mathematics

Article

Spectral Salt-and-Pepper Patch Masking for Self-Supervised Speech Representation Learning

June-Woo Kim [1], Hoon Chung [2] and Ho-Young Jung [1,*]

[1] Department of Artificial Intelligence, Kyungpook National University, Daegu 41566, Republic of Korea; kaen2891@knu.ac.kr
[2] Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea; hchung@etri.re.kr
* Correspondence: hoyjung@knu.ac.kr

Abstract: Recent advanced systems in the speech recognition domain use large Transformer neural networks that have been pretrained on massive speech data. General methods in the deep learning area have been frequently shared across various domains, and the Transformer model can also be used effectively across speech and image. In this paper, we introduce a novel masking method for self-supervised speech representation learning with salt-and-pepper (S&P) mask which is commonly used in computer vision. The proposed scheme includes consecutive quadrilateral-shaped S&P patches randomly contaminating the input speech spectrum. Furthermore, we modify the standard S&P mask to make it appropriate for the speech domain. In order to validate the effect of the proposed spectral S&P patch masking for the self-supervised representation learning approach, we conduct the pretraining and downstream experiments with two languages, English and Korean. To this end, we pretrain the speech representation model using each dataset and evaluate the pretrained models for feature extraction and fine-tuning performance on varying downstream tasks, respectively. The experimental outcomes clearly illustrate that the proposed spectral S&P patch masking is effective for various downstream tasks when combined with the conventional masking methods.

Keywords: self-supervised learning; speech representation learning; salt-and-pepper masking; spectrum patch masking

MSC: 68T10

Citation: Kim, J.-W.; Chung, H.; Jung, H.-Y. Spectral Salt-and-Pepper Patch Masking for Self-Supervised Speech Representation Learning. *Mathematics* **2023**, *11*, 3418. https://doi.org/10.3390/math11153418

Academic Editors: Grigoreta-Sofia Cojocar, Adriana-Mihaela Guran, Laura-Silvia Dioşan and Yang Liu

Received: 14 June 2023
Revised: 21 July 2023
Accepted: 3 August 2023
Published: 5 August 2023

1. Introduction

The majority of recent speech representation models typically depend on large Transformer neural networks [1] that are pretrained using self-supervised learning methods with thousands of hours of speech data. In general, self-supervised speech representation learning utilizes the structure of the input speech itself for the learning process without any annotations. Through speech representation pretraining with massive speech datasets, researchers have been able to achieve state-of-the-art performance on a diverse set of speech-related tasks, such as speech recognition (ASR), phoneme recognition, emotion recognition, and speaker verification [2–6]. Overall, the pretraining of large Transformer networks using self-supervised learning techniques has become a key strategy for advancing state-of-the-art speech processing technology.

Various promising outcomes of neural network models and sophisticated methodologies are often adapted to various other domains as well. Specifically, the Transformer network was first proposed in natural language processing (NLP) tasks such as machine translation and language modeling and has become popular in various domains, including computer vision, speech, and signal processing. Moreover, the masked language modeling for self-supervised learning introduced in BERT [7] has found extensive use in speech

representation learning tasks [3,6,8–12]. Contrastive Predictive Coding (CPC) [13], a well-known technique for representation learning, is extensively applied in speech [2,4,10,14,15], NLP [16,17], and computer vision domains [18–20], even in a supervised setting [21].

Especially, speech spectrograms share similar data formats with images, and consequently, there has been a mutual influence between data preprocessing techniques and data training neural networks. For example, several applications of transfer learning from visual to audio tasks have been demonstrated to be effective [22,23]. The vision Transformer [24], derived from the Transformer for word sequences, has enabled patch embedding in audio streams [25,26].

SpecAugment [27] is a speech data augmentation technique inspired by "Cutout" [28], an augmentation method proposed in the computer vision domain. Within the domain of self-supervised speech representation learning, masking techniques based on SpecAugment are mainly used in reconstruction tasks [3,6,8–12]. Similar to image processing techniques, SpecAugment performs masking over continuous time–frequency regions of a given input spectrogram by drawing continuous blocks with zero values. By reconstructing these masked regions to their original forms, the pretrained model can learn more robust speech representations and has outperformed conventional techniques in several downstream tasks [3,6,8,9,11,12].

In the computer vision domain, salt-and-pepper (S&P) noise refers to a type of impulse contamination in an image, where random white and black dots appear. This kind of noise can often be eliminated by using techniques such as denoising autoencoders [29,30] and convolutional-neural-network-based median layers [31,32]. These denoising pretext tasks aim to remove or reduce the noise to improve their quality and make them more useful for downstream tasks.

Inspired by this, we introduce a novel self-supervised speech representation learning strategy that utilizes the S&P mask. The proposed masking method involves consecutive quadrilateral-shaped S&P patches that contaminate the speech spectrogram by a randomly determined percentage. The S&P noise in computer vision, however, cannot be effectively applied to the speech domain due to the difference in resolution or scale between the spectrograms and images. To cope with this problem, the proposed scheme uses the S&P mask modifying the standard S&P noise to make it suitable for the speech domain. Figure 1 presents the overall framework of this paper.

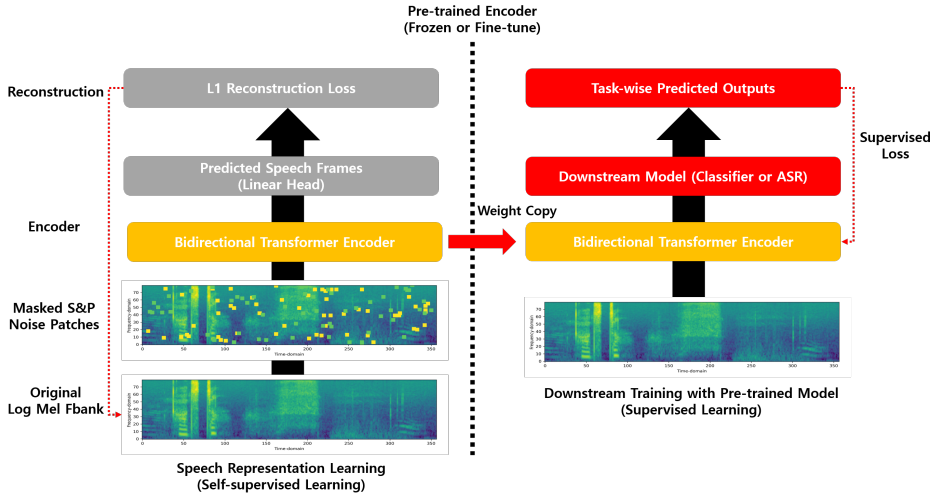


Figure 1. The overall framework of this paper consists of two parts. The left side illustrates the architecture of the speech representation model that uses the proposed spectral S&P patch masking for self-supervised learning. On the right side, labeled speech data are finally trained using the pretrained speech representation model. In other words, the pretrained encoder is connected to the downstream model, which can be used as a feature extraction (weight frozen) or fine-tuning (gradient flow) approach.

To assess the efficacy of the proposed spectral S&P patch masking method for self-supervised speech representation learning, pretraining experiments are performed separately in two languages: English [33] and Korean [34]. The pretrained model is then evaluated in two ways, namely feature extraction and fine-tuning, on several downstream tasks, which include speech recognition for both the English and Korean datasets, phoneme classification, keyword spotting, and speaker identification. Furthermore, a comparative analysis is conducted to determine the effectiveness of the spectral S&P patch masking method on its own, as well as in combination with other conventional masking approaches. Our findings confirm that the proposed spectral S&P patch masking, when utilized in conjunction with conventional masking techniques, yields superior results for various speech-related downstream tasks. These results indicate that the proposed method can serve as a valuable supplement to existing self-supervised learning techniques, potentially leading to improvements in speech representation learning.

The main contributions of this paper can be summarized as follows:

- We propose a straightforward and novel masking method for self-supervised speech representation learning with consecutive quadrilateral-shaped S&P patch blocks. S&P masking has not been attempted before for speech representation learning.
- Due to the difference in resolution or scale between the spectrogram and the image, applying S&P noise directly is not a useful method. To this end, we demonstrate that modifying S&P noise is more applicable for reconstruction objectives of self-supervised speech representation learning.
- We show that the combination of the proposed spectral S&P patch method with the conventional reconstruction-based speech representation learning approach is more effective in several speech downstream tasks compared with using the traditional masking methods alone.

The rest of this paper is organized as follows: A concise overview of related works on S&P noise and masking reconstruction for self-supervised speech representation learning is provided in Section 2. In Section 3, the details of the proposed spectral S&P patch masking and pretraining method are introduced. In Section 4, detailed information about the experimental setting is provided, including various downstream tasks such as feature extraction, fine-tuning, and the datasets used. In Section 5, extensive experimental results are presented to validate the effectiveness of the proposed method. Finally, the discussion and conclusion of this paper are given in Sections 6 and 7.

2. Related Work

In this section, we briefly review the S&P noise in the computer vision domain and the masking-based reconstruction method for self-supervised speech representation learning.

2.1. Salt-and-Pepper Noise

S&P noise is a common type of image distortion caused by impulse contamination in the field of computer vision, where white and black pixels are randomly caused throughout the image, resembling grains of salt and pepper. In the conventional approaches, the removal of this kind of noise involves using median filtering, which replaces each pixel in the image with the median value of the neighboring pixels [35,36] and CNN-based median layers [31,32]. Furthermore, recent approaches to the utilization of S&P noise as a pretext task for unsupervised learning have been demonstrated to be effective [30,37,38]. As a result, pretrained models are able to obtain improved performance in downstream tasks by learning to extract stable and consistent features, achieved through reconstructing the original input data from noisy and contaminated data. In contrast to existing research focused on the image domain, our study introduces a novel modification to the conventional pointwise S&P noise technique by transforming it into quadrilateral-shaped patch masking. This adaptation ensures its suitability for the speech domain, and we utilize it as a reconstruction objective for self-supervised speech representation learning. Furthermore, the pretrained speech representation model with the proposed spectral S&P patch masking

demonstrates synergistic effects when combined with conventional masking approaches across various downstream tasks.

