition failure in human-robot dialogue with elderly people. *Appl. Sci.* **2020**, *10*, 1522. [CrossRef]
7. Dahl, G.E.; Yu, D.; Deng, L.; Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *20*, 30–42. [CrossRef]
8. Droppo, J.; Acero, A. Joint discriminative front end and backend training for improved speech recognition accuracy. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toulouse, France, 14–19 May 2006; pp. 281–284. [CrossRef]
9. Li, L.; Kang, Y.; Shi, Y.; Kürzinger, L.; Watzel, T.; Rigoll, G. Adversarial joint training with self-attention mechanism for robust end-to-end speech recognition. *EURASIP J. Audio Speech Music Process.* **2021**, *26*, 26. [CrossRef]
10. Seltzer, M.L.; Yu, D.; Wang, Y. An investigation of deep neural networks for noise robust speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013; pp. 7398–7402. [CrossRef]
11. Kinoshita, K.; Ochiai, T.; Delcroix, M.; Nakatani, T. Improving noise robust automatic speech recognition with single-channel time-domain enhancement network. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7009–7013. [CrossRef]
12. Wang, Z.-Q.; Wang, P.; Wang, D. Complex spectral mapping for single-and multi-channel speech enhancement and robust ASR. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1778–1787. [CrossRef] [PubMed]
13. Shimada, K.; Bando, Y.; Mimura, M.; Itoyama, K.; Yoshii, K.; Kawahara, T. Unsupervised speech enhancement based on multichannel NMF-informed beamforming for noise-robust automatic speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 960–971. [CrossRef]
14. Schuller, B.; Weninger, F.; Wöllmer, M.; Sun, Y.; Rigoll, G. Non-negative matrix factorization as noise-robust feature extractor for speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Dallas, TX, USA, 14–19 March 2010; pp. 4562–4565. [CrossRef]
15. Ma, D.; Hou, N.; Xu, H.; Chng, E.S. Multitask-based joint learning approach to robust ASR for radio communication speech. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, 14–17 December 2021; pp. 497–502. Available online: https://ieeexplore.ieee.org/abstract/document/9689671 (accessed on 5 March 2024).
16. Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; Finn, C. Gradient surgery for multi-task learning. In Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 5824–5836. Available online: https://proceedings.neurips.cc/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf (accessed on 5 March 2024).
17. Guangyuan, S.H.I.; Li, Q.; Zhang, W.; Chen, J.; Wu, X.M. Recon: Reducing conflicting gradients from the root for multi-task learning. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022; Available online: https://openreview.net/forum?id=ivwZO-HnzG_ (accessed on 5 March 2024).
18. Lee, G.W.; Kim, H.K. Two-step joint optimization with auxiliary loss function for noise-robust speech recognition. *Sensors* **2022**, *22*, 5381. [CrossRef]
19. Pandey, A.; Liu, C.; Wang, Y.; Saraf, Y. Dual application of speech enhancement for automatic speech recognition. In Proceedings of the IEEE Spoken Language Technology (SLT) Workshop, Shenzhen, China, 19–22 January 2021; pp. 223–228. [CrossRef]
20. Hu, Y.; Chen, C.; Li, R.; Zhu, Q.; Chng, E.S. Gradient remedy for multi-task learning in end-to-end noise-robust speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5. [CrossRef]
21. Lee, C.C.; Tsao, Y.; Wang, H.M.; Chen, C.S. D4AM: A general denoising framework for downstream acoustic models. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023; Available online: https://openreview.net/forum?id=5fvXH49wk2 (accessed on 5 March 2024).
22. Lee, G.W.; Kim, H.K. Knowledge distillation-based training of speech enhancement for noise-robust automatic speech recognition. *IEEE Access*, 2024; *under revision*.
23. Chai, L.; Du, J.; Liu, Q.F.; Lee, C.H. A cross-entropy-guided measure (CEGM) for assessing speech recognition performance and optimizing DNN-based speech enhancement. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *29*, 106–117. [CrossRef]
24. Zhu, Q.S.; Zhang, J.; Zhang, Z.Q.; Dai, L.R. A joint speech enhancement and self-supervised representation learning framework for noise-robust speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2023**, *31*, 1927–1939. [CrossRef]

25. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. Wav2vec 2.0: A framework for self-supervised learning of speech representations. In Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 12449–12460. Available online: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html (accessed on 5 March 2024).
