electronics

MDPI

Article

Adapting Pre-Trained Self-Supervised Learning Model for Speech Recognition with Light-Weight Adapters

Xianghu Yue [1,2], Xiaoxue Gao [1,*], Xinyuan Qian [3] and Haizhou Li [1,2,4]

[1] Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583, Singapore; xianghu.yue@u.nus.edu (X.Y.); haizhouli@cuhk.edu.cn (H.L.)
[2] School of Data Science, The Chinese University of Hong Kong, Shenzhen 518172, China
[3] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China; qianxy@ustb.edu.cn
[4] Shenzhen Research Institute of Big Data, Shenzhen 518172, China
* Correspondence: xiaoxue.gao@u.nus.edu

Abstract: Self-supervised learning (SSL) is an effective way of learning rich and transferable speech representations from unlabeled data to benefit downstream tasks. However, effectively incorporating a pre-trained SSL model into an automatic speech recognition (ASR) system remains challenging. In this paper, we propose a network architecture with light-weight adapters to adapt a pre-trained SSL model for an end-to-end (E2E) ASR. An adapter is introduced in each SSL network layer and trained on the downstream ASR task, while the parameters of the pre-trained SSL network layers remain unchanged. By carrying over all pre-trained parameters, we avoid the catastrophic forgetting problem. At the same time, we allow the network to quickly adapt to ASR task with light-weight adapters. The experiments using LibriSpeech and Wall Street Journal (WSJ) datasets show that (1) the proposed adapter-based fine-tuning consistently outperforms full-fledged training in low-resource scenarios, with up to 17.5%/12.2% relative word error rate (WER) reduction on the 10 min LibriSpeech split; (2) the adapter-based adaptation also shows competitive performance in high-resource scenarios, which further validates the effectiveness of the adapters.

Keywords: self-supervised learning; automatic speech recognition; domain adaptation

Citation: Yue, X.; Gao, X.; Qian, X.; Li, H. Adapting Pre-Trained Self-Supervised Learning Model for Speech Recognition with Light-Weight Adapters. *Electronics* **2024**, *13*, 190. https://doi.org/10.3390/electronics13010190

Academic Editor: Chiman Kwan

Received: 17 November 2023
Revised: 22 December 2023
Accepted: 28 December 2023
Published: 1 January 2024

1. Introduction

In recent years, end-to-end (E2E) automatic speech recognition (ASR) has garnered extensive attention [1,2]. It replaces the traditional hybrid ASR pipeline, which typically consists of several independent components, with a single, unified deep neural network (DNN), and directly transcribes speech into text transcription without the need for predefined alignment and a phonetic lexicon [3–5]. E2E ASR models represent a novel modeling process, reducing the complexity inherent in the traditional ASR pipeline. However, despite their advantages, these models typically require a large amount of labeled speech data to achieve acceptable performance. This poses a considerable challenge, especially for low-resource languages, where such extensive transcribed speech data are not readily available.

Self-supervised learning (SSL) has proven its efficacy in capturing rich and transferable representations from unannotated speech data, thereby significantly benefiting downstream speech-related tasks [6–8]. This approach has been proven particularly effective for the pre-training of E2E ASR models on large amounts of untranscribed speech data, which are subsequently fine-tuned on parallel labeled speech data for target applications [9–12]. SSL leverages unsupervised pre-training strategy that allows it to learn high-level semantic information from unlabeled data. The speech representations derived in this manner are typically more robust and versatile than those acquired through supervised learning methods, which can be biased towards the training data [13].

There have been many successful self-supervised methods for learning speech representations, such as Autoregressive Predictive Coding (APC) [10,14], wav2vec [6,15], Problem-agnostic Speech Encoder (PASE) [16,17], Speech SimCLR [18], Speech XL-Net [19], TERA [20], WavLM [21], and data2vec [8], to name a few.

Through self-supervised pre-training, these pre-trained speech models (PSMs) could be applied to downstream tasks through adaptation, typically including feature-based speech representation transfer and fine-tuning. When PSMs are re-used as features extractors, they generate rich and high-level speech representations that capture complex patterns and structures in the speech, thereby reducing the complexity of downstream models and ultimately boosting the downstream performance. On the other hand, the fine-tuning based approaches involve copying the weights from PSMs and initializing part of the downstream models for the subsequent supervised training. Usually, the pre-trained parameters of PSMs provide particularly effective initialization for the encoders of ASR models [9,11,12,22,23], and outperform the feature-based transfer method. In this way, PSMs have achieved great success for E2E ASR, especially in low-resource speech datasets [24,25].

Despite significant progress, few studies have investigated how to effectively adapt a pre-trained self-supervised speech model; that is, the general speech representations, towards a specific ASR task. Simply initializing the ASR encoder with the PSMs parameters and then fine-tuning can lead to a catastrophic forgetting problem [26] and sub-optimal performance. More importantly, fine-tuning of over-parameterized models on a small amount of adaptation data, i.e., a low-resource dataset, is ineffective [27]. Last but not least, fine-tuning the entire model is not parameter efficient, considering the enormous amount of parameters in PSMs [6,20]. For example, the base wav2vec2.0 model contains 12 transformer layers and the large wav2vec2.0 model contains 24 transformer layers. Fine-tuning all these layers can be computationally expensive and inefficient. In the field of natural language processing (NLP), adapters [28,29] have been introduced as an alternative lightweight fine-tuning strategy, consisting of a small set of newly initialized weights at every layer of the transformer. These weights are trained during the fine-tuning process, while the pre-trained parameters remain fixed. This strategy allows for efficient parameter sharing and achieves competitive performance in many text-related tasks. Despite their success in NLP, there has been limited investigation into the applicability of adapters for speech-related tasks.