2.2. Masked Reconstruction for Self-Supervised Speech Representation Learning

Inspired by BERT [7], masking-based reconstruction is one of the most commonly used in self-supervised speech representation learning techniques [2–6,8,9,11,12]. The objective of the masking-based speech representation model is to restore the original speech frames from the masked ones using a reconstruction loss function. To this end, continuous time frames of the given spectrogram are randomly chosen and then masked by zero value or replaced with other frames. Similar to SpecAugment [27], a more recent method includes masking both time and frequency regions of spectrograms. This process enables the pretrained model to restore the contaminated input speech features while also acquiring robust speech representation. In this study, we introduce a straightforward spectral S&P patch masking method for self-supervised speech representation learning that randomly masks selected regions with consecutive quadrilateral-shaped S&P blocks, without any complex operations. Adding the proposed method to the previous studies [3,6,12] will demonstrate more effectiveness for various speech downstream tasks compared with conventional masking methods only.

3. Method

3.1. Modified S&P for Speech Representation Learning

Typically, image pixels usually have integer values between 0 to 255, where 0 and 255 represent black and white color in the image, respectively. S&P noise in computer vision refers to a type of impulse contamination that results in random white and black dots appearing in image pixels. Directly applying the S&P noise technique originally designed for image data in the computer vision domain to spectrograms would not yield effective results. This is primarily because of the inherent differences in data scales between spectrogram and image. Generally, spectrograms consist of floating-point values, while image data are represented using that of an integer. As a consequence, applying the S&P noise method to spectrograms as a masking strategy that operates on a different scale may lead to suboptimal outcomes when we perform self-supervised speech representation learning. Therefore, it is crucial to consider the specific characteristics and requirements of spectrogram data to enhance speech representation learning effectively. In this work, the proposal is made to modify S&P noise to enhance its suitability for speech samples.

Let $x_{f,t}$, for $(f,t) \in \mathcal{S} \equiv \{1,\ldots,F\} \times \{1,\ldots,T\}$, be the original F-by-T spectrogram x at pixel location (f,t) and $[v_{min}, v_{max}]$ be the dynamic vector range of x, i.e., $v_{min} \leq x_{f,t} \leq v_{max}$ for all $(f,t) \in \mathcal{S}$. Consequently, a noisy spectrogram is denoted as y. Therefore, the proposed S&P noise for speech samples at pixel location (f,t) is given by

$$y_{f,t} = \begin{cases} v_{max}, & \text{with probability } s \\ v_{min}, & \text{with probability } p \\ x_{f,t}, & \text{with probability } 1-p-s \end{cases} \quad (1)$$

where s and p are the probability of salt and pepper noise, respectively, with α representing the noise level defined as the sum of s and p. In this work, we set s and p to 0.002, resulting in α being equal to 0.004. Note that the speech data are normalized with zero mean and unit variance before obtaining the salt values used for self-supervised learning.

3.2. Consecutive Quadrilateral-Shaped Spectral S&P Patch Masking for Self-Supervised Learning

Unlike image data, speech has continuous characteristics and is larger than the image in scale (time frames). Scale issues may prevent accurate self-supervised speech representation learning when the original S&P noise is applied to the masking strategy. Figure 2 shows the different point sizes of the S&P noise applied to masking the spectrograms. As shown in Figure 2b, the original point-shaped S&P noise (frame-level noise) is a very small

portion of the given spectrogram. As a result, conventional point-shaped S&P noise is not very effective for self-supervised pretext tasks in the speech domain, since the single pointwise noise is a tiny fraction of the spectrogram (e.g., 10 s is 1000 frames and the pointwise 1 frame noise is 25 ms), that is, typically extracted with a window size of 25 ms.

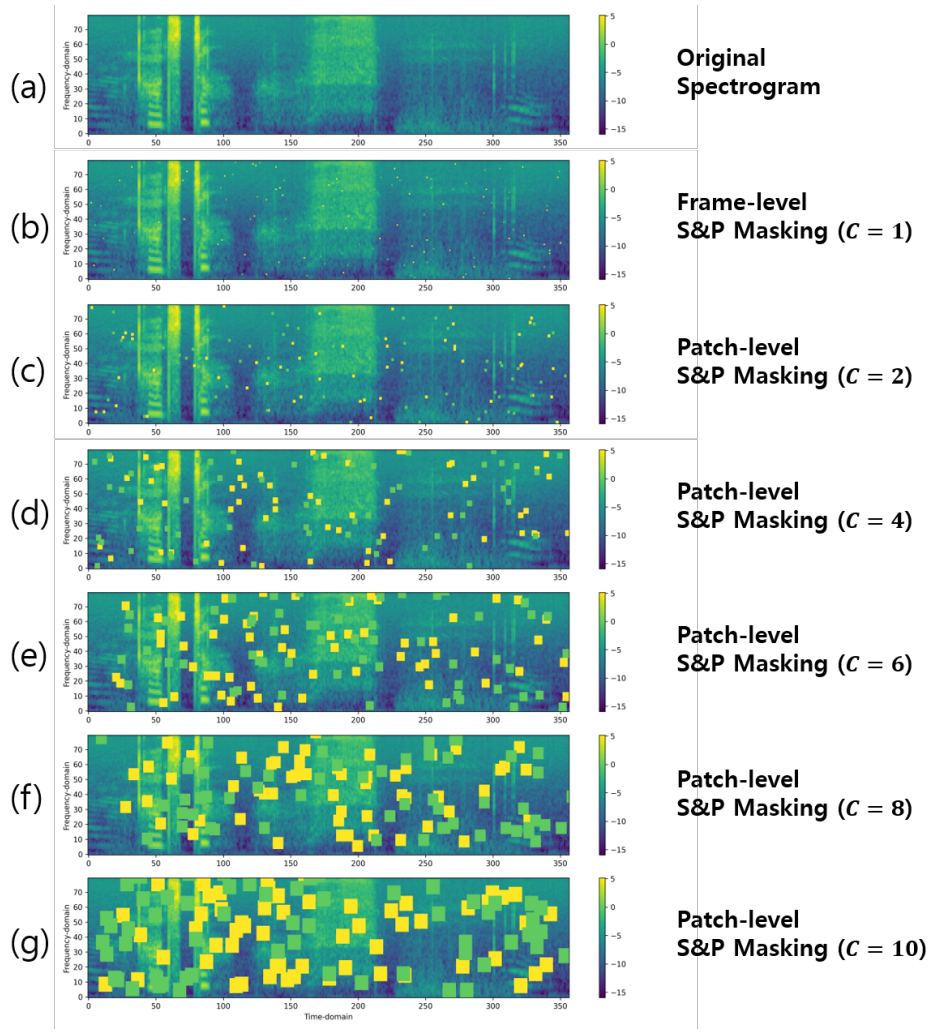


Figure 2. Illustration of the S&P patch masking on the spectrogram with various consecutive factor C but same total noise masking level $\alpha = 0.004$. (**a**) shows the input original spectrogram while (**b**–**g**) illustrates the masked spectrogram on different hyperparameters $C = 1, 2, 4, 6, 8, 10$, respectively.

To consider the specific scale characteristics of spectrogram data and improve the effectiveness of the masking strategy using S&P for speech representation learning, we propose a solution that substitutes consecutive quadrilateral-shaped patches for point-shaped noise, illustrated in Figure 2. To this end, a consecutive factor C is employed, which determines the number of frames to be masked during pretraining. This encourages the model to learn contextualized representations from the spectrogram structure. Specifically, the initial step of the process entails the random selection of a spectrum point denoted as $y_{f,t}$ according to Equation (1). Subsequently, a quadrilateral-shaped region with a side length of C is masked starting from the selected point, where C represents a certain value. For instance, if C is assigned a value of 3, the masking process involves 9 points within the square region, resulting in a square size of 9×9. The termination condition for each random point $y_{f,t}$ is when it intersects with either the endpoint of a horizontal or vertical point within the spectrogram or when it is moved by C.

When a larger value of C is used (Figure 2e,f), the model is encouraged to learn relationships between more distant parts of the spectrogram. This can be advantageous

for capturing higher-level structures and dependencies that span across larger temporal contexts. On the other hand, when a smaller value of C is employed (Figure 2c,d), the model is focused more on local relationships within shorter temporal windows. This can be beneficial for capturing fine-grained details and local patterns within the spectrogram.

Using excessively small or large values of C, however, can lead to challenges in accurate speech representation learning. Extremely small values (Figure 2b) might overly constrain the model's ability to learn meaningful representations, limiting its capacity to capture the relevant information within the spectrogram. Conversely, excessively large values might (Figure 2f) hinder the model's capability to faithfully reconstruct the original input spectrogram. Therefore, finding an appropriate range for the value of C is crucial for ensuring accurate speech representation learning. We provide an exposition on how the consecutive factor C affects the learning of speech representation for its performance of downstream tasks in Section 5.

3.3. Pretraining with S&P Patch Masking for Self-Supervised Learning

To pretrain the speech representation model in self-supervision manner, the first step involves applying the proposed spectral S&P patch masking to the original input spectrogram. Figure 3 provides an overview of the masking process using the proposed spectral S&P patch.