26. Feng, L.; Shu, S.; Lin, Z.; Lv, F.; Li, L.; An, B. Can cross entropy loss be robust to label noise? In Proceedings of the International Conference on International Joint Conferences on Artificial Intelligence (IJCAI), Virtual, 19–26 August 2021; pp. 2206–2212. [CrossRef]
27. Boudiaf, M.; Rony, J.; Ziko, I.M.; Granger, E.; Pedersoli, M.; Piantanida, P.; Ayed, I.B. A unifying mutual information view of metric learning: Cross-entropy vs. pairwise losses. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual, 23–28 August 2020; pp. 548–564. [CrossRef]
28. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. In Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS), Virtual conference, 6–12 December 2020; pp. 18661–18673. Available online: https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html (accessed on 5 March 2024).
29. Oord, A.V.D.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748. [CrossRef]
30. Hu, Y.; Liu, Y.; Lv, S.; Xing, M.; Zhang, S.; Fu, Y.; Wu, J.; Zhang, B.; Xie, L. DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement. In Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech), Shanghai, China, 25–29 October 2020; pp. 2472–2476. [CrossRef]
31. Gulati, A.; Qin, J.; Chiu, C.C.; Parmar, N.; Zhang, Y.; Yu, J.; Pang, R. Conformer: Convolution-augmented transformer for speech recognition. In Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech), Shanghai, China, 25–29 October 2020; pp. 5036–5040. [CrossRef]
32. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. Available online: https://jmlr.org/papers/v12/pedregosa11a.html (accessed on 5 March 2024).
33. McFee, B.; Raffel, C.; Liang, D.; Ellis, D.P.; McVicar, M.; Battenberg, E.; Nieto, O. Librosa: Audio and music signal analysis in python. In Proceedings of the 14th Python in Science Conference, Austin, TX, USA, 6–12 July 2015; pp. 18–25. [CrossRef]
34. Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In Proceedings of the International Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 1857–1865. Available online: https://dl.acm.org/doi/10.5555/3157096.3157304 (accessed on 5 March 2024).
35. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 14–19 June 2020; pp. 9729–9738. [CrossRef]
36. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Pires, B.A.; Guo, Z.D.; Azar, M.G.; et al. Bootstrap your own latent-a new approach to self-supervised learning. In Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 21271–21284. Available online: https://papers.nips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html (accessed on 5 March 2024).
37. Chen, M.; Fu, D.Y.; Narayan, A.; Zhang, M.; Song, Z.; Fatahalian, K.; Ré, C. Perfectly balanced: Improving transfer and robustness of supervised contrastive learning. In Proceedings of the International Conference on Machine Learning (ICML), Baltimore, MD, USA, 17–23 July 2023; pp. 3090–3122. Available online: https://proceedings.mlr.press/v162/chen22d (accessed on 5 March 2024).
38. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. LibriSpeech: An ASR corpus based on public domain audio books. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 5206–5210. [CrossRef]
39. Reddy, C.K.; Gopal, V.; Cutler, R.; Beyrami, E.; Cheng, R.; Dubey, H.; Gehrke, J. The INTERSPEECH 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results. In Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech), Shanghai, China, 25–29 October 2020; pp. 2492–2496. [CrossRef]
40. Kudo, T.; Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP), Brussels, Belgium, 2–4 November 2018; pp. 66–71. [CrossRef]
41. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Zheng, X. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 265–283. Available online: https://dl.acm.org/doi/10.5555/3026877.3026899 (accessed on 5 March 2024).
42. ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs. 2005. Available online: https://www.itu.int/rec/T-REC-P.862 (accessed on 3 March 2024).
43. Taal, C.H.; Hendriks, R.C.; Heusdens, R.; Jensen, J. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Dallas, TX, USA, 14–19 March 2010; pp. 4214–4217. [CrossRef]

44. Hu, Y.; Loizou, P.C. Evaluation of objective quality measures for speech enhancement. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *16*, 229–238. [CrossRef]
45. Yamada, T.; Kumakura, M.; Kitawaki, N. Performance estimation of speech recognition system under noise conditions using objective quality measures and artificial voice. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 2006–2013. [CrossRef]
46. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–15 October 2016; pp. 499–515. [CrossRef]
47. Kim, S.; Kim, D.; Cho, M.; Kwak, S. Proxy anchor loss for deep metric learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 21–24 July 2022; pp. 3238–3247. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.