To mitigate the aforementioned challenges and fill the gap, in this study, we propose a novel approach that will allow E2E ASR models to effectively benefit from SSL pre-trained models. Specifically, our method introduces a light-weight neural network module, termed *adapter*, which serves as a new component inserted into each transformer encoder layer of the pre-trained model to achieve efficient adaptation to the ASR task. During the fine-tuning process, we keep the parameters of the pre-trained model frozen and only update the adapter module. The proposed modeling process is referred to as *fine-tuning with adapters* or FTA. It seeks to obtain the following two primary benefits. (1) By introducing the adapter module, the parameters of the pre-trained model can be disentangled from the task-specific ASR adapters. This leads to a high degree of parameter sharing while simultaneously avoiding the issue of catastrophic forgetting. (2) Due to the light-weight nature of the adapter module, the adaptation process can effectively and efficiently leverage a smaller amount of adaptation data. Regarding the model efficiency, we consider our work related to the quantum tensor approach for speech processing [30], in which the low-complexity hybrid tensor networks are designed for speech enhancement and spoken command recognition tasks, and teacher–student transfer learning methods for speech recognition [31]. The authors of [30] design a low-rank tensor-train deep neural network with fewer model parameters for practical application, while our proposed method aims to efficiently adapt the existing SSL pre-trained models for the downstream ASR task.

We evaluate the proposed FTA modeling process on Libri-light [32], LibriSpeech [33] and Wall Street Journal (WSJ) [34] datasets. Our experimental results show that, compared with the standard full fine-tuning approach, FTA reduces the word error rate (WER) by up

to 17.5%/12.2% on the 10 min subset of LibriSpeech when decoding without a language model (LM) in a low-resource scenario. Furthermore, in high-resource scenarios, our proposed method also delivers competitive results, while introducing only a minimal number of additional trainable parameters.

The rest of the paper is structured as follows. Section 2 introduces related works and background knowledge. In Section 3, we formulate the fine-tuning with adapters (FTA) modeling process in detail. In Section 4, we outline our experimental setup, followed by a comprehensive discussion and analysis of our experimental results in Section 5. Lastly, we conclude this paper in Section 6.

2. Related Work

2.1. Pre-Trained Speech Models

Pre-trained speech models (PSMs) aim to learn robust and contextually rich speech representations from extensive large unlabeled speech corpora through self-supervised learning [6,8–10,14,20,35,36]. They have significantly enhanced the performance of various speech-related tasks; typical examples are speech recognition [11,12], speaker recognition [37,38], and speech translation [39,40]. Among these methodologies, wav2vec2.0 [6] is one of the most prevalent pre-training approaches. Its pre-training objective is composed of contrastive loss [35] and diversity loss. The principle of contrastive learning has been widely applied to a variety of other tasks, including speaker recognition [41], image [35,42] and text [43] representation learning.

In wav2vec2.0, which is described in Figure 1, the self-supervised loss utilized for pre-training can be interpreted as a contrastive predictive coding (CPC) [35] loss. The task involves predicting masked encoder features [44] (e.g., BERT) rather than predicting near-future encoder features given previous ones. Specifically, given an input raw audio sequence $\mathbf{x}$, the convolutional encoder $f : X \rightarrow Z$, first transforms $\mathbf{x}$ into latent speech features $\mathbf{z}$ at a 20 ms stride with a receptive field 30 ms. These latent features $\mathbf{z}$, which are masked with a certain proportion, are then processed by a Transformer-based context network $g : Z \rightarrow C$. This process results in contextualized representations $\mathbf{c}$, which encapsulate information from the whole sequence. Those masked latent speech representations are subsequently quantized into q_t using the quantization module $q : Z \rightarrow Q$ to provide the labels for the masked time step t in the contrastive loss. The contrastive loss is formulated as follows:

$$\mathcal{L}_m = -\log \frac{\exp(sim(c_t, q_t)/k)}{\sum_{\tilde{q} \sim Q_t} \exp(sim(c_t, \tilde{q})/k)} \tag{1}$$

where $sim(a, b) = a^T b / \|a\| \|b\|$ represents the cosine similarity between context representations and discrete latent speech representations, Q_t denotes a collection of $K + 1$ discrete candidate representations $\tilde{q} \in Q_t$, which is composed of q_t and K distractors.

Besides contrastive learning-based pre-training methods, there is an another emerging branch of speech pre-training technique that focuses on reconstruction losses. Among these techniques, the Autoregressive Predictive Coding (APC) method stands out as a notable example, taking inspiration from language models (LMs) commonly used in text processing. The APC model can be viewed as a speech-specific counterpart to an LM, as it employs an autoregressive model to forecast future speech frames based on the temporal information of previous acoustic sequences, similar to recurrent LMs [45]. In [14], an auxiliary objective is further introduced to extend the APC objective to perform the multi-target pre-training. This auxiliary objective works as a regularization term to enhance the generalization of the task of the future frame prediction. Moreover, inspired by techniques like the masked language model (MLM) introduced in BERT [44] and the permutation language model (PLM) introduced in XLNet [46], some works [11,12] have explored the use of BERT-style mask-predict approaches to pre-train ASR speech encoders. These methods adjust natural language processing (NLP) techniques, which are designed for discrete word tokens, to accommodate continuous audio data. Despite yielding remarkable results for a range of speech-related tasks, these models usually require large-scale training datasets, making

them both time-consuming and resource-intensive. In contrast, the focus of this paper is on leveraging publicly available pre-trained speech models (e.g., wav2vec2.0) for E2E ASR, aiming to improve ASR performance by utilizing existing pre-trained models and reducing the need for extensive additional training.