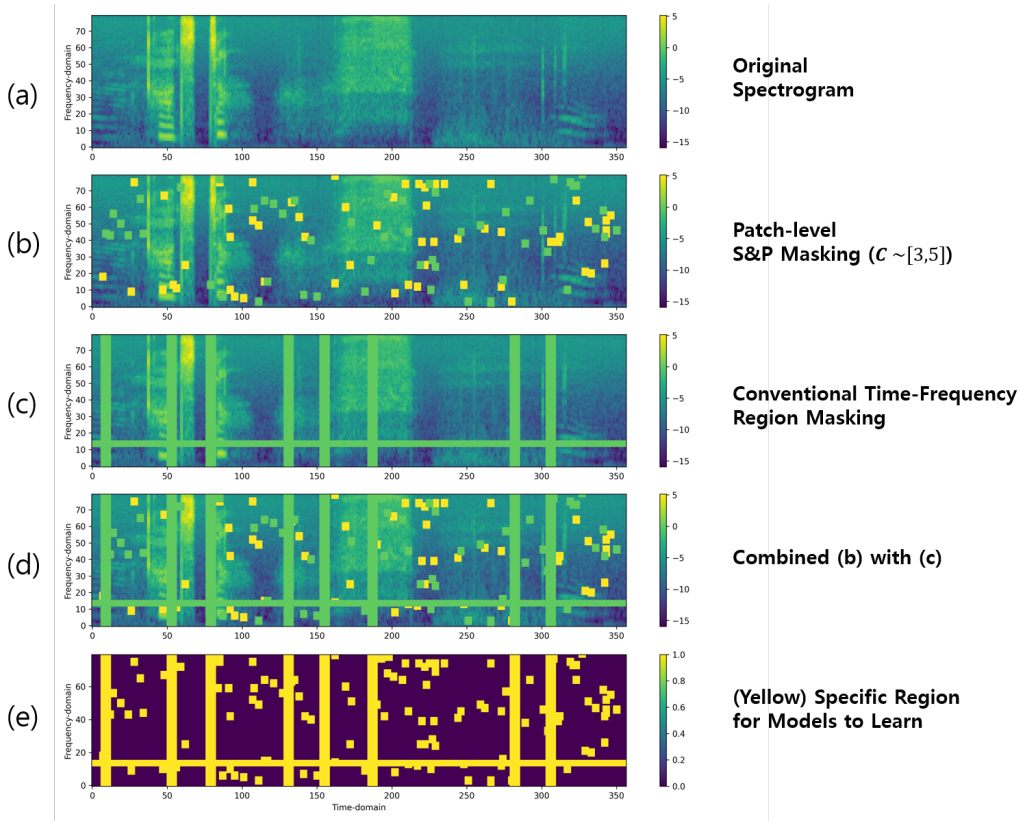


Figure 3. The overall process of the proposed spectral S&P patch masking with $\alpha = 0.004$ and $C \sim [3,5]$. (**a**) shows the input original spectrogram while (**b**) and (**c**) demonstrate the spectrogram after being masked with the S&P patches and conventional time-frequency regions masking respectively, and (**d**) shows the combination of both (**b**) and (**c**). In (**d**), both the pepper and conventional masking regions (green) are masked with zero value and the salt (yellow) regions consist of the maximum value in the given spectrogram. (**e**) shows that the yellow area in the spectrogram denoted where the model will be learned during the reconstruction pretext tasks. The masked input spectrogram (**d**) is subsequently fed into the speech feature representation model. The model is trained with the objective of accurately reconstructing the masked spectrogram (**d**) to the original spectrogram (**a**) using the reconstruction loss.

Initially, the input original spectrogram (Figure 3a) is masked by applying S&P values. The process of setting the region masked by the S&P patch involves assigning a value of zero to half of the region and the maximum value of the given spectrogram to the other half, depending on the total masking value α. This ensures that half of the masked region contains no information (represented by a value of zero), while the other half retains the maximum value from the original spectrogram. This approach helps to introduce randomness and variability in the masked region, promoting robust feature learning during the self-supervised training process. In Figure 3b, the pepper and salt patch regions are depicted in green and yellow colors, respectively. We use a random value of C between 3 and 5 (denoted $C \sim [3, 5]$) with $\alpha = 0.004$ condition during pretraining to encourage the model to learn contextualized representations from spectrogram structure. Note that the quadrilateral-shaped spectral S&P patches can be rectangle or square and overlapped with each other.

In addition, the proposed spectral S&P patch masking approach can be easily integrated with other masking methods. Figure 3c depicts the conventional time–frequency region masking method [6], while Figure 3d illustrates the combination of Figure 3b,c. Note that the existing masking methods can be time, frequency, or time–frequency regions for reconstruction [3,6,12]. In Figure 3e, the yellow area in the spectrogram represents the specific region where the model focuses on reconstruction for pretraining tasks.

To reconstruct the masked spectrogram shown in Figure 3d back to its original form in Figure 3a, 3-layer bidirectional Transformer encoders with 768 hidden sizes and position encoding, 3072-dimensional feed-forward networks, and 12 multihead attentions are used. Speech sequence length is limited to under 1500 to fast model training, which is approximately 15 s. The proposed pretrained model utilizes the masked spectrogram as input and generates a reconstructed version using L1 loss as the objective function for minimizing between the original spectrogram and predicted outputs.

In order to perform the aforementioned process using the proposed spectral S&P patch masking, we use 80-dimensional log Mel Fbank (filter bank) features, which were extracted with a window size of 25 ms and an overlap size of 10 ms. These speech features are normalized with a mean of zero and a unit variance to use as input to the model. To optimize the pretrained model, the AdamW optimizer with a learning rate scheduler is used, which increases from 0 up to a maximum value of 0.0002 after 7% of the training steps have been completed and then decreases back to 0. In the pretraining experiments, 32 batch size and 4 gradient accumulation steps are used until 1,000,000 steps, which are approximately 100 epochs, to learn optimal model parameters that minimize the L1 loss for reconstruction. For reproducibility reasons, we use the same configuration as described in the S3PRL toolkit [39].

To summarize the pretraining stage, we aim to let the pretrained model learn representations that capture essential information from the masked spectrogram in Figure 3d in order to successfully reconstruct the original spectrogram in Figure 3a using the reconstruction loss.

3.4. Training Pretrained Model on Downstream Tasks

To evaluate the effectiveness of the pretrained model using the proposed S&P patch masking, there are two ways to measure several downstream tasks.

(1) Speech representation extraction: To obtain speech representations from the pretrained model, we extract the hidden states of the last layer of the model, which correspond to the deepest Transformer encoder layer. Subsequently, these extracted representations are utilized as inputs for the downstream model, effectively replacing the speech features (e.g., log Mel Fbank). By feeding the speech representations of the pretrained model to the downstream model, we enable it to leverage the rich and meaningful information encoded in the representations for performing various tasks. Note that when training the downstream tasks in this manner, the parameters of the pretrained model are frozen. In other words, the parameters of the pretrained model are not updated during the training process of the downstream tasks. This allows the pretrained model to serve as a fixed

speech feature extractor, providing stable and reliable representations for the downstream tasks. This method is represented as "feature extraction" in later experimental tables.

(2) Fine-tuning: Another approach for utilizing the pretrained model is through fine-tuning in conjunction with downstream models. In this method, the output of the pretrained model is connected to a downstream model, and the downstream model can be of any type, depending on the specific task at hand. Initially, while the pretrained model retains its learned knowledge, the parameters of the downstream model are randomly initialized. During fine-tuning, both the pretrained model and the downstream model are updated together. The fine-tuning process involves jointly optimizing the parameters of both models using task-specific training data. This method is denoted as "fine-tuning" in subsequent experimental tables.

4. Experimental Setup

In this section, a comprehensive evaluation of the proposed spectral S&P patch masking for self-supervised speech representation learning by performing six downstream tasks is conducted.

4.1. Dataset Description

We standardize the audio sampling rate to 16 kHz to ensure that all speech data used in the experiments have a consistent quality. Table 1 shows all the datasets used in this paper. Details of various datasets used in this paper are as below.

Table 1. Speech datasets summary used in the pretraining and downstream experiments in this paper. The symbols ✗ and ✓ denoted yes and no.

Dataset Specific		Used For	
Name	**Hours**	**Pretraining**	**Downstream Task**
LibriSpeech [33]	960	✓	Phoneme Classification (100 h) English ASR (100 h)
TIMIT [40]	5.4	✗	Phoneme Classification (All)
Speech Commands [41]	18	✗	Keyword Spotting (All)
VoxCeleb1 [42]	352	✗	Speaker Identification (All)
KsponSpeech [34]	1000	✓	Korean ASR (All)

(1) LibriSpeech: The LibriSpeech dataset [33] is one of the widely used benchmarks for speech recognition research. This dataset encompasses a large-scale collection of English speech recordings totaling approximately 960 h from audiobooks. The training set comprises three subsets: train-clean-100, train-clean-360, and train-other-500. The "clean" designation indicates the absence of noise, while "other" denotes the presence of noise. The numbers 100, 360, and 500 refer to the respective hour durations of the subsets. For evaluation purposes, the dataset includes the dev-clean, dev-other, test-clean, and test-other subsets. In our experiments, we utilize a total of 960 h from the LibriSpeech dataset for pretraining. For the downstream tasks, the train-clean-100 subset is used for training the phoneme classification and English ASR tasks, and the dev-clean and test-clean subsets are used for evaluation.

(2) TIMIT: The TIMIT [40] dataset is a well-known corpus used for speech recognition and acoustic–phonetic studies. This dataset consists of recordings from 630 American English speakers pronouncing phonetically rich sentences. In our experiments, we only use this dataset for conducting phoneme classification for downstream tasks. The TIMIT dataset is divided into three subsets: "train", "dev", and "test". During our experiments, we train the downstream tasks using the training set and determine the best-performing checkpoint on the dev set to assess the performance on the test set.

(3) Speech Commands: The Speech Commands [41] dataset is usually used in the field of keyword spotting, which is specifically designed to recognize and classify spoken commands of keywords. The dataset comprises recordings of short spoken commands from a diverse set of speakers, covering different categories such as "yes", "no", "up", "down", "left", "right", "stop", "go", and more. In our experiments, we train the keyword spotting downstream model using the training set and then evaluate the performance of the model on the development set and test set. Note that the Speech Commands dataset is not utilized for pretraining in our experiments.

(4) VoxCeleb1: The VoxCeleb1 [42] dataset is an audio–visual dataset that contains more than 100,000 utterances from 1251 celebrities. These utterances are extracted from videos on YouTube. The VoxCeleb1 dataset is widely used for tasks such as speech recognition and speaker identification. For conducting the speaker identification downstream task, we utilize the VoxCeleb1 training and test set split provided within the dataset itself. The VoxCeleb1 dataset is not used for pretraining in this paper.

(5) KsponSpeech: To explore the effectiveness of the proposed method in languages other than English, the KsponSpeech [34] dataset is used for both pretraining and ASR downstream tasks. By using a non-English dataset, we can evaluate the performance and generalizability of the proposed method in different linguistic contexts. This allows us to investigate the applicability and effectiveness of the method beyond the English language. The KsponSpeech dataset is widely used in the Korean ASR domain and comprises around 1000 h of speech from native Korean adult males and females, providing a total of 620,000 training examples. For pretraining, the speech samples that are shorter than 3 s were excluded, so only 517,144 samples were utilized for self-supervised learning with the proposed method. In contrast, all the KsponSpeech training samples were used to measure ASR performance in the downstream task. In the Korean ASR experiment, the ASR performance is reported on the KsponSpeech dev set.

4.2. Downstream Tasks Details

In this subsection, the respective six downstream tasks setup and training details are explained.

(1) LibriSpeech phoneme classification: The framewise phoneme prediction performance is evaluated using classifiers trained on the last hidden state of representations for both the feature extraction and fine-tuning stages. To ensure reproducibility, this downstream task follows previous work as described in [39]. For the phoneme classification task on the LibriSpeech [33] dataset, the train-clean-100 subset includes 41 possible phoneme classes used for training. To ensure consistency, we make use of aligned phoneme labels, train/test split, and the development set derived from 10% of the training set as provided in [39]. In evaluating the LibriSpeech phoneme classification task, the phoneme classification accuracy (%) on the development and test sets employing two measurement approaches are provided: *1-linear classifier* and *2-linear classifier*. In the 1-linear classifier approach, a single linear classifier is employed to evaluate the linear separability of phonemes. This setting is denoted as "1-linear classifier". Furthermore, the incorporated classifiers with a single hidden layer of 768 dimensions are referred to as the "2-linear classifier" setting. During training, the AdamW [43] optimizer with a learning rate of 0.0002 and a batch size of 32 is used. The training process continues until 500,000 steps.

(2) TIMIT phoneme classification: For the TIMIT [40] phoneme classification task, framewise phoneme predictions are estimated based on the manual phoneme transcriptions provided in the TIMIT dataset. To ensure reproducibility, the procedures outlined in previous work [39] are followed. In experiments, the phoneme classification task is conducted using the TIMIT training set, which comprises 39 phoneme classes, as described in [39,44]. The phoneme classification accuracy (%) on the test set is reported using three different measurement approaches: *conv-bank classifier*, *1-linear classifier*, and *2-linear classifier*. In the conv-bank classifier approach, a 64-dimensional hidden layer is utilized, along with three 1D-CNN layers with kernel sizes of [3, 5, 7] and channel sizes of [32, 32, 32]. This is

followed by a 96-dimensional hidden layer and a phoneme classifier. This specific setting is referred to as the "conv-bank classifier" approach. The structures of both the 1-linear classifier and the 2-linear classifier are equivalent to the phoneme classification setting used for the LibriSpeech task. For training, the AdamW optimizer is employed with a learning rate of 0.0002 and a batch size of 16. The training process for this task continues until 500,000 steps.

(3) Keyword spotting: To evaluate the quality of representations for the keyword spotting task in both the feature extraction and fine-tuning stages, the Speech Commands [41] dataset is used, following the setup employed in the S3PRL toolkit [39]. In this setup, keyword spotting is treated as a 12-class classification problem. A two 256-dimensional hidden layer feed-forward classifier is employed for this task. Prior to the output layer, mean pooling over time is applied to the representations, as described in [39]. The evaluation metric reported is the keyword classification accuracy (%) on the test set. This evaluation allows for the assessment of the pretrained model's representation transferability across different domains. During the keyword spotting training, the Adam optimizer [45] is used with a learning rate of 0.0001 and a batch size of 32. The training process for this task continues until 200,000 steps.

(4) Speaker identification: For evaluating the speaker identification task in both the feature extraction and fine-tuning stages, the VoxCeleb1 dataset [42] is used. A frame-wise linear transformation is applied, followed by mean-pooling across all sequences. This is then followed by another linear transformation with a cross-entropy loss for the utterance-level task. This setting is consistent with the approach described in [39,46]. The evaluation metric used for this task is accuracy (%), which measures the percentage of correctly identified speakers. During training, the Adam optimizer is utilized with a learning rate of 0.0001 and a batch size of 8. Additionally, 4 gradient accumulation steps are employed until reaching 200,000 training steps.

(5) English ASR: To evaluate the English Automatic Speech Recognition (ASR) downstream performance in both the feature extraction and fine-tuning stages, the LibriSpeech dataset [33] is utilized. Specifically, the train-clean-100, dev-clean, and test-clean subsets of LibriSpeech are used for training, validation, and testing, respectively. The performance of two types of deep neural network settings, BiGRU and BiLSTM, is measured as described in [39,46]. For both the BiGRU and BiLSTM settings, a 2-layer bidirectional GRU and bidirectional LSTM with a dimensionality of 1024 are used. The models are optimized using the Connectionist Temporal Classification [47] loss on 32 English characters. Once trained, the ASR models are decoded using the LibriSpeech official 4-gram language model performed by KenLM [48] and the Wav2letter toolkit [49]. The evaluation metric reported for English ASR modeling on the LibriSpeech dataset is the word error rate (WER, %) on the test-clean subset. During the training stage, the Adam optimizer is used with a learning rate of 0.0001, a batch size of 32, and a beam size of 5 until 200,000 steps.

(6) Korean ASR: To evaluate the performance of Korean ASR downstream performance for feature extraction, the KsponSpeech dataset [34] is employed. In order to compare the performance of feature extraction, the weights of various pretrained models are kept frozen. The speech representations are extracted from the hidden state of the final bidirectional Transformer encoder. These representations are then fed into the ASR model architecture described in [50]. Specifically, an ESPNet [51]-like Transformer model is used for ASR architecture, which includes 7-layer CNNs with 8 subsampling operations, followed by 3 Transformer encoder layers and 6 Transformer decoder layers. The maximum input length for the training set is set to 25 s, and no additional normalization or preprocessing is applied. During training, the AdamW optimizer is utilized with a learning rate of 0.001, and a Transformer learning rate scheduler [1] is employed. The model is trained using a total batch size of 64 on 4 TITAN RTX GPUs until 50 epochs. Label smoothing [52] is also applied during training.

Unlike English characters, the KsponSpeech dataset includes 2311 symbols, including special tokens such as "start", "end", "mask", "pad", and "unk". This makes the Korean ASR

downstream task more challenging to predict accurately. To evaluate the performance of the Korean ASR downstream task, the character error rate (CER, %) is used as a commonly employed metric in Korean ASR. The CER is computed at the syllable level by measuring the Levenshtein distance [53] between the ASR-predicted results and the corresponding labels.

4.3. Software and Hardware Details

In all the experiments of this paper, the following software and hardware configurations are utilized. By using these specific versions of the software and libraries, the reproducibility and consistency of the experimental setup are ensured.

- Python version: 3.9.15
- GPU server: The pretraining and downstream experiments are conducted on an NVIDIA RTX A6000 GPU (48GB) server and TITAN RTX GPU (24GB) server running Ubuntu 18.04.6 LTS, respectively.
- Deep learning framework: PyTorch [54] version 1.12.1 with CUDA version 11.3 and CuDNN version 8.21. These libraries enable efficient GPU acceleration for training and inference.
- Speech preprocessing: For speech preprocessing tasks, we rely on the TorchAudio [55] library (version 0.12.1) for audio-related operations. Additionally, Numpy [56] (version 1.23.5) and SoundFile (version 0.11.0) are used for numerical computations and reading and writing audio files, respectively.
- English ASR downstream: For English ASR tasks, both KenLM [48] (Available online: https://github.com/kpu/kenlm, accessed on 14 June 2023) and Wav2letter++ [49] (Available online: https://github.com/flashlight/wav2letter, accessed on 14 June 2023) libraries are employed. KenLM is used for language modeling, while Wav2letter++ provides useful ASR functionality.
- Korean ASR downstream: In the case of Korean ASR tasks, the python-Levenshtein library (version 0.20.9) is utilized to compute the edit distance metric.

5. Results

In all the downstream tasks, the effectiveness of the proposed spectral S&P patch masking alone or combined with previous speech representations learning methods is compared.

Table 2 presents seven selected methods, including diverse self-supervised representation learning techniques (Fbank, APC [15], NPC [10], Mockingjay [3], Audio ALBERT [12], TERA [6], and the proposed spectral S&P Patch). Note that the Fbank refers to the input that is directly converted from the original speech without using any pretrained speech feature representation. These methods have been chosen to provide a comprehensive evaluation of the proposed S&P patch masking in relation to existing approaches. In Table 2, the designation of * models indicates the utilization of their pretrained weights from the S3PRL [39] toolkit, which are employed for our downstream tasks without modification. On the contrary, the † models listed in Table 2 signify the pretrained speech representation models that we implement ourselves using the S3PRL toolkit.

Furthermore, the experimental results are presented by integrating their approaches with contemporary self-supervision masking methods. In our experiments, note that these combined experiments exclusively focus on parallel network-type models that are composed of Transformer-based architectures [3,6,12], namely Mockingjay + Ours, Audio ALBERT + Ours, and TERA + Ours.