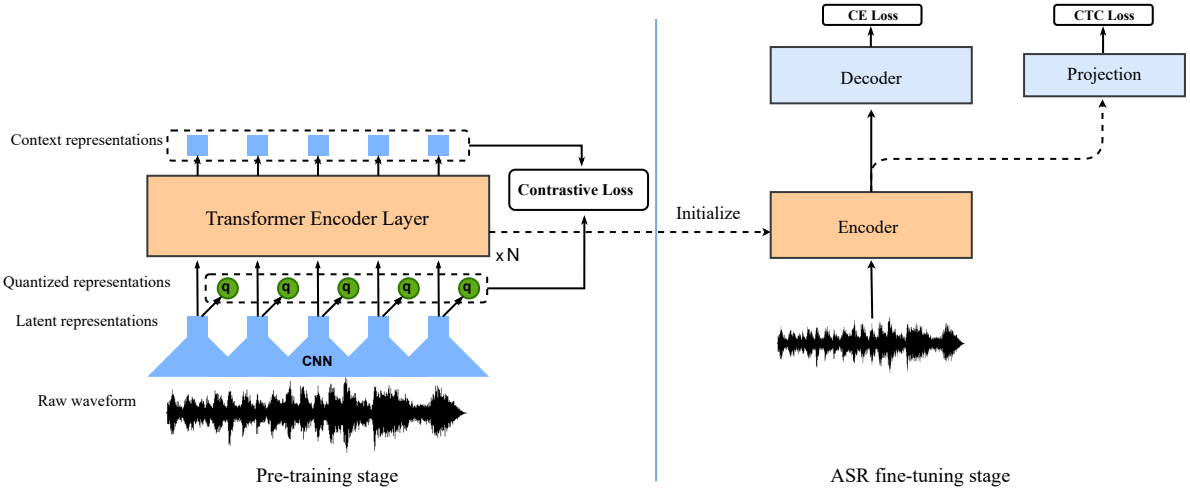

Figure 1. (**Left**:) the architecture of wav2vec2.0 model and its pre-training objective. The model contains a multi-layer convolutional feature encoder and a transformer encoder for learning contextualized speech representations. (**Right**): the method of adapting pre-trained wav2vec2.0 into E2E ASR, including CTC-based and encoder–decoder-based frameworks.

2.2. PSMs in End-to-End Speech Recognition

Several prior studies show that E2E ASR systems benefit from effective feature representations derived from PSMs. In [10,20], PSMs are first pre-trained and then used as feature extractors to replace surface features, such as Mel-Frequency Cepstral Coefficients (MFCC) and filter-bank features, for ASR systems. In [11,12], a BERT-style masked reconstruction strategy is used to pre-train a transformer-based ASR encoder model by leveraging a large amount of unannotated speech data, which are then fine-tuned with a randomly initialized decoder on labeled speech data with cross-entropy loss. In [6,15,47], the ASR encoder is first pre-trained using a contrastive loss and fine-tuned with a CTC loss.

To adapt PSMs towards the ASR tasks, most studies simply fine-tune the whole PSMs with some domain data. To fully benefit from PSMs, how to efficiently and effectively incorporate PSM into an E2E ASR system must be determined. First, fine-tuning PSMs requires careful hyper-parameters tuning; for example, the learning rate can be sensitive and unstable, especially in low-resource datasets. In addition, PSMs are usually very large; a full-fledged fine-tuning of every parameter requires a large dataset.

Motivated by the related work, we propose introducing adapters to improve the fine-tuning process. The idea of light-weight adapters has been successfully applied in fine-tuning pre-trained vision models [48–50] and pre-trained language models [28,29,51]. In this paper, we further extend the idea of *adapters* and explore their application to adapt PSMs within an E2E ASR framework.

3. Adaptation with Light-Weight Adapters

3.1. Transformer Encoder

The main architecture of wav2vec2.0 [6] is the Transformer encoder [52], as shown in Figure 2, which is composed of multi-head attention (MHA), feed-forward layers, layer normalization [53], and residual connections [54]. The MHA module is a crucial component of the encoder, which enables the model to understand the connections between queries,

keys, and values across various representation subspaces positions. The basic unit of the MHA module is the self-attention module, which is computed using the following equation:

$$\text{SelfAtt}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}}) \tag{2}$$

where $Q \in \mathbb{R}^{t_q \times d_q}$, $K \in \mathbb{R}^{t_k \times d_k}$, and $V \in \mathbb{R}^{t_v \times d_v}$ denotes queries, keys and values, respectively. t_* are the element numbers in different inputs and d_* are the corresponding element dimensions.

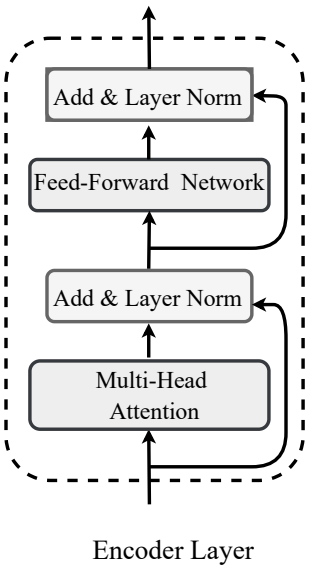

Figure 2. The architecture of transformer encoder layer.

To introduce variations in the attention scores, the mechanism of self-attention can be extended to the MHA version. In the MHA module, multiple sets of queries, keys, and values are generated through linear projections of the input. These projections allow the model to capture different aspects or perspectives of the input data.

$$\text{MHA}(Q, K, V) = \text{Concat}(h_1, \cdots, h_H)W^o$$
$$h_i = \text{SelfAtt}(QW_i^q, KW_i^k, VW_i^v)$$

The queries, keys, and values are converted into subspaces through parameter matrices W_i^q, W_i^k, W_i^v, where i is the index of a head. Then, the self-attention is calculated based on the transformed inputs. Finally, the outputs are concatenated together and multiplied with W^o. h_i is one attention head; H is the number of heads.