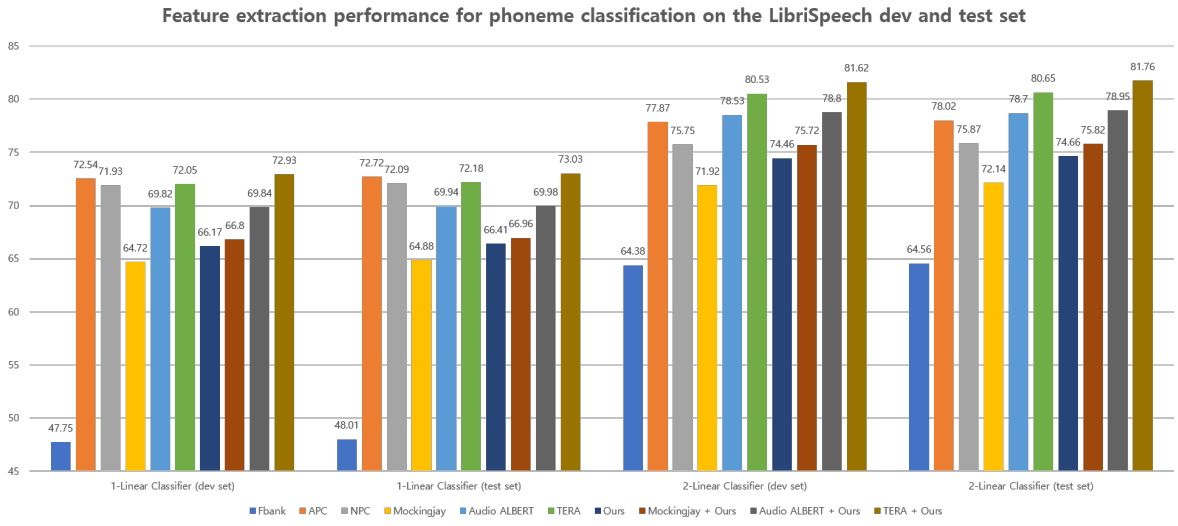
Table 2. Details of various self-supervised speech representation approaches. Nonparallel and Parallel denote the using Non-Transformer and Transformer-based neural network architecture for pretraining, respectively. * represents the pretrained model provided in the S3PRL toolkit. † denotes the pretrained model we implemented ourselves using the S3PRL toolkit.

Representations	Network Type	No. Model Paramaters
Fbank *	-	0
APC * [15]	Non-parallel	9,107,024
NPC * [10]	Non-parallel	19,380,560
Mockingjay * [3]	Non-parallel	22,226,928
Audio ALBERT * [12]	Non-parallel	7,805,264
TERA * [6]	Non-parallel	21,981,008
S&P Patch † (Ours)	Non-parallel	21,981,008
Combined with other representations		
Mockingjay + Ours †	Parallel	22,226,928
Audio ALBERT + Ours †	Parallel	7,805,264
TERA + Ours †	Parallel	21,981,008

5.1. LibriSpeech Phoneme Classification Results

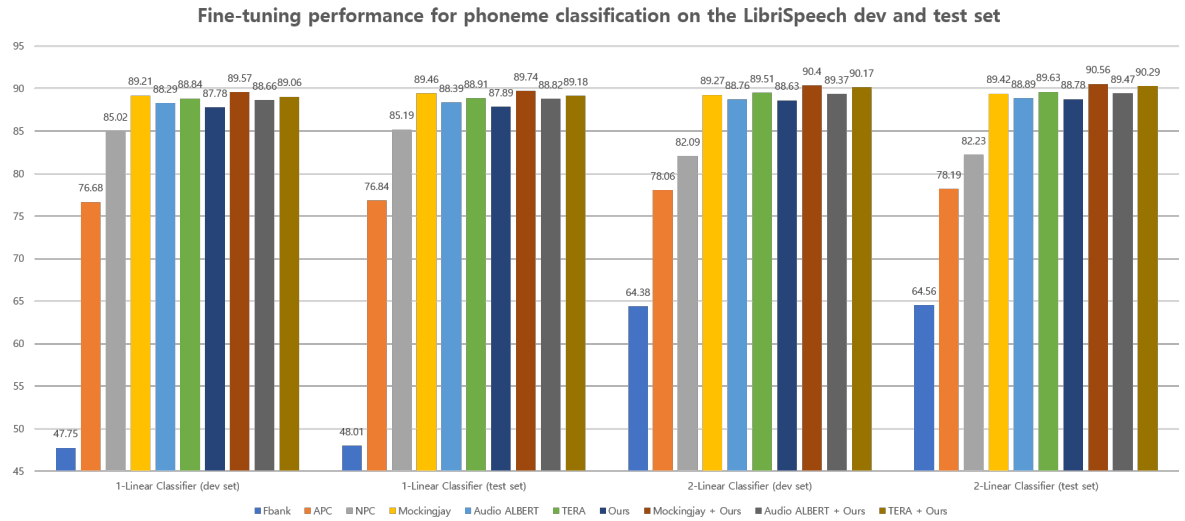
Figure 4 demonstrates the performance of both feature extraction and fine-tuning for the LibriSpeech [33] phoneme classification task using various representations. All the pretrained models of these features are extracted from the final layer. For the feature extraction, all the pretrained model parameters are kept fixed when conducting the downstream task, and the representations are provided as input to the downstream model. In contrast to the feature extraction, all the pretrained model parameters are updated during the fine-tuning experiments on the downstream task.

In Figure 4a, the proposed method outperforms the Mockingjay [3], considering the average results from two methods: 1-linear classifier and 2-linear classifier. It is also observed that combining the proposed method with representations from previous parallel-based approaches yields better performance on average than when they are used alone. Notably, the relatively lower performance of Mockingjay is significantly improved when integrated with the proposed method (Mockingjay + Ours).



(**a**) Feature extraction performance.

Figure 4. *Cont.*



(**b**) Fine-tuning performance.

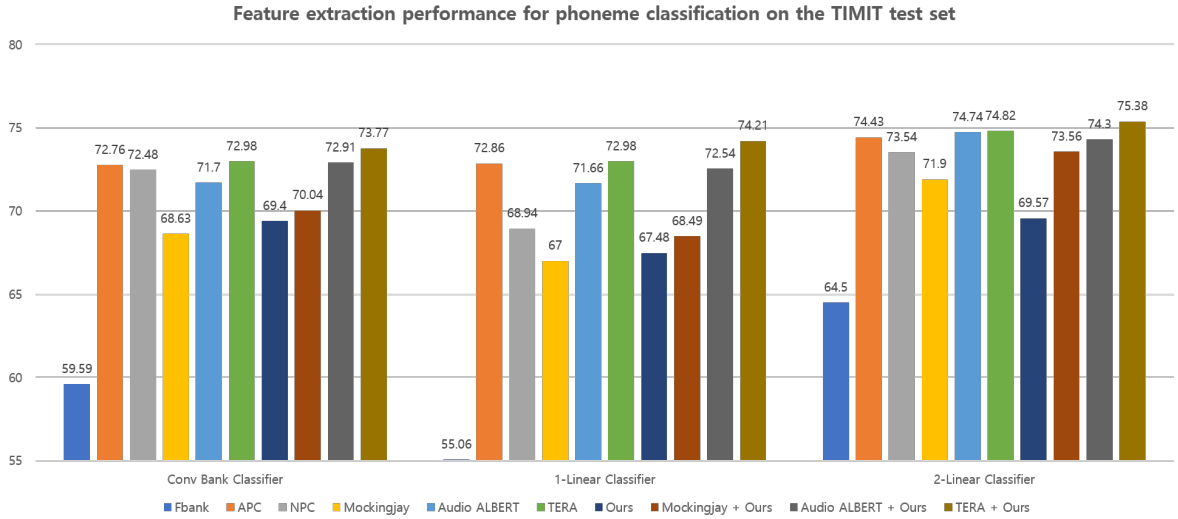
Figure 4. Feature extraction (**a**) and fine-tuning (**b**) performance on LibriSpeech [33] phoneme classification task between different representations. The higher the better.

According to the results presented in Figure 4, the fine-tuning performance (Figure 4b) of all representations surpasses the performance observed on the feature extraction performance (Figure 4a). Particularly, the proposed method demonstrates notable improvement when compared with its performance in the feature extraction stage when used alone. Furthermore, it is observed that the proposed method exhibits significant enhancement compared with feature extraction, outperforming the APC [15] and NPC [10] approaches in terms of average results from the 1-linear and 2-linear classifier. Additionally, it is discovered that the proposed method shows synergistic performance when combined with other methods for fine-tuning, similar to its performance in the feature extraction stage.

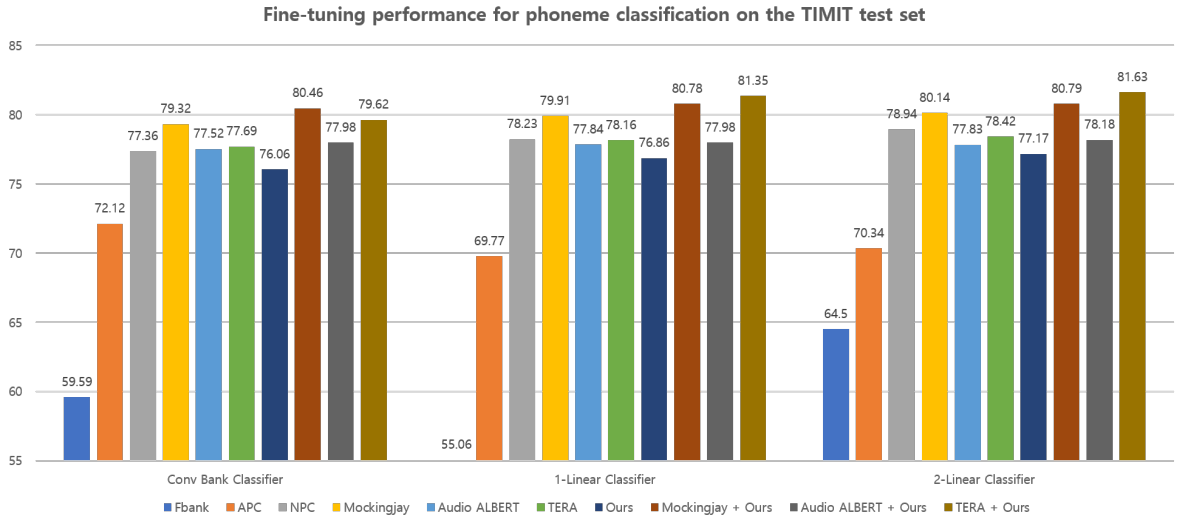
5.2. TIMIT Phoneme Classification Results

The performance of both feature extraction and fine-tuning for the TIMIT [40] phoneme classification task using different representations are presented in Figure 5. As mentioned previously, the conv bank classifier results for both the feature extraction and fine-tuning stages are added to Figure 5. Among the various pretrained representations, the proposed spectral S&P patch masking achieves the lowest performance among other pretraining approaches on the feature extraction performance (Figure 5a) but outperforms the APC on the fine-tuning performance (Figure 5b). However, when combined with other methods, it exhibits effective performance, resulting in improved overall performance. Particularly, it is found that when the pretrained models using the proposed method on the LibriSpeech dataset are applied to a downstream task with a different domain (TIMIT dataset), they still demonstrate a synergistic effect.

Furthermore, it is observed that all representations exhibit improved performance when used with a deeper downstream model, both in the context of feature extraction and fine-tuning. This can be a restricted labeled data environment, as observed in the TIMIT dataset. Specifically, the performance achieved with the 2-linear classifier outperforms that of the conv bank classifier and the 1-linear classifier for all representations. This observation suggests that the 2-linear classifier is capable of extracting more informative features compared with 1-linear classifier, leading to enhanced performance across all representations. As a result, it is worth noticing that the pretrained model combined with the proposed method and previous approaches for both feature extraction and fine-tuning, even with limited labeled data (TIMIT dataset), shows synergistically to further improve the overall performance.



(**a**) Feature extraction performance.



(**b**) Fine-tuning performance.

Figure 5. Feature extraction (**a**) and fine-tuning (**b**) performance on TIMIT [40] phoneme classification task between different representations. The higher the better.

5.3. Keyword Spotting Results

Figure 6 provides an overview of the keyword classification performance on the Speech Commands [41] dataset, considering both feature extraction and fine-tuning tasks. Overall, the results indicate that fine-tuning demonstrates higher performance compared with feature extraction. Interestingly, the proposed method achieves the lowest keyword spotting performance (80.40%) among the pretrained representations in the feature extraction but achieves 92.86% accuracy, which outperforms the APC [15] (90.45%) and NPC [10] (90.34%) methods when applied in the fine-tuning stage. Furthermore, a substantial improvement is noted when the proposed method is combined with other representations, compared with using them individually. Specifically, it is observed that the combination of the proposed method with TERA [6], Audio ALBERT [12], and Mockingjay [3] leads to an average performance boost, respectively, compared with using TERA, Audio ALBERT, and Mockingjay alone.

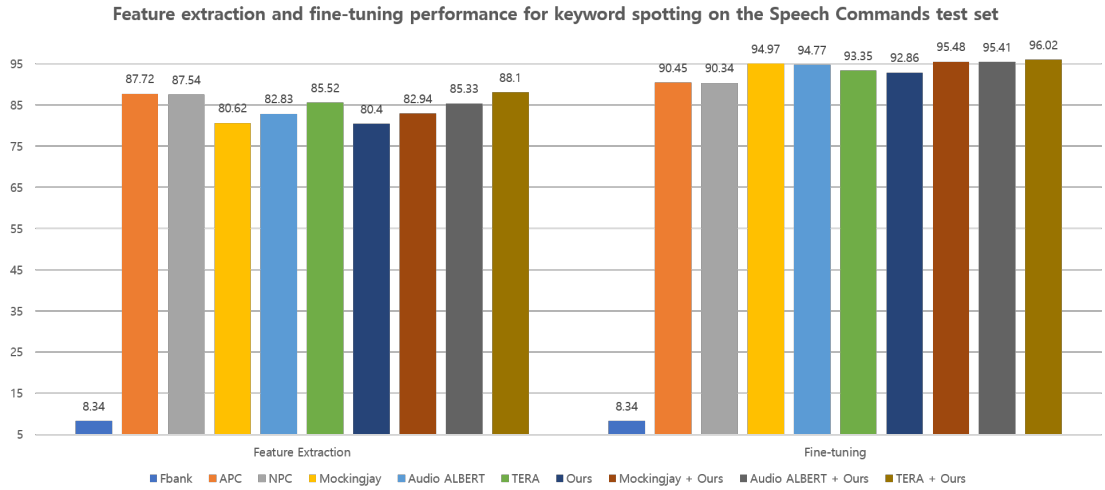


Figure 6. Feature extraction and fine-tuning performance for keyword spotting task on the Speech Commands [41] dataset between different representations, respectively. The higher the better.

5.4. Speaker Identification Results

Figure 7 summarizes the speaker identification results for the VoxCeleb1 [42] dataset across feature extraction and fine-tuning experiments. During feature extraction, the proposed spectral S&P patch masking showed significantly outperforms the non-parallel-based architectures of APC [15] and NPC [10]. Surprisingly, all representations exhibit performance degradation when fine-tuning is applied to speaker identification downstream tasks. Despite following the same settings as described in [39,46], it is conjectured that the chosen hyperparameters for fine-tuning may not be suitable. Interestingly, the fine-tuning performance using the proposed method outperforms that of NPC [10], as well as the parallel-based architectures of Mockingjay [3] and Audio ALBERT [12]. Moreover, experimental results indicate that combining the proposed method with other feature representations leads to improved overall performance. This suggests that the proposed spectral S&P patch masking can serve as simple yet effective self-supervised speech representation learning techniques for speaker identification tasks.

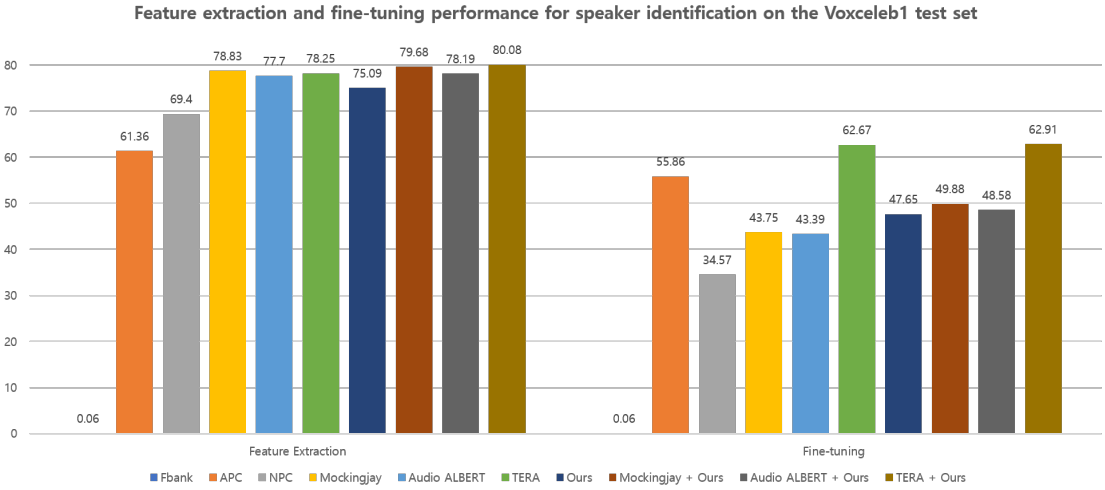


Figure 7. Feature extraction and fine-tuning performance for speaker identification task on the Voxceleb1 [42] dataset between different representations, respectively. The higher the better.

5.5. English ASR Results

In this section, the proposed method and the comparison results in terms of the Word Error Rate (WER) metric are presented. The ASR models are trained using feature extraction and fine-tuning approaches, respectively. All methods are pretrained with

960 h of the LibriSpeech [33] dataset. For decoding and rescoring, the setup described in Section 4.2 is employed for all representations, using both BiGRU and BiLSTM models as ASR architectures.

In Tables 3 and 4, a summary of the results obtained from the feature extraction and fine-tuning approaches using both BiGRU and BiLSTM frameworks is presented. These tables include results from previous literature, as well as the proposed method. Overall, when comparing the results from feature extraction and fine-tuning, it is consistently observed that fine-tuning yields superior performance compared with using the extracted speech feature representations for ASR downstream tasks. Similar to the findings in the LibriSpeech phoneme classification task described in Figure 4, a significant improvement in the performance of the proposed method is observed when transferring from feature extraction to fine-tuning, especially outperforming the non-parallel-based approaches [10,15].

Table 3. Feature extraction and fine-tuning performance on LibriSpeech [33] ASR downstream task using BiGRU network between different representations, respectively. The lower the better (↓).

Representations	Feature Extraction (↓)		Fine-Tuning (↓)		Average (↓)	
	WER	Rescore	WER	Rescore	WER	Rescore
Fbank	27.90	18.42	27.90	18.42	27.90	18.42
APC [15]	23.66	16.58	21.44	15.37	22.55	15.98
NPC [10]	24.18	16.25	21.20	14.55	22.69	15.40
Mockingjay [3]	26.45	17.59	19.48	14.43	22.97	16.01
Audio ALBERT [12]	24.32	16.14	19.16	14.27	21.74	15.21
TERA [6]	22.47	14.96	19.95	14.16	21.21	14.56
Ours	26.35	16.83	20.39	14.52	23.37	15.68
Combined with other representations						
Mockingjay + Ours	26.35	16.83	20.39	14.52	23.37	15.68
Audio ALBERT + Ours	26.23	17.30	19.25	14.05	22.74	15.68
TERA + Ours	21.74	14.04	17.78	13.03	19.76	13.54

Table 4. Feature extraction and fine-tuning performance on LibriSpeech [33] ASR downstream task using BiLSTM network between different representations, respectively. The lower the better (↓).