The position-wise feed-forward neural network (FFN) transforms the output of the attention at each position with a ReLU activation:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{3}$$

where x is a vector at one position, W_1, W_2, b_1, b_2 are learnable parameters.

The Transformer encoder lacks recurrence and a convolutional layer; therefore, the positional encoding is applied to provide the model with knowledge about the relative positions of the input acoustic sequence. Specifically, the input frames are first projected into the model's dimension and subsequently added with the positional encoding. Rather than using fixed positional embeddings that encode absolute positions, a convolutional layer, akin to those in [22,55], is utilized to provide relatively positional embedding. The convolution's output is then added to the inputs followed by the layer normalization.

3.2. Light-Weight Adapters

To efficiently and effectively transfer the learned knowledge of the pre-trained wav2vec2.0 to E2E ASR models, we introduce light-weight adapters and insert them into each wav2vec2.0 encoder layer. During downstream ASR training, only the parameters of the adapters are updated, while the parameters of wav2vec2.0 model are kept frozen, allowing more flexible architectural modifications to adapt the pre-trained model for the specific downstream task. Therefore, the adapter-based adaptation could mitigate the issue of forgetting. Furthermore, unlike standard fine-tuning, which introduces an entirety new model for the downstream task, the adapter-based adaptation yields a compact model with only a few trainable parameters.

The architecture of a adapter is shown in Figure 3. It consists of a stack of down- and up-scale neural networks that perform dimensionality transformation. The adapter maps the input vector h from dimension d to a lower dimension m, and then re-maps it back to dimension d. The hidden size m of the adapter allows for flexible control over the capacity and efficiency of the adapter layers. To ensure the adapter module approximates an identity function when the projection layers' parameters are close to zero, a residual connection is employed within the adapter network. This connection enables the adapter to preserve the original information by bypassing the projection layers. Formally, given an input hidden vector h, the output vector h' is computed as follows:

$$h' = f_2(\text{ReLU} f_1(h)) + h \tag{4}$$

in which $f_1(\cdot)$ and $f_2(\cdot)$ are the down- and up-projection layers. As discussed in Section 3.1, the information learned by the pre-trained model is mainly preserved in multi-head attention and feed-forward modules. Therefore, two adapters are inserted for each encoder layer of wav2vec2.0 model. Specifically, one adapter is placed after the multi-head attention layer and another follows the feed-forward layer. During adaptation, only the parameters of these adapters, along with the normalization layers and the final projection layer (CTC-based ASR) or the decoder (encoder–decoder-based ASR), are updated.

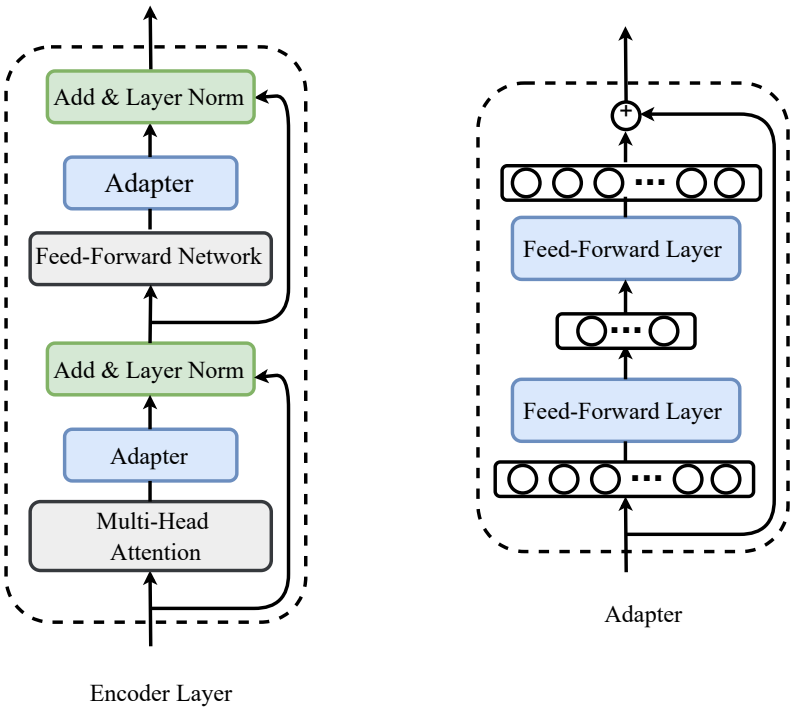

Figure 3. The structure of the proposed adapter module and its combination with the wav2vec2.0 Transformer encoder layer.

The proposed adapter-based adaptation differs from two common adaptation methods, e.g., feature-based transfer learning and full fine-tuning, in the following ways. Given a pre-trained model with parameters $\mathbf{w}$, represented as $\varphi_{\mathbf{w}}(\mathbf{x})$, feature-based transfer involves composing $\varphi_{\mathbf{w}}$ with a newly introduced function $\phi_{\mathbf{u}}$, then obtaining $\phi_{\mathbf{u}}(\varphi_{\mathbf{w}})$. During training, only parameters of the task-specific model, $\mathbf{u}$, will be updated, while parameters of the pre-trained model remain fixed. On the other hand, fine-tuning tries to initialize the task-specific model with pre-trained parameters and then adjust them. For adapter-based adaptation, it defines a new function, $\psi_{\mathbf{w},\mathbf{v}}$, where $\mathbf{w}$ is directly copied from the pre-trained model and remains fixed during tuning. Typically, the adapter parameters, $\mathbf{v}$ are much less than $\mathbf{w}$, yielding light-weight adaptation and fast convergence, especially in low-resource settings.

4. Experiments

4.1. Datasets

To verify the effectiveness of the proposed FTA method, we conduct the experiments using three standard datasets: Libri-light [32], LibriSpeech [33] and Wall Street Journal (WSJ) [34], including low-resource and high-resource scenarios. We use the limited-resource training sets (10 min, 1 h, 10 h) of Libri-light for our low-resource experiments. WSJ and LibriSpeech are considered for high-resource experiments.

4.2. Experimental Setup

We mainly use the publicly released pre-trained base wav2vec2.0 model (https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt, accessed on 24 October 2020) for our experiments, which is pre-trained on 960 h of the full LibriSpeech dataset. This base model comprises 12 transformer encoder layers, 12 self-attention heads, attention dimension 768, and a total of 95 M parameters. The structure of our proposed adapter is a simple two-layer feed-forward layer network with a residual connection. All the code and experiments are performed in *fairseq* toolkit. All our experimental results are obtained without LM decoding.

In CTC-based E2E ASR, the pre-trained wav2vec2.0 is fine-tuned by adding a randomly initialized linear layer on top of the transformer encoder, which maps the output into C classes, corresponding to the vocabulary. In encoder–decoder-based E2E ASR, pre-trained wav2vec2.0 are fine-tuned with a randomly initialized decoder. For the decoder, the number of layer is 6, the number of head of MHA is 8, the attention dimension is 768, and the dimension of the intermediate feed-forward network is 3072. These specifications were mainly chosen based on preliminary experiments and previous successful implementations in related works [6,56]. For both E2E ASR systems, we utilize 29 tokens for character targets, alongside a token that denotes word boundaries. SpecAugment is a simple data augmentation technique used in speech recognition that manipulates the spectrogram of the speech data. It involves time warping, frequency masking, and time masking to create variations in the data, thereby enhancing the model's ability to generalize. We employ the same SpecAugment method in [6], where the mask is applied along the time and channel dimensions during fine-tuning on LibriSpeech and Libri-light, which could delay the overfitting problem and improve the final word error rates (WER) for a fair comparison. For WSJ, no data augmentation method is used in all experiments.

We follow the default setup in [6], in which wav2vec2.0 is fine-tuned for 80 k steps on the 100 h subset, 20 k steps on the 10 h subset, 13 k steps on the 1 h subset and 10 min subset. For WSJ, the number of fine-tuning steps is 50 k. The learning rate increases linearly for the initial 10% of the steps, stays constant for the next 40% of the steps, and then decreases exponentially for the remaining 50% of the steps using Adam algorithm in standard full fine-tuning. For adapter-based adaptation, we use the Adam optimizer

with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$ and adjust the learning rate throughout the training process, following the formula below:

$$lr = d_{model}^{-0.5} \cdot min(step^{-0.5}, step \cdot warmup^{-1.5}) \quad (5)$$

where the warm-up step is set to 5000.

5. Results and Discussion

5.1. Main Results

In Table 1, we investigate the effectiveness of the proposed FTA method in low-resource settings, in which the 10 min, 1 h, and 10 h splits of Libri-light are used. The hidden size of the adapter here is 256, the pre-trained model is wav2vec2.0 base model, and we perform experiments using the CTC-based E2E ASR framework. Here, we mainly report the results of experiments conducted specifically under the CTC-based E2E ASR framework.

We also present the results of Continuous BERT and Discrete BERT [22] here; they are also first pre-trained on the full 960 h LibriSpeech dataset and then fine-tuned using these low-resource settings. As we do not have access to their pre-trained models, we did not apply our proposed FTA for these two models. The results of DeCoAR2.0 [57] and WavLM [21] are also provided; however, it should be noted that both of them are full fine-tuned models for ASR.

From the table, we observe that the proposed FTA method is very effective in low-resource scenarios. For the 10 min split, the proposed FTA achieves 17.5%/12.2% relative WER reduction (38.7% vs. 46.9% on test-clean set, and 50.9% vs. 44.7% on the test-other set). During a 1h split, FTA obtains 21.6%/11.6% relative WER reduction (19.2% vs. 24.5% for the test-clean set, and 26.7% vs. 30.2% for the test-other set). These results suggest that FTA clearly outperforms the whole-model fine-tuning on the low-resource dataset.

Table 1. WER (%) results on the test sets of LibriSpeech when the model is fine-tuned on the 10 min, 1 hour, 10 h subsets of Libri-light and the clean 100h subset of LibriSpeech. The hidden size of the adapter is 256.

Method	Fine-Tuning Data	Test-Clean	Test-Other
Continuous BERT + LM [22]	10 min	49.5	66.3
	1 h	22.4	44.0
	10 h	14.1	34.3
Continuous BERT + LM [22]	10 min	16.3	25.2
	1 h	9.0	17.6
	10 h	5.9	14.1
DeCoAR 2.0 + LM [57]	1 h	13.8	29.1
	10 h	5.4	11.3
	100 h	5.0	12.1
WavLM [21]	1 h	24.5	29.2
	10 h	9.8	16.0
	100 h	5.7	12.0
Whole model fine-tuning [6]	10 min	46.9	50.9
	1 h	24.5	30.2
	10 h	9.4	16.6
	100 h	6.1	13.3
Proposed FTA	10 min	38.7	44.7
	1 h	19.2	26.7
	10 h	9.4	17.0
	100 h	5.4	13.5

5.2. Ablation Study

Another finding from Table 1 is that, as the quantity of training data increases, the benefit of adapters diminishes. We conjecture that the whole model fine-tuning has a more severe overfitting problem in low-resource settings, since it has much more tunable parameters than FTA. The size of the adapter, which is the only adapter-specific hyperparameter, offers a straightforward way to balance the performance against the parameter efficiency. Smaller adapters imply fewer parameters, but this might potentially compromise the performance. In order to investigate this trade-off, we experiment with varying the size of adapter on the 100 h split of LibriSpeech. In Table 2, we show the effect of adapter hidden size, where we find that increasing size leads to marginal performance gains.