Representations	Feature Extraction (↓)		Fine-Tuning (↓)		Average (↓)	
	WER	Rescore	WER	Rescore	WER	Rescore
Fbank	22.89	15.35	22.89	15.35	22.89	15.35
APC [15]	21.94	15.32	19.18	13.26	20.56	14.29
NPC [10]	22.21	15.45	19.42	13.36	20.82	14.41
Mockingjay [3]	21.56	15.31	17.75	12.52	19.66	13.92
Audio ALBERT [12]	21.13	14.54	17.20	12.25	19.17	13.40
TERA [6]	19.90	13.33	17.18	12.06	18.54	12.70
Ours	22.03	15.89	17.88	13.02	19.96	14.46
Combined with other representations						
Mockingjay + Ours	21.22	15.06	17.31	12.42	19.27	13.74
Audio ALBERT + Ours	20.58	14.24	17.03	12.02	18.81	13.13
TERA + Ours	18.02	12.94	16.37	11.51	17.20	12.23

In conclusion, it is found that the overall performance of the BiLSTM-based ASR model in Table 3 exhibits notably better performance compared with the BiGRU model

in Table 4. Furthermore, it is also observed that combining the proposed method with other approaches leads to increased ASR performance in both the feature extraction and fine-tuning stages.

5.6. Korean ASR Results

To measure the proposed spectral S&P patch masking on the Korean ASR downstream task, we compare previous works that were pretrained on the KsponSpeech [34] dataset. For training the Korean ASR downstream task, each pretrained model is utilized as a feature extractor for the respective ASR model. All the pretrained comparisons experimental results use the S3PRL toolkit [46]. We report the CER (%) on the KsponSpeech dev set.

Table 5 demonstrates the performance of feature extraction for the overall Korean ASR performance, including the proposed method. According to our experimental results, the ASR performance using the proposed spectral S&P patch masking alone achieves a 14.66% CER, which outperforms the NPC [10] (14.78%), Mockingjay [3] (17.25%), and Audio ALBERT [12] (16.95%), respectively. This indicates that the proposed spectral S&P patch can be simple yet effective for masking strategy, which can be useful for ASR downstream tasks.

Table 5. Feature extraction performance on Korean ASR downstream task. The lower the better (↓).

Representations	**CER (↓)**
Fbank	15.31
APC [15]	13.36
NPC [10]	14.78
Mockingjay [3]	16.95
Audio ALBERT [12]	17.25
TERA [6]	13.86
SVR1K [50]	12.32
Ours	14.66
Combined with other representations	
Mockingjay + Ours	14.83
Audio ALBERT + Ours	15.87
TERA + Ours	12.14

In addition, while the ASR performances of conventional Mockingjay [3], Audio ALBERT [12], and TERA [6] are 16.95%, 17.25%, and 13.86% of CER, respectively, each method combined with the proposed S&P patch masking obtain performances of 14.83%, 15.87%, and 12.14%, which implies a relative average improvement of 10%. In particular, when the proposed method was added to TERA (TERA + Ours), it outperformed the results of SVR1K [50], which had the best ASR performance among all speech representations. As a result, our results demonstrate that combining the proposed spectral S&P patch with conventional masking methods is a useful supplement to the existing self-supervised techniques for speech representation learning.

5.7. Ablation: Impact of Two S&P Patch Masking Hyperparameters

In this section, an ablation study is performed to further explore the effectiveness of the proposed spectral S&P patch masking for self-supervised speech pretraining. Specifically, we experiment and report on two hyperparameter factors: the total amount of the S&P patch α and the consecutive noise patches factor C. The goal is to investigate the contribution of quadrilateral-shaped noise patches in the speech domain. To this end, ablation studies are conducted on the LibriSpeech phoneme classification task for both feature extraction and fine-tuning, as well as on the Korean ASR task for the feature extraction task.

First, an investigation is conducted on six different values of α and eight different values of C in the TERA + Ours setting for the LibriSpeech phoneme classification tasks. According to the phoneme classification accuracy results obtained from the LibriSpeech

test-clean set, as presented in Table 6a,b, our experimental findings indicate that when using the parameters $\alpha = 0.004$ and $C \sim [3, 5]$, the most promising outcomes are achieved for both feature extraction and fine-tuning processes. In our findings, it is observed that there is no significant difference in the fine-tuning performance when varying the hyperparameters. However, a notable gap is noticed in the feature extraction performance across different hyperparameter settings, especially in C. This observation supports the hypothesis that a point-shaped S&P patch ($C = 1$) is not effective for learning speech feature representation.

Table 6. Ablation study on hyperparameters of the proposed S&P noise patches. The higher the better (↑). (**a**) LibriSpeech phoneme classification for both feature extraction and fine-tuning performance comparison according to the various α. (**b**) LibriSpeech phoneme classification for both feature extraction and fine-tuning performance comparison according to the various C. Bold denotes the best result.

(**a**)			
α	C	**Feature Extraction (↑)**	**Fine-Tuning (↑)**
0.001	$C \sim [3, 5]$	68.91	88.06
0.002		70.29	88.46
0.004		**73.03**	**89.18**
0.006		71.07	88.85
0.008		69.97	88.75
0.01		69.02	88.14
(**b**)			
α	C	**Feature Extraction (↑)**	**Fine-Tuning (↑)**
0.004	$C = 1$	70.84	87.89
	$C \sim [1, 3]$	71.82	88.13
	$C = 3$	71.70	88.49
	$\boldsymbol{C \sim [3, 5]}$	**73.03**	**89.18**
	$C \sim [3, 8]$	72.10	88.87
	$C = 5$	71.59	88.49
	$C \sim [1, 10]$	68.85	88.44
	$C \sim [3, 10]$	71.76	88.86

As shown in Table 7, we also explore five different values of α and four different values of C in the TERA + Ours setting for the Korean ASR downstream task. Similar to Table 6, the best ASR performance is obtained when using the $\alpha = 0.004$ and $C \sim [3, 5]$ settings. In particular, it is observed that too small (point-shaped) or too large amounts of S&P patches are not suitable for self-supervised pretext tasks. Additionally, considering that the conventional pointwise ($C = 1$) S&P patch has a CER of 17.25%, the results provide evidence that the proposed spectral S&P patch masking is highly effective. These findings indicate that the consecutive patch masking factor C plays a crucial role in shaping the effectiveness of the spectral S&P patch masking for speech representation learning.

According to the ablation study results for English and Korean ASR presented in Tables 6 and 7, respectively, the optimal hyperparameter configuration that yielded the best performance is $\alpha = 0.004$ and $C \sim [3, 5]$. Indeed, the results obtained from using the representations of the fixed pretrained model without any parameter updates provide support for the hypothesis that a point-shaped S&P patch with $C = 1$ is not effective for speech representation learning. We also highlight the importance of using the proposed quadrilateral-shaped patches to achieve effective speech feature representation learning.

Table 7. Ablation study on hyperparameters of the proposed spectral S&P patch masking. The lower the better (↓). (**a**) ASR performance comparison according to the various α. (**b**) ASR performance comparison according to the various C. **Bold** denotes the best result.

(**a**)		
α	C	**CER (↓)**
0.002	$C \sim [3, 5]$	13.84
0.004		12.14
0.006		13.58
0.008		13.75
0.01		13.34
(**b**)		
α	C	**CER (↓)**
0.004	$C \sim [1, 3]$	17.25
	$\boldsymbol{C \sim [3, 5]}$	12.14
	$C \sim [3, 8]$	13.51
	$C \sim [3, 10]$	15.82

6. Discussion

The primary goal of this paper is to successfully apply the spectral S&P patch masking method for self-supervised speech representation learning. To the best of our knowledge, this study is the first attempt to utilize the S&P patch as a masking strategy for self-supervised speech representation learning. To address the difference in resolution or scale between speech and image data, we have introduced the novel salt value and consecutive quadrilateral-shaped patches for masking, which allows for the useful extraction of continuous information from the speech input.

In our experiments, we conducted an extensive investigation of a diverse range of speech downstream tasks using the proposed method. Firstly, we obtained two pretrained speech representation models on both English and Korean datasets. To ensure a fair comparison of the proposed method, various pretrained weights (models) available in the S3PRL toolkit were directly utilized, and the performance gap across different downstream tasks was reported. As a result, the proposed method achieved similar performance to other recent approaches when used alone for various downstream tasks. Furthermore, we have shown that combining the proposed spectral S&P patch masking with conventional methods for self-supervised speech representation learning leads to effective results in various downstream tasks.

The primary limitation of our study lies in the fact that the proposed spectral S&P patch masking method can only be utilized or combined with masking-based speech representation learning approaches. To gain a more profound understanding of speech representation pretraining, further research is required to extend the proposed method, such as investigating its applicability to directly extracting speech using CNNs like wav2vec 2.0 [2] and HuBERT [5]. Second, in our study, we limited the pretraining to 960 h of the LibriSpeech [33] dataset for the English ASR downstream task. To ensure fair comparisons with other conventional masking-based approaches, we utilized the S3PRL framework using 960 h of LibriSpeech dataset, as described in TERA and SUPERB. However, the amount of data used for pretraining has a significant impact on the performance of models in various downstream tasks. Even when using a large-scale speech dataset such as 60,000 h of Libri-Light [57], further validation is required so verify whether the proposed method shows reliable results. This will enable us to estimate the method's scalability and its potential to yield robust results under more data-rich conditions. Third, we observed that keeping the pretrained weights frozen worked better than fine-tuning them during the training of the Korean ASR model. In the future, we plan to concentrate on addressing these limitations and potentially striving to improve the performance further.

In conclusion, our findings demonstrated that the proposed spectral S&P patch masking shows reasonable performance on several downstream tasks, but it significantly enhanced effectiveness particularly when combined with other conventional approaches. We believe that the proposed methods can be extended to various benchmarks and downstream tasks.