In Table 2, we also include other reported results for comparison, including noisy student training [58] and joint training of CPC-CTC [59] here, where the 100 h split subset of LibriSpeech are used as labeled data; the rest are used as unlabeled data. We observe that the proposed FTA is as competitive as whole model fine-tuning, indicating the effectiveness of the adapter in a high-resource setting. In addition, the proposed FTA is more parameter-efficient, with only 14 M tunable parameters when the adapter size is 256, while whole model fine-tuning involves 95 M parameters.

Table 2. WER (%) on the LibriSpeech test sets with different adapter hidden sizes. The model is trained on the clean 100 h subset of LibriSpeech.

Method/WER	**# Params**	**Test-Clean**	**Test-Other**
Noisy student + LM [58]	-	4.2	8.6
Joint CPC-CTC [59]	-	6.2	13.9
Whole model fine-tuning [6]	95 M	6.1	13.3
Proposed FTA			
Adapter_{256}	14 M	5.4	13.5
Adapter_{128}	9 M	6.3	13.7
Adapter_{64}	7 M	6.4	14.1

5.3. Transfer to Encoder–Decoder-Based E2E ASR

The above experiments are based on CTC-based E2E ASR, in which a randomly initialized linear projection is added on top of the pre-trained wav2vec2.0. To appreciate the adaptation ability of the adapters, we use the adapters to incorporate the pre-trained wav2vec2.0 into encoder–decoder-based E2E ASR, in which, instead, a randomly initialized transformer decoder is combined with the pre-trained encoder. During tuning, only the adapters in the encoder module and the decoder module are updated. To explore this, we perform the experiments on the WSJ dataset and the recognition results are shown in Table 3. We also present the results of the state-of-the-art transformer, which is trained only on WSJ data using the ESPnet toolkit [60]. It should be noted that all our results are without LM decoding.

The results suggest that the adapters work well in encoder–decoder E2E ASR by simply incorporating the pre-trained ASR encoder. The FTA method achieves almost the same recognition results with the whole model fine-tuning. In the whole model fine-tuning settings, we also experiment with only fine-tuning the top eight, top six, and top four layers of the pre-trained wav2vec2.0 encoder and the decoder. Obviously, the performance decreases dramatically when fewer layers are fine-tuned. In contrast, when we only tune the adapters of the top eight, top six, and top four layers of the wav2vec2.0 encoder and the decoder, the performance is well maintained, confirming the adaptation ability of the proposed adapters. This also suggests that pruning some adapters layers will not severely hurt the performance; hence, we will perform adapter layer selection in our future work to further reduce parameters. In addition, we observe the same phenomena that increasing the hidden size will lead to marginal performance gains, except too-small hidden size 64.

Table 3. WER (%) on WSJ with different hidden adapter sizes. The wav2vec2.0 base model is used as the pre-trained speech model, which is fine-tuned for encoder–decoder-based E2E ASR. Adapter$_{k-256}$ means that adapters of the top k encoder layers and the decoder are trained.

Method/WER	test_dev93	test_eval92
Transformer (ESPnet) [60]	18.6	14.8
+ LM [60]	8.8	5.6
Whole model fine-tuning		
top 12 layers (full)	7.7	6.4
top 8 layers	9.2	8.4
top 6 layers	10.0	8.8
top 4 layers	15.0	13.2
Proposed FTA		
Adapter$_{512}$	7.7	6.1
Adapter$_{256}$	7.6	6.5
Adapter$_{128}$	8.0	6.5
Adapter$_{64}$	8.2	7.1
Adapter$_{8-256}$	7.9	6.2
Adapter$_{6-256}$	7.9	6.4
Adapter$_{4-256}$	10.7	9.2

6. Conclusions

In this study, we introduced the light-weight adapters, a simple and intuitive adaptation technique in concept, for adapting pre-trained self-supervised learning models for E2E ASR systems. Our experiments demonstrated the effectiveness and parameter efficiency of the adapters. Specifically, we found that the proposed FTA tends to outperform the full fine-tuning on low-resource settings, achieving a relative WER reduction of 17.5%/12.2% on the 10 min LibriSpeech split. In high-resource settings, the FTA achieves comparable recognition performance but with a significant reduction in parameter usage. Furthermore, we explored the trade-off between parameters of adapters and the resulting recognition performance, suggesting that a carefully designed adapter could provide further improvements in ASR performance while maintaining parameter efficiency. In future work, we plan to further explore the potential applications of this adaptation technique. One interesting direction is the integration of pre-trained speech models and pre-trained language models within the E2E ASR framework. We believe this approach could potentially lead to a more comprehensive and efficient ASR system. Additionally, we will also explore how the light-weight adapters can be combined with other methods to further enhance the performance. This could include exploring how the adapters can be used in conjunction with other types of model adaptation techniques.

Author Contributions: Methodology, X.Y., X.G. and H.L.; Validation, X.Y. and X.Q.; Writing—original draft, X.Y. and X.G.; Writing—review & editing, X.Q. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CCF-Tencent Rhino-Bird Open Research Fund, the National Natural Science Foundation of China (Grant No. 62306029 and No. 62271432), and Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen (Grant No. B10120210117-KP02).

Data Availability Statement: Data are contained within the article.