7. Conclusions

In this paper, we introduced simple yet effective spectral S&P patch masking for self-supervised speech representation learning. In order to handle the difference in resolution or scale between spectrograms and images, we also suggested consecutive quadrilateral-shaped patches that extract the continuous information from spectrograms. Experimental results demonstrate the effectiveness of the proposed method on several speech downstream tasks and show it can be a useful supplement to existing self-supervised techniques for speech representation learning.

In our experiments, we observed that keeping the pretrained weights frozen worked better than fine-tuning them during the training of the Korean ASR model. Moreover, further studies are required to ensure that the proposed spectral S&P method can surpass masking-based approaches and encompass direct waveform extraction using CNNs or pretraining with quantizers. In the future, we plan to focus on these issues and potentially improve the performance further.

Author Contributions: Conceptualization, J.-W.K.; methodology, J.-W.K.; software, J.-W.K.; validation, J.-W.K., H.C., and H.-Y.J.; formal analysis, J.-W.K.; investigation, J.-W.K.; resources, J.-W.K. and H.-Y.J.; data curation, J.-W.K.; writing—original draft preparation, J.-W.K.; writing—review and editing, J.-W.K. and H.-Y.J.; visualization, J.-W.K.; supervision, J.-W.K. and H.-Y.J.; project administration, J.-W.K., H.C., and H.-Y.J.; funding acquisition, H.C. and H.-Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Research Foundation (NRF), Korea, under project BK21 FOUR, in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2020-0-01808) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation), and in part by IITP grant funded by the Korea government (MSIT) (2019-0-00004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The LibriSpeech dataset is available at [33], the TIMIT dataset is available at [40], the Speech Commands dataset is available at [41], the VoxCeleb1 dataset is available at [42], and the KsponSpeech dataset is available at [34].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
2. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12449–12460.
3. Liu, A.T.; Yang, S.w.; Chi, P.H.; Hsu, P.c.; Lee, H.y. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6419–6423.
4. Chung, Y.A.; Zhang, Y.; Han, W.; Chiu, C.C.; Qin, J.; Pang, R.; Wu, Y. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Cartagena, Colombia, 13–17 December 2021; pp. 244–250.
5. Hsu, W.N.; Bolte, B.; Tsai, Y.H.H.; Lakhotia, K.; Salakhutdinov, R.; Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3451–3460. [CrossRef]

6. Liu, A.T.; Li, S.W.; Lee, H.y. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 2351–2366. [CrossRef]
7. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
8. Ling, S.; Liu, Y.; Salazar, J.; Kirchhoff, K. Deep contextualized acoustic representations for semi-supervised speech recognition. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6429–6433.
9. Wang, W.; Tang, Q.; Livescu, K. Unsupervised pre-training of bidirectional speech encoders via masked reconstruction. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6889–6893.
10. Liu, A.H.; Chung, Y.A.; Glass, J. Non-Autoregressive Predictive Coding for Learning Speech Representations from Local Dependencies. *Proc. Interspeech 2021* **2021**, 3730–3734. [CrossRef]
11. Ling, S.; Liu, Y. Decoar 2.0: Deep contextualized acoustic representations with vector quantization. *arXiv* **2020**, arXiv:2012.06659.
12. Chi, P.H.; Chung, P.H.; Wu, T.H.; Hsieh, C.C.; Chen, Y.H.; Li, S.W.; Lee, H.y. Audio albert: A lite bert for self-supervised learning of audio representation. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19–22 January 2021; pp. 344–350.
13. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
14. Schneider, S.; Baevski, A.; Collobert, R.; Auli, M. wav2vec: Unsupervised Pre-Training for Speech Recognition. *Proc. Interspeech 2019* **2019**, 3465–3469. [CrossRef]
15. Chung, Y.A.; Hsu, W.N.; Tang, H.; Glass, J. An Unsupervised Autoregressive Model for Speech Representation Learning. *Proc. Interspeech 2019* **2019**, 146–150.
16. Gunel, B.; Du, J.; Conneau, A.; Stoyanov, V. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. *arXiv* **2020**, arXiv:2011.01403.
17. Kim, T.; Yoo, K.M.; Lee, S.g. Self-Guided Contrastive Learning for BERT Sentence Representations. *arXiv* **2021**, arXiv:2106.07345.
18. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International conference on machine learning, PMLR, Virtual, 13–18 July 2020; pp. 1597–1607.
19. 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, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
20. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap your own latent-a new approach to self-supervised learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21271–21284.
21. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18661–18673.
22. Palanisamy, K.; Singhania, D.; Yao, A. Rethinking CNN models for audio classification. *arXiv* **2020**, arXiv:2007.11154.
23. Gong, Y.; Chung, Y.A.; Glass, J. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3292–3306. [CrossRef]
24. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
25. Gong, Y.; Chung, Y.A.; Glass, J. AST: Audio Spectrogram Transformer. *Proc. Interspeech 2021* **2021**, 571–575. [CrossRef]
26. Gong, Y.; Lai, C.I.; Chung, Y.A.; Glass, J. Ssast: Self-supervised audio spectrogram transformer. *AAAI Conf. Artif. Intell.* **2022**, *36*, 10699–10709. [CrossRef]
27. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. *Proc. Interspeech 2019* **2019**, 2613–2617. [CrossRef]
28. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552.
29. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A.; Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.
30. Agostinelli, F.; Anderson, M.R.; Lee, H. Adaptive multi-column deep neural networks with application to robust image denoising. *Adv. Neural Inf. Process. Syst.* **2013**, *26*.
31. Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; Aila, T. Noise2Noise: Learning Image Restoration without Clean Data. *arXiv* **2018**, arXiv:1803.04189.
32. Liang, L.; Deng, S.; Gueguen, L.; Wei, M.; Wu, X.; Qin, J. Convolutional neural network with median layers for denoising salt-and-pepper contaminations. *Neurocomputing* **2021**, *442*, 26–35. [CrossRef]
33. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An asr corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 5206–5210.
34. Bang, J.U.; Yun, S.; Kim, S.H.; Choi, M.Y.; Lee, M.K.; Kim, Y.J.; Kim, D.H.; Park, J.; Lee, Y.J.; Kim, S.H. Ksponspeech: Korean spontaneous speech corpus for automatic speech recognition. *Appl. Sci.* **2020**, *10*, 6936. [CrossRef]
35. Chan, R.H.; Ho, C.W.; Nikolova, M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Trans. Image Process.* **2005**, *14*, 1479–1485. [CrossRef]

36. Esakkirajan, S.; Veerakumar, T.; Subramanyam, A.N.; PremChand, C. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. *IEEE Signal Process. Lett.* **2011**, *18*, 287–290. [CrossRef]
37. Erhan, D.; Courville, A.; Bengio, Y.; Vincent, P. Why does unsupervised pre-training help deep learning? In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 201–208.
38. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
39. S3PRL Speech Toolkit (S3PRL). Github. Available online: https://github.com/s3prl/s3prl (accessed on 9 June 2023).
40. Garofolo, J.S.; Lamel, L.F.; Fisher, W.M.; Fiscus, J.G.; Pallett, D.S. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Tech. Rep. N* **1993**, *93*, 27403.
41. Warden, P. Speech Commands: A Public Dataset for Single-Word Speech Recognition. Available online: http://download.tensorflow.org/data/speech_commands_v0 (accessed on 9 June 2023).
42. Nagrani, A.; Chung, J.S.; Xie, W.; Zisserman, A. Voxceleb: Large-scale speaker verification in the wild. *Comput. Speech Lang.* **2020**, *60*, 101027. [CrossRef]
43. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101.
44. Lee, K.F.; Hon, H.W. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 1641–1648. [CrossRef]
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
46. wen Yang, S.; Chi, P.H.; Chuang, Y.S.; Lai, C.I.J.; Lakhotia, K.; Lin, Y.Y.; Liu, A.T.; Shi, J.; Chang, X.; Lin, G.T.; et al. SUPERB: Speech Processing Universal PERformance Benchmark. *Proc. Interspeech 2021* **2021**, 1194–1198. [CrossRef]
47. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376.
48. Heafield, K. KenLM: Faster and smaller language model queries. In Proceedings of the Sixth Workshop on Statistical Machine Translation, Edinburgh, UK, 30–31 July 2011; pp. 187–197.
49. Pratap, V.; Hannun, A.; Xu, Q.; Cai, J.; Kahn, J.; Synnaeve, G.; Liptchinsky, V.; Collobert, R. Wav2letter++: A fast open-source speech recognition system. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 6460–6464.
50. Kim, J.W.; Chung, H.; Jung, H.Y. Unsupervised Representation Learning with Task-Agnostic Feature Masking for Robust End-to-End Speech Recognition. *Mathematics* **2023**, *11*, 622. [CrossRef]
51. Watanabe, S.; Hori, T.; Karita, S.; Hayashi, T.; Nishitoba, J.; Unno, Y.; Enrique Yalta Soplin, N.; Heymann, J.; Wiesner, M.; Chen, N.; et al. ESPnet: End-to-End Speech Processing Toolkit. *Proc. Interspeech 2018* **2018**, 2207–2211. [CrossRef]
52. Müller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
53. Yujian, L.; Bo, L. A normalized Levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1091–1095. [CrossRef]
54. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
55. Yang, Y.Y.; Hira, M.; Ni, Z.; Astafurov, A.; Chen, C.; Puhrsch, C.; Pollack, D.; Genzel, D.; Greenberg, D.; Yang, E.Z.; et al. Torchaudio: Building blocks for audio and speech processing. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 6982–6986.
56. Harris, C.R.; Millman, K.J.; Van Der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]
57. Kahn, J.; Rivière, M.; Zheng, W.; Kharitonov, E.; Xu, Q.; Mazaré, P.E.; Karadayi, J.; Liptchinsky, V.; Collobert, R.; Fuegen, C.; et al. Libri-light: A benchmark for asr with limited or no supervision. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7669–7673.