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

References

1. Watanabe, S.; Hori, T.; Kim, S.; Hershey, J.R.; Hayashi, T. Hybrid CTC/attention Architecture for End-to-End Speech Recognition. *IEEE J. Sel. Top. Signal Process.* **2017**, *11*, 1240–1253. [CrossRef]
2. Chan, W.; Jaitly, N.; Le, Q.V.; Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4960–4964.
3. Graves, A.; Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. In Proceedings of the International Conference on Machine Learning (ICML), Beijing, China, 21–26 June 2014; pp. 1764–1772.
4. Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 577–585.
5. Chiu, C.C.; Sainath, T.N.; Wu, Y.; Prabhavalkar, R.; Nguyen, P.; Chen, Z. State-of-the-Art Speech Recognition with Sequence-to-Sequence Models. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AL, Canada, 15–20 April 2018; pp. 4774–4778.
6. Baevski, A.; Zhou, H.; Mohamed, A.; Auli, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020.
7. Chiu, C.C.; Qin, J.; Zhang, Y.; Yu, J.; Wu, Y. Self-supervised learning with random-projection quantizer for speech recognition. In Proceedings of the International Conference on Machine Learning (ICML), Baltimore, MD, USA, 17–23 July 2022; pp. 3915–3924.
8. Baevski, A.; Hsu, W.N.; Xu, Q.; Babu, A.; Gu, J.; Auli, M. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language. In Proceedings of the International Conference on Machine Learning (ICML), Baltimore, MD, USA, 17–23 July 2022; pp. 1298–1312.
9. Liu, A.; 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 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6419–6423.
10. Chung, Y.A.; Hsu, W.N.; Tang, H.; Glass, J. Generative pre-training for speech with autoregressive predictive coding. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3497–3501.
11. Jiang, D.; Lei, X.; Li, W.; Luo, N.; Hu, Y.; Zou, W.; Li, X. Improving Transformer-based Speech Recognition Using Unsupervised Pre-training. *arXiv* **2019**, arXiv:1910.09932.
12. Jiang, D.; Li, W.; Zhang, R.; Cao, M.; Luo, N.; Han, Y.; Zou, W.; Li, X. A Further Study of Unsupervised Pre-training for Transformer Based Speech Recognition. *arXiv* **2020**, arXiv:2005.09862.
13. Chorowski, J.; Weiss, R.J.; Bengio, S.; van den Oord, A. Unsupervised speech representation learning using WaveNet autoencoders. *IEEE ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 2041–2053. [CrossRef]
14. Chung, Y.A.; Glass, J. Improved speech representations with multi-target autoregressive predictive coding. *arXiv* **2020**, arXiv:2004.05274.
15. Schneider, S.; Baevski, A.; Collobert, R.; Auli, M. wav2vec: Unsupervised pre-training for speech recognition. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019; pp. 3465–3469.
16. Pascual, S.; Ravanelli, M.; Serrà, J.; Bonafonte, A.; Bengio, Y. Learning Problem-Agnostic Speech Representations from Multiple Self-Supervised Tasks. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019; pp. 161–165.
17. Ravanelli, M.; Zhong, J.; Pascual, S.; Swietojanski, P.; Monteiro, J.; Trmal, J.; Bengio, Y. Multi-Task Self-Supervised Learning for Robust Speech Recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6989–6993.
18. Jiang, D.; Li, W.; Cao, M.; Zhang, R.; Zou, W.; Han, K.; Li, X. Speech SIMCLR: Combining Contrastive and Reconstruction Objective for Self-supervised Speech Representation Learning. *arXiv* **2020**, arXiv:2010.13991.
19. Song, X.; Wang, G.; Huang, Y.; Wu, Z.; Su, D.; Meng, H. Speech-XLNet: Unsupervised Acoustic Model Pretraining for Self-Attention Networks. In Proceedings of the INTERSPEECH, Shanghai, China, 25–29 October 2020; pp. 3765–3769.
20. Liu, A.; Li, S.W.; Lee, H.Y. TERA: Self-supervised learning of Transformer encoder representation for speech. *IEEE ACM Trans. Audio Speech Lang. Process.* **2020**, *27*, 2351–2366. [CrossRef]
21. Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 1505–1518. [CrossRef]
22. Baevski, A.; Auli, M.; Rahman Mohamed, A. Effectiveness of self-supervised pre-training for speech recognition. *arXiv* **2019**, arXiv:1911.03912.
23. Wang, W.; Tang, Q.; Livescu, K. Unsupervised Pre-Training of Bidirectional Speech Encoders via Masked Reconstruction. In Proceedings of the ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6889–6893.
24. Yi, C.; Wang, J.; Cheng, N.; Zhou, S.; Xu, B. Applying Wav2vec2.0 to Speech Recognition in Various Low-resource Languages. *arXiv* **2020**, arXiv:2012.12121.
25. Cheng, Y.; Shiyu, Z.; Bo, X. Efficiently Fusing Pretrained Acoustic and Linguistic Encoders for Low-Resource Speech Recognition. *IEEE Signal Process. Lett.* **2021**, *28*, 788–792.
26. McCloskey, M.; Cohen, N.J. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychol. Learn. Motiv.* **1989**, *24*, 109–165.

27. Dodge, J.; Ilharco, G.; Schwartz, R.; Farhadi, A.; Hajishirzi, H.; Smith, N. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv* **2020**, arXiv:2002.06305.
28. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q. Parameter-Efficient Transfer Learning for NLP. *arXiv* **2019**, arXiv:1902.00751.
29. Wang, R.; Tang, D.; Duan, N.; Wei, Z.; Huang, X.; Ji, J.; Cao, G.; Jiang, D.; Zhou, M. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. *arXiv* **2020**, arXiv:2002.01808.
30. Qi, J.; Yang, C.H.H.; Chen, P.Y.; Tejedor, J. Exploiting Low-Rank Tensor-Train Deep Neural Networks Based on Riemannian Gradient Descent with Illustrations of Speech Processing. *IEEE ACM Trans. Audio Speech Lang. Process.* **2023**, *31*, 633–642. [CrossRef]
31. Yang, C.H.H.; Qi, J.; Siniscalchi, S.M.; Lee, C.H. An Ensemble Teacher-Student Learning Approach with Poisson Sub-sampling to Differential Privacy Preserving Speech Recognition. In Proceedings of the 2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP), Singapore, 11–14 December 2022; pp. 1–5.
32. Kahn, J.; Riviere, M.; Zheng, W.; Kharitonov, E.; Xu, Q.; Mazare, P. Libri-Light: A Benchmark for ASR with Limited or No Supervision. *arXiv* **2019**, arXiv:1912.07875.
33. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the ICASSP, Brisbane, QL, Australia, 19–24 April 2015; pp. 5206–5210.
34. Paul, D.B.; Baker, J. The Design for the wall street journal-based CSR Corpus. In Proceedings of the workshop on Speech and Natural Language, Harriman, NY, USA, 23–26 February 1992; pp. 357–362.
35. van den Oord, A.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
36. Chen, S.; Wu, Y.; Wang, C.; Chen, Z.; Chen, Z.; Liu, S.; Wu, J.; Qian, Y.; Wei, F.; Li, J.; et al. Unispeech-Sat: Universal Speech Representation Learning With Speaker Aware Pre-Training. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 7–13 May 2022; pp. 6152–6156.
37. Vaessen, N.; Van Leeuwen, D.A. Fine-Tuning Wav2Vec2 for Speaker Recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 7–13 May 2022; pp. 7967–7971.
38. Huang, Z.; Watanabe, S.; Yang, S.w.; García, P.; Khudanpur, S. Investigating Self-Supervised Learning for Speech Enhancement and Separation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 7–13 May 2022; pp. 6837–6841.
39. Nguyen, H.; Bougares, F.; Tomashenko, N.; Estève, Y.; Besacier, L. Investigating Self-Supervised Pre-Training for End-to-End Speech Translation. In Proceedings of the INTERSPEECH, Shanghai, China, 25–29 October 2020; pp. 1466–1470.
40. Fukuda, R.; Sudoh, K.; Nakamura, S. Improving Speech Translation Accuracy and Time Efficiency with Fine-tuned wav2vec 2.0-based Speech Segmentation. *IEEE ACM Trans. Audio Speech Lang. Process.* **2023**, 1 –12. [CrossRef]
41. Huh, J.; Heo, H.S.; Kang, J.; Watanabe, S.; Chung, J.S. Augmentation adversarial training for self-supervised speaker recognition. *arXiv* **2020**, arXiv:2007.12085.
42. Chen, X.; He, K. Exploring Simple Siamese Representation Learning. *arXiv* **2020**, arXiv:2011.10566.
43. Giorgi, J.M.; Nitski, O.; Bader, G.D.; Wang, B. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. *arXiv* **2020**, arXiv:2006.03659.
44. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
45. Mikolov, T.; Karafiát, M.; Burget, L.; Cernocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the INTERSPEECH, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
46. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
47. Rivière, M.; Joulin, A.; Mazaré, P.E.; Dupoux, E. Unsupervised Pretraining Transfers Well Across Languages. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 414–7418.
48. Rebuffi, S.A.; Bilen, H.; Vedaldi, A. Learning multiple visual domains with residual adapters. In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 506–516.
49. Sung, Y.L.; Cho, J.; Bansal, M. VL-Adapter: Parameter-Efficient Transfer Learning for Vision-and-Language Tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LO, USA, 18–24 June 2022; pp. 5227–5237.
50. Chen, T.; Zhu, L.; Deng, C.; Cao, R.; Wang, Y.; Zhang, S.; Li, Z.; Sun, L.; Zang, Y.; Mao, P. SAM-Adapter: Adapting Segment Anything in Underperformed Scenes. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Paris, France, 2–3 October 2023; pp. 3367–3375.
51. Shah, A.; Thapa, S.; Jain, A.; Huang, L. ADEPT: Adapter-based Efficient Prompt Tuning Approach for Language Models. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Toronto, ON, Canada, 9–14 July 2023; pp. 121–128.

52. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 3–8 December 2018; pp. 6000–6010.
53. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
55. Wu, F.; Fan, A.; Baevski, A.; Dauphin, Y.N.; Auli, M. Pay Less Attention with Lightweight and Dynamic Convolutions. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
56. Ao, J.; Zhang, Z.; Zhou, L.; Liu, S.; Li, H.; Ko, T.; Dai, L.; Li, J.; Qian, Y.; Wei, F. Pre-Training Transformer Decoder for End-to-End ASR Model with Unpaired Speech Data. In Proceedings of the INTERSPEECH, Virtual, 14 September 2022.
57. Ling, S.; Liu, Y. DeCoAR 2.0: Deep Contextualized Acoustic Representations with Vector Quantization. *arXiv* **2020**, arXiv:2012.06659.
58. Park, D.S.; Zhang, Y.; Jia, Y.; Han, W.; Chiu, C.C.; Li, B.; Wu, Y.; Le, Q.V. Improved Noisy Student Training for Automatic Speech Recognition. In Proceedings of the INTERSPEECH, Shanghai, China, 25–29 October 2020; pp. 2817–2821.
59. Talnikar, C.; Likhomanenko, T.; Collobert, R.; Synnaeve, G. Joint Masked CPC and CTC Training for ASR. *arXiv* **2020**, arXiv:2011.00093.
60. Watanabe, S.; Hori, T.; Karita, S.; Hayashi, T.; Nishitoba, J. ESPnet: End-toend speech processing toolkit. In Proceedings of the INTERSPEECH, Hyderabad, India, 2–6 September 2018; pp. 2207–2